



POINT GREY

FlyCapture 2.1

Managed API Programming Reference

Revised September 16, 2010

Point Grey Research Inc.

12051 Riverside Way • Richmond, BC • Canada • V6W 1K7 • T (604) 242-9937 •
www.ptgrey.com

Software Warranty

Point Grey Research warrants to the Original Purchaser, for a period of one (1) year from date of purchase that:

1. The diskette on which the Software is furnished and the accompanying documentation are not defective;
2. The Software is properly recorded upon the diskettes enclosed;
3. The documentation is substantially complete and contains all the information Point Grey Research deems necessary to use the Software;
4. The Software functions substantially as described in the documentation.

Point Grey Research, Inc.'s entire liability and the Original Purchaser's exclusive remedy shall be the replacement of any diskette or documentation not meeting these warranties. On such an occasion, a copy of the paid receipt accompanied with the faulty diskette or documentation must be returned to Point Grey Research, Inc. or an authorized dealer.

Point Grey Research, Inc. expressly disclaims and excludes all other warranties, express, implied and statutory, including, but without limitation, warranty of merchantability and fitness for a particular application or purpose. In no event shall Point Grey Research, Inc. be liable to the Original Purchaser or any third party for direct, indirect, incidental, consequential, special or accidental damages, including without limitation damages for business interruption, loss of profits, revenue, data or bodily injury or death.

Software License Agreement

The FlyCapture® Software Development Kit (the "Software") is owned and copyrighted by Point Grey Research, Inc. All rights are reserved. The Original Purchaser is granted a license to use the Software subject to the following restrictions and limitations.

1. The license is to the Original Purchaser only, and is nontransferable unless you have received written permission of Point Grey Research, Inc.
2. The Original Purchaser may use the Software only with Point Grey Research, Inc. cameras owned by the Original Purchaser, including but not limited to, Firefly®, Firefly®2, Firefly® MV, Flea®, Scorpion™, Dragonfly®, Dragonfly®2, Dragonfly Express™, Grasshopper™ or Chameleon™ Camera Modules.
3. The Original Purchaser may make back-up copies of the Software for his or her own use only, subject to the use limitations of this license.
4. Subject to s.5 below, the Original Purchaser may not engage in, nor permit third parties to engage in, any of the following:
 - A. Providing or disclosing the Software to third parties.
 - B. Making alterations or copies of any kind of the Software (except as specifically permitted in s.3 above).
 - C. Attempting to un-assemble, de-compile or reverse engineer the Software in any way.
 - D. Granting sublicenses, leases or other rights in the Software to others.
5. Original Purchasers who are Original Equipment Manufacturers may make Derivative Products with the Software. Derivative Products are new software products developed, in whole or in part, using the Software and other Point Grey Research, Inc. products. Point Grey Research, Inc. hereby grants a license to Original Equipment Manufacturers to incorporate and distribute the libraries found in the Software with the Derivative Products. The components of any Derivative Product that contain the Software libraries may only be used with Point Grey Research, Inc. products, or images derived from such products.
 - 5.1 By the distribution of the Software libraries with Derivative Products, Original Purchasers agree to:
 - A. not permit further redistribution of the Software libraries by end-user customers;
 - B. include a valid copyright notice on any Derivative Product; and
 - C. indemnify, hold harmless, and defend Point Grey Research, Inc. from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of any Derivative Product.

Point Grey Research, Inc. reserves the right to terminate this license if there are any violations of its terms or if there is a default committed by the Original Purchaser. Upon termination, for any reason, all copies of the Software must be immediately returned to Point Grey Research, Inc. and the Original Purchaser shall be liable to Point Grey Research, Inc. for any and all damages suffered as a result of the violation or default.

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	Module Documentation	9
5.1	Enumerations	9
5.1.1	Enumeration Type Documentation	15
5.1.1.1	BandwidthAllocation	15
5.1.1.2	BayerTileFormat	16
5.1.1.3	BusSpeed	16
5.1.1.4	ByteOrder	17
5.1.1.5	ColorProcessingAlgorithm	17
5.1.1.6	ErrorType	17
5.1.1.7	FrameRate	18
5.1.1.8	GigEPropertyType	19
5.1.1.9	GrabMode	19
5.1.1.10	GrabTimeout	19
5.1.1.11	ImageFileFormat	20
5.1.1.12	InterfaceType	20
5.1.1.13	ManagedCallbackType	20
5.1.1.14	Mode	21
5.1.1.15	OSType	22

5.1.1.16	PixelFormat	22
5.1.1.17	PropertyType	22
5.1.1.18	StatisticsChannel	23
5.1.1.19	VideoMode	24
5.2	Structures	25
5.3	Image saving structures.	27
5.3.1	Detailed Description	27
6	Namespace Documentation	29
6.1	FlyCapture2 Namespace Reference	29
6.2	FlyCapture2Managed Namespace Reference	30
6.2.1	Function Documentation	39
6.2.1.1	EnumCallback	39
6.2.1.2	ImageCallbackDelegate	39
6.2.1.3	ImageEventCallback	39
6.3	FlyCapture2Managed::Gui Namespace Reference	40
7	Class Documentation	41
7.1	AviOption Struct Reference	41
7.1.1	Detailed Description	41
7.1.2	Constructor & Destructor Documentation	41
7.1.2.1	AviOption	41
7.1.3	Property Documentation	41
7.1.3.1	frameRate	41
7.2	CameraControlDialog Class Reference	42
7.2.1	Detailed Description	42
7.2.2	Constructor & Destructor Documentation	42
7.2.2.1	CameraControlDialog	42
7.2.2.2	~CameraControlDialog	42
7.2.3	Member Function Documentation	42
7.2.3.1	Connect	42
7.2.3.2	Disconnect	42
7.2.3.3	Hide	42
7.2.3.4	IsVisible	42
7.2.3.5	Show	42
7.3	CameraInfo Struct Reference	43
7.3.1	Detailed Description	44

7.3.2	Property Documentation	44
7.3.2.1	bayerTileFormat	44
7.3.2.2	configROM	44
7.3.2.3	defaultGateway	44
7.3.2.4	driverName	45
7.3.2.5	firmwareBuildTime	45
7.3.2.6	firmwareVersion	45
7.3.2.7	gigEMajorVersion	45
7.3.2.8	gigEMinorVersion	45
7.3.2.9	iidcVersion	45
7.3.2.10	interfaceType	45
7.3.2.11	ipAddress	45
7.3.2.12	isColorCamera	45
7.3.2.13	macAddress	45
7.3.2.14	maximumBusSpeed	45
7.3.2.15	modelName	46
7.3.2.16	sensorInfo	46
7.3.2.17	sensorResolution	46
7.3.2.18	serialNumber	46
7.3.2.19	subnetMask	46
7.3.2.20	userDefinedName	46
7.3.2.21	vendorName	46
7.3.2.22	xmlURL1	46
7.3.2.23	xmlURL2	46
7.4	CameraProperty Struct Reference	47
7.4.1	Detailed Description	47
7.4.2	Constructor & Destructor Documentation	48
7.4.2.1	CameraProperty	48
7.4.2.2	CameraProperty	48
7.4.3	Property Documentation	48
7.4.3.1	absControl	48
7.4.3.2	absValue	48
7.4.3.3	autoManualMode	48
7.4.3.4	onePush	48
7.4.3.5	onOff	48
7.4.3.6	present	48

7.4.3.7	type	48
7.4.3.8	valueA	48
7.4.3.9	valueB	48
7.5	CameraPropertyInfo Struct Reference	49
7.5.1	Detailed Description	50
7.5.2	Constructor & Destructor Documentation	50
7.5.2.1	CameraPropertyInfo	50
7.5.2.2	CameraPropertyInfo	50
7.5.3	Property Documentation	50
7.5.3.1	absMax	50
7.5.3.2	absMin	50
7.5.3.3	absValSupported	50
7.5.3.4	autoSupported	50
7.5.3.5	manualSupported	50
7.5.3.6	max	50
7.5.3.7	min	50
7.5.3.8	onePushSupported	51
7.5.3.9	onOffSupported	51
7.5.3.10	present	51
7.5.3.11	readOutSupported	51
7.5.3.12	type	51
7.5.3.13	unitAbbr	51
7.5.3.14	units	51
7.6	CameraSelectionDialog Class Reference	52
7.6.1	Detailed Description	52
7.6.2	Constructor & Destructor Documentation	52
7.6.2.1	CameraSelectionDialog	52
7.6.2.2	~CameraSelectionDialog	52
7.6.3	Member Function Documentation	52
7.6.3.1	GetSelectedCameraGuids	52
7.6.3.2	ShowModal	52
7.7	ConfigROM Struct Reference	53
7.7.1	Detailed Description	53
7.7.2	Property Documentation	53
7.7.2.1	chipIdHi	53
7.7.2.2	chipIdLo	54

7.7.2.3	keyword	54
7.7.2.4	nodeVendorId	54
7.7.2.5	unitSpecId	54
7.7.2.6	unitSubSWVer	54
7.7.2.7	unitSWVer	54
7.7.2.8	vendorUniqueInfo0	54
7.7.2.9	vendorUniqueInfo1	54
7.7.2.10	vendorUniqueInfo2	54
7.7.2.11	vendorUniqueInfo3	54
7.8	EmbeddedImageInfo Struct Reference	55
7.8.1	Detailed Description	55
7.8.2	Constructor & Destructor Documentation	56
7.8.2.1	EmbeddedImageInfo	56
7.8.3	Property Documentation	56
7.8.3.1	brightness	56
7.8.3.2	exposure	56
7.8.3.3	frameCounter	56
7.8.3.4	gain	56
7.8.3.5	GPIOPinState	56
7.8.3.6	ROIPosition	56
7.8.3.7	shutter	56
7.8.3.8	strobePattern	56
7.8.3.9	timestamp	56
7.8.3.10	whiteBalance	56
7.9	EmbeddedImageInfoProperty Struct Reference	57
7.9.1	Detailed Description	57
7.9.2	Property Documentation	57
7.9.2.1	available	57
7.9.2.2	onOff	57
7.10	FC2Config Struct Reference	58
7.10.1	Detailed Description	58
7.10.2	Constructor & Destructor Documentation	58
7.10.2.1	FC2Config	58
7.10.3	Property Documentation	58
7.10.3.1	asyncBusSpeed	58
7.10.3.2	bandwidthAllocation	59

7.10.3.3	grabMode	59
7.10.3.4	grabTimeout	59
7.10.3.5	isochBusSpeed	59
7.10.3.6	numBuffers	59
7.10.3.7	numImageNotifications	59
7.11	FC2Exception Class Reference	60
7.11.1	Detailed Description	60
7.11.2	Constructor & Destructor Documentation	61
7.11.2.1	FC2Exception	61
7.11.2.2	FC2Exception	61
7.11.2.3	FC2Exception	61
7.11.2.4	~FC2Exception	61
7.11.2.5	FC2Exception	61
7.11.2.6	FC2Exception	61
7.11.3	Property Documentation	61
7.11.3.1	CauseType	61
7.11.3.2	Type	61
7.12	FC2Version Struct Reference	62
7.12.1	Detailed Description	62
7.12.2	Property Documentation	62
7.12.2.1	build	62
7.12.2.2	major	62
7.12.2.3	minor	62
7.12.2.4	type	62
7.13	Format7ImageSettings Struct Reference	63
7.13.1	Detailed Description	63
7.13.2	Property Documentation	63
7.13.2.1	height	63
7.13.2.2	mode	63
7.13.2.3	offsetX	63
7.13.2.4	offsetY	63
7.13.2.5	pixelFormat	64
7.13.2.6	width	64
7.14	Format7Info Struct Reference	65
7.14.1	Detailed Description	65
7.14.2	Property Documentation	66

7.14.2.1	imageHStepSize	66
7.14.2.2	imageVStepSize	66
7.14.2.3	maxHeight	66
7.14.2.4	maxPacketSize	66
7.14.2.5	maxWidth	66
7.14.2.6	minPacketSize	66
7.14.2.7	mode	66
7.14.2.8	offsetHStepSize	66
7.14.2.9	offsetVStepSize	66
7.14.2.10	packetSize	66
7.14.2.11	percentage	66
7.14.2.12	pixelFormatBitField	67
7.15	Format7PacketInfo Struct Reference	68
7.15.1	Detailed Description	68
7.15.2	Property Documentation	68
7.15.2.1	maxBytesPerPacket	68
7.15.2.2	recommendedBytesPerPacket	68
7.15.2.3	unitBytesPerPacket	68
7.16	GigEConfig Struct Reference	69
7.16.1	Detailed Description	69
7.16.2	Property Documentation	69
7.16.2.1	channels	69
7.16.2.2	numChannels	69
7.17	GigEImageSettings Struct Reference	70
7.17.1	Detailed Description	70
7.17.2	Property Documentation	70
7.17.2.1	height	70
7.17.2.2	offsetX	70
7.17.2.3	offsetY	70
7.17.2.4	pixelFormat	70
7.17.2.5	width	70
7.18	GigEImageSettingsInfo Struct Reference	71
7.18.1	Detailed Description	71
7.18.2	Property Documentation	71
7.18.2.1	imageHStepSize	71
7.18.2.2	imageVStepSize	71

7.18.2.3	maxHeight	71
7.18.2.4	maxWidth	72
7.18.2.5	offsetHStepSize	72
7.18.2.6	offsetVStepSize	72
7.18.2.7	pixelFormatBitField	72
7.19	GigEProperty Struct Reference	73
7.19.1	Detailed Description	73
7.19.2	Property Documentation	73
7.19.2.1	isReadable	73
7.19.2.2	isWritable	73
7.19.2.3	max	73
7.19.2.4	min	73
7.19.2.5	propType	74
7.19.2.6	value	74
7.20	GigEStreamChannel Struct Reference	75
7.20.1	Detailed Description	75
7.20.2	Property Documentation	75
7.20.2.1	destinationIpAddress	75
7.20.2.2	doNotFragment	75
7.20.2.3	hostPost	75
7.20.2.4	interPacketDelay	76
7.20.2.5	networkInterfaceIndex	76
7.20.2.6	packetSize	76
7.20.2.7	sourcePort	76
7.21	ImageMetadata Struct Reference	77
7.21.1	Detailed Description	77
7.21.2	Property Documentation	77
7.21.2.1	embeddedBrightness	77
7.21.2.2	embeddedExposure	78
7.21.2.3	embeddedFrameCounter	78
7.21.2.4	embeddedGain	78
7.21.2.5	embeddedGPIOPinState	78
7.21.2.6	embeddedROIPosition	78
7.21.2.7	embeddedShutter	78
7.21.2.8	embeddedStrobePattern	78
7.21.2.9	embeddedTimeStamp	78

7.21.2.10 embeddedWhiteBalance	78
7.22 JpegOption Struct Reference	79
7.22.1 Detailed Description	79
7.22.2 Constructor & Destructor Documentation	79
7.22.2.1 JpegOption	79
7.22.3 Property Documentation	79
7.22.3.1 progressive	79
7.22.3.2 quality	79
7.23 Jpg2Option Struct Reference	80
7.23.1 Detailed Description	80
7.23.2 Constructor & Destructor Documentation	80
7.23.2.1 Jpg2Option	80
7.23.3 Property Documentation	80
7.23.3.1 quality	80
7.24 LutData Struct Reference	81
7.24.1 Detailed Description	81
7.24.2 Property Documentation	81
7.24.2.1 enabled	81
7.24.2.2 inputBitDepth	81
7.24.2.3 numBanks	81
7.24.2.4 numChannels	82
7.24.2.5 numEntries	82
7.24.2.6 outputBitDepth	82
7.24.2.7 supported	82
7.25 ManagedAVIRecorder Class Reference	83
7.25.1 Detailed Description	83
7.25.2 Constructor & Destructor Documentation	83
7.25.2.1 ManagedAVIRecorder	83
7.25.2.2 ~ManagedAVIRecorder	83
7.25.3 Member Function Documentation	83
7.25.3.1 AVIAppend	83
7.25.3.2 AVIClose	83
7.25.3.3 AVIOpen	84
7.26 ManagedBusManager Class Reference	85
7.26.1 Detailed Description	86
7.26.2 Constructor & Destructor Documentation	87

7.26.2.1	ManagedBusManager	87
7.26.2.2	~ManagedBusManager	87
7.26.2.3	!ManagedBusManager	87
7.26.3	Member Function Documentation	87
7.26.3.1	ConvertToManagedGuid	87
7.26.3.2	ConvertToNativeGuid	87
7.26.3.3	DiscoverGigECameras	87
7.26.3.4	FireBusReset	87
7.26.3.5	ForceIPAddressToCamera	88
7.26.3.6	GetCameraFromIndex	88
7.26.3.7	GetCameraFromIPAddress	88
7.26.3.8	GetCameraFromSerialNumber	88
7.26.3.9	GetCameraSerialNumberFromIndex	89
7.26.3.10	GetDeviceFromIndex	89
7.26.3.11	GetInterfaceTypeFromGuid	89
7.26.3.12	GetNumOfCameras	90
7.26.3.13	GetNumOfDevices	90
7.26.3.14	ReadPhyRegister	90
7.26.3.15	RegisterCallback	90
7.26.3.16	RescanBus	91
7.26.3.17	UnregisterCallback	91
7.26.3.18	WritePhyRegister	91
7.27	ManagedCamera Class Reference	92
7.27.1	Detailed Description	93
7.27.2	Constructor & Destructor Documentation	94
7.27.2.1	ManagedCamera	94
7.27.2.2	~ManagedCamera	94
7.27.2.3	!ManagedCamera	94
7.27.3	Member Function Documentation	94
7.27.3.1	GetFormat7Configuration	94
7.27.3.2	GetFormat7Info	94
7.27.3.3	GetVideoModeAndFrameRate	95
7.27.3.4	GetVideoModeAndFrameRateInfo	95
7.27.3.5	SetFormat7Configuration	95
7.27.3.6	SetFormat7Configuration	96
7.27.3.7	SetVideoModeAndFrameRate	96

7.27.3.8	ValidateFormat7Settings	96
7.28	ManagedCameraBase Class Reference	97
7.28.1	Detailed Description	100
7.28.2	Constructor & Destructor Documentation	101
7.28.2.1	ManagedCameraBase	101
7.28.2.2	~ManagedCameraBase	101
7.28.3	Member Function Documentation	101
7.28.3.1	Connect	101
7.28.3.2	Disconnect	101
7.28.3.3	EnableLUT	101
7.28.3.4	FireSoftwareTrigger	101
7.28.3.5	GetActiveLUTBank	102
7.28.3.6	GetCameraInfo	102
7.28.3.7	GetConfiguration	102
7.28.3.8	GetEmbeddedImageInfo	102
7.28.3.9	GetGPIOPinDirection	102
7.28.3.10	GetLUTBankInfo	103
7.28.3.11	GetLUTChannel	103
7.28.3.12	GetLUTInfo	103
7.28.3.13	GetMemoryChannel	104
7.28.3.14	GetMemoryChannelInfo	104
7.28.3.15	GetNativeCamera	104
7.28.3.16	GetProperty	104
7.28.3.17	GetPropertyInfo	105
7.28.3.18	GetStrobe	105
7.28.3.19	GetStrobeInfo	105
7.28.3.20	GetTriggerMode	106
7.28.3.21	GetTriggerModeInfo	106
7.28.3.22	IsConnected	106
7.28.3.23	OnNativeCallback	106
7.28.3.24	ReadRegister	106
7.28.3.25	ReadRegisterBlock	107
7.28.3.26	RestoreFromMemoryChannel	107
7.28.3.27	RetrieveBuffer	107
7.28.3.28	SaveToMemoryChannel	108
7.28.3.29	SetActiveLUTBank	108

7.28.3.30	SetCallback	108
7.28.3.31	SetConfiguration	108
7.28.3.32	SetEmbeddedImageInfo	109
7.28.3.33	SetGPIOPinDirection	109
7.28.3.34	SetGPIOPinDirection	109
7.28.3.35	SetLUTChannel	109
7.28.3.36	SetProperty	110
7.28.3.37	SetProperty	110
7.28.3.38	SetStrobe	110
7.28.3.39	SetTriggerMode	111
7.28.3.40	StartCapture	111
7.28.3.41	StartCapture	111
7.28.3.42	StopCapture	112
7.28.3.43	WaitForBufferEvent	112
7.28.3.44	WriteRegister	112
7.28.3.45	WriteRegister	112
7.28.3.46	WriteRegisterBlock	113
7.28.4	Member Data Documentation	113
7.28.4.1	m_externalDelegate	113
7.28.4.2	m_internalDelegate	113
7.28.4.3	m_pNativeCamBase	113
7.29	ManagedGigECamera Class Reference	114
7.29.1	Detailed Description	116
7.29.2	Constructor & Destructor Documentation	117
7.29.2.1	ManagedGigECamera	117
7.29.2.2	~ManagedGigECamera	117
7.29.2.3	!ManagedGigECamera	117
7.29.3	Member Function Documentation	117
7.29.3.1	DiscoverGigEPacketSize	117
7.29.3.2	GetGigEImageBinningSettings	117
7.29.3.3	GetGigEImageSettings	117
7.29.3.4	GetGigEImageSettingsInfo	117
7.29.3.5	GetGigEImagingMode	118
7.29.3.6	GetGigEProperty	118
7.29.3.7	GetGigEStreamChannelInfo	118
7.29.3.8	GetNumStreamChannels	118

7.29.3.9	QueryGigEImagingMode	118
7.29.3.10	ReadGVCPMemory	119
7.29.3.11	ReadGVCPRegister	119
7.29.3.12	ReadGVCPRegisterBlock	119
7.29.3.13	SetGigEImageBinningSettings	119
7.29.3.14	SetGigEImageSettings	119
7.29.3.15	SetGigEImagingMode	120
7.29.3.16	SetGigEProperty	120
7.29.3.17	SetGigEStreamChannelInfo	120
7.29.3.18	WriteGVCPMemory	120
7.29.3.19	WriteGVCPRegister	120
7.29.3.20	WriteGVCPRegister	121
7.29.3.21	WriteGVCPRegisterBlock	121
7.30	ManagedImage Class Reference	122
7.30.1	Detailed Description	124
7.30.2	Constructor & Destructor Documentation	125
7.30.2.1	ManagedImage	125
7.30.2.2	ManagedImage	125
7.30.2.3	ManagedImage	125
7.30.2.4	ManagedImage	125
7.30.2.5	ManagedImage	125
7.30.2.6	ManagedImage	125
7.30.2.7	ManagedImage	125
7.30.2.8	~ManagedImage	125
7.30.2.9	ManagedImage	125
7.30.2.10	!ManagedImage	125
7.30.3	Member Function Documentation	125
7.30.3.1	CalculateStatistics	125
7.30.3.2	Convert	125
7.30.3.3	Convert	125
7.30.3.4	DetermineBitsPerPixel	126
7.30.3.5	GetNativeImage	126
7.30.3.6	GetRawNativeImagePointer	126
7.30.3.7	IsNativeImageValid	126
7.30.3.8	ReleaseBuffer	126
7.30.3.9	Save	126

7.30.3.10	Save	126
7.30.3.11	Save	127
7.30.3.12	Save	127
7.30.3.13	Save	127
7.30.3.14	Save	127
7.30.3.15	Save	127
7.30.3.16	Save	128
7.30.3.17	SetData	128
7.30.3.18	SetDimensions	128
7.30.4	Property Documentation	128
7.30.4.1	bayerTileFormat	128
7.30.4.2	bitmap	128
7.30.4.3	bitsPerPixel	128
7.30.4.4	colorProcessingAlgorithm	129
7.30.4.5	cols	129
7.30.4.6	data	129
7.30.4.7	defaultColorProcessingAlgorithm	129
7.30.4.8	defaultOutputPixelFormat	129
7.30.4.9	imageMetadata	129
7.30.4.10	pixelFormat	129
7.30.4.11	rows	129
7.30.4.12	stride	129
7.30.4.13	timeStamp	129
7.31	ManagedImageStatistics Class Reference	130
7.31.1	Constructor & Destructor Documentation	131
7.31.1.1	ManagedImageStatistics	131
7.31.1.2	~ManagedImageStatistics	131
7.31.2	Member Function Documentation	131
7.31.2.1	DisableAll	131
7.31.2.2	EnableAll	131
7.31.2.3	EnableGreyOnly	131
7.31.2.4	EnableHSLOnly	131
7.31.2.5	EnableRGBOnly	131
7.31.2.6	GetChannelStatus	131
7.31.2.7	GetHistogram	131
7.31.2.8	GetMean	131

7.31.2.9	GetNativeImageStatistics	131
7.31.2.10	GetNumPixelValues	131
7.31.2.11	GetPixelValueRange	131
7.31.2.12	GetRange	131
7.31.2.13	GetStatistics	131
7.31.2.14	SetChannelStatus	131
7.32	ManagedPGRGuid Class Reference	132
7.32.1	Detailed Description	132
7.32.2	Constructor & Destructor Documentation	132
7.32.2.1	ManagedPGRGuid	132
7.32.2.2	ManagedPGRGuid	133
7.32.2.3	ManagedPGRGuid	133
7.32.3	Member Function Documentation	133
7.32.3.1	Equals	133
7.32.3.2	GetHashCode	133
7.32.3.3	operator!=	133
7.32.3.4	operator=	133
7.32.3.5	operator==	133
7.32.4	Member Data Documentation	133
7.32.4.1	value0	133
7.32.4.2	value1	133
7.32.4.3	value2	133
7.32.4.4	value3	133
7.33	ManagedUtilities Class Reference	134
7.33.1	Member Function Documentation	134
7.33.1.1	LaunchBrowser	134
7.33.1.2	LaunchCommand	134
7.33.1.3	LaunchHelp	134
7.33.2	Property Documentation	134
7.33.2.1	libraryVersion	134
7.33.2.2	systemInfo	134
7.34	PgmOption Struct Reference	135
7.34.1	Detailed Description	135
7.34.2	Constructor & Destructor Documentation	135
7.34.2.1	PgmOption	135
7.34.3	Property Documentation	135

7.34.3.1	binaryFile	135
7.35	PngOption Struct Reference	136
7.35.1	Detailed Description	136
7.35.2	Constructor & Destructor Documentation	136
7.35.2.1	PngOption	136
7.35.3	Property Documentation	136
7.35.3.1	compressionLevel	136
7.35.3.2	interlaced	136
7.36	PpmOption Struct Reference	137
7.36.1	Detailed Description	137
7.36.2	Constructor & Destructor Documentation	137
7.36.2.1	PpmOption	137
7.36.3	Property Documentation	137
7.36.3.1	binaryFile	137
7.37	StrobeControl Struct Reference	138
7.37.1	Detailed Description	138
7.37.2	Property Documentation	138
7.37.2.1	delay	138
7.37.2.2	duration	138
7.37.2.3	onOff	138
7.37.2.4	polarity	138
7.37.2.5	source	138
7.38	StrobeInfo Struct Reference	139
7.38.1	Detailed Description	139
7.38.2	Property Documentation	139
7.38.2.1	maxValue	139
7.38.2.2	minValue	139
7.38.2.3	onOffSupported	139
7.38.2.4	polaritySupported	140
7.38.2.5	present	140
7.38.2.6	readOutSupported	140
7.38.2.7	source	140
7.39	SystemInfo Struct Reference	141
7.39.1	Detailed Description	141
7.39.2	Property Documentation	141
7.39.2.1	byteOrder	141

7.39.2.2	cpuDescription	142
7.39.2.3	driverList	142
7.39.2.4	gpuDescription	142
7.39.2.5	libraryList	142
7.39.2.6	numCpuCores	142
7.39.2.7	osDescription	142
7.39.2.8	osType	142
7.39.2.9	screenHeight	142
7.39.2.10	screenWidth	142
7.39.2.11	systemMemorySize	142
7.40	TiffOption Struct Reference	143
7.40.1	Detailed Description	143
7.40.2	Member Enumeration Documentation	143
7.40.2.1	CompressionMethod	143
7.40.3	Constructor & Destructor Documentation	144
7.40.3.1	TiffOption	144
7.40.4	Property Documentation	144
7.40.4.1	compression	144
7.41	TimeStamp Struct Reference	145
7.41.1	Detailed Description	145
7.41.2	Property Documentation	145
7.41.2.1	cycleCount	145
7.41.2.2	cycleOffset	145
7.41.2.3	cycleSeconds	145
7.41.2.4	microSeconds	145
7.41.2.5	seconds	145
7.42	Translate Class Reference	146
7.42.1	Member Function Documentation	149
7.42.1.1	ToMgd	149
7.42.1.2	ToMgd	149
7.42.1.3	ToMgd	149
7.42.1.4	ToMgd	149
7.42.1.5	ToMgd	149
7.42.1.6	ToMgd	149
7.42.1.7	ToMgd	149
7.42.1.8	ToMgd	149

7.42.1.9 ToMgd	149
7.42.1.10 ToMgd	149
7.42.1.11 ToMgd	149
7.42.1.12 ToMgd	149
7.42.1.13 ToMgd	149
7.42.1.14 ToMgd	149
7.42.1.15 ToMgd	149
7.42.1.16 ToMgd	149
7.42.1.17 ToMgd	149
7.42.1.18 ToMgd	149
7.42.1.19 ToMgd	149
7.42.1.20 ToMgd	149
7.42.1.21 ToMgd	149
7.42.1.22 ToMgd	149
7.42.1.23 ToMgd	149
7.42.1.24 ToMgd	149
7.42.1.25 ToMgd	149
7.42.1.26 ToMgd	149
7.42.1.27 ToMgd	149
7.42.1.28 ToMgd	149
7.42.1.29 ToMgd	149
7.42.1.30 ToMgd	149
7.42.1.31 ToMgd	149
7.42.1.32 ToNative	149
7.42.1.33 ToNative	149
7.42.1.34 ToNative	149
7.42.1.35 ToNative	149
7.42.1.36 ToNative	149
7.42.1.37 ToNative	149
7.42.1.38 ToNative	149
7.42.1.39 ToNative	149
7.42.1.40 ToNative	149
7.42.1.41 ToNative	149
7.42.1.42 ToNative	149
7.42.1.43 ToNative	149
7.42.1.44 ToNative	149

7.42.1.45 ToNative	149
7.42.1.46 ToNative	149
7.42.1.47 ToNative	149
7.42.1.48 ToNative	149
7.42.1.49 ToNative	149
7.42.1.50 ToNative	149
7.42.1.51 ToNative	149
7.42.1.52 ToNative	149
7.42.1.53 ToNative	149
7.42.1.54 ToNative	149
7.42.1.55 ToNative	149
7.42.1.56 ToNative	149
7.42.1.57 ToNative	149
7.42.1.58 ToNative	149
7.42.1.59 ToNative	149
7.42.1.60 ToNative	149
7.42.1.61 ToNative	149
7.42.1.62 translate	149
7.42.1.63 translate	149
7.42.1.64 translate	149
7.42.1.65 translate	149
7.42.1.66 translate	149
7.42.1.67 translate	149
7.42.1.68 translate	149
7.42.1.69 translate	149
7.42.1.70 translate	149
7.42.1.71 translate	149
7.42.1.72 translate	149
7.42.1.73 translate	149
7.42.1.74 translate	149
7.42.1.75 translate	149
7.42.1.76 translate	149
7.42.1.77 translate	149
7.42.1.78 translate	149
7.42.1.79 translate	149
7.42.1.80 translate	149

7.42.1.81	translate	149
7.42.1.82	translate	149
7.42.1.83	translate	149
7.42.1.84	translate	149
7.42.1.85	translate	149
7.42.1.86	translate	149
7.42.1.87	translate	149
7.42.1.88	translate	149
7.42.1.89	translate	149
7.42.1.90	translate	149
7.42.1.91	translate	149
7.42.1.92	translate	149
7.42.1.93	translate	149
7.42.1.94	translate	149
7.42.1.95	translate	149
7.42.1.96	translate	149
7.42.1.97	translate	149
7.43	TriggerMode Struct Reference	150
7.43.1	Detailed Description	150
7.43.2	Property Documentation	150
7.43.2.1	mode	150
7.43.2.2	onOff	150
7.43.2.3	parameter	150
7.43.2.4	polarity	150
7.43.2.5	source	150
7.44	TriggerModeInfo Struct Reference	151
7.44.1	Detailed Description	151
7.44.2	Property Documentation	151
7.44.2.1	modeMask	151
7.44.2.2	onOffSupported	151
7.44.2.3	polaritySupported	151
7.44.2.4	present	151
7.44.2.5	readOutSupported	152
7.44.2.6	softwareTriggerSupported	152
7.44.2.7	sourceMask	152
7.44.2.8	valueReadable	152

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Enumerations	9
Structures	25
Image saving structures.	27

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

FlyCapture2	29
FlyCapture2Managed	30
FlyCapture2Managed::Gui	40

Chapter 3

Class Index

3.1 Class Hierarchy

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

AviOption	41
CameraControlDialog	42
CameraInfo	43
CameraProperty	47
CameraPropertyInfo	49
CameraSelectionDialog	52
ConfigROM	53
EmbeddedImageInfo	55
EmbeddedImageInfoProperty	57
FC2Config	58
FC2Exception	60
FC2Version	62
Format7ImageSettings	63
Format7Info	65
Format7PacketInfo	68
GigEConfig	69
GigEImageSettings	70
GigEImageSettingsInfo	71
GigEProperty	73
GigEStreamChannel	75
ImageMetadata	77
JpegOption	79
Jpg2Option	80
LutData	81
ManagedAVIRecorder	83
ManagedBusManager	85
ManagedCameraBase	97
ManagedCamera	92
ManagedGigECamera	114
ManagedImage	122
ManagedImageStatistics	130
ManagedPGRGuid	132
ManagedUtilities	134

PgmOption	135
PngOption	136
PpmOption	137
StrobeControl	138
StrobeInfo	139
SystemInfo	141
TiffOption	143
TimeStamp	145
Translate	146
TriggerMode	150
TriggerModeInfo	151

Chapter 4

Class Index

4.1 Class List

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

AviOption (Options for saving AVI files)	41
CameraControlDialog (CameraControlDialog : managed wrapper of FlyCap- ture2::CameraControlDialog (see for details))	42
CameraInfo (Camera information)	43
CameraProperty (A specific camera property)	47
CameraPropertyInfo (Information about a specific camera property)	49
CameraSelectionDialog (CameraControlDialog : managed wrapper of FlyCap- ture2::CameraSelectionDialog (see for details))	52
ConfigROM (Camera configuration ROM)	53
EmbeddedImageInfo (Properties of the possible embedded image information)	55
EmbeddedImageInfoProperty (Properties of a single embedded image info property)	57
FC2Config (Configuration for a camera)	58
FC2Exception (Exception that is thrown when an error is encountered)	60
FC2Version (The current version of the library)	62
Format7ImageSettings (Format 7 image settings)	63
Format7Info (Format 7 information for a single mode)	65
Format7PacketInfo (Format 7 packet information)	68
GigEConfig (Configuration for a GigE camera)	69
GigEImageSettings (Image settings for a GigE camera)	70
GigEImageSettingsInfo (Format 7 information for a single mode)	71
GigEProperty (A GigE property)	73
GigEStreamChannel (Information about a single GigE stream channel)	75
ImageMetadata (Metadata related to an image)	77
JpegOption (Options for saving JPEG image)	79
Jpg2Option (Options for saving JPEG2000 image)	80
LutData (Information about the camera's look up table)	81
ManagedAVIRecorder (ManagedAVIRecorder provides the functionality for the user to record images to an AVI file)	83
ManagedBusManager (ManagedBusManager provides the functionality for the user to get an PGRGuid for a desired camera or device easily)	85
ManagedCamera (ManagedCamera represents a physical camera that uses the IIDC register set)	92
ManagedCameraBase (Abstract base class that represents a generic camera that defines a general interface to a camera)	97

ManagedGigECamera (The GigECamera object represents a physical Gigabit Ethernet camera)	114
ManagedImage (The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk)	122
ManagedImageStatistics	130
ManagedPGRGuid (Managed version of a PGRGuid)	132
ManagedUtilities	134
PgmOption (Options for saving PGM images)	135
PngOption (Options for saving PNG images)	136
PpmOption (Options for saving PPM images)	137
StrobeControl (A camera strobe)	138
StrobeInfo (A camera strobe property)	139
SystemInfo (Description of the system)	141
TiffOption (Options for saving TIFF images)	143
TimeStamp (Timestamp information)	145
Translate	146
TriggerMode (A camera trigger)	150
TriggerModeInfo (Information about a camera trigger property)	151

Chapter 5

Module Documentation

5.1 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`,
 `IidcFailed`,

StrobeFailed,
TriggerFailed,
PropertyFailed,
PropertyNotPresent,
RegisterFailed,
ReadRegisterFailed,
WriteRegisterFailed,
IsochFailed,
IsochAlreadyStarted,
IsochNotStarted,
IsochStartFailed,
IsochRetrieveBufferFailed,
IsochStopFailed,
IsochSyncFailed,
IsochBandwidthExceeded,
ImageConversionFailed,
ImageLibraryFailure,
BufferTooSmall,
ImageConsistencyError }

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`,
 `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,
NumberOfPixelFormat = 15 }

Pixel formats available for Format7 modes.

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

- enum `ColorProcessingAlgorithm` {
 `Default`,
 `NoColorProcessing`,
 `NearestNeighbor`,
 `EdgeSensing`,
 `HQLinear`,
 `Rigorous` }
 Color processing algorithms.

- enum `BayerTileFormat` {
 `None = 0`,
 `RGGB`,
 `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 `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.

- enum `GigEPropertyType` {
Heartbeat,
HeartbeatTimeout,
PacketSize,
PacketDelay }

Possible properties that can be queried from the camera.

5.1.1 Enumeration Type Documentation

5.1.1.1 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.

5.1.1.2 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.

5.1.1.3 enum BusSpeed

Bus speeds.

Enumerator:

- S100* 100Mbps/sec.
- S200* 200Mbps/sec.
- S400* 400Mbps/sec.
- S480* 480Mbps/sec.
Only for USB cameras.
- S800* 800Mbps/sec.
- S1600* 1600Mbps/sec.
- S3200* 3200Mbps/sec.
- 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 bus speed.

5.1.1.4 enum ByteOrder

Possible byte orders.

Enumerator:

LittleEndian

BigEndian

5.1.1.5 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.

5.1.1.6 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.

5.1.1.7 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.

5.1.1.8 enum GigEPropertyType

Possible properties that can be queried from the camera.

Enumerator:

Heartbeat

HeartbeatTimeout

PacketSize

PacketDelay

5.1.1.9 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 RetrieveBuffer() 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.

5.1.1.10 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.

5.1.1.11 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.

5.1.1.12 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.
GigE GigE.
Unknown Unknown interface.
Unknown bus speed.

5.1.1.13 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.

5.1.1.14 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

5.1.1.15 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.

5.1.1.16 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.

NumberOfPixelFormat Number of pixel formats.

5.1.1.17 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.

5.1.1.18 enum StatisticsChannel

Channels that allow statistics to be calculated.

Enumerator:

Grey

Red

Green

Blue

Hue Hue.

Saturation Saturation.

Lightness

NumberOfStatisticsChannels

5.1.1.19 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.

5.2 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 [TriggerModeInfo](#)
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 [PngOption](#)
Options for saving PNG images.

Modules

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

5.3 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 [AviOption](#)
Options for saving AVI files.
- struct [SystemInfo](#)
Description of the system.

5.3.1 Detailed Description

These structures define various parameters used for saving images.

Chapter 6

Namespace Documentation

6.1 FlyCapture2 Namespace Reference

6.2 FlyCapture2Managed Namespace Reference

Namespaces

- namespace [Gui](#)

Classes

- class [FC2Exception](#)
Exception that is thrown when an error is encountered.
- class [ManagedAVIRecorder](#)
ManagedAVIRecorder provides the functionality for the user to record images to an AVI file.
- 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.
- 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 [TriggerModeInfo](#)
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 [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 [AviOption](#)
Options for saving AVI files.
- struct [SystemInfo](#)
Description of the system.
- 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 [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,
IidcFailed,
StrobeFailed,
TriggerFailed,
PropertyFailed,
PropertyNotPresent,
RegisterFailed,
ReadRegisterFailed,
WriteRegisterFailed,
IsochFailed,
IsochAlreadyStarted,
IsochNotStarted,
IsochStartFailed,
IsochRetrieveBufferFailed,
IsochStopFailed,
IsochSyncFailed,
IsochBandwidthExceeded,
ImageConversionFailed,
ImageLibraryFailure,
BufferTooSmall,
ImageConsistencyError }

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`,
 `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,  
NumberOfPixelFormats = 15 }
```

Pixel formats available for Format7 modes.

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

Bus speeds.

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

Color processing algorithms.

- enum [BayerTileFormat](#) {
 None = 0,
 RGGB,
 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 [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.

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

Possible properties that can be queried from the camera.

Functions

- public delegate void [EnumCallback](#) (System::IntPtr parameter, unsigned int serialNumber)
Bus event callback function prototype.
- public delegate void [ImageEventCallback](#) ([ManagedImage](#)[^] 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 proper external callback for users.

6.2.1 Function Documentation

6.2.1.1 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().

6.2.1.2 protected delegate void FlyCapture2Managed::ImageCallbackDelegate (FlyCapture2::Image * *image*, void * *data*)

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

6.2.1.3 public delegate void FlyCapture2Managed::ImageEventCallback ([ManagedImage](#)[^] *image*)

The external callback that will be used by managed consumers.

6.3 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 7

Class Documentation

7.1 AviOption Struct Reference

Options for saving AVI files.

Public Member Functions

- [AviOption \(\)](#)

Properties

- float [frameRate](#)
Frame rate of the stream.

7.1.1 Detailed Description

Options for saving AVI files.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 [AviOption \(\)](#) [inline]

7.1.3 Property Documentation

7.1.3.1 float [frameRate](#)

Frame rate of the stream.

7.2 CameraControlDialog Class Reference

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

Public Member Functions

- [CameraControlDialog](#) ()
- [~CameraControlDialog](#) ()
- void [Connect](#) ([ManagedCamera](#)[^] camera)
- void [Disconnect](#) ()
- void [Show](#) ()
- void [Hide](#) ()
- bool [IsVisible](#) ()

7.2.1 Detailed Description

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

7.2.2 Constructor & Destructor Documentation

7.2.2.1 [CameraControlDialog](#) ()

7.2.2.2 [~CameraControlDialog](#) ()

7.2.3 Member Function Documentation

7.2.3.1 void [Connect](#) ([FlyCapture2Managed::ManagedCamera](#)[^] camera)

7.2.3.2 void [Disconnect](#) ()

7.2.3.3 void [Hide](#) ()

7.2.3.4 bool [IsVisible](#) ()

7.2.3.5 void [Show](#) ()

7.3 CameraInfo Struct Reference

Camera information.

Properties

- unsigned int [serialNumber](#)
Device serial number.
- [InterfaceType](#) [interfaceType](#)
Interface 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.
- [BayerTileFormat](#) [bayerTileFormat](#)
Bayer tile format.

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.

7.3.1 Detailed Description

Camera information.

7.3.2 Property Documentation

7.3.2.1 BayerTileFormat bayerTileFormat

Bayer tile format.

7.3.2.2 ConfigROM configROM

Configuration ROM data.

7.3.2.3 System:: Net:: IPAddress^ defaultGateway

Default gateway.

7.3.2.4 System:: String^ driverName

Driver name of driver being used.

7.3.2.5 System:: String^ firmwareBuildTime

Firmware build time.

7.3.2.6 System:: String^ firmwareVersion

Firmware version of camera.

7.3.2.7 unsigned int gigEMajorVersion

GigE Vision version.

7.3.2.8 unsigned int gigEMinorVersion

GigE Vision minor version.

7.3.2.9 unsigned int iidcVersion

DCAM version.

7.3.2.10 InterfaceType interfaceType

Interface type.

7.3.2.11 System:: Net:: IPAddress^ ipAddress

IP address.

7.3.2.12 bool isColorCamera

Flag indicating if this is a color camera.

7.3.2.13 System:: Net:: NetworkInformation:: PhysicalAddress^ macAddress

MAC address.

7.3.2.14 BusSpeed maximumBusSpeed

Maximum bus speed.

7.3.2.15 System:: String^ modelName

Device model name.

7.3.2.16 System:: String^ sensorInfo

String detailing the sensor information.

7.3.2.17 System:: String^ sensorResolution

String providing the sensor resolution.

7.3.2.18 unsigned int serialNumber

Device serial number.

7.3.2.19 System:: Net:: IPAddress^ subnetMask

Subnet mask.

7.3.2.20 System:: String^ userDefinedName

User defined name.

7.3.2.21 System:: String^ vendorName

Device vendor name.

7.3.2.22 System:: String^ xmlURL1

XML URL 1.

7.3.2.23 System:: String^ xmlURL2

XML URL 2.

7.4 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.

7.4.1 Detailed Description

A specific camera property.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 CameraProperty () `[inline]`

7.4.2.2 CameraProperty (PropertyType *type*) `[inline]`

7.4.3 Property Documentation

7.4.3.1 bool absControl

Flag controlling absolute mode.

7.4.3.2 float absValue

Floating point value.

7.4.3.3 bool autoManualMode

Flag controlling auto.

7.4.3.4 bool onePush

Flag controlling one push.

7.4.3.5 bool onOff

Flag controlling on/off.

7.4.3.6 bool present

Flag indicating if the property is present.

7.4.3.7 PropertyType type

Property info type.

7.4.3.8 unsigned int valueA

Value A (integer).

7.4.3.9 unsigned int valueB

Value B (integer).

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

7.5 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.

7.5.1 Detailed Description

Information about a specific camera property.

This structure is also used as the `TriggerDelayInfo` structure.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `CameraPropertyInfo ()` `[inline]`

7.5.2.2 `CameraPropertyInfo (PropertyType type)` `[inline]`

7.5.3 Property Documentation

7.5.3.1 `float absMax`

Maximum value (as a floating point value).

7.5.3.2 `float absMin`

Minimum value (as a floating point value).

7.5.3.3 `bool absValSupported`

Flag indicating if absolute mode is supported.

7.5.3.4 `bool autoSupported`

Flag indicating if auto is supported.

7.5.3.5 `bool manualSupported`

Flag indicating if manual is supported.

7.5.3.6 `unsigned int max`

Maximum value (as an integer).

7.5.3.7 `unsigned int min`

Minimum value (as an integer).

7.5.3.8 bool onePushSupported

Flag indicating if one push is supported.

7.5.3.9 bool onOffSupported

Flag indicating if on/off is supported.

7.5.3.10 bool present

Flag indicating if the property is present.

7.5.3.11 bool readOutSupported

Flag indicating if property value can be read out.

7.5.3.12 PropertyType type

Property info type.

7.5.3.13 System::String^ unitAbbr

Abbreviated textual description of units.

7.5.3.14 System::String^ units

Textual description of units.

7.6 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\(\)](#).

7.6.1 Detailed Description

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

7.6.2 Constructor & Destructor Documentation

7.6.2.1 CameraSelectionDialog ()

7.6.2.2 ~CameraSelectionDialog ()

7.6.3 Member Function Documentation

7.6.3.1 array< ManagedPGRGuid ^ > GetSelectedCameraGuids ()

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

Returns:

Array of PGRGuids identifying the selected cameras.

7.6.3.2 bool ShowModal ()

Show the CameraSelectionDlg.

Returns:

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

7.7 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^` `keyword`
Keyword.

7.7.1 Detailed Description

Camera configuration ROM.

7.7.2 Property Documentation

7.7.2.1 unsigned int chipIdHi

Chip ID (high part).

7.7.2.2 unsigned int chipIdLo

Chip ID (low part).

7.7.2.3 System:: String^ keyword

Keyword.

7.7.2.4 unsigned int nodeVendorId

Vendor ID of a node.

7.7.2.5 unsigned int unitSpecId

Unit Spec ID, usually 0xa02d.

7.7.2.6 unsigned int unitSubSWVer

Unit sub software version.

7.7.2.7 unsigned int unitSWVer

Unit software version.

7.7.2.8 unsigned int vendorUniqueInfo0

Vendor unique info 0.

7.7.2.9 unsigned int vendorUniqueInfo1

Vendor unique info 1.

7.7.2.10 unsigned int vendorUniqueInfo2

Vendor unique info 2.

7.7.2.11 unsigned int vendorUniqueInfo3

Vendor unique info 3.

7.8 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](#)

7.8.1 Detailed Description

Properties of the possible embedded image information.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `EmbeddedImageInfo()` `[inline]`

7.8.3 Property Documentation

7.8.3.1 `EmbeddedImageInfoProperty^ brightness`

7.8.3.2 `EmbeddedImageInfoProperty^ exposure`

7.8.3.3 `EmbeddedImageInfoProperty^ frameCounter`

7.8.3.4 `EmbeddedImageInfoProperty^ gain`

7.8.3.5 `EmbeddedImageInfoProperty^ GPIOPinState`

7.8.3.6 `EmbeddedImageInfoProperty^ ROIPosition`

7.8.3.7 `EmbeddedImageInfoProperty^ shutter`

7.8.3.8 `EmbeddedImageInfoProperty^ strobePattern`

7.8.3.9 `EmbeddedImageInfoProperty^ timestamp`

7.8.3.10 `EmbeddedImageInfoProperty^ whiteBalance`

7.9 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.

7.9.1 Detailed Description

Properties of a single embedded image info property.

7.9.2 Property Documentation

7.9.2.1 bool available

Whether this property is available.

7.9.2.2 bool onOff

Whether this property is on or off.

7.10 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](#)
This is the number of notifications per image that will be triggered.
- 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.
- [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.

7.10.1 Detailed Description

Configuration for a camera.

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

7.10.2 Constructor & Destructor Documentation

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

7.10.3 Property Documentation

7.10.3.1 [BusSpeed](#) [asyncBusSpeed](#)

Asynchronous bus speed.

7.10.3.2 BandwidthAllocation bandwidthAllocation

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

7.10.3.3 GrabMode grabMode

Grab mode for the camera.

The default is DROP_FRAMES.

7.10.3.4 int grabTimeout

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

7.10.3.5 BusSpeed isochBusSpeed

Isochronous bus speed.

7.10.3.6 unsigned int numBuffers

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

7.10.3.7 unsigned int numImageNotifications

This is the number of notifications per image that will be triggered.

The default case is 1 notification at the end of a image. Setting this parameter to 2 will result in notifications after the first packet and at the end of image. Setting this parameter to anything more then 2 will divide the notifications equally throughout the buffer. The maximum number of notifications possible is bufferSize/packetSize since notifications need to land on packet boundaries.

7.11 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]

7.11.1 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.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 **FC2Exception ()**

7.11.2.2 **FC2Exception (String[^] *string*)**

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

7.11.2.4 **~FC2Exception ()**

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

7.11.2.6 **FC2Exception (FlyCapture2::Error *error*)** [package]

7.11.3 Property Documentation

7.11.3.1 **ErrorType CauseType** [get]

7.11.3.2 **ErrorType Type** [get]

7.12 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.

7.12.1 Detailed Description

The current version of the library.

7.12.2 Property Documentation

7.12.2.1 unsigned int build

Build version number.

7.12.2.2 unsigned int major

Major version number.

7.12.2.3 unsigned int minor

Minor version number.

7.12.2.4 unsigned int type

Type version number.

7.13 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.

7.13.1 Detailed Description

Format 7 image settings.

7.13.2 Property Documentation

7.13.2.1 unsigned int height

Height of image.

7.13.2.2 Mode mode

Format 7 mode.

7.13.2.3 unsigned int offsetX

Horizontal image offset.

7.13.2.4 unsigned int offsetY

Vertical image offset.

7.13.2.5 PixelFormat pixelFormat

Pixel format of image.

7.13.2.6 unsigned int width

Width of image.

7.14 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 [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.

7.14.1 Detailed Description

Format 7 information for a single mode.

7.14.2 Property Documentation

7.14.2.1 unsigned int imageHStepSize

Horizontal step size for the image.

7.14.2.2 unsigned int imageVStepSize

Vertical step size for the image.

7.14.2.3 unsigned int maxHeight

Maximum image height.

7.14.2.4 unsigned int maxPacketSize

Maximum packet size in bytes for current mode.

7.14.2.5 unsigned int maxWidth

Maximum image width.

7.14.2.6 unsigned int minPacketSize

Minimum packet size in bytes for current mode.

7.14.2.7 Mode mode

Format 7 mode.

7.14.2.8 unsigned int offsetHStepSize

Horizontal step size for the offset.

7.14.2.9 unsigned int offsetVStepSize

Vertical step size for the offset.

7.14.2.10 unsigned int packetSize

Current packet size in bytes.

7.14.2.11 float percentage

Current packet size as a percentage of maximum packet size.

7.14.2.12 unsigned int pixelFormatBitField

Supported pixel formats in a bit field.

7.15 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.

7.15.1 Detailed Description

Format 7 packet information.

7.15.2 Property Documentation

7.15.2.1 unsigned int maxBytesPerPacket

Maximum bytes per packet.

7.15.2.2 unsigned int recommendedBytesPerPacket

Recommended bytes per packet.

7.15.2.3 unsigned int unitBytesPerPacket

Minimum bytes per packet.

7.16 GigEConfig Struct Reference

Configuration for a GigE camera.

Properties

- unsigned int [numChannels](#)
Number of stream channels.
- array< [GigEStreamChannel](#) ^> ^ [channels](#)
Array of stream channel data.

7.16.1 Detailed Description

Configuration for a GigE camera.

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

7.16.2 Property Documentation

7.16.2.1 array< GigEStreamChannel ^> ^ channels

Array of stream channel data.

7.16.2.2 unsigned int numChannels

Number of stream channels.

Read only.

7.17 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.

7.17.1 Detailed Description

Image settings for a GigE camera.

7.17.2 Property Documentation

7.17.2.1 unsigned int height

Height of image.

7.17.2.2 unsigned int offsetX

Horizontal image offset.

7.17.2.3 unsigned int offsetY

Vertical image offset.

7.17.2.4 PixelFormat pixelFormat

Pixel format of image.

7.17.2.5 unsigned int width

Width of image.

7.18 GigEImageSettingsInfo 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.

7.18.1 Detailed Description

Format 7 information for a single mode.

7.18.2 Property Documentation

7.18.2.1 unsigned int imageHStepSize

Horizontal step size for the image.

7.18.2.2 unsigned int imageVStepSize

Vertical step size for the image.

7.18.2.3 unsigned int maxHeight

Maximum image height.

7.18.2.4 unsigned int maxWidth

Maximum image width.

7.18.2.5 unsigned int offsetHStepSize

Horizontal step size for the offset.

7.18.2.6 unsigned int offsetVStepSize

Vertical step size for the offset.

7.18.2.7 unsigned int pixelFormatBitField

Supported pixel formats in a bit field.

7.19 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.

7.19.1 Detailed Description

A GigE property.

7.19.2 Property Documentation

7.19.2.1 bool isReadable

Whether the property is readable.

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

7.19.2.2 bool isWritable

Whether the property is writable.

7.19.2.3 unsigned int max

Maximum value.

7.19.2.4 unsigned int min

Minimum value.

7.19.2.5 GigEPropertyType propType

The type of property.

7.19.2.6 unsigned int value

Current value.

7.20 GigEStreamChannel Struct Reference

Information about a single GigE stream channel.

Properties

- unsigned int [networkInterfaceIndex](#)
Network interface index used (or to use).
- unsigned int [hostPost](#)
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.

7.20.1 Detailed Description

Information about a single GigE stream channel.

7.20.2 Property Documentation

7.20.2.1 System::Net:: IPAddress^ destinationIpAddress

Destination IP address.

It can be a multicast or unicast address.

7.20.2.2 bool doNotFragment

Disable IP fragmentation of packets.

7.20.2.3 unsigned int hostPost

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

7.20.2.4 unsigned int interPacketDelay

Inter packet delay, in timestamp counter units.

7.20.2.5 unsigned int networkInterfaceIndex

Network interface index used (or to use).

7.20.2.6 unsigned int packetSize

Packet size, in bytes.

7.20.2.7 unsigned int sourcePort

Source UDP port of the stream channel.

Read only.

7.21 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.

7.21.1 Detailed Description

Metadata related to an image.

7.21.2 Property Documentation

7.21.2.1 unsigned int embeddedBrightness

Embedded brightness.

7.21.2.2 unsigned int embeddedExposure

Embedded exposure.

7.21.2.3 unsigned int embeddedFrameCounter

Embedded frame counter.

7.21.2.4 unsigned int embeddedGain

Embedded gain.

7.21.2.5 unsigned int embeddedGPIOPinState

Embedded GPIO pin state.

7.21.2.6 unsigned int embeddedROIPosition

Embedded ROI position.

7.21.2.7 unsigned int embeddedShutter

Embedded shutter.

7.21.2.8 unsigned int embeddedStrobePattern

Embedded strobe pattern.

7.21.2.9 unsigned int embeddedTimeStamp

Embedded timestamp.

7.21.2.10 unsigned int embeddedWhiteBalance

Embedded white balance.

7.22 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).

7.22.1 Detailed Description

Options for saving JPEG image.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 [JpegOption \(\)](#) `[inline]`

7.22.3 Property Documentation

7.22.3.1 bool [progressive](#)

Whether to save as a progressive JPEG file.

7.22.3.2 unsigned int [quality](#)

JPEG image quality in range (0-100).

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

7.23 Jpg2Option Struct Reference

Options for saving JPEG2000 image.

Public Member Functions

- [Jpg2Option](#) ()

Properties

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

7.23.1 Detailed Description

Options for saving JPEG2000 image.

7.23.2 Constructor & Destructor Documentation

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

7.23.3 Property Documentation

7.23.3.1 unsigned int [quality](#)

JPEG saving quality in range (1-512).

7.24 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.

7.24.1 Detailed Description

Information about the camera's look up table.

7.24.2 Property Documentation

7.24.2.1 bool enabled

Flag indicating if LUT is enabled.

7.24.2.2 unsigned int inputBitDepth

The input bit depth of the LUT.

7.24.2.3 unsigned int numBanks

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

7.24.2.4 unsigned int numChannels

The number of LUT channels per bank available.

7.24.2.5 unsigned int numEntries

The number of entries in the LUT.

7.24.2.6 unsigned int outputBitDepth

The output bit depth of the LUT.

7.24.2.7 bool supported

Flag indicating if LUT is supported.

7.25 ManagedAVIRecorder Class Reference

[ManagedAVIRecorder](#) provides the functionality for the user to record images to an AVI file.

Public Member Functions

- [ManagedAVIRecorder](#) ()
- [~ManagedAVIRecorder](#) ()
- void [AVIOpen](#) (System::String^ fileName, [AviOption](#)^ option)
Open an AVI file in preparation for writing Images to disk.
- void [AVIAppend](#) ([ManagedImage](#)^ image)
Append an image to the AVI file.
- void [AVIClose](#) ()
Close the AVI file.

7.25.1 Detailed Description

[ManagedAVIRecorder](#) provides the functionality for the user to record images to an AVI file.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 ManagedAVIRecorder ()

7.25.2.2 ~ManagedAVIRecorder ()

7.25.3 Member Function Documentation

7.25.3.1 void AVIAppend (ManagedImage^ image)

Append an image to the AVI file.

Parameters:

image The [ManagedImage](#) to append.

7.25.3.2 void AVIClose ()

Close the AVI file.

See also:

[AVIOpen\(\)](#)

7.25.3.3 void AVIOpen (System::String^ *fileName*, AviOption^ *option*)

Open an AVI file in preparation for writing Images to disk.

The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters:

fileName The filename of the AVI file.

option Options to apply to the AVI file.

See also:

[AVIClose\(\)](#)

7.26 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.
- void [RescanBus](#) ()
Force a rescan of the buses.
- System::IntPtr [RegisterCallback](#) (EnumCallback^ hCallbackDelegate, [ManagedCallbackType](#) 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.

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

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

Protected Member Functions

- [!ManagedBusManager](#) ()

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](#).

7.26.1 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.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 `ManagedBusManager ()`

7.26.2.2 `~ManagedBusManager ()`

7.26.2.3 `!ManagedBusManager ()` [protected]

7.26.3 Member Function Documentation

7.26.3.1 `void ConvertToManagedGuid (FlyCapture2::PGRGuid * pgrGuid, ManagedPGRGuid^ mgdPGRGuid)` [inline, static, package]

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

Parameters:

pgrGuid The native PGRGuid.

mgdPGRGuid The [ManagedPGRGuid](#).

7.26.3.2 `void ConvertToNativeGuid (ManagedPGRGuid^ mgdPGRGuid, FlyCapture2::PGRGuid * pgrGuid)` [inline, static, package]

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

Parameters:

mgdPGRGuid The [ManagedPGRGuid](#).

pgrGuid The native PGRGuid.

7.26.3.3 `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.

7.26.3.4 `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:

guid [ManagedPGRGuid](#) of the camera or the device to cause bus reset.

7.26.3.5 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:

macAddress MAC address of the camera.
ipAddress IP address to set on the camera.
subnetMask Subnet mask to set on the camera.
defaultGateway Default gateway to set on the camera.

7.26.3.6 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:

index Zero based index of camera.

Returns:

Unique [ManagedPGRGuid](#) for the camera.

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

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

Parameters:

ipAddress IP address to get [ManagedPGRGuid](#) for.

Returns:

Unique [ManagedPGRGuid](#) for the camera.

7.26.3.8 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:

serialNumber Serial number of camera.

See also:

[GetCameraFromIndex\(\)](#)

Returns:

Unique [ManagedPGRGuid](#) for the camera.

7.26.3.9 unsigned int GetCameraSerialNumberFromIndex (unsigned int *index*)

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

Parameters:

index Zero based index of desired camera.

Returns:

Serial number of camera.

7.26.3.10 ManagedPGRGuid GetDeviceFromIndex (unsigned int *index*)

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

It uniquely identifies the device specified by the index.

Parameters:

index Zero based index of device.

See also:

[GetNumOfDevices\(\)](#)

Returns:

Unique [ManagedPGRGuid](#) for the device.

7.26.3.11 InterfaceType GetInterfaceTypeFromGuid (ManagedPGRGuid^ *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:

guid The [ManagedPGRGuid](#) to get the interface for.

Returns:

The interface type of the PGRGuid.

7.26.3.12 unsigned int GetNumOfCameras ()

Gets the number of cameras attached to the PC.

Returns:

The number of cameras attached.

7.26.3.13 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.

7.26.3.14 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:

guid [ManagedPGRGuid](#) of the device to read from.

page Page to read from.

port Port to read from.

address Address to read from.

Returns:

Value read from the phy register.

7.26.3.15 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:

hCallbackDelegate Handle to EnumCallback function to receive the callback.

callbackType Type of callback to register for.

parameter Callback parameter to be passed to callback.

See also:

[UnregisterCallback\(\)](#)

Returns:

Unique callback handle used for unregistering callback.

7.26.3.16 void RescanBus ()

Force a rescan of the buses.

This does not trigger a bus reset. However, any current connections to a [ManagedCamera](#) object will be invalidated.

7.26.3.17 void UnregisterCallback (System::IntPtr *callbackHandle*)

Unregister a callback function.

Parameters:

callbackHandle Unique callback handle.

See also:

[RegisterCallback\(\)](#)

7.26.3.18 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:

guid [ManagedPGRGuid](#) of the device to write to.

page Page to write to.

port Port to write to.

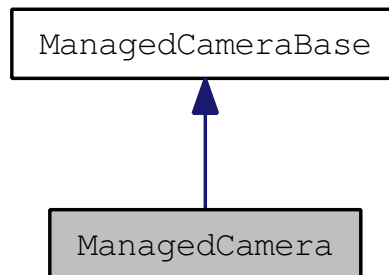
address Address to write to.

regVal Value to write to phy register.

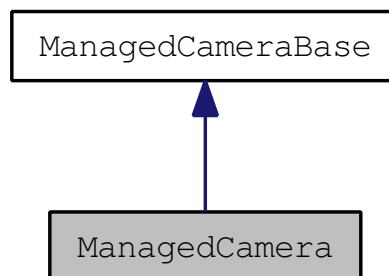
7.27 ManagedCamera Class Reference

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

Inheritance diagram for ManagedCamera:



Collaboration diagram for ManagedCamera:



Public Member Functions

- [ManagedCamera](#) ()
- [~ManagedCamera](#) ()

Protected Member Functions

- [!ManagedCamera](#) ()

DCAM Formats

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

- 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](#)[^] [GetFormat7Info](#) ([Mode](#) mode, bool% supported)

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

- [Format7PacketInfo](#)[^] [ValidateFormat7Settings](#) ([Format7ImageSettings](#)[^] imageSettings, bool% settingsAreValid)

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

- void [GetFormat7Configuration](#) ([Format7ImageSettings](#)[^] imageSettings, unsigned int% packetSize, float% percentSpeed)

Get the current Format7 configuration from the camera.

- void [SetFormat7Configuration](#) ([Format7ImageSettings](#)[^] imageSettings, unsigned int recommendedPacketSize)

Set the current Format7 configuration to the camera.

- void [SetFormat7Configuration](#) ([Format7ImageSettings](#)[^] imageSettings, float recommendedPercentSpeed)

Set the current Format7 configuration to the camera.

7.27.1 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.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 ManagedCamera ()

7.27.2.2 ~ManagedCamera ()

7.27.2.3 !ManagedCamera () [protected]

7.27.3 Member Function Documentation

7.27.3.1 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:

imageSettings Current image settings.

packetSize Current packet size.

percentSpeed Current packet size as a percentage.

See also:

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

7.27.3.2 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:

mode Format7 mode to query.

supported 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.

7.27.3.3 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:

videoMode Current video mode.

frameRate Current frame rate.

See also:

[GetVideoModeAndFrameRateInfo\(\)](#)

[SetVideoModeAndFrameRate\(\)](#)

7.27.3.4 bool GetVideoModeAndFrameRateInfo (VideoMode *videoMode*, FrameRate *frameRate*)

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

Parameters:

videoMode Video mode to check.

frameRate Frame rate to check.

See also:

[GetVideoModeAndFrameRate\(\)](#)

[SetVideoModeAndFrameRate\(\)](#)

Returns:

Whether the video mode and frame rate is supported.

7.27.3.5 void SetFormat7Configuration (Format7ImageSettings^ *imageSettings*, float *recommendedPercentSpeed*)

Set the current Format7 configuration to the camera.

Parameters:

imageSettings Image settings to be written to the camera.

recommendedPercentSpeed Percentage of packet size to be written to the camera.

See also:

[GetFormat7Info\(\)](#)

[ValidateFormat7Settings\(\)](#)

[GetFormat7Configuration\(\)](#)

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

Set the current Format7 configuration to the camera.

Parameters:

imageSettings Image settings to be written to the camera.

recommendedPacketSize Packet size to be written to the camera.

See also:

[GetFormat7Info\(\)](#)

[ValidateFormat7Settings\(\)](#)

[GetFormat7Configuration\(\)](#)

7.27.3.7 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:

videoMode Video mode to set to camera.

frameRate Frame rate to set to camera.

See also:

[GetVideoModeAndFrameRateInfo\(\)](#)

[GetVideoModeAndFrameRate\(\)](#)

7.27.3.8 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:

imageSettings Structure containing the image settings.

settingsAreValid Whether the settings are valid.

See also:

[GetFormat7Info\(\)](#)

[GetFormat7Configuration\(\)](#)

[SetFormat7Configuration\(\)](#)

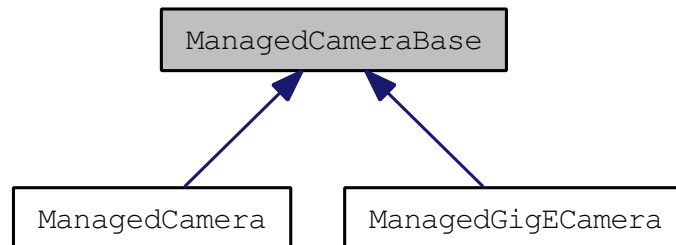
Returns:

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

7.28 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

- [ManagedCameraBase](#) ()
- virtual [~ManagedCameraBase](#) ()

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 currently connected to a physical camera.
- 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 eventNumber)
Retrieves the next image event containing the next part of the image.
- 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 [TriggerModeInfo](#)^ [GetTriggerModeInfo](#) ()
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.

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.

- 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.

Protected Member Functions

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

Protected Attributes

- FlyCapture2::CameraBase * [m_pNativeCamBase](#)
- ImageEventCallback^ [m_externalDelegate](#)
- ImageCallbackDelegate^ [m_internalDelegate](#)

Package Functions

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

7.28.1 Detailed Description

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

7.28.2 Constructor & Destructor Documentation

7.28.2.1 `ManagedCameraBase ()` [inline]

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

7.28.3 Member Function Documentation

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

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

Parameters:

mgdPGRGuid The unique identifier for a specific camera on the PC.

See also:

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

7.28.3.2 `void Disconnect ()` [virtual]

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

This allows another physical camera to be connected to the [ManagedCamera](#) object.

See also:

[Connect\(\)](#)

7.28.3.3 `void EnableLUT (bool on)` [virtual]

Enable or disable LUT functionality on the camera.

Parameters:

on Whether to enable or disable LUT.

See also:

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

7.28.3.4 `void FireSoftwareTrigger (bool broadcast)` [virtual]

Fire the software trigger according to the DCAM specifications.

Parameters:

broadcast Whether the action should be broadcast.

7.28.3.5 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.

7.28.3.6 CameraInfo GetCameraInfo () [virtual]

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

Returns:

Structure containing camera information.

7.28.3.7 FC2Config GetConfiguration () [virtual]

Get the configuration associated with the camera object.

See also:

[SetConfiguration\(\)](#)

Returns:

Current configuration.

7.28.3.8 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.

7.28.3.9 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:

pin Pin to get the direction for.

See also:

[SetGPIOPinDirection\(\)](#)

Returns:

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

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

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

Parameters:

bank The bank to query.

readSupported Whether reading from the bank is supported.

writeSupported Whether writing to the bank is supported.

7.28.3.11 `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:

bank Bank to retrieve.

channel Channel to retrieve.

sizeEntries Number of entries in LUT table to read.

entries Array to store LUT entries in.

See also:

[GetLUTInfo\(\)](#)

[EnableLUT\(\)](#)

[SetLUTChannel\(\)](#)

7.28.3.12 `LutData GetLUTInfo ()` [virtual]

Query if LUT support is available on the camera.

See also:

[EnableLUT\(\)](#)

[GetLUTChannel\(\)](#)

[SetLUTChannel\(\)](#)

Returns:

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

7.28.3.13 unsigned int GetMemoryChannel () [virtual]

Retrieve the current memory channel from the camera.

See also:

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

Returns:

Currently selected memory channel.

7.28.3.14 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.

7.28.3.15 FlyCapture2::CameraBase * GetNativeCamera () [package]

7.28.3.16 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:

type The PropertyType to retrieve information about.

See also:

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

Returns:

Property structure containing property information.

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

Retrieves information about the specified camera property.

Parameters:

type The PropertyType to retrieve information about.

See also:

[GetProperty\(\)](#)

[SetProperty\(\)](#)

Returns:

PropertyInfo structure containing property information.

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

Retrieve current strobe settings from the camera.

Parameters:

source Source pin for strobe information.

See also:

[GetStrobeInfo\(\)](#)

[SetStrobe\(\)](#)

Returns:

[StrobeControl](#) structure containing strobe information.

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

Retrieve strobe information from the camera.

Parameters:

source Source pin for strobe information.

See also:

[GetStrobe\(\)](#)

[SetStrobe\(\)](#)

Returns:

[StrobeInfo](#) structure containing strobe information.

7.28.3.20 TriggerMode GetTriggerMode () [virtual]

Retrieve current trigger settings from the camera.

See also:

[GetTriggerModeInfo\(\)](#)
[SetTriggerMode\(\)](#)

Returns:

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

7.28.3.21 TriggerModeInfo GetTriggerModeInfo () [virtual]

Retrieve trigger information from the camera.

See also:

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

Returns:

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

7.28.3.22 bool IsConnected () [virtual]

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

See also:

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

Returns:

Whether the [ManagedCamera](#) object is connected to a physical camera.

7.28.3.23 void OnNativeCallback (FlyCapture2::Image * *pImage*, void * *pCallbackData*)
[protected]**7.28.3.24 unsigned int ReadRegister (unsigned int *address*)** [virtual]

Read the specified register from the camera.

Parameters:

address DCAM address to be read from.

See also:

[WriteRegister\(\)](#)

Returns:

The register value that is read.

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

Read from the specified register block on the camera.

Parameters:

addressHigh Top 16 bits of the 48 bit absolute address to read from.

addressLow Bottom 32 bits of the 48 bits absolute address to read from.

buffer Array to store read data.

See also:

[WriteRegisterBlock\(\)](#)

7.28.3.26 `void RestoreFromMemoryChannel (unsigned int channel)` [virtual]

Restore the specified current memory channel.

Parameters:

channel Memory channel to restore from.

See also:

[GetMemoryChannel\(\)](#)

[SaveToMemoryChannel\(\)](#)

[GetMemoryChannelInfo\(\)](#)

7.28.3.27 `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 re-queue 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:

image [ManagedImage](#) object to store image data.

See also:

[StartCapture\(\)](#)

[StopCapture\(\)](#)

[WaitForBufferEvent\(\)](#)

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

Save the current settings to the specified current memory channel.

Parameters:

channel Memory channel to save to.

See also:

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

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

Set the LUT bank that will be used.

Parameters:

activeBank The bank to be set as active.

7.28.3.30 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:

hCallbackDelegate 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.

7.28.3.31 void SetConfiguration (FC2Config^ *config*) [virtual]

Set the configuration associated with the camera object.

Parameters:

config Configuration structure to be used.

See also:

[GetConfiguration\(\)](#)

7.28.3.32 void SetEmbeddedImageInfo (EmbeddedImageInfo^ *info*) [virtual]

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

Parameters:

info Structure to be used.

See also:

[GetEmbeddedImageInfo\(\)](#)

7.28.3.33 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:

pin Pin to get the direction for.

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

broadcast Whether the action should be broadcast.

See also:

[GetGPIOPinDirection\(\)](#)

7.28.3.34 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:

pin Pin to get the direction for.

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

See also:

[GetGPIOPinDirection\(\)](#)

7.28.3.35 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:

bank Bank to set.
channel Channel to set.
sizeEntries Number of entries in LUT table to write.
entries Array containing LUT entries to write.

See also:

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

7.28.3.36 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:

camProperty [CameraProperty](#) structure to be used.
broadcast Whether the action should be broadcast.

See also:

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

7.28.3.37 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:

camProperty [CameraProperty](#) structure to be used.

See also:

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

7.28.3.38 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:

strobeControl Structure providing strobe settings.

See also:

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

7.28.3.39 void SetTriggerMode (TriggerMode^ triggerMode) [virtual]

Set the specified trigger settings to the camera.

Parameters:

triggerMode Structure providing trigger mode settings.

See also:

[GetTriggerModeInfo\(\)](#)
[GetTriggerMode\(\)](#)

7.28.3.40 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:

hCallbackDelegate A function to be called when a new image is received.

See also:

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

7.28.3.41 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\(\)](#)

7.28.3.42 void StopCapture () [virtual]

Stops isochronous image transfer and cleans up all associated resources.

See also:

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

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

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

Parameters:

pImage [ManagedImage](#) object to store image data.
eventNumber The event number to wait for.

See also:

[RetrieveBuffer](#)

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

Write to the specified register on the camera.

Parameters:

address DCAM address to be written to.
value The value to be written.
broadcast Whether the action should be broadcast.

See also:

[ReadRegister\(\)](#)

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

Write to the specified register on the camera.

Parameters:

address DCAM address to be written to.
value The value to be written.

See also:

[ReadRegister\(\)](#)

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

Write to the specified register block on the camera.

Parameters:

addressHigh Top 16 bits of the 48 bit absolute address to write to.

addressLow Bottom 32 bits of the 48 bits absolute address to write to.

buffer Array containing data to be written.

See also:

[ReadRegisterBlock\(\)](#)

7.28.4 Member Data Documentation

7.28.4.1 ImageEventCallback ^ m_externalDelegate [protected]

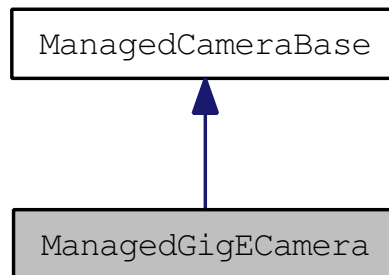
7.28.4.2 ImageCallbackDelegate ^ m_internalDelegate [protected]

7.28.4.3 FlyCapture2::CameraBase* m_pNativeCamBase [protected]

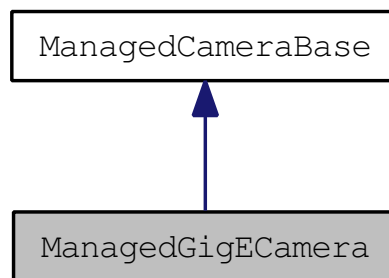
7.29 ManagedGigECamera Class Reference

The GigECamera object represents a physical Gigabit Ethernet camera.

Inheritance diagram for ManagedGigECamera:



Collaboration diagram for ManagedGigECamera:



Public Member Functions

- [ManagedGigECamera](#) ()
- [~ManagedGigECamera](#) ()

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 >^buffer)
Write a GVCP register block.
- void [ReadGVCPRegisterBlock](#) (unsigned int address, array< unsigned int >^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.

7.29.1 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.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 ManagedGigECamera ()

7.29.2.2 ~ManagedGigECamera ()

7.29.2.3 !ManagedGigECamera () [protected]

7.29.3 Member Function Documentation

7.29.3.1 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.

7.29.3.2 void GetGigEImageBinningSettings (unsigned int% *horzBinningValue*, unsigned int% *vertBinningValue*)

Get the current binning settings on the camera.

Parameters:

horzBinningValue Current horizontal binning value.

vertBinningValue Current vertical binning value.

7.29.3.3 GigEImageSettings GetGigEImageSettings ()

Get the current image settings on the camera.

Returns:

Current image settings on camera.

7.29.3.4 GigEImageSettingsInfo GetGigEImageSettingsInfo ()

Get information about the image settings possible on the camera.

Returns:

Image settings information.

7.29.3.5 Mode GetGigEImagingMode ()

Get the current imaging mode on the camera.

Returns:

Current imaging mode on the camera.

7.29.3.6 GigEProperty GetGigEProperty (GigEPropertyType *propType*)

Get the specified [GigEProperty](#).

Returns:

The GigE property to get.

7.29.3.7 GigEStreamChannel GetGigEStreamChannelInfo (unsigned int *channel*)

Get the stream channel information for the specified channel.

Parameters:

channel Channel number to use.

Returns:

Stream channel information for the specified channel.

7.29.3.8 unsigned int GetNumStreamChannels ()

Get the number of stream channels present on the camera.

Returns:

Number of stream channels present.

7.29.3.9 bool QueryGigEImagingMode (Mode *mode*)

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

Parameters:

mode The mode to check.

Returns:

Whether the mode is supported.

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

Read a GVCP memory block.

Parameters:

address GVCP address to be read from.

buffer Array for data to be read into.

length Size of array, in quadlets.

7.29.3.11 unsigned int ReadGVCPRegister (unsigned int *address*)

Read a GVCP register.

Parameters:

address GVCP address to be read from.

Returns:

The value that is read.

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

Read a GVCP register block.

Parameters:

address GVCP address to be read from.

buffer Array for data to be read into.

length Size of array, in quadlets.

7.29.3.13 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:

horzBinningValue Horizontal binning value.

vertBinningValue Vertical binning value.

7.29.3.14 void SetGigEImageSettings (GigEImageSettings^ *settings*)

Set the image settings specified to the camera.

Parameters:

settings Image settings to set to camera.

7.29.3.15 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:

mode Imaging mode to set to the camera.

7.29.3.16 void SetGigEProperty (GigEProperty^ *prop*)

Set the specified [GigEProperty](#).

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

Parameters:

prop The GigE property to set.

7.29.3.17 void SetGigEStreamChannelInfo (unsigned int *channel*, GigEStreamChannel^ *channelInfo*)

Set the stream channel information for the specified channel.

Parameters:

channel Channel number to use.

channelInfo Stream channel information to use for the specified channel.

7.29.3.18 void WriteGVCPMemory (unsigned int *address*, array< unsigned char >^ *buffer*)

Write a GVCP memory block.

Parameters:

address GVCP address to be write to.

buffer Array containing data to be written.

length Size of array, in quadlets.

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

Write a GVCP register.

Parameters:

address GVCP address to be written to.

value The value to be written.

broadcast Whether the action should be broadcast.

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

Write a GVCP register.

Parameters:

address GVCP address to be written to.

value The value to be written.

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

Write a GVCP register block.

Parameters:

address GVCP address to be write to.

buffer Array containing data to be written.

length Size of array, in quadlets.

7.30 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, [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 [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 [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 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.

- [ImageMetadata](#)^ [imageMetadata](#) [get]

Get the metadata associated with the image.

- [TimeStamp](#)^ [timeStamp](#) [get]

Get the timestamp data associated with the image.

- System::Drawing::Bitmap^ [bitmap](#) [get]

Get the internal bitmap representation associated with the image.

7.30.1 Detailed Description

The ManagedImageImage 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.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 ManagedImage ()

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

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

7.30.2.4 ManagedImage (unsigned char * *pData*, unsigned int *dataSize*)

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

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

7.30.2.7 ManagedImage (ManagedImage^ *image*)

7.30.2.8 ~ManagedImage ()

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

7.30.2.10 !ManagedImage () [protected]

7.30.3 Member Function Documentation

7.30.3.1 void CalculateStatistics (ManagedImageStatistics^ *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:

statistics The [ManagedImageStatistics](#) object to hold the statistics.

7.30.3.2 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:

format Output format of the converted image.

destImage Destination image.

7.30.3.3 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:

destImage Destination image.

7.30.3.4 unsigned int DetermineBitsPerPixel (PixelFormat *format*) [static]

Calculate the bits per pixel for the specified pixel format.

Parameters:

format The pixel format.

Returns:

The bits per pixel.

7.30.3.5 FlyCapture2::Image * GetNativeImage () [package]

7.30.3.6 void * GetRawNativeImagePointer ()

7.30.3.7 bool IsNativeImageValid () [package]

7.30.3.8 void ReleaseBuffer ()

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

If no buffer is associated, the function does nothing.

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

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

Parameters:

fileName Filename to save image with.

option Options to use while saving image.

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

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

Parameters:

fileName Filename to save image with.

option Options to use while saving image.

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

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

Parameters:

fileName Filename to save image with.

option Options to use while saving image.

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

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

Parameters:

fileName Filename to save image with.

option Options to use while saving image.

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

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

Parameters:

fileName Filename to save image with.

option Options to use while saving image.

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

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

Parameters:

fileName Filename to save image with.

option Options to use while saving image.

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

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

Parameters:

fileName Filename to save image with.

format File format to save in.

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

Save the image to the specified file name.

Parameters:

fileName Filename to save image with.

7.30.3.17 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:

pData Pointer to the image buffer.

dataSize Size of the image buffer.

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

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

Parameters:

rows Number of rows to set.

cols Number of cols to set.

stride Stride to set.

pixelFormat Pixel format to set.

bayerFormat Bayer tile format to set.

7.30.4 Property Documentation**7.30.4.1 BayerTileFormat bayerTileFormat [get]**

Bayer tile format of the image.

7.30.4.2 System::Drawing::Bitmap^ bitmap [get]

Get the internal bitmap representation associated with the image.

Returns:

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

7.30.4.3 unsigned int bitsPerPixel [get]

Number of bits per pixel in the image.

7.30.4.4 ColorProcessingAlgorithm colorProcessingAlgorithm [get, set]

Color processing algorithm to be used.

7.30.4.5 unsigned int cols [get]

Number of columns in the image.

7.30.4.6 unsigned char* data [get]

Raw pointer to image data.

7.30.4.7 ColorProcessingAlgorithm defaultColorProcessingAlgorithm [static, get, set]

The default color processing algorithm to be used.

7.30.4.8 PixelFormat defaultOutputPixelFormat [static, get, set]

The default output pixel format to be used.

7.30.4.9 ImageMetadata^ imageMetadata [get]

Get the metadata associated with the image.

This includes embedded image information.

Returns:

Metadata associated with the image.

7.30.4.10 PixelFormat pixelFormat [get]

Pixel format of the image.

7.30.4.11 unsigned int rows [get]

Number of rows in the image.

7.30.4.12 unsigned int stride [get]

Number of bytes between rows in the image.

7.30.4.13 TimeStamp^ timeStamp [get]

Get the timestamp data associated with the image.

Returns:

Timestamp data associated with the image.

7.31 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](#) ()

7.31.1 Constructor & Destructor Documentation

7.31.1.1 ManagedImageStatistics ()

7.31.1.2 ~ManagedImageStatistics ()

7.31.2 Member Function Documentation

7.31.2.1 void DisableAll ()

7.31.2.2 void EnableAll ()

7.31.2.3 void EnableGreyOnly ()

7.31.2.4 void EnableHSLOnly ()

7.31.2.5 void EnableRGBOnly ()

7.31.2.6 bool GetChannelStatus (StatisticsChannel *channel*)

7.31.2.7 void GetHistogram (StatisticsChannel *channel*, array< int >^ *histogram*)

7.31.2.8 void GetMean (StatisticsChannel *channel*, float% *mean*)

7.31.2.9 FlyCapture2::ImageStatistics * GetNativeImageStatistics () [package]

7.31.2.10 void GetNumPixelValues (StatisticsChannel *channel*, unsigned int% *numPixelValues*)

7.31.2.11 void GetPixelValueRange (StatisticsChannel *channel*, unsigned int% *pixelValueMin*, unsigned int% *pixelValueMax*)

7.31.2.12 void GetRange (StatisticsChannel *channel*, unsigned int% *min*, unsigned int% *max*)

7.31.2.13 void GetStatistics (StatisticsChannel *channel*, unsigned int% *rangeMin*, unsigned int% *rangeMax*, unsigned int% *pixelValueMin*, unsigned int% *pixelValueMax*, unsigned int% *numPixelValues*, float% *mean*, array< int >^ *histogram*)

7.31.2.14 void SetChannelStatus (StatisticsChannel *channel*, bool *enabled*)

7.32 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.

Package Attributes

- unsigned int [value0](#)
- unsigned int [value1](#)
- unsigned int [value2](#)
- unsigned int [value3](#)

7.32.1 Detailed Description

Managed version of a PGRGuid.

It is used to uniquely identify a camera.

7.32.2 Constructor & Destructor Documentation

7.32.2.1 [ManagedPGRGuid](#) () [inline]

Constructor.

7.32.2.2 ManagedPGRGuid (ManagedPGRGuid^ *managedGuid*) [inline]

Copy constructor.

7.32.2.3 ManagedPGRGuid (ManagedPGRGuid% *managedGuid*) [inline]

Copy constructor.

7.32.3 Member Function Documentation**7.32.3.1 virtual bool Equals (Object^ *obj*)** [inline, override, virtual]**7.32.3.2 virtual int GetHashCode ()** [inline, override, virtual]**7.32.3.3 static bool operator!= (ManagedPGRGuid% *left*, ManagedPGRGuid% *right*)**
[inline, static]

Inequality operator.

7.32.3.4 ManagedPGRGuid % operator= (ManagedPGRGuid% *managedGuid*) [inline]

Assignment operator.

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

Equality operator.

7.32.4 Member Data Documentation**7.32.4.1 unsigned int value0** [package]**7.32.4.2 unsigned int value1** [package]**7.32.4.3 unsigned int value2** [package]**7.32.4.4 unsigned int value3** [package]

7.33 ManagedUtilities Class Reference

Static Public Member Functions

- static void [LaunchBrowser](#) (System::String^ address)
- static void [LaunchHelp](#) (System::String^ fileName)
- static void [LaunchCommand](#) (System::String^ command)

Properties

- static [SystemInfo](#)^ [systemInfo](#) [get]
- static [FC2Version](#)^ [libraryVersion](#) [get]

7.33.1 Member Function Documentation

7.33.1.1 void [LaunchBrowser](#) (System::String^ *address*) [static]

7.33.1.2 void [LaunchCommand](#) (System::String^ *command*) [static]

7.33.1.3 void [LaunchHelp](#) (System::String^ *fileName*) [static]

7.33.2 Property Documentation

7.33.2.1 [FC2Version](#)^ [libraryVersion](#) [static, get]

7.33.2.2 [SystemInfo](#)^ [systemInfo](#) [static, get]

7.34 PgmOption Struct Reference

Options for saving PGM images.

Public Member Functions

- [PgmOption](#) ()

Properties

- bool [binaryFile](#)

Whether to save the PPM as a binary file.

7.34.1 Detailed Description

Options for saving PGM images.

7.34.2 Constructor & Destructor Documentation

7.34.2.1 [PgmOption](#) () `[inline]`

7.34.3 Property Documentation

7.34.3.1 bool [binaryFile](#)

Whether to save the PPM as a binary file.

7.35 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).

7.35.1 Detailed Description

Options for saving PNG images.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 [PngOption](#) () `[inline]`

7.35.3 Property Documentation

7.35.3.1 unsigned int [compressionLevel](#)

Compression level (0-9).

0 is no compression, 9 is best compression.

7.35.3.2 bool [interlaced](#)

Whether to save the PNG as interlaced.

7.36 PpmOption Struct Reference

Options for saving PPM images.

Public Member Functions

- [PpmOption](#) ()

Properties

- bool [binaryFile](#)

Whether to save the PPM as a binary file.

7.36.1 Detailed Description

Options for saving PPM images.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 [PpmOption](#) () `[inline]`

7.36.3 Property Documentation

7.36.3.1 bool [binaryFile](#)

Whether to save the PPM as a binary file.

7.37 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).

7.37.1 Detailed Description

A camera strobe.

7.37.2 Property Documentation

7.37.2.1 float delay

Signal delay (in ms).

7.37.2.2 float duration

Signal duration (in ms).

7.37.2.3 bool onOff

Flag controlling on/off.

7.37.2.4 unsigned int polarity

Signal polarity.

7.37.2.5 unsigned int source

Source value.

7.38 StrobeInfo 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.

7.38.1 Detailed Description

A camera strobe property.

7.38.2 Property Documentation

7.38.2.1 float [maxValue](#)

Maximum value.

7.38.2.2 float [minValue](#)

Minimum value.

7.38.2.3 bool [onOffSupported](#)

Flag indicating if on/off is supported.

7.38.2.4 bool polaritySupported

Flag indicating if polarity is supported.

7.38.2.5 bool present

Presence of strobe.

7.38.2.6 bool readOutSupported

Flag indicating if strobe value can be read out.

7.38.2.7 unsigned int source

Source value.

7.39 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.

7.39.1 Detailed Description

Description of the system.

7.39.2 Property Documentation

7.39.2.1 ByteOrder byteOrder

Byte order of the system.

7.39.2.2 System:: String^ cpuDescription

Detailed description of the CPU.

7.39.2.3 System:: String^ driverList

List of drivers used.

7.39.2.4 System:: String^ gpuDescription

Detailed description of the GPU.

7.39.2.5 System:: String^ libraryList

List of libraries used.

7.39.2.6 unsigned int numCpuCores

Number of cores on all CPUs on the system.

7.39.2.7 System:: String^ osDescription

Detailed description of the operating system.

7.39.2.8 OSType osType

Operating system type as described by OSType.

7.39.2.9 unsigned int screenHeight

Screen resolution height in pixels.

7.39.2.10 unsigned int screenWidth

Screen resolution width in pixels.

7.39.2.11 unsigned int systemMemorySize

Amount of memory available on the system.

7.40 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.

7.40.1 Detailed Description

Options for saving TIFF images.

7.40.2 Member Enumeration Documentation

7.40.2.1 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.

7.40.3 Constructor & Destructor Documentation

7.40.3.1 TiffOption () [inline]

7.40.4 Property Documentation

7.40.4.1 CompressionMethod compression

Compression method to use for encoding TIFF images.

7.41 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.

7.41.1 Detailed Description

Timestamp information.

7.41.2 Property Documentation

7.41.2.1 unsigned int cycleCount

1394 cycle time count.

7.41.2.2 unsigned int cycleOffset

1394 cycle time offset.

7.41.2.3 unsigned int cycleSeconds

1394 cycle time seconds.

7.41.2.4 unsigned int microSeconds

Microseconds.

7.41.2.5 long long seconds

Seconds.

7.42 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 [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 [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::TriggerModeInfo *pNative, [TriggerModeInfo](#)^ mgd)
- static void [ToNative](#) ([TriggerModeInfo](#)^ mgd, FlyCapture2::TriggerModeInfo *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::JPEGOOption *pNative, [JpegOption](#)^ mgd)
- static void [ToNative](#) ([JpegOption](#)^ mgd, FlyCapture2::JPEGOOption *pNative)
- static void [ToMgd](#) (FlyCapture2::JPG2Option *pNative, [Jpg2Option](#)^ mgd)
- static void [ToNative](#) ([Jpg2Option](#)^ mgd, FlyCapture2::JPG2Option *pNative)
- static void [ToNative](#) ([AviOption](#)^ mgd, FlyCapture2::AVIOption *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 [ToMgd](#) (FlyCapture2::GigEStreamChannel *pNative, [GigEStreamChannel](#)^ mgd)
- static void [ToNative](#) ([GigEStreamChannel](#)^ mgd, FlyCapture2::GigEStreamChannel *pNative)

7.42.1 Member Function Documentation

7.42.1.1 void ToMgd (FlyCapture2::GigEStreamChannel * *pNative*, GigEStreamChannel^ *mgd*) [static, package]

7.42.1.2 void ToMgd (FlyCapture2::GigEImageSettings * *pNative*, GigEImageSettings^ *mgd*) [static, package]

7.42.1.3 void ToMgd (FlyCapture2::GigEImageSettingsInfo * *pNative*, GigEImageSettingsInfo^ *mgd*) [static, package]

7.42.1.4 void ToMgd (FlyCapture2::GigEProperty * *pNative*, GigEProperty^ *mgd*) [static, package]

7.42.1.5 void ToMgd (FlyCapture2::MACAddress * *pNative*, System::Net::NetworkInformation::PhysicalAddress^ % *mgd*) [static, package]

7.42.1.6 void ToMgd (FlyCapture2::IPAddress * *pNative*, System::Net::IPAddress^ % *mgd*) [static, package]

7.42.1.7 void ToMgd (FlyCapture2::FC2Version * *pNative*, FC2Version^ *mgd*) [static, package]

7.42.1.8 void ToMgd (FlyCapture2::SystemInfo * *pNative*, SystemInfo^ *mgd*) [static, package]

7.42.1.9 void ToMgd (FlyCapture2::JPG2Option * *pNative*, Jpg2Option^ *mgd*) [static, package]

7.42.1.10 void ToMgd (FlyCapture2::JPEGOption * *pNative*, JpegOption^ *mgd*) [static, package]

7.42.1.11 void ToMgd (FlyCapture2::TIFFOption * *pNative*, TiffOption^ *mgd*) [static, package]

7.42.1.12 void ToMgd (FlyCapture2::PGMOption * *pNative*, PgmOption^ *mgd*) [static, package]

7.42.1.13 void ToMgd (FlyCapture2::PPMOption * *pNative*, PpmOption^ *mgd*) [static, package]

7.42.1.14 void ToMgd (FlyCapture2::PNGOption * *pNative*, PngOption^ *mgd*) [static, package]

7.42.1.15 void ToMgd (FlyCapture2::EmbeddedImageInfo * *pNative*, EmbeddedImageInfo^ *mgd*) [static, package]

7.42.1.16 void ToMgd (FlyCapture2::EmbeddedImageInfoProperty * *pNative*, EmbeddedImageInfoProperty^ *mgd*) [static, package]

7.42.1.17 void ToMgd (FlyCapture2::LUTData * *pNative*, LutData^ *mgd*) [static, package]

7.42.1.18 void ToMgd (FlyCapture2::ImageMetadata * *pNative*, ImageMetadata^ *mgd*) [static, package]

7.42.1.19 void ToMgd (FlyCapture2::CameraInfo * *pNative*, CameraInfo^ *mgd*) [static, package]

Generated by Doxygen 1.9.1
7.42.1.20 void ToMgd (FlyCapture2::ConfigROM * *pNative*, ConfigROM^ *mgd*) [static, package]

7.42.1.21 void ToMgd (FlyCapture2::TimeStamp * *pNative*, TimeStamp^ *mgd*) [static, package]

7.42.1.22 void ToMgd (FlyCapture2::Format7PacketInfo * *pNative*, Format7PacketInfo^

7.43 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.

7.43.1 Detailed Description

A camera trigger.

7.43.2 Property Documentation

7.43.2.1 unsigned int mode

Mode value.

7.43.2.2 bool onOff

Flag controlling on/off.

7.43.2.3 unsigned int parameter

Parameter value.

7.43.2.4 unsigned int polarity

Polarity value.

7.43.2.5 unsigned int source

Source value.

7.44 TriggerModeInfo 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.

7.44.1 Detailed Description

Information about a camera trigger property.

7.44.2 Property Documentation

7.44.2.1 unsigned int modeMask

Mode mask.

7.44.2.2 bool onOffSupported

Flag indicating if on/off is supported.

7.44.2.3 bool polaritySupported

Flag indicating if polarity is supported.

7.44.2.4 bool present

Presence of trigger mode.

7.44.2.5 bool readOutSupported

Flag indicating if trigger value can be read out.

7.44.2.6 bool softwareTriggerSupported

Flag indicating if software trigger is supported.

7.44.2.7 unsigned int sourceMask

Source mask.

7.44.2.8 bool valueReadable

Flag indicating if the value is readable.

Index

- ~CameraControlDialog
 - FlyCapture2Managed::Gui::CameraControlDialog, [42](#)
- ~CameraSelectionDialog
 - FlyCapture2Managed::Gui::CameraSelectionDialog, [52](#)
- ~FC2Exception
 - FlyCapture2Managed::FC2Exception, [61](#)
- ~ManagedAVIRecorder
 - FlyCapture2Managed::ManagedAVIRecorder, [83](#)
- ~ManagedBusManager
 - FlyCapture2Managed::ManagedBusManager, [87](#)
- ~ManagedCamera
 - FlyCapture2Managed::ManagedCamera, [94](#)
- ~ManagedCameraBase
 - FlyCapture2Managed::ManagedCameraBase, [103](#)
- ~ManagedGigECamera
 - FlyCapture2Managed::ManagedGigECamera, [113](#)
- ~ManagedImage
 - FlyCapture2Managed::ManagedImage, [125](#)
- ~ManagedImageStatistics
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- absControl
 - FlyCapture2Managed::CameraProperty, [48](#)
- absMax
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
- absMin
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
- absValSupported
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
- absValue
 - FlyCapture2Managed::CameraProperty, [48](#)
- AdobeDeflate
 - FlyCapture2Managed::TiffOption, [143](#)
- Any
 - Enumerations, [16](#)
- Arrival
 - Enumerations, [20](#)
- asyncBusSpeed
 - FlyCapture2Managed::FC2Config, [58](#)
- AutoExposure
 - Enumerations, [22](#)
- autoManualMode
 - FlyCapture2Managed::CameraProperty, [48](#)
- autoSupported
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
- available
 - FlyCapture2Managed::EmbeddedImageInfoProperty, [57](#)
- AVIAppend
 - FlyCapture2Managed::ManagedAVIRecorder, [83](#)
- AVIClose
 - FlyCapture2Managed::ManagedAVIRecorder, [83](#)
- AVIOpen
 - FlyCapture2Managed::ManagedAVIRecorder, [83](#)
- AviOption
 - FlyCapture2Managed::AviOption, [41](#)
- BandwidthAllocation
 - Enumerations, [15](#)
- bandwidthAllocation
 - FlyCapture2Managed::FC2Config, [58](#)
- BayerTileFormat
 - Enumerations, [16](#)
- bayerTileFormat
 - FlyCapture2Managed::CameraInfo, [44](#)
- BGGR
 - FlyCapture2Managed::ManagedImage, [128](#)
- Enumerations, [16](#)
- BigEndian
 - Enumerations, [17](#)
- binaryFile
 - FlyCapture2Managed::PgmOption, [135](#)
 - FlyCapture2Managed::PpmOption, [137](#)
- bitmap
 - FlyCapture2Managed::ManagedImage, [128](#)
- bitsPerPixel
 - FlyCapture2Managed::ManagedImage, [128](#)
- Blue
 - Enumerations, [23](#)
- Bmp
 - Enumerations, [20](#)
- Brightness
 - Enumerations, [22](#)
- brightness
 - FlyCapture2Managed::EmbeddedImageInfo, [56](#)
- BufferFrames
 - Enumerations, [19](#)
- BufferTooSmall
 - Enumerations, [18](#)
- build
 - FlyCapture2Managed::FC2Version, [62](#)

- BusMasterFailed
 - Enumerations, 18
- BusReset
 - Enumerations, 20
- BusSpeed
 - Enumerations, 16
- ByteOrder
 - Enumerations, 16
- byteOrder
 - FlyCapture2Managed::SystemInfo, 141
- CalculateStatistics
 - FlyCapture2Managed::ManagedImage, 125
- CameraControlDialog
 - FlyCapture2Managed::Gui::CameraControlDialog, 42
- CameraProperty
 - FlyCapture2Managed::CameraProperty, 48
- CameraPropertyInfo
 - FlyCapture2Managed::CameraPropertyInfo, 50
- CameraSelectionDialog
 - FlyCapture2Managed::Gui::CameraSelectionDialog, 52
- CauseType
 - FlyCapture2Managed::FC2Exception, 61
- CcittFax3
 - FlyCapture2Managed::TiffOption, 143
- CcittFax4
 - FlyCapture2Managed::TiffOption, 143
- channels
 - FlyCapture2Managed::GigEConfig, 69
- chipIdHi
 - FlyCapture2Managed::ConfigROM, 53
- chipIdLo
 - FlyCapture2Managed::ConfigROM, 53
- ColorProcessingAlgorithm
 - Enumerations, 17
- colorProcessingAlgorithm
 - FlyCapture2Managed::ManagedImage, 128
- cols
 - FlyCapture2Managed::ManagedImage, 129
- compression
 - FlyCapture2Managed::TiffOption, 144
- compressionLevel
 - FlyCapture2Managed::PngOption, 136
- CompressionMethod
 - FlyCapture2Managed::TiffOption, 143
- configROM
 - FlyCapture2Managed::CameraInfo, 44
- Connect
 - FlyCapture2Managed::Gui::CameraControlDialog, 42
 - FlyCapture2Managed::ManagedCameraBase, 10
- Convert
 - FlyCapture2Managed::ManagedImage, 125
- ConvertToManagedGuid
 - FlyCapture2Managed::ManagedBusManager, 87
- ConvertToNativeGuid
 - FlyCapture2Managed::ManagedBusManager, 87
- cpuDescription
 - FlyCapture2Managed::SystemInfo, 141
- cycleCount
 - FlyCapture2Managed::TimeStamp, 145
- cycleOffset
 - FlyCapture2Managed::TimeStamp, 145
- cycleSeconds
 - FlyCapture2Managed::TimeStamp, 145
- data
 - FlyCapture2Managed::ManagedImage, 129
- Default
 - Enumerations, 17
- defaultColorProcessingAlgorithm
 - FlyCapture2Managed::ManagedImage, 129
- defaultGateway
 - FlyCapture2Managed::CameraInfo, 44
- defaultOutputPixelFormat
 - FlyCapture2Managed::ManagedImage, 129
- Deflate
 - FlyCapture2Managed::TiffOption, 143
- delay
 - FlyCapture2Managed::StrobeControl, 138
- destinationIpAddress
 - FlyCapture2Managed::GigEStreamChannel, 75
- DetermineBitsPerPixel
 - FlyCapture2Managed::ManagedImage, 126
- DisableAll
 - FlyCapture2Managed::ManagedImageStatistics, 131
- Disconnect
 - FlyCapture2Managed::Gui::CameraControlDialog, 42
 - FlyCapture2Managed::ManagedCameraBase, 101
- DiscoverGigECameras
 - FlyCapture2Managed::ManagedBusManager, 87
- DiscoverGigEPacketSize
 - FlyCapture2Managed::ManagedGigECamera, 117
- doNotFragment
 - FlyCapture2Managed::GigEStreamChannel, 75
- driverList
 - FlyCapture2Managed::SystemInfo, 142
- driverName
 - FlyCapture2Managed::CameraInfo, 44
- DropFrames
 - Enumerations, 19
- duration
 - FlyCapture2Managed::StrobeControl, 138
- EdgeSensing
 - Enumerations, 17
- embeddedBrightness
 - FlyCapture2Managed::ImageMetadata, 77
- embeddedExposure

- FlyCapture2Managed::ImageMetadata, 77
- embeddedFrameCounter
 - FlyCapture2Managed::ImageMetadata, 78
- embeddedGain
 - FlyCapture2Managed::ImageMetadata, 78
- embeddedGPIOPinState
 - FlyCapture2Managed::ImageMetadata, 78
- EmbeddedImageInfo
 - FlyCapture2Managed::EmbeddedImageInfo, 56
- embeddedROIPosition
 - FlyCapture2Managed::ImageMetadata, 78
- embeddedShutter
 - FlyCapture2Managed::ImageMetadata, 78
- embeddedStrobePattern
 - FlyCapture2Managed::ImageMetadata, 78
- embeddedTimeStamp
 - FlyCapture2Managed::ImageMetadata, 78
- embeddedWhiteBalance
 - FlyCapture2Managed::ImageMetadata, 78
- EnableAll
 - FlyCapture2Managed::ManagedImageStatistics, 131
- enabled
 - FlyCapture2Managed::LutData, 81
- EnableGreyOnly
 - FlyCapture2Managed::ManagedImageStatistics, 131
- EnableHSLOnly
 - FlyCapture2Managed::ManagedImageStatistics, 131
- EnableLUT
 - FlyCapture2Managed::ManagedCameraBase, 101
- EnableRGBOnly
 - FlyCapture2Managed::ManagedImageStatistics, 131
- EnumCallback
 - FlyCapture2Managed, 39
- Enumerations, 9
 - Any, 16
 - Arrival, 20
 - AutoExposure, 22
 - BandwidthAllocation, 15
 - BayerTileFormat, 16
 - BGGR, 16
 - BigEndian, 17
 - Blue, 23
 - Bmp, 20
 - Brightness, 22
 - BufferFrames, 19
 - BufferTooSmall, 18
 - BusMasterFailed, 18
 - BusReset, 20
 - BusSpeed, 16
 - ByteOrder, 16
 - ColorProcessingAlgorithm, 17
 - Default, 17
 - DropFrames, 19
 - EdgeSensing, 17
 - ErrorType, 17
 - Failed, 17
 - FailedBusMasterConnection, 17
 - FailedGuid, 17
 - Fastest, 16
 - Focus, 23
 - FrameRate, 18, 23
 - FrameRate120, 18
 - FrameRate15, 18
 - FrameRate1_875, 18
 - FrameRate240, 18
 - FrameRate30, 18
 - FrameRate3_75, 18
 - FrameRate60, 18
 - FrameRate7_5, 18
 - FrameRateFormat7, 18
 - FromFileExtension, 20
 - Gain, 23
 - Gamma, 23
 - GBRG, 16
 - GigE, 20
 - GigE_10000Base_T, 16
 - GigE_1000Base_T, 16
 - GigE_100Base_T, 16
 - GigE_10Base_T, 16
 - GigEPropertyType, 18
 - GrabMode, 19
 - GrabTimeout, 19
 - GRBG, 16
 - Green, 23
 - Grey, 23
 - HeartbeatTimeout, 19
 - Heartbeat, 19
 - HQLinear, 17
 - Hue, 23
 - Ieee1394, 20
 - IidcFailed, 18
 - ImageConsistencyError, 18
 - ImageConversionFailed, 18
 - ImageFileFormat, 20
 - ImageLibraryFailure, 18
 - Infinite, 19
 - InitFailed, 17
 - InterfaceType, 20
 - InvalidBuManager, 17
 - InvalidGeneration, 18
 - InvalidMode, 18
 - InvalidPacketSize, 18
 - InvalidParameter, 17
 - InvalidSettings, 17
 - Iris, 23
 - IsochAlreadyStarted, 18
 - IsochBandwidthExceeded, 18
 - IsochFailed, 18

- IsochNotStarted, [18](#)
- IsochRetrieveBufferFailed, [18](#)
- IsochStartFailed, [18](#)
- IsochStopFailed, [18](#)
- IsochSyncFailed, [18](#)
- Jpeg, [20](#)
- Jpeg2000, [20](#)
- Lightness, [23](#)
- LinuxX64, [22](#)
- LinuxX86, [22](#)
- LittleEndian, [17](#)
- LowLevelFailure, [17](#)
- LutFailed, [18](#)
- Mac, [22](#)
- ManagedCallbackType, [20](#)
- MemoryAllocationFailed, [17](#)
- Mode, [20](#)
- Mode0, [21](#)
- Mode1, [21](#)
- Mode10, [21](#)
- Mode11, [21](#)
- Mode12, [21](#)
- Mode13, [21](#)
- Mode14, [21](#)
- Mode15, [21](#)
- Mode16, [21](#)
- Mode17, [21](#)
- Mode18, [21](#)
- Mode19, [21](#)
- Mode2, [21](#)
- Mode20, [21](#)
- Mode21, [21](#)
- Mode22, [21](#)
- Mode23, [21](#)
- Mode24, [21](#)
- Mode25, [21](#)
- Mode26, [21](#)
- Mode27, [21](#)
- Mode28, [21](#)
- Mode29, [21](#)
- Mode3, [21](#)
- Mode30, [21](#)
- Mode31, [21](#)
- Mode4, [21](#)
- Mode5, [21](#)
- Mode6, [21](#)
- Mode7, [21](#)
- Mode8, [21](#)
- Mode9, [21](#)
- NearestNeighbor, [17](#)
- NoColorProcessing, [17](#)
- None, [16](#), [19](#)
- NotConnected, [17](#)
- NotFound, [17](#)
- NotImplemented, [17](#)
- NotInFormat7, [18](#)
- NotInitialized, [17](#)
- NotSupported, [18](#)
- NumberOfFrameRates, [18](#)
- NumberOfModes, [21](#)
- NumberOfPixelFormat, [22](#)
- NumberOfStatisticsChannels, [23](#)
- NumberOfVideoModes, [24](#)
- Off, [16](#)
- Ok, [17](#)
- On, [16](#)
- OSType, [21](#)
- PacketDelay, [19](#)
- PacketSize, [19](#)
- Pan, [23](#)
- Pgm, [20](#)
- PixelFormat, [22](#)
- PixelFormat411Yuv8, [22](#)
- PixelFormat422Yuv8, [22](#)
- PixelFormat444Yuv8, [22](#)
- PixelFormatBgr, [22](#)
- PixelFormatBgru, [22](#)
- PixelFormatMono12, [22](#)
- PixelFormatMono16, [22](#)
- PixelFormatMono8, [22](#)
- PixelFormatRaw12, [22](#)
- PixelFormatRaw16, [22](#)
- PixelFormatRaw8, [22](#)
- PixelFormatRgb, [22](#)
- PixelFormatRgb16, [22](#)
- PixelFormatRgb8, [22](#)
- PixelFormatRgba, [22](#)
- PixelFormatSignedMono16, [22](#)
- PixelFormatSignedRgb16, [22](#)
- Png, [20](#)
- Ppm, [20](#)
- PropertyFailed, [18](#)
- PropertyNotPresent, [18](#)
- PropertyType, [22](#)
- Raw, [20](#)
- ReadRegisterFailed, [18](#)
- Red, [23](#)
- RegisterFailed, [18](#)
- Removal, [20](#)
- RGBA, [16](#)
- Rigorous, [17](#)
- S100, [16](#)
- S1600, [16](#)
- S200, [16](#)
- S3200, [16](#)
- S400, [16](#)
- S480, [16](#)
- S800, [16](#)

- Saturation, [23](#)
- Sharpness, [23](#)
- Shutter, [23](#)
- StatisticsChannel, [23](#)
- StrobeFailed, [18](#)
- Temperature, [23](#)
- Tiff, [20](#)
- Tilt, [23](#)
- Timeout, [18](#)
- TriggerDelay, [23](#)
- TriggerFailed, [18](#)
- TriggerMode, [23](#)
- Undefined, [17](#)
- Unknown, [16](#), [20](#)
- UnknownOS, [22](#)
- Unspecified, [16](#), [19](#), [23](#)
- Unsupported, [16](#)
- Usb2, [20](#)
- VideoMode, [23](#)
- VideoMode1024x768Rgb, [24](#)
- VideoMode1024x768Y16, [24](#)
- VideoMode1024x768Y8, [24](#)
- VideoMode1024x768Yuv422, [24](#)
- VideoMode1280x960Rgb, [24](#)
- VideoMode1280x960Y16, [24](#)
- VideoMode1280x960Y8, [24](#)
- VideoMode1280x960Yuv422, [24](#)
- VideoMode1600x1200Rgb, [24](#)
- VideoMode1600x1200Y16, [24](#)
- VideoMode1600x1200Y8, [24](#)
- VideoMode1600x1200Yuv422, [24](#)
- VideoMode160x120Yuv444, [24](#)
- VideoMode320x240Yuv422, [24](#)
- VideoMode640x480Rgb, [24](#)
- VideoMode640x480Y16, [24](#)
- VideoMode640x480Y8, [24](#)
- VideoMode640x480Yuv411, [24](#)
- VideoMode640x480Yuv422, [24](#)
- VideoMode800x600Rgb, [24](#)
- VideoMode800x600Y16, [24](#)
- VideoMode800x600Y8, [24](#)
- VideoMode800x600Yuv422, [24](#)
- VideoModeFormat7, [24](#)
- WhiteBalance, [23](#)
- WindowsX64, [22](#)
- WindowsX86, [22](#)
- WriteRegisterFailed, [18](#)
- Zoom, [23](#)
- Equals
 - FlyCapture2Managed::ManagedPGRGuid, [133](#)
- ErrorType
 - Enumerations, [17](#)
- exposure
 - FlyCapture2Managed::EmbeddedImageInfo, [56](#)
- Failed
 - Enumerations, [17](#)
- FailedBusMasterConnection
 - Enumerations, [17](#)
- FailedGuid
 - Enumerations, [17](#)
- Fastest
 - Enumerations, [16](#)
- FC2Config
 - FlyCapture2Managed::FC2Config, [58](#)
- FC2Exception
 - FlyCapture2Managed::FC2Exception, [61](#)
- FireBusReset
 - FlyCapture2Managed::ManagedBusManager, [87](#)
- FireSoftwareTrigger
 - FlyCapture2Managed::ManagedCameraBase, [101](#)
- firmwareBuildTime
 - FlyCapture2Managed::CameraInfo, [45](#)
- firmwareVersion
 - FlyCapture2Managed::CameraInfo, [45](#)
- FlyCapture2, [29](#)
- FlyCapture2Managed, [30](#)
 - EnumCallback, [39](#)
 - ImageCallbackDelegate, [39](#)
 - ImageEventCallback, [39](#)
- FlyCapture2Managed::AviOption, [41](#)
 - AviOption, [41](#)
 - frameRate, [41](#)
- FlyCapture2Managed::CameraInfo, [43](#)
 - bayerTileFormat, [44](#)
 - configROM, [44](#)
 - defaultGateway, [44](#)
 - driverName, [44](#)
 - firmwareBuildTime, [45](#)
 - firmwareVersion, [45](#)
 - gigEMajorVersion, [45](#)
 - gigEMinorVersion, [45](#)
 - iidcVersion, [45](#)
 - interfaceType, [45](#)
 - ipAddress, [45](#)
 - isColorCamera, [45](#)
 - macAddress, [45](#)
 - maximumBusSpeed, [45](#)
 - modelName, [45](#)
 - sensorInfo, [46](#)
 - sensorResolution, [46](#)
 - serialNumber, [46](#)
 - subnetMask, [46](#)
 - userDefinedName, [46](#)
 - vendorName, [46](#)
 - xmlURL1, [46](#)
 - xmlURL2, [46](#)
- FlyCapture2Managed::CameraProperty, [47](#)
 - absControl, [48](#)

- absValue, 48
- autoManualMode, 48
- CameraProperty, 48
- onePush, 48
- onOff, 48
- present, 48
- type, 48
- valueA, 48
- valueB, 48
- FlyCapture2Managed::CameraPropertyInfo, 49
 - absMax, 50
 - absMin, 50
 - absValSupported, 50
 - autoSupported, 50
 - CameraPropertyInfo, 50
 - manualSupported, 50
 - max, 50
 - min, 50
 - onePushSupported, 50
 - onOffSupported, 51
 - present, 51
 - readOutSupported, 51
 - type, 51
 - unitAbbr, 51
 - units, 51
- FlyCapture2Managed::ConfigROM, 53
 - chipIdHi, 53
 - chipIdLo, 53
 - keyword, 54
 - nodeVendorId, 54
 - unitSpecId, 54
 - unitSubSWVer, 54
 - unitSWVer, 54
 - vendorUniqueInfo0, 54
 - vendorUniqueInfo1, 54
 - vendorUniqueInfo2, 54
 - vendorUniqueInfo3, 54
- FlyCapture2Managed::EmbeddedImageInfo, 55
 - brightness, 56
 - EmbeddedImageInfo, 56
 - exposure, 56
 - frameCounter, 56
 - gain, 56
 - GPIOPinState, 56
 - ROIPosition, 56
 - shutter, 56
 - strobePattern, 56
 - timestamp, 56
 - whiteBalance, 56
- FlyCapture2Managed::EmbeddedImageInfoProperty, 57
 - available, 57
 - onOff, 57
- FlyCapture2Managed::FC2Config, 58
 - asyncBusSpeed, 58
 - bandwidthAllocation, 58
 - FC2Config, 58
 - grabMode, 59
 - grabTimeout, 59
 - isochBusSpeed, 59
 - numBuffers, 59
 - numImageNotifications, 59
- FlyCapture2Managed::FC2Exception, 60
 - ~FC2Exception, 61
 - CauseType, 61
 - FC2Exception, 61
 - Type, 61
- FlyCapture2Managed::FC2Version, 62
 - build, 62
 - major, 62
 - minor, 62
 - type, 62
- FlyCapture2Managed::Format7ImageSettings, 63
 - height, 63
 - mode, 63
 - offsetX, 63
 - offsetY, 63
 - pixelFormat, 63
 - width, 64
- FlyCapture2Managed::Format7Info, 65
 - imageHStepSize, 66
 - imageVStepSize, 66
 - maxHeight, 66
 - maxPacketSize, 66
 - maxWidth, 66
 - minPacketSize, 66
 - mode, 66
 - offsetHStepSize, 66
 - offsetVStepSize, 66
 - packetSize, 66
 - percentage, 66
 - pixelFormatBitField, 66
- FlyCapture2Managed::Format7PacketInfo, 68
 - maxBytesPerPacket, 68
 - recommendedBytesPerPacket, 68
 - unitBytesPerPacket, 68
- FlyCapture2Managed::GigEConfig, 69
 - channels, 69
 - numChannels, 69
- FlyCapture2Managed::GigEImageSettings, 70
 - height, 70
 - offsetX, 70
 - offsetY, 70
 - pixelFormat, 70
 - width, 70
- FlyCapture2Managed::GigEImageSettingsInfo, 71
 - imageHStepSize, 71
 - imageVStepSize, 71
 - maxHeight, 71

- maxWidth, 71
- offsetHStepSize, 72
- offsetVStepSize, 72
- pixelFormatBitField, 72
- FlyCapture2Managed::GigEProperty, 73
 - isReadable, 73
 - isWritable, 73
 - max, 73
 - min, 73
 - propType, 73
 - value, 74
- FlyCapture2Managed::GigEStreamChannel, 75
 - destinationIpAddress, 75
 - doNotFragment, 75
 - hostPort, 75
 - interPacketDelay, 75
 - networkInterfaceIndex, 76
 - packetSize, 76
 - sourcePort, 76
- FlyCapture2Managed::Gui, 40
- FlyCapture2Managed::Gui::CameraControlDialog, 42
 - ~CameraControlDialog, 42
 - CameraControlDialog, 42
 - Connect, 42
 - Disconnect, 42
 - Hide, 42
 - IsVisible, 42
 - Show, 42
- FlyCapture2Managed::Gui::CameraSelectionDialog, 52
 - ~CameraSelectionDialog, 52
 - CameraSelectionDialog, 52
 - GetSelectedCameraGuids, 52
 - ShowModal, 52
- FlyCapture2Managed::ImageMetadata, 77
 - embeddedBrightness, 77
 - embeddedExposure, 77
 - embeddedFrameCounter, 78
 - embeddedGain, 78
 - embeddedGPIOPinState, 78
 - embeddedROIPosition, 78
 - embeddedShutter, 78
 - embeddedStrobePattern, 78
 - embeddedTimeStamp, 78
 - embeddedWhiteBalance, 78
- FlyCapture2Managed::JpegOption, 79
 - JpegOption, 79
 - progressive, 79
 - quality, 79
- FlyCapture2Managed::Jpg2Option, 80
 - Jpg2Option, 80
 - quality, 80
- FlyCapture2Managed::LutData, 81
 - enabled, 81
 - inputBitDepth, 81
 - numBanks, 81
 - numChannels, 81
 - numEntries, 82
 - outputBitDepth, 82
 - supported, 82
- FlyCapture2Managed::ManagedAVIRecorder, 83
 - ~ManagedAVIRecorder, 83
 - AVIAppend, 83
 - AVIClose, 83
 - AVIOpen, 83
 - ManagedAVIRecorder, 83
- FlyCapture2Managed::ManagedBusManager, 85
 - ~ManagedBusManager, 87
 - ConvertToManagedGuid, 87
 - ConvertToNativeGuid, 87
 - DiscoverGigECameras, 87
 - FireBusReset, 87
 - ForceIPAddressToCamera, 87
 - GetCameraFromIndex, 88
 - GetCameraFromIPAddress, 88
 - GetCameraFromSerialNumber, 88
 - GetCameraSerialNumberFromIndex, 89
 - GetDeviceFromIndex, 89
 - GetInterfaceTypeFromGuid, 89
 - GetNumOfCameras, 89
 - GetNumOfDevices, 90
 - ManagedBusManager, 87
 - ReadPhyRegister, 90
 - RegisterCallback, 90
 - RescanBus, 90
 - UnregisterCallback, 91
 - WritePhyRegister, 91
- FlyCapture2Managed::ManagedCamera, 92
 - ~ManagedCamera, 94
 - GetFormat7Configuration, 94
 - GetFormat7Info, 94
 - GetVideoModeAndFrameRate, 94
 - GetVideoModeAndFrameRateInfo, 95
 - ManagedCamera, 94
 - SetFormat7Configuration, 95
 - SetVideoModeAndFrameRate, 96
 - ValidateFormat7Settings, 96
- FlyCapture2Managed::ManagedCameraBase, 97
 - ~ManagedCameraBase, 101
 - Connect, 101
 - Disconnect, 101
 - EnableLUT, 101
 - FireSoftwareTrigger, 101
 - GetActiveLUTBank, 101
 - GetCameraInfo, 102
 - GetConfiguration, 102
 - GetEmbeddedImageInfo, 102
 - GetGPIOPinDirection, 102
 - GetLUTBankInfo, 103

- GetLUTChannel, 103
- GetLUTInfo, 103
- GetMemoryChannel, 103
- GetMemoryChannelInfo, 104
- GetNativeCamera, 104
- GetProperty, 104
- GetPropertyInfo, 104
- GetStrobe, 105
- GetStrobeInfo, 105
- GetTriggerMode, 105
- GetTriggerModeInfo, 106
- IsConnected, 106
- m_externalDelegate, 113
- m_internalDelegate, 113
- m_pNativeCamBase, 113
- ManagedCameraBase, 101
- OnNativeCallback, 106
- ReadRegister, 106
- ReadRegisterBlock, 106
- RestoreFromMemoryChannel, 107
- RetrieveBuffer, 107
- SaveToMemoryChannel, 107
- SetActiveLUTBank, 108
- SetCallback, 108
- SetConfiguration, 108
- SetEmbeddedImageInfo, 108
- SetGPIOPinDirection, 109
- SetLUTChannel, 109
- SetProperty, 110
- SetStrobe, 110
- SetTriggerMode, 111
- StartCapture, 111
- StopCapture, 111
- WaitForBufferEvent, 112
- WriteRegister, 112
- WriteRegisterBlock, 112
- FlyCapture2Managed::ManagedGigECamera, 114
 - ~ManagedGigECamera, 117
 - DiscoverGigEPacketSize, 117
 - GetGigEImageBinningSettings, 117
 - GetGigEImageSettings, 117
 - GetGigEImageSettingsInfo, 117
 - GetGigEImagingMode, 117
 - GetGigEProperty, 118
 - GetGigEStreamChannelInfo, 118
 - GetNumStreamChannels, 118
 - ManagedGigECamera, 117
 - QueryGigEImagingMode, 118
 - ReadGVCPMemory, 118
 - ReadGVCPRegister, 119
 - ReadGVCPRegisterBlock, 119
 - SetGigEImageBinningSettings, 119
 - SetGigEImageSettings, 119
 - SetGigEImagingMode, 119
 - SetGigEProperty, 120
 - SetGigEStreamChannelInfo, 120
 - WriteGVCPMemory, 120
 - WriteGVCPRegister, 120
 - WriteGVCPRegisterBlock, 121
- FlyCapture2Managed::ManagedImage, 122
 - ~ManagedImage, 125
 - bayerTileFormat, 128
 - bitmap, 128
 - bitsPerPixel, 128
 - CalculateStatistics, 125
 - colorProcessingAlgorithm, 128
 - cols, 129
 - Convert, 125
 - data, 129
 - defaultColorProcessingAlgorithm, 129
 - defaultOutputPixelFormat, 129
 - DetermineBitsPerPixel, 126
 - GetNativeImage, 126
 - GetRawNativeImagePointer, 126
 - imageMetadata, 129
 - IsNativeImageValid, 126
 - ManagedImage, 125
 - pixelFormat, 129
 - ReleaseBuffer, 126
 - rows, 129
 - Save, 126, 127
 - SetData, 128
 - SetDimensions, 128
 - stride, 129
 - timeStamp, 129
- FlyCapture2Managed::ManagedImageStatistics, 130
 - ~ManagedImageStatistics, 131
 - DisableAll, 131
 - EnableAll, 131
 - EnableGreyOnly, 131
 - EnableHSLOnly, 131
 - EnableRGBOnly, 131
 - GetChannelStatus, 131
 - GetHistogram, 131
 - GetMean, 131
 - GetNativeImageStatistics, 131
 - GetNumPixelValues, 131
 - GetPixelValueRange, 131
 - GetRange, 131
 - GetStatistics, 131
 - ManagedImageStatistics, 131
 - SetChannelStatus, 131
- FlyCapture2Managed::ManagedPGRGuid, 132
 - Equals, 133
 - GetHashCode, 133
 - ManagedPGRGuid, 132, 133
 - operator=, 133
 - operator==, 133

- value0, [133](#)
- value1, [133](#)
- value2, [133](#)
- value3, [133](#)
- FlyCapture2Managed::ManagedUtilities, [134](#)
 - LaunchBrowser, [134](#)
 - LaunchCommand, [134](#)
 - LaunchHelp, [134](#)
 - libraryVersion, [134](#)
 - systemInfo, [134](#)
- FlyCapture2Managed::PgmOption, [135](#)
 - binaryFile, [135](#)
 - PgmOption, [135](#)
- FlyCapture2Managed::PngOption, [136](#)
 - compressionLevel, [136](#)
 - interlaced, [136](#)
 - PngOption, [136](#)
- FlyCapture2Managed::PpmOption, [137](#)
 - binaryFile, [137](#)
 - PpmOption, [137](#)
- FlyCapture2Managed::StrobeControl, [138](#)
 - delay, [138](#)
 - duration, [138](#)
 - onOff, [138](#)
 - polarity, [138](#)
 - source, [138](#)
- FlyCapture2Managed::StrobeInfo, [139](#)
 - maxValue, [139](#)
 - minValue, [139](#)
 - onOffSupported, [139](#)
 - polaritySupported, [139](#)
 - present, [140](#)
 - readOutSupported, [140](#)
 - source, [140](#)
- FlyCapture2Managed::SystemInfo, [141](#)
 - byteOrder, [141](#)
 - cpuDescription, [141](#)
 - driverList, [142](#)
 - gpuDescription, [142](#)
 - libraryList, [142](#)
 - numCpuCores, [142](#)
 - osDescription, [142](#)
 - osType, [142](#)
 - screenHeight, [142](#)
 - screenWidth, [142](#)
 - systemMemorySize, [142](#)
- FlyCapture2Managed::TiffOption, [143](#)
 - AdobeDeflate, [143](#)
 - CcittFax3, [143](#)
 - CcittFax4, [143](#)
 - compression, [144](#)
 - CompressionMethod, [143](#)
 - Deflate, [143](#)
 - Jpeg, [143](#)
 - Lzw, [143](#)
 - None, [143](#)
 - PackBits, [143](#)
 - TiffOption, [144](#)
- FlyCapture2Managed::TimeStamp, [145](#)
 - cycleCount, [145](#)
 - cycleOffset, [145](#)
 - cycleSeconds, [145](#)
 - microSeconds, [145](#)
 - seconds, [145](#)
- FlyCapture2Managed::Translate, [146](#)
 - ToMgd, [149](#)
 - ToNative, [149](#)
 - translate, [149](#)
- FlyCapture2Managed::TriggerMode, [150](#)
 - mode, [150](#)
 - onOff, [150](#)
 - parameter, [150](#)
 - polarity, [150](#)
 - source, [150](#)
- FlyCapture2Managed::TriggerModeInfo, [151](#)
 - modeMask, [151](#)
 - onOffSupported, [151](#)
 - polaritySupported, [151](#)
 - present, [151](#)
 - readOutSupported, [151](#)
 - softwareTriggerSupported, [152](#)
 - sourceMask, [152](#)
 - valueReadable, [152](#)
- Focus
 - Enumerations, [23](#)
- ForceIPAddressToCamera
 - FlyCapture2Managed::ManagedBusManager, [87](#)
- frameCounter
 - FlyCapture2Managed::EmbeddedImageInfo, [56](#)
- FrameRate
 - Enumerations, [18, 23](#)
- frameRate
 - FlyCapture2Managed::AviOption, [41](#)
- FrameRate120
 - Enumerations, [18](#)
- FrameRate15
 - Enumerations, [18](#)
- FrameRate1_875
 - Enumerations, [18](#)
- FrameRate240
 - Enumerations, [18](#)
- FrameRate30
 - Enumerations, [18](#)
- FrameRate3_75
 - Enumerations, [18](#)
- FrameRate60
 - Enumerations, [18](#)
- FrameRate7_5

- Enumerations, [18](#)
- FrameRateFormat7
 - Enumerations, [18](#)
- FromFileExtension
 - Enumerations, [20](#)
- Gain
 - Enumerations, [23](#)
- gain
 - FlyCapture2Managed::EmbeddedImageInfo, [56](#)
- Gamma
 - Enumerations, [23](#)
- GBRG
 - Enumerations, [16](#)
- GetActiveLUTBank
 - FlyCapture2Managed::ManagedCameraBase, [101](#)
- GetCameraFromIndex
 - FlyCapture2Managed::ManagedBusManager, [88](#)
- GetCameraFromIPAddress
 - FlyCapture2Managed::ManagedBusManager, [88](#)
- GetCameraFromSerialNumber
 - FlyCapture2Managed::ManagedBusManager, [88](#)
- GetCameraInfo
 - FlyCapture2Managed::ManagedCameraBase, [102](#)
- GetCameraSerialNumberFromIndex
 - FlyCapture2Managed::ManagedBusManager, [89](#)
- GetChannelStatus
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetConfiguration
 - FlyCapture2Managed::ManagedCameraBase, [102](#)
- GetDeviceFromIndex
 - FlyCapture2Managed::ManagedBusManager, [89](#)
- GetEmbeddedImageInfo
 - FlyCapture2Managed::ManagedCameraBase, [102](#)
- GetFormat7Configuration
 - FlyCapture2Managed::ManagedCamera, [94](#)
- GetFormat7Info
 - FlyCapture2Managed::ManagedCamera, [94](#)
- GetGigEImageBinningSettings
 - FlyCapture2Managed::ManagedGigECamera, [117](#)
- GetGigEImageSettings
 - FlyCapture2Managed::ManagedGigECamera, [117](#)
- GetGigEImageSettingsInfo
 - FlyCapture2Managed::ManagedGigECamera, [117](#)
- GetGigEImagingMode
 - FlyCapture2Managed::ManagedGigECamera, [117](#)
- GetGigEProperty
 - FlyCapture2Managed::ManagedGigECamera, [118](#)
- GetGigEStreamChannelInfo
 - FlyCapture2Managed::ManagedGigECamera, [118](#)
- GetGPIOPinDirection
 - FlyCapture2Managed::ManagedCameraBase, [102](#)
- GetHashCode
 - FlyCapture2Managed::ManagedPGRGuid, [133](#)
- GetHistogram
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetInterfaceTypeFromGuid
 - FlyCapture2Managed::ManagedBusManager, [89](#)
- GetLUTBankInfo
 - FlyCapture2Managed::ManagedCameraBase, [103](#)
- GetLUTChannel
 - FlyCapture2Managed::ManagedCameraBase, [103](#)
- GetLUTInfo
 - FlyCapture2Managed::ManagedCameraBase, [103](#)
- GetMean
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetMemoryChannel
 - FlyCapture2Managed::ManagedCameraBase, [103](#)
- GetMemoryChannelInfo
 - FlyCapture2Managed::ManagedCameraBase, [104](#)
- GetNativeCamera
 - FlyCapture2Managed::ManagedCameraBase, [104](#)
- GetNativeImage
 - FlyCapture2Managed::ManagedImage, [126](#)
- GetNativeImageStatistics
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetNumOfCameras
 - FlyCapture2Managed::ManagedBusManager, [89](#)
- GetNumOfDevices
 - FlyCapture2Managed::ManagedBusManager, [90](#)
- GetNumPixelValues
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetNumStreamChannels
 - FlyCapture2Managed::ManagedGigECamera, [118](#)
- GetPixelValueRange
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetProperty
 - FlyCapture2Managed::ManagedCameraBase, [104](#)
- GetPropertyInfo
 - FlyCapture2Managed::ManagedCameraBase, [104](#)
- GetRange
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetRawNativeImagePointer
 - FlyCapture2Managed::ManagedImage, [126](#)
- GetSelectedCameraGuids
 - FlyCapture2Managed::Gui::CameraSelectionDialog, [52](#)
- GetStatistics
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- GetStrobe
 - FlyCapture2Managed::ManagedCameraBase, [105](#)
- GetStrobeInfo
 - FlyCapture2Managed::ManagedCameraBase, [105](#)
- GetTriggerMode
 - FlyCapture2Managed::ManagedCameraBase, [105](#)
- GetTriggerModeInfo
 - FlyCapture2Managed::ManagedCameraBase, [106](#)
- GetVideoModeAndFrameRate
 - FlyCapture2Managed::ManagedCamera, [94](#)

- GetVideoModeAndFrameRateInfo
 - FlyCapture2Managed::ManagedCamera, 95
- GigE
 - Enumerations, 20
- GigE_10000Base_T
 - Enumerations, 16
- GigE_1000Base_T
 - Enumerations, 16
- GigE_100Base_T
 - Enumerations, 16
- GigE_10Base_T
 - Enumerations, 16
- gigEMajorVersion
 - FlyCapture2Managed::CameraInfo, 45
- gigEMinorVersion
 - FlyCapture2Managed::CameraInfo, 45
- GigEPropertyType
 - Enumerations, 18
- GPIOPinState
 - FlyCapture2Managed::EmbeddedImageInfo, 56
- gpuDescription
 - FlyCapture2Managed::SystemInfo, 142
- GrabMode
 - Enumerations, 19
- grabMode
 - FlyCapture2Managed::FC2Config, 59
- GrabTimeout
 - Enumerations, 19
- grabTimeout
 - FlyCapture2Managed::FC2Config, 59
- GRBG
 - Enumerations, 16
- Green
 - Enumerations, 23
- Grey
 - Enumerations, 23
- HeartbeatTimeout
 - Enumerations, 19
- Heartbeat
 - Enumerations, 19
- height
 - FlyCapture2Managed::Format7ImageSettings, 63
 - FlyCapture2Managed::GigEImageSettings, 70
- Hide
 - FlyCapture2Managed::Gui::CameraControlDialog, 42
- hostPost
 - FlyCapture2Managed::GigEStreamChannel, 75
- HQLinear
 - Enumerations, 17
- Hue
 - Enumerations, 23
- Ieee1394
 - Enumerations, 20
- IidcFailed
 - Enumerations, 18
- iidcVersion
 - FlyCapture2Managed::CameraInfo, 45
- Image saving structures., 27
- ImageCallbackDelegate
 - FlyCapture2Managed, 39
- ImageConsistencyError
 - Enumerations, 18
- ImageConversionFailed
 - Enumerations, 18
- ImageEventCallback
 - FlyCapture2Managed, 39
- ImageFileFormat
 - Enumerations, 20
- imageHStepSize
 - FlyCapture2Managed::Format7Info, 66
 - FlyCapture2Managed::GigEImageSettingsInfo, 71
- ImageLibraryFailure
 - Enumerations, 18
- imageMetadata
 - FlyCapture2Managed::ManagedImage, 129
- imageVStepSize
 - FlyCapture2Managed::Format7Info, 66
 - FlyCapture2Managed::GigEImageSettingsInfo, 71
- Infinite
 - Enumerations, 19
- InitFailed
 - Enumerations, 17
- inputBitDepth
 - FlyCapture2Managed::LutData, 81
- InterfaceType
 - Enumerations, 20
- interfaceType
 - FlyCapture2Managed::CameraInfo, 45
- interlaced
 - FlyCapture2Managed::PngOption, 136
- interPacketDelay
 - FlyCapture2Managed::GigEStreamChannel, 75
- InvalidBuManager
 - Enumerations, 17
- InvalidGeneration
 - Enumerations, 18
- InvalidMode
 - Enumerations, 18
- InvalidPacketSize
 - Enumerations, 18
- InvalidParameter
 - Enumerations, 17
- InvalidSettings
 - Enumerations, 17
- ipAddress
 - FlyCapture2Managed::CameraInfo, 45

- Iris
 - Enumerations, [23](#)
- isColorCamera
 - FlyCapture2Managed::CameraInfo, [45](#)
- IsConnected
 - FlyCapture2Managed::ManagedCameraBase, [106](#)
- IsNativeImageValid
 - FlyCapture2Managed::ManagedImage, [126](#)
- IsochAlreadyStarted
 - Enumerations, [18](#)
- IsochBandwidthExceeded
 - Enumerations, [18](#)
- isochBusSpeed
 - FlyCapture2Managed::FC2Config, [59](#)
- IsochFailed
 - Enumerations, [18](#)
- IsochNotStarted
 - Enumerations, [18](#)
- IsochRetrieveBufferFailed
 - Enumerations, [18](#)
- IsochStartFailed
 - Enumerations, [18](#)
- IsochStopFailed
 - Enumerations, [18](#)
- IsochSyncFailed
 - Enumerations, [18](#)
- isReadable
 - FlyCapture2Managed::GigEProperty, [73](#)
- IsVisible
 - FlyCapture2Managed::Gui::CameraControlDialog, [42](#)
- isWritable
 - FlyCapture2Managed::GigEProperty, [73](#)
- Jpeg
 - Enumerations, [20](#)
 - FlyCapture2Managed::TiffOption, [143](#)
- Jpeg2000
 - Enumerations, [20](#)
- JpegOption
 - FlyCapture2Managed::JpegOption, [79](#)
- Jpg2Option
 - FlyCapture2Managed::Jpg2Option, [80](#)
- keyword
 - FlyCapture2Managed::ConfigROM, [54](#)
- LaunchBrowser
 - FlyCapture2Managed::ManagedUtilities, [134](#)
- LaunchCommand
 - FlyCapture2Managed::ManagedUtilities, [134](#)
- LaunchHelp
 - FlyCapture2Managed::ManagedUtilities, [134](#)
- libraryList
 - FlyCapture2Managed::SystemInfo, [142](#)
- libraryVersion
 - FlyCapture2Managed::ManagedUtilities, [134](#)
- Lightness
 - Enumerations, [23](#)
- LinuxX64
 - Enumerations, [22](#)
- LinuxX86
 - Enumerations, [22](#)
- LittleEndian
 - Enumerations, [17](#)
- LowLevelFailure
 - Enumerations, [17](#)
- LutFailed
 - Enumerations, [18](#)
- Lzw
 - FlyCapture2Managed::TiffOption, [143](#)
- m_externalDelegate
 - FlyCapture2Managed::ManagedCameraBase, [113](#)
- m_internalDelegate
 - FlyCapture2Managed::ManagedCameraBase, [113](#)
- m_pNativeCamBase
 - FlyCapture2Managed::ManagedCameraBase, [113](#)
- Mac
 - Enumerations, [22](#)
- macAddress
 - FlyCapture2Managed::CameraInfo, [45](#)
- major
 - FlyCapture2Managed::FC2Version, [62](#)
- ManagedAVIRecorder
 - FlyCapture2Managed::ManagedAVIRecorder, [83](#)
- ManagedBusManager
 - FlyCapture2Managed::ManagedBusManager, [87](#)
- ManagedCallbackType
 - Enumerations, [20](#)
- ManagedCamera
 - FlyCapture2Managed::ManagedCamera, [94](#)
- ManagedCameraBase
 - FlyCapture2Managed::ManagedCameraBase, [101](#)
- ManagedGigECamera
 - FlyCapture2Managed::ManagedGigECamera, [117](#)
- ManagedImage
 - FlyCapture2Managed::ManagedImage, [125](#)
- ManagedImageStatistics
 - FlyCapture2Managed::ManagedImageStatistics, [131](#)
- ManagedPGRGuid
 - FlyCapture2Managed::ManagedPGRGuid, [132](#), [133](#)
- manualSupported
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
- max
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
 - FlyCapture2Managed::GigEProperty, [73](#)
- maxBytesPerPacket
 - FlyCapture2Managed::Format7PacketInfo, [68](#)

- maxHeight
 - FlyCapture2Managed::Format7Info, [66](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [71](#)
- maximumBusSpeed
 - FlyCapture2Managed::CameraInfo, [45](#)
- maxPacketSize
 - FlyCapture2Managed::Format7Info, [66](#)
- maxValue
 - FlyCapture2Managed::StrobeInfo, [139](#)
- maxWidth
 - FlyCapture2Managed::Format7Info, [66](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [71](#)
- MemoryAllocationFailed
 - Enumerations, [17](#)
- microSeconds
 - FlyCapture2Managed::TimeStamp, [145](#)
- min
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
 - FlyCapture2Managed::GigEProperty, [73](#)
- minor
 - FlyCapture2Managed::FC2Version, [62](#)
- minPacketSize
 - FlyCapture2Managed::Format7Info, [66](#)
- minValue
 - FlyCapture2Managed::StrobeInfo, [139](#)
- Mode
 - Enumerations, [20](#)
- mode
 - FlyCapture2Managed::Format7ImageSettings, [63](#)
 - FlyCapture2Managed::Format7Info, [66](#)
 - FlyCapture2Managed::TriggerMode, [150](#)
- Mode0
 - Enumerations, [21](#)
- Mode1
 - Enumerations, [21](#)
- Mode10
 - Enumerations, [21](#)
- Mode11
 - Enumerations, [21](#)
- Mode12
 - Enumerations, [21](#)
- Mode13
 - Enumerations, [21](#)
- Mode14
 - Enumerations, [21](#)
- Mode15
 - Enumerations, [21](#)
- Mode16
 - Enumerations, [21](#)
- Mode17
 - Enumerations, [21](#)
- Mode18
 - Enumerations, [21](#)
- Mode19
 - Enumerations, [21](#)
- Mode2
 - Enumerations, [21](#)
- Mode20
 - Enumerations, [21](#)
- Mode21
 - Enumerations, [21](#)
- Mode22
 - Enumerations, [21](#)
- Mode23
 - Enumerations, [21](#)
- Mode24
 - Enumerations, [21](#)
- Mode25
 - Enumerations, [21](#)
- Mode26
 - Enumerations, [21](#)
- Mode27
 - Enumerations, [21](#)
- Mode28
 - Enumerations, [21](#)
- Mode29
 - Enumerations, [21](#)
- Mode3
 - Enumerations, [21](#)
- Mode30
 - Enumerations, [21](#)
- Mode31
 - Enumerations, [21](#)
- Mode4
 - Enumerations, [21](#)
- Mode5
 - Enumerations, [21](#)
- Mode6
 - Enumerations, [21](#)
- Mode7
 - Enumerations, [21](#)
- Mode8
 - Enumerations, [21](#)
- Mode9
 - Enumerations, [21](#)
- modelName
 - FlyCapture2Managed::CameraInfo, [45](#)
- modeMask
 - FlyCapture2Managed::TriggerModeInfo, [151](#)
- NearestNeighbor
 - Enumerations, [17](#)
- networkInterfaceIndex
 - FlyCapture2Managed::GigEStreamChannel, [76](#)
- NoColorProcessing
 - Enumerations, [17](#)
- nodeVendorId
 - FlyCapture2Managed::ConfigROM, [54](#)

- None
 - Enumerations, [16](#), [19](#)
 - FlyCapture2Managed::TiffOption, [143](#)
- NotConnected
 - Enumerations, [17](#)
- NotFound
 - Enumerations, [17](#)
- NotImplemented
 - Enumerations, [17](#)
- NotInFormat7
 - Enumerations, [18](#)
- NotInitialized
 - Enumerations, [17](#)
- NotSupported
 - Enumerations, [18](#)
- numBanks
 - FlyCapture2Managed::LutData, [81](#)
- NumberOfFrameRates
 - Enumerations, [18](#)
- NumberOfModes
 - Enumerations, [21](#)
- NumberOfPixelFormats
 - Enumerations, [22](#)
- NumberOfStatisticsChannels
 - Enumerations, [23](#)
- NumberOfVideoModes
 - Enumerations, [24](#)
- numBuffers
 - FlyCapture2Managed::FC2Config, [59](#)
- numChannels
 - FlyCapture2Managed::GigEConfig, [69](#)
 - FlyCapture2Managed::LutData, [81](#)
- numCpuCores
 - FlyCapture2Managed::SystemInfo, [142](#)
- numEntries
 - FlyCapture2Managed::LutData, [82](#)
- numImageNotifications
 - FlyCapture2Managed::FC2Config, [59](#)
- Off
 - Enumerations, [16](#)
- offsetHStepSize
 - FlyCapture2Managed::Format7Info, [66](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [72](#)
- offsetVStepSize
 - FlyCapture2Managed::Format7Info, [66](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [72](#)
- offsetX
 - FlyCapture2Managed::Format7ImageSettings, [63](#)
 - FlyCapture2Managed::GigEImageSettings, [70](#)
- offsetY
 - FlyCapture2Managed::Format7ImageSettings, [63](#)
 - FlyCapture2Managed::GigEImageSettings, [70](#)
- Ok
 - Enumerations, [17](#)
- On
 - Enumerations, [16](#)
- onePush
 - FlyCapture2Managed::CameraProperty, [48](#)
- onePushSupported
 - FlyCapture2Managed::CameraPropertyInfo, [50](#)
- OnNativeCallback
 - FlyCapture2Managed::ManagedCameraBase, [106](#)
- onOff
 - FlyCapture2Managed::CameraProperty, [48](#)
 - FlyCapture2Managed::EmbeddedImageInfoProperty, [57](#)
 - FlyCapture2Managed::StrobeControl, [138](#)
 - FlyCapture2Managed::TriggerMode, [150](#)
- onOffSupported
 - FlyCapture2Managed::CameraPropertyInfo, [51](#)
 - FlyCapture2Managed::StrobeInfo, [139](#)
 - FlyCapture2Managed::TriggerModeInfo, [151](#)
- operator=
 - FlyCapture2Managed::ManagedPGRGuid, [133](#)
- operator==
 - FlyCapture2Managed::ManagedPGRGuid, [133](#)
- osDescription
 - FlyCapture2Managed::SystemInfo, [142](#)
- OSType
 - Enumerations, [21](#)
- osType
 - FlyCapture2Managed::SystemInfo, [142](#)
- outputBitDepth
 - FlyCapture2Managed::LutData, [82](#)
- PackBits
 - FlyCapture2Managed::TiffOption, [143](#)
- PacketDelay
 - Enumerations, [19](#)
- PacketSize
 - Enumerations, [19](#)
- packetSize
 - FlyCapture2Managed::Format7Info, [66](#)
 - FlyCapture2Managed::GigEStreamChannel, [76](#)
- Pan
 - Enumerations, [23](#)
- parameter
 - FlyCapture2Managed::TriggerMode, [150](#)
- percentage
 - FlyCapture2Managed::Format7Info, [66](#)
- Pgm
 - Enumerations, [20](#)
- PgmOption
 - FlyCapture2Managed::PgmOption, [135](#)
- PixelFormat
 - Enumerations, [22](#)
- pixelFormat
 - FlyCapture2Managed::Format7ImageSettings, [63](#)

- FlyCapture2Managed::GigEImageSettings, 70
- FlyCapture2Managed::ManagedImage, 129
- PixelFormat411Yuv8
 - Enumerations, 22
- PixelFormat422Yuv8
 - Enumerations, 22
- PixelFormat444Yuv8
 - Enumerations, 22
- PixelFormatBgr
 - Enumerations, 22
- PixelFormatBgru
 - Enumerations, 22
- pixelFormatBitField
 - FlyCapture2Managed::Format7Info, 66
 - FlyCapture2Managed::GigEImageSettingsInfo, 72
- PixelFormatMono12
 - Enumerations, 22
- PixelFormatMono16
 - Enumerations, 22
- PixelFormatMono8
 - Enumerations, 22
- PixelFormatRaw12
 - Enumerations, 22
- PixelFormatRaw16
 - Enumerations, 22
- PixelFormatRaw8
 - Enumerations, 22
- PixelFormatRgb
 - Enumerations, 22
- PixelFormatRgb16
 - Enumerations, 22
- PixelFormatRgb8
 - Enumerations, 22
- PixelFormatRgbu
 - Enumerations, 22
- PixelFormatSignedMono16
 - Enumerations, 22
- PixelFormatSignedRgb16
 - Enumerations, 22
- Png
 - Enumerations, 20
- PngOption
 - FlyCapture2Managed::PngOption, 136
- polarity
 - FlyCapture2Managed::StrobeControl, 138
 - FlyCapture2Managed::TriggerMode, 150
- polaritySupported
 - FlyCapture2Managed::StrobeInfo, 139
 - FlyCapture2Managed::TriggerModeInfo, 151
- Ppm
 - Enumerations, 20
- PpmOption
 - FlyCapture2Managed::PpmOption, 137
- present
 - FlyCapture2Managed::CameraProperty, 48
 - FlyCapture2Managed::CameraPropertyInfo, 51
 - FlyCapture2Managed::StrobeInfo, 140
 - FlyCapture2Managed::TriggerModeInfo, 151
- progressive
 - FlyCapture2Managed::JpegOption, 79
- PropertyFailed
 - Enumerations, 18
- PropertyNotPresent
 - Enumerations, 18
- PropertyType
 - Enumerations, 22
- propType
 - FlyCapture2Managed::GigEProperty, 73
- quality
 - FlyCapture2Managed::JpegOption, 79
 - FlyCapture2Managed::Jpg2Option, 80
- QueryGigEImagingMode
 - FlyCapture2Managed::ManagedGigECamera, 118
- Raw
 - Enumerations, 20
- ReadGVCPMemory
 - FlyCapture2Managed::ManagedGigECamera, 118
- ReadGVCPRegister
 - FlyCapture2Managed::ManagedGigECamera, 119
- ReadGVCPRegisterBlock
 - FlyCapture2Managed::ManagedGigECamera, 119
- readOutSupported
 - FlyCapture2Managed::CameraPropertyInfo, 51
 - FlyCapture2Managed::StrobeInfo, 140
 - FlyCapture2Managed::TriggerModeInfo, 151
- ReadPhyRegister
 - FlyCapture2Managed::ManagedBusManager, 90
- ReadRegister
 - FlyCapture2Managed::ManagedCameraBase, 106
- ReadRegisterBlock
 - FlyCapture2Managed::ManagedCameraBase, 106
- ReadRegisterFailed
 - Enumerations, 18
- recommendedBytesPerPacket
 - FlyCapture2Managed::Format7PacketInfo, 68
- Red
 - Enumerations, 23
- RegisterCallback
 - FlyCapture2Managed::ManagedBusManager, 90
- RegisterFailed
 - Enumerations, 18
- ReleaseBuffer
 - FlyCapture2Managed::ManagedImage, 126
- Removal
 - Enumerations, 20
- RescanBus

- FlyCapture2Managed::ManagedBusManager, 90
- RestoreFromMemoryChannel
 - FlyCapture2Managed::ManagedImage, 128
- FlyCapture2Managed::ManagedCameraBase, 107
- RetrieveBuffer
 - FlyCapture2Managed::ManagedImage, 128
- FlyCapture2Managed::ManagedCameraBase, 107
- RGGB
 - Enumerations, 16
- Rigorous
 - Enumerations, 17
- ROIPosition
 - FlyCapture2Managed::EmbeddedImageInfo, 56
- rows
 - FlyCapture2Managed::ManagedImage, 129
- S100
 - Enumerations, 16
- S1600
 - Enumerations, 16
- S200
 - Enumerations, 16
- S3200
 - Enumerations, 16
- S400
 - Enumerations, 16
- S480
 - Enumerations, 16
- S800
 - Enumerations, 16
- Saturation
 - Enumerations, 23
- Save
 - FlyCapture2Managed::ManagedImage, 126, 127
- SaveToMemoryChannel
 - FlyCapture2Managed::ManagedCameraBase, 107
- screenHeight
 - FlyCapture2Managed::SystemInfo, 142
- screenWidth
 - FlyCapture2Managed::SystemInfo, 142
- seconds
 - FlyCapture2Managed::TimeStamp, 145
- sensorInfo
 - FlyCapture2Managed::CameraInfo, 46
- sensorResolution
 - FlyCapture2Managed::CameraInfo, 46
- serialNumber
 - FlyCapture2Managed::CameraInfo, 46
- SetActiveLUTBank
 - FlyCapture2Managed::ManagedCameraBase, 108
- SetCallback
 - FlyCapture2Managed::ManagedCameraBase, 108
- SetChannelStatus
 - FlyCapture2Managed::ManagedImageStatistics, 131
- SetConfiguration
 - FlyCapture2Managed::ManagedCameraBase, 108
- SetData
 - FlyCapture2Managed::ManagedImage, 128
- SetDimensions
 - FlyCapture2Managed::ManagedImage, 128
- SetEmbeddedImageInfo
 - FlyCapture2Managed::ManagedCameraBase, 108
- SetFormat7Configuration
 - FlyCapture2Managed::ManagedCamera, 95
- SetGigEImageBinningSettings
 - FlyCapture2Managed::ManagedGigECamera, 119
- SetGigEImageSettings
 - FlyCapture2Managed::ManagedGigECamera, 119
- SetGigEImagingMode
 - FlyCapture2Managed::ManagedGigECamera, 119
- SetGigEProperty
 - FlyCapture2Managed::ManagedGigECamera, 120
- SetGigEStreamChannelInfo
 - FlyCapture2Managed::ManagedGigECamera, 120
- SetGPIOPinDirection
 - FlyCapture2Managed::ManagedCameraBase, 109
- SetLUTChannel
 - FlyCapture2Managed::ManagedCameraBase, 109
- SetProperty
 - FlyCapture2Managed::ManagedCameraBase, 110
- SetStrobe
 - FlyCapture2Managed::ManagedCameraBase, 110
- SetTriggerMode
 - FlyCapture2Managed::ManagedCameraBase, 111
- SetVideoModeAndFrameRate
 - FlyCapture2Managed::ManagedCamera, 96
- Sharpness
 - Enumerations, 23
- Show
 - FlyCapture2Managed::Gui::CameraControlDialog, 42
- ShowModal
 - FlyCapture2Managed::Gui::CameraSelectionDialog, 52
- Shutter
 - Enumerations, 23
- shutter
 - FlyCapture2Managed::EmbeddedImageInfo, 56
- softwareTriggerSupported
 - FlyCapture2Managed::TriggerModeInfo, 152
- source
 - FlyCapture2Managed::StrobeControl, 138
 - FlyCapture2Managed::StrobeInfo, 140
 - FlyCapture2Managed::TriggerMode, 150
- sourceMask
 - FlyCapture2Managed::TriggerModeInfo, 152
- sourcePort
 - FlyCapture2Managed::GigEStreamChannel, 76
- StartCapture
 - FlyCapture2Managed::ManagedCameraBase, 111
- StatisticsChannel
 - Enumerations, 23

- StopCapture
 - FlyCapture2Managed::ManagedCameraBase, 11
- stride
 - FlyCapture2Managed::ManagedImage, 129
- StrobeFailed
 - Enumerations, 18
- strobePattern
 - FlyCapture2Managed::EmbeddedImageInfo, 56
- Structures, 25
- subnetMask
 - FlyCapture2Managed::CameraInfo, 46
- supported
 - FlyCapture2Managed::LutData, 82
- systemInfo
 - FlyCapture2Managed::ManagedUtilities, 134
- systemMemorySize
 - FlyCapture2Managed::SystemInfo, 142
- Temperature
 - Enumerations, 23
- Tiff
 - Enumerations, 20
- TiffOption
 - FlyCapture2Managed::TiffOption, 144
- Tilt
 - Enumerations, 23
- Timeout
 - Enumerations, 18
- timeStamp
 - FlyCapture2Managed::ManagedImage, 129
- timestamp
 - FlyCapture2Managed::EmbeddedImageInfo, 56
- ToMgd
 - FlyCapture2Managed::Translate, 149
- ToNative
 - FlyCapture2Managed::Translate, 149
- translate
 - FlyCapture2Managed::Translate, 149
- TriggerDelay
 - Enumerations, 23
- TriggerFailed
 - Enumerations, 18
- TriggerMode
 - Enumerations, 23
- Type
 - FlyCapture2Managed::FC2Exception, 61
- type
 - FlyCapture2Managed::CameraProperty, 48
 - FlyCapture2Managed::CameraPropertyInfo, 51
 - FlyCapture2Managed::FC2Version, 62
- Undefined
 - Enumerations, 17
- unitAbbr
 - FlyCapture2Managed::CameraPropertyInfo, 51
- unitBytesPerPacket
 - FlyCapture2Managed::Format7PacketInfo, 68
- units
 - FlyCapture2Managed::CameraPropertyInfo, 51
- unitSpecId
 - FlyCapture2Managed::ConfigROM, 54
- unitSubSWVer
 - FlyCapture2Managed::ConfigROM, 54
- unitSWVer
 - FlyCapture2Managed::ConfigROM, 54
- Unknown
 - Enumerations, 16, 20
- UnknownOS
 - Enumerations, 22
- UnregisterCallback
 - FlyCapture2Managed::ManagedBusManager, 91
- Unspecified
 - Enumerations, 16, 19, 23
- Unsupported
 - Enumerations, 16
- Usb2
 - Enumerations, 20
- userDefinedName
 - FlyCapture2Managed::CameraInfo, 46
- ValidateFormat7Settings
 - FlyCapture2Managed::ManagedCamera, 96
- value
 - FlyCapture2Managed::GigEProperty, 74
- value0
 - FlyCapture2Managed::ManagedPGRGuid, 133
- value1
 - FlyCapture2Managed::ManagedPGRGuid, 133
- value2
 - FlyCapture2Managed::ManagedPGRGuid, 133
- value3
 - FlyCapture2Managed::ManagedPGRGuid, 133
- valueA
 - FlyCapture2Managed::CameraProperty, 48
- valueB
 - FlyCapture2Managed::CameraProperty, 48
- valueReadable
 - FlyCapture2Managed::TriggerModeInfo, 152
- vendorName
 - FlyCapture2Managed::CameraInfo, 46
- vendorUniqueInfo0
 - FlyCapture2Managed::ConfigROM, 54
- vendorUniqueInfo1
 - FlyCapture2Managed::ConfigROM, 54
- vendorUniqueInfo2
 - FlyCapture2Managed::ConfigROM, 54
- vendorUniqueInfo3
 - FlyCapture2Managed::ConfigROM, 54

- VideoMode
 - Enumerations, [23](#)
- VideoMode1024x768Rgb
 - Enumerations, [24](#)
- VideoMode1024x768Y16
 - Enumerations, [24](#)
- VideoMode1024x768Y8
 - Enumerations, [24](#)
- VideoMode1024x768Yuv422
 - Enumerations, [24](#)
- VideoMode1280x960Rgb
 - Enumerations, [24](#)
- VideoMode1280x960Y16
 - Enumerations, [24](#)
- VideoMode1280x960Y8
 - Enumerations, [24](#)
- VideoMode1280x960Yuv422
 - Enumerations, [24](#)
- VideoMode1600x1200Rgb
 - Enumerations, [24](#)
- VideoMode1600x1200Y16
 - Enumerations, [24](#)
- VideoMode1600x1200Y8
 - Enumerations, [24](#)
- VideoMode1600x1200Yuv422
 - Enumerations, [24](#)
- VideoMode160x120Yuv444
 - Enumerations, [24](#)
- VideoMode320x240Yuv422
 - Enumerations, [24](#)
- VideoMode640x480Rgb
 - Enumerations, [24](#)
- VideoMode640x480Y16
 - Enumerations, [24](#)
- VideoMode640x480Y8
 - Enumerations, [24](#)
- VideoMode640x480Yuv411
 - Enumerations, [24](#)
- VideoMode640x480Yuv422
 - Enumerations, [24](#)
- VideoMode800x600Rgb
 - Enumerations, [24](#)
- VideoMode800x600Y16
 - Enumerations, [24](#)
- VideoMode800x600Y8
 - Enumerations, [24](#)
- VideoMode800x600Yuv422
 - Enumerations, [24](#)
- VideoModeFormat7
 - Enumerations, [24](#)
- WaitForBufferEvent
 - FlyCapture2Managed::ManagedCameraBase, [112](#)
- WhiteBalance
 - Enumerations, [23](#)
- whiteBalance
 - FlyCapture2Managed::EmbeddedImageInfo, [56](#)
- width
 - FlyCapture2Managed::Format7ImageSettings, [64](#)
 - FlyCapture2Managed::GigEImageSettings, [70](#)
- WindowsX64
 - Enumerations, [22](#)
- WindowsX86
 - Enumerations, [22](#)
- WriteGVCPMemory
 - FlyCapture2Managed::ManagedGigECamera, [120](#)
- WriteGVCPRegister
 - FlyCapture2Managed::ManagedGigECamera, [120](#)
- WriteGVCPRegisterBlock
 - FlyCapture2Managed::ManagedGigECamera, [121](#)
- WritePhyRegister
 - FlyCapture2Managed::ManagedBusManager, [91](#)
- WriteRegister
 - FlyCapture2Managed::ManagedCameraBase, [112](#)
- WriteRegisterBlock
 - FlyCapture2Managed::ManagedCameraBase, [112](#)
- WriteRegisterFailed
 - Enumerations, [18](#)
- xmlURL1
 - FlyCapture2Managed::CameraInfo, [46](#)
- xmlURL2
 - FlyCapture2Managed::CameraInfo, [46](#)
- Zoom
 - Enumerations, [23](#)