



POINT GREY

ActiveFlyCap

ActiveX Control for FlyCapture
Reference

Version 2.0

Revised October 7, 2009



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.

Trademarks

Point Grey Research, PGR, the Point Grey Research, Inc. logo, Chameleon, Digiclops, Dragonfly, Dragonfly Express, Firefly, Flea, FlyCapture, Grasshopper, Ladybug, and Triclops are trademarks or registered trademarks of Point Grey Research, Inc. in Canada and other countries.

Table of Contents

| | |
|---|-----------|
| Table of Contents | 3 |
| 1 Overview | 6 |
| 1.1 Supported Programming Languages | 6 |
| 1.2 Manual Registration..... | 6 |
| 1.3 Supporting downloads | 6 |
| 1.4 Adding the Control..... | 7 |
| 1.4.1 Visual Studio 2005 | 7 |
| 1.4.2 Visual C++ 6 | 8 |
| 1.4.3 Visual Basic 6 | 9 |
| 1.5 Grab Modes | 10 |
| 1.5.1 Free Running..... | 10 |
| 1.5.2 LockLatest..... | 10 |
| 1.5.3 LockNext | 10 |
| 2 Declarations | 11 |
| 2.1 Structures | 11 |
| 2.1.1 CameraInfo..... | 11 |
| 2.1.2 ImageInfo | 11 |
| 2.1.3 CameraPropertyInfo | 11 |
| 2.1.4 CameraProperty | 12 |
| 2.1.5 TriggerStruct..... | 12 |
| 2.1.6 StrobeStruct | 12 |
| 2.1.7 Format7Info | 12 |
| 2.1.8 Format7ImageSettings | 13 |
| 2.1.9 Format7PacketInfo | 13 |
| 3 Properties | 14 |
| 3.1 Control Properties..... | 14 |
| 3.1.1 EnableRightClickMenu | 14 |
| 3.1.2 Display | 14 |
| 3.1.3 AutoResize | 14 |
| 3.1.4 Font | 14 |
| 3.2 General Camera Properties..... | 14 |
| 3.2.1 Camera..... | 14 |
| 3.2.2 Start..... | 15 |
| 3.2.3 VideoMode | 15 |
| 3.2.4 FrameRate | 15 |
| 3.2.5 AsyncBusSpeed | 15 |
| 3.2.6 IsochBusSpeed | 15 |
| 3.2.7 GrabTimeout | 15 |
| 4 Methods | 16 |

| | | |
|--------|-----------------------------------|----|
| 4.1 | Control Methods | 16 |
| 4.1.1 | DrawSingleImage | 16 |
| 4.2 | GUI Methods..... | 16 |
| 4.2.1 | ToggleCamCtlDlg | 16 |
| 4.2.2 | IsCamCtlDlgVisible..... | 16 |
| 4.2.3 | ShowCamSlnModal..... | 16 |
| 4.3 | Camera Methods | 17 |
| 4.3.1 | GetCameraInfo | 17 |
| 4.3.2 | CheckVideoMode | 17 |
| 4.3.3 | GetFormat7PacketInfo | 17 |
| 4.3.4 | GetPacketInfo..... | 17 |
| 4.3.5 | GetImageInformation..... | 17 |
| 4.3.6 | GetImageData | 18 |
| 4.3.7 | GetImagePtr | 18 |
| 4.4 | 1394 Bus Methods..... | 18 |
| 4.4.1 | GetNumOfCameras..... | 18 |
| 4.4.2 | GetCameraList | 18 |
| 4.5 | Grab Methods | 18 |
| 4.5.1 | SetGrabMode | 18 |
| 4.5.2 | GetGrabMode..... | 19 |
| 4.5.3 | GrabImage | 19 |
| 4.6 | Image Saving Methods..... | 19 |
| 4.6.1 | SaveImage | 19 |
| 4.6.2 | SetJPEGCompressionQuality | 19 |
| 4.6.3 | GetColorProcessingAlgorithm..... | 19 |
| 4.6.4 | SetColorProcessingAlgorithm | 20 |
| 4.7 | Format 7 Methods..... | 20 |
| 4.7.1 | GetFormat7Info | 20 |
| 4.7.2 | GetFormat7Configuration..... | 20 |
| 4.7.3 | SetFormat7Configuration | 20 |
| 4.7.4 | GetFormat7PacketSize | 20 |
| 4.8 | Camera Property Methods | 21 |
| 4.8.1 | GetProperty | 21 |
| 4.8.2 | SetProperty | 21 |
| 4.8.3 | GetPropertyInfo | 21 |
| 4.9 | Trigger / Strobe Methods..... | 21 |
| 4.9.1 | GetTrigger | 21 |
| 4.9.2 | SetTrigger..... | 21 |
| 4.9.3 | SetTriggerBroadcast | 22 |
| 4.9.4 | GetStrobe | 22 |
| 4.9.5 | SetStrobe | 22 |
| 4.9.6 | SetStrobeBroadcast | 22 |
| 4.10 | Memory Channel Methods | 22 |
| 4.10.1 | GetNumMemoryChannels..... | 22 |
| 4.10.2 | GetCurrentMemoryChannel | 22 |
| 4.10.3 | SaveToMemoryChannel..... | 23 |
| 4.10.4 | RestoreFromMemoryChannel..... | 23 |
| 4.11 | Register Methods..... | 23 |
| 4.11.1 | ReadRegister | 23 |
| 4.11.2 | WriteRegister..... | 23 |
| 4.11.3 | ReadRegisterBlock..... | 23 |
| 4.11.4 | WriteRegisterBlock..... | 24 |
| 4.12 | Drawing Methods..... | 24 |
| 4.12.1 | DrawEllipse | 24 |

| | | |
|----------|---|-----------|
| 4.12.2 | DrawLine | 24 |
| 4.12.3 | DrawPixel | 24 |
| 4.12.4 | DrawRectangle | 25 |
| 4.12.5 | DrawText | 25 |
| 5 | Events..... | 26 |
| 5.1 | General Events | 26 |
| 5.1.1 | CameraArrival | 26 |
| 5.1.2 | CameraRemoval | 26 |
| 5.1.3 | BusReset..... | 26 |
| 5.1.4 | Image..... | 26 |
| 6 | Additional Resources..... | 27 |
| 6.1 | Getting Started Manual..... | 27 |
| 6.2 | Technical Reference Manual..... | 27 |
| 6.3 | Point Grey Digital Camera Register Reference..... | 27 |
| 7 | Technical Support Resources | 28 |
| 8 | Contacting Point Grey Research Inc..... | 29 |

1 Overview

The ActiveFlyCap control allows users to easily insert an ActiveX control into a GUI application and control Point Grey cameras easily without having to worry about writing additional code to draw images onto the screen.

1.1 Supported Programming Languages

The following programming languages are supported:

- Visual Basic 6
- Visual Basic.Net
- Visual C#
- Visual C++

Any language capable of interacting with ActiveX controls (e.g. in a Windows Form) should be able to interface with ActiveFlyCap.

1.2 Manual Registration

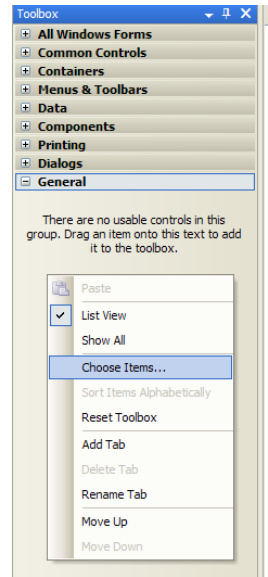
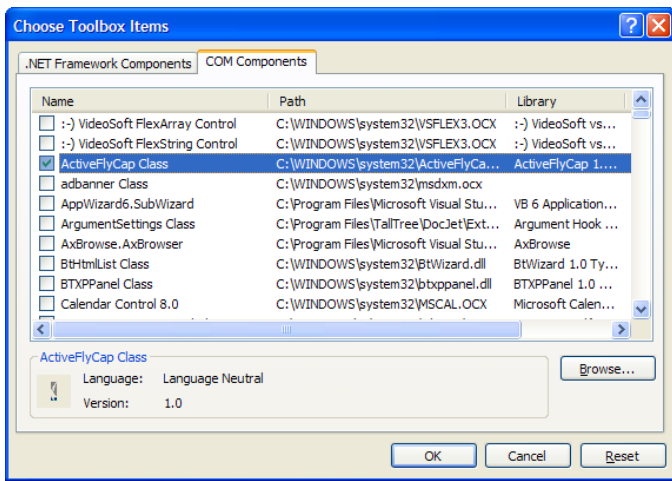
In the event that the control must be manually registered, call regsvr32 with the name of the control (ActiveFlyCap.dll). The FlyCapture2 library DLLs (FlyCapture2.dll and FlyCapture2GUI.dll) should also be present in the system path.

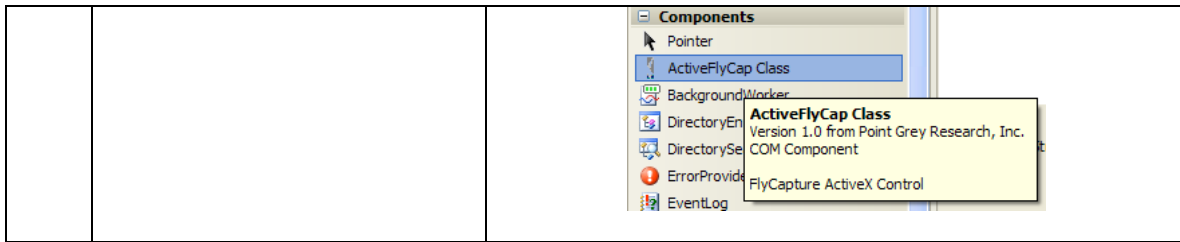
1.3 Supporting downloads

It may be necessary to install the Microsoft Visual C++ 2005 SP1 Redistributable Package. The package can be downloaded directly from Microsoft from the following URL: <http://www.microsoft.com/downloads/details.aspx?FamilyID=200b2fd9-ae1a-4a14-984d-389c36f85647&displaylang=en>.

1.4 Adding the Control

1.4.1 Visual Studio 2005

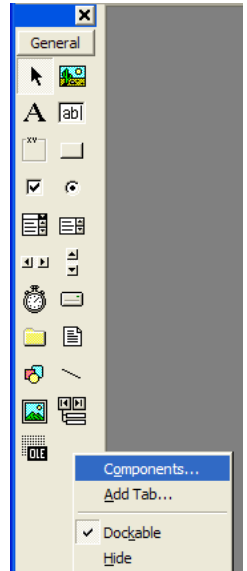
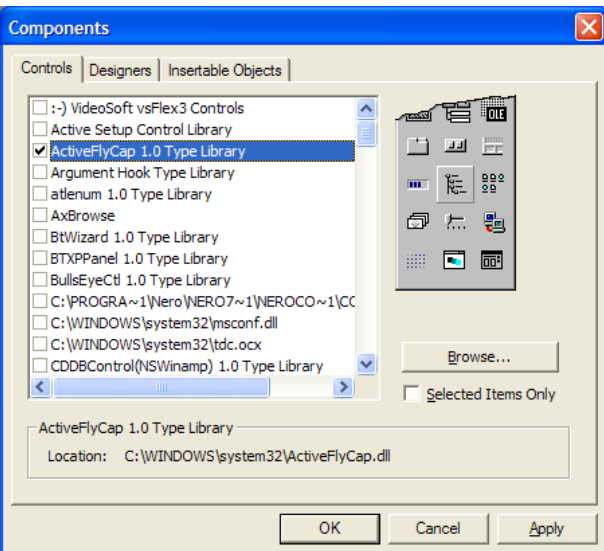
| Step | Instruction | Screenshot |
|------|---|--|
| 1 | In a GUI-based application, right click the <i>Toolbox</i> and select <i>Choose Items</i> . |  |
| 2 | Under <i>COM Components</i> , make sure that the <i>ActiveFlyCap</i> class is selected. |  |
| 3 | The <i>ActiveFlyCap</i> control component will now appear in the <i>Toolbox</i> . | |

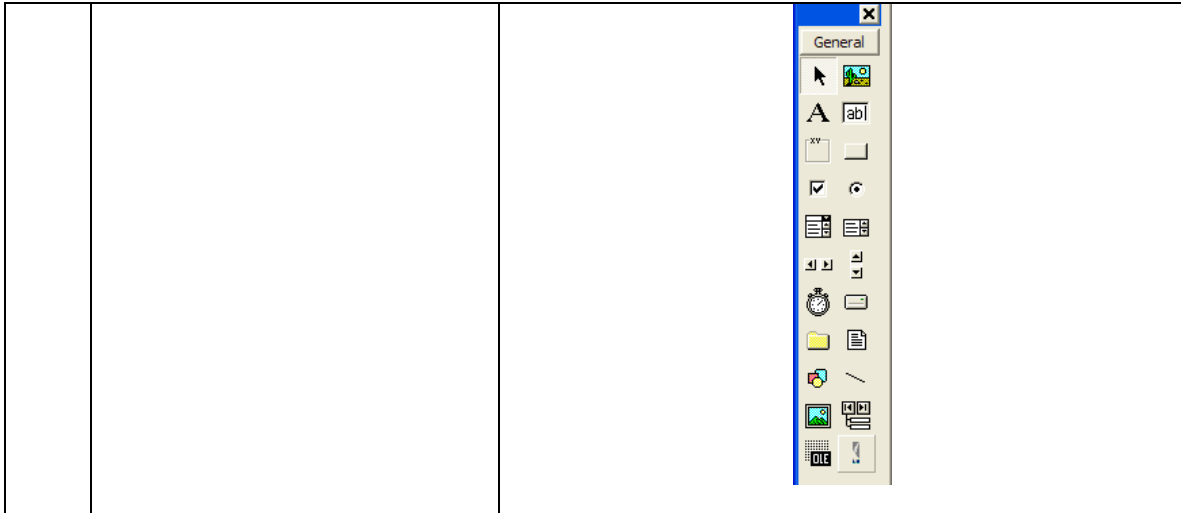


1.4.2 Visual C++ 6

| Step | Instruction | Screenshot |
|------|--|------------|
| 1 | Right click a dialog in the dialog editor and select <i>Insert ActiveX Control</i> . | |
| 2 | Select the ActiveFlyCap class and click <i>Ok</i> . | |

1.4.3 Visual Basic 6

| Step | Instruction | Screenshot |
|------|--|---|
| 1 | Right click the <i>Toolbox</i> and select <i>Components</i> . |  |
| 2 | Select the ActiveFlyCap Type Library and click <i>Ok</i> . |  |
| 3 | The ActiveFlyCap component will now appear in the <i>Toolbox</i> . | |



1.5 Grab Modes

There are 3 grab modes available for use in ActiveFlyCap.

1.5.1 Free Running

This mode simply starts the camera and automatically draws images to the screen. This mode is ideal if accessing the image data directly is not required.

1.5.2 LockLatest

This mode locks the latest image that hasn't been seen by the user. This means that there may be a possibility that images may be lost if the PC is unable to perform processing quickly enough.

1.5.3 LockNext

This mode locks the oldest image that hasn't been seen by the user. Provided that the PC is fast enough to process images faster than the frame rate of the camera, this mode guarantees that no images are lost.

2 Declarations

2.1 Structures

In order to pass multiple arguments to the control, structures are used to contain data that will be passed between the application, the ActiveFlyCap control and the FlyCapture library. Structures do not have to be populated before use. Any incoming parameters must be supplied as a separate argument when the function is called.

2.1.1 CameraInfo

| Type | Name | Description |
|---------------|-------------------|---|
| LONG | serialNumber | Serial number of the camera |
| InterfaceType | interfaceType | Interface type |
| BOOL | isColorCamera | Whether it is a color or mono camera |
| BSTR | modelName | Model name |
| BSTR | vendorName | Vendor name |
| BSTR | sensorInfo | Sensor info |
| BSTR | sensorResolution | Sensor resolution |
| BSTR | driverName | Driver name |
| BSTR | firmwareVersion | Firmware version |
| BSTR | firmwareBuildTime | Firmware build time |
| LONG | iidcVer | DCAM version. Divide this number by 100 for the actual number |
| BusSpeed | maxBusSpeed | Maximum bus speed |

2.1.2 ImageInfo

| Type | Name | Description |
|---------------|-----------|--------------------------------------|
| LONG | rows | Number of rows in the image |
| LONG | cols | Number of columns in the image |
| LONG | rowInc | Number of bytes per row in the image |
| LONG | numImages | Number of images contained within |
| DCAMVideoMode | videoMode | Video mode |
| PixelFormat | pixelFmt | Pixel format |

2.1.3 CameraPropertyInfo

| Type | Name | Description |
|------|------------------|--|
| BOOL | present | Whether the property is present |
| BOOL | autoSupported | Whether auto is supported |
| BOOL | manualSupported | Whether manual is supported |
| BOOL | onOffSupported | Whether on / off is supported |
| BOOL | onePushSupported | Whether one push is supported |
| BOOL | absValSupported | Whether absolute mode is supported |
| BOOL | readOutSupported | Whether the property value can be read out |
| LONG | min | Minimum value |
| LONG | max | Maximum value |

| | | |
|-------|-----------|--|
| FLOAT | absMin | Minimum absolute value |
| FLOAT | absMax | Maximum absolute value |
| BSTR | units | Textual description of units |
| BSTR | unitsAbbr | Abbreviated textual description of units |

2.1.4 CameraProperty

| Type | Name | Description |
|-------|----------------|---------------------------------|
| BOOL | present | Whether the property is present |
| BOOL | absControl | Absolute mode control status |
| BOOL | onePush | One push status |
| BOOL | onOff | On / off status |
| BOOL | autoManualMode | Auto status |
| LONG | valueA | Value A |
| LONG | valueB | Value B |
| FLOAT | absValue | Absolute value |

2.1.5 TriggerStruct

| Type | Name | Description |
|------|-----------|-----------------|
| BOOL | isOnOff | On / off status |
| LONG | polarity | Polarity |
| LONG | source | Source |
| LONG | mode | Mode |
| LONG | parameter | Parameter |

2.1.6 StrobeStruct

| Type | Name | Description |
|------|----------|-----------------|
| BOOL | isOnOff | On / off status |
| LONG | polarity | Polarity |
| LONG | delay | Delay |
| LONG | duration | Duration |

2.1.7 Format7Info

| Type | Name | Description |
|-------|----------------------|--|
| BOOL | isSupported | Whether the mode is available in Format7 |
| LONG | maxWidth | Maximum width |
| LONG | maxHeight | Maximum height |
| LONG | imageHStepSize | Horizontal step size of the Format 7 image |
| LONG | imageVStepSize | Vertical step size of the Format 7 image |
| LONG | offsetHStepSize | Horizontal step size of offset of the Format 7 image |
| LONG | offsetVStepSize | Vertical step size of the offset of the Format 7 image |
| LONG | pixelFormatBitField | Bit field containing supported pixel formats |
| LONG | packetSize | Current packet size in bytes |
| LONG | minPacketSize | Minimum packet size in bytes |
| LONG | maxPacketSize | Maximum packet size in bytes |
| FLOAT | packetSizePercentage | Current packet size as a percentage of maximum packet size |

2.1.8 Format7ImageSettings

| Type | Name | Description |
|-------------|----------------------|-----------------------------------|
| LONG | mode | Format 7 mode |
| LONG | offsetX | Left offset of the Format 7 image |
| LONG | offsetY | Top offset of the Format 7 image |
| LONG | width | Width of the Format 7 image |
| LONG | height | Height of the Format 7 image |
| LONG | packetSize | Packet size in bytes |
| FLOAT | packetSizePercentage | Percentage of bandwidth |
| PixelFormat | pixelFormat | Pixel format |

2.1.9 Format7PacketInfo

| Type | Name | Description |
|------|---------------------------|----------------------------------|
| LONG | recommendedBytesPerPacket | Recommended packet size in bytes |
| LONG | maxBytesPerPacket | Maximum packet size in bytes |
| LONG | unitBytesPerPacket | Minimum packet size in bytes |

3 Properties

The property get/set functions always return a HRESULT in C++. The HRESULT can have several possible values:

| Name | Description |
|--------------|------------------|
| S_OK | Success |
| E_FAIL | Failure |
| E_INVALIDARG | Invalid argument |
| E_POINTER | Invalid pointer |

In VB and C#, the return value is the value of the property itself. An error or exception handler should always be used appropriately.

3.1 Control Properties

3.1.1 EnableRightClickMenu

Enables or disables the right click menu. The right click menu contains basic functionality such as starting and stopping the camera, as well as showing the camera control dialog. It is set to 0 (off) by default.

3.1.2 Display

Draw all incoming images onto control.

3.1.3 AutoResize

Automatically resize the image to fit the control. If the value is `FALSE`, then scroll bars will appear to enable movement around the image.

3.1.4 Font

The font to be used when drawing text to the screen.

3.2 General Camera Properties

3.2.1 Camera

Get or set the index of the currently selected camera. This **does not** start the camera. -1 is returned if there is no camera currently selected.

3.2.2 Start

Start grabbing images. This will enable isochronous data transmission from the camera to the PC. Depending on the grab mode, this may or may not automatically grab images. This **does not** draw the image to screen.

Setting this to 0 will stop the camera, ending isochronous data transmission.

3.2.3 VideoMode

Get or set the current video mode.

Setting the video mode is only possible if the Start property is 0.

3.2.4 FrameRate

Get or set the current frame rate.

Setting the frame rate is only possible if the Start property is 0.

3.2.5 AsyncBusSpeed

Get or set the current asynchronous bus speed (S100, S200, S400, S800).

Setting the bus speed is only possible if the Start property is 0.

3.2.6 IsochBusSpeed

Get or set the current isochronous bus speed (S100, S200, S400, S800).

Setting the bus speed is only possible if the Start property is 0.

3.2.7 GrabTimeout

Get or set the grab timeout.

0 is a non-blocking grab call.

-1 is equivalent to FLYCAPTURE_INFINITE.

4 Methods

4.1 Control Methods

4.1.1 DrawSingleImage

| Argument | Type | Description |
|----------|-----------|---------------------------------------|
| bConvert | [in] BOOL | Perform color processing on raw image |

Draw the latest image to the control. This is only used when the Display property is set to 0. If the internal raw image has been modified after using GetImageData() or GetImagePtr(), then the bConvert argument should be set to TRUE to perform color processing on the raw image again.

4.2 GUI Methods

4.2.1 ToggleCamCtlDlg

| Argument | Type | Description |
|----------|------|-------------|
| N/A | N/A | N/A |

Toggle the display of the camera control dialog.

4.2.2 IsCamCtlDlgVisible

| Argument | Type | Description |
|----------|---------------------|--|
| bShowing | [out, retval] BOOL* | Whether the camera control dialog is currently being shown |

Get the status of the camera control dialog.

4.2.3 ShowCamSelModal

| Argument | Type | Description |
|---------------|---------------------|--|
| bDialogStatus | [out, retval] BOOL* | Whether the Ok or Cancel button was clicked. TRUE for Ok, FALSE for Cancel |

Show the camera selection dialog. If the camera associated with this control is grabbing images, it will be stopped. Selecting a camera from the dialog will set the active camera to the selected one.

4.3 Camera Methods

4.3.1 GetCameraInfo

| Argument | Type | Description |
|----------|------------------------------|--|
| pInfo | [out, retval] CameraInfo* | A CameraInfo structure containing the camera information |

Get camera info.

4.3.2 CheckVideoMode

| Argument | Type | Description |
|-------------|---------------------|---|
| videoMode | [in] DCAMVideoMode | The DCAM video mode to test against |
| frameRate | [in] DCAMFrameRate | The DCAM frame rate to test against |
| pbSupported | [out, retval] BOOL* | Whether the video mode is supported by the camera |

Check if the specified video mode and frame rate combination is supported.

4.3.3 GetFormat7PacketInfo

| Argument | Type | Description |
|-------------|------------------------------|--|
| lMode | [in] LONG | The Format 7 mode to check |
| lWidth | [in] LONG | The width of the Format 7 image |
| lHeight | [in] LONG | The height of the Format 7 image |
| pixelFormat | [in] PixelFormat | The pixel format of the Format 7 image |
| pPacketInfo | [out, retval] PacketInfo* | A PacketInfo structure containing the packet information |

Get the Format 7 packet info for a particular mode, image size and pixel format.

4.3.4 GetPacketInfo

| Argument | Type | Description |
|-------------|------------------------------|--|
| videoMode | [in] DCAMVideoMode | The DCAM video mode to test against |
| frameRate | [in] DCAMFrameRate | The DCAM frame rate to test against |
| pPacketInfo | [out, retval] PacketInfo* | A PacketInfo structure containing the packet information |

Get the packet info for a particular DCAM mode

4.3.5 GetImageInformation

| Argument | Type | Description |
|------------|-----------------------------|--|
| pImageInfo | [out, retval] ImageInfo* | A ImageInfo structure containing the image information |

Get image information about the current image.

4.3.6 GetImageData

| Argument | Type | Description |
|----------|---------------------------|--|
| type | [in] ImageType | The type of image to return (raw or color processed) |
| pArray | [out, retval] VARIANT* | An array containing the image data |

Get the image data for the current image. This should not be used in .NET languages as a copy of the data will be automatically made instead.

4.3.7 GetImagePtr

| Argument | Type | Description |
|----------|---------------------------|--|
| type | [in] ImageType | The type of image to return (raw or color processed) |
| pValue | [out, retval] VARIANT* | An pointer to the array containing the image data |

Get a pointer to the image data. Although the value returned is a LONG, it can be cast to a pointer to access the image data directly.

4.4 1394 Bus Methods

4.4.1 GetNumOfCameras

| Argument | Type | Description |
|--------------|---------------------|--------------------------------|
| plNumCameras | [out, retval] LONG* | The number of cameras detected |

Get the number of cameras on the bus.

4.4.2 GetCameraList

| Argument | Type | Description |
|----------|---------------------------|--|
| pList | [out, retval] VARIANT* | An array containing the model names and serial numbers of the cameras on the bus |

Get an array containing a list of cameras on the bus. The contents of the array are strings containing the camera name as well as the serial number in parentheses.

4.5 Grab Methods

4.5.1 SetGrabMode

| Argument | Type | Description |
|----------|---------------|--|
| mode | [in] GrabMode | The grab mode that the control is to be set to |

Set the grab mode from the following:

- FreeRunning
- LockLatest
- LockNext

The default grab mode is FreeRunning.

4.5.2 GetGrabMode

| Argument | Type | Description |
|----------|----------------------------|-----------------------|
| pMode | [out, retval] GrabMode* | The current grab mode |

Get the current grab mode.

4.5.3 GrabImage

| Argument | Type | Description |
|----------|-----------------------------|----------------------------------|
| pError | [out, retval] GrabError* | The error returned from the grab |

Grab the next image from the camera. The image that is returned depends on the grab mode that is set. If LockLatest mode is set, then the newest image will be captured. If LockNext is set, then the oldest image will be captured. This call is not valid if the camera is in free running mode.

4.6 Image Saving Methods

4.6.1 SaveImage

| Argument | Type | Description |
|--------------|----------------------|--------------------------------------|
| bstrFilename | [in] BSTR | The path to save the file to |
| fileFormat | [in] ImageFileFormat | The file format to save the image in |

Save the current image.

4.6.2 SetJPEGCompressionQuality

| Argument | Type | Description |
|----------|-----------|----------------------------------|
| lQuality | [in] LONG | JPEG compression quality (0-100) |

Set the JPEG compression quality.

4.6.3 GetColorProcessingAlgorithm

| Argument | Type | Description |
|------------|---|----------------------------|
| pAlgorithm | [out, retval] ColorProcessingAlgorithm | Color processing algorithm |

Get the color processing algorithm being used.

4.6.4 SetColorProcessingAlgorithm

| Argument | Type | Description |
|-----------|-----------|----------------------------|
| algorithm | [in] LONG | Color processing algorithm |

Set the color processing algorithm to be used.

4.7 Format 7 Methods

4.7.1 GetFormat7Info

| Argument | Type | Description |
|----------|-------------------------------|--|
| lMode | [in] LONG | The Format 7 mode to retrieve information from |
| pStruct | [out, retval] Format7Info* | A Format7Info containing Format 7 information for the specified mode |

Get the possible settings for a particular Format 7 mode.

4.7.2 GetFormat7Configuration

| Argument | Type | Description |
|-----------------|--|---|
| pSettingsStruct | [out, retval] Format7ImageSettings* | A Format7ImageSettings containing the current Format 7 settings |

Get the current Format 7 settings.

4.7.3 SetFormat7Configuration

| Argument | Type | Description |
|-----------------|-------------------------------|---|
| pSettingsStruct | [in] Format7ImageSettings* | A Format7ImageSettings containing the Format 7 settings to be set |

Set the current Format 7 settings. If the image offsets are the only parameter being changed, then this can be done when the camera is started. Otherwise, the camera must be stopped in order for the call to succeed.

4.7.4 GetFormat7PacketSize

| Argument | Type | Description |
|-------------|-------------------------------------|--|
| pPacketInfo | [out, retval] Format7PacketInfo* | A Format7PacketInfo containing the current Format 7 settings |

Get the current Format 7 settings.

4.8 Camera Property Methods

These methods enable the user to get the camera properties.

4.8.1 GetProperty

| Argument | Type | Description |
|----------|----------------------------------|---|
| camProp | [in] CameraProperty | The camera property to get |
| pProp | [out, retval] CameraProperty* | A CameraProperty containing the data about the specified property |

Get detailed absolute property.

4.8.2 SetProperty

| Argument | Type | Description |
|----------|-----------------------|---|
| camProp | [in] CameraProperty | The camera property to set |
| pProp | [in] CameraProperty * | A CameraProperty containing the data to be set for the specified property |

Set detailed absolute property.

4.8.3 GetPropertyInfo

| Argument | Type | Description |
|-----------|--------------------------------------|---|
| camProp | [in] CameraProperty | The camera property to get |
| pPropInfo | [out, retval] CameraPropertyInfo* | A CameraProperty containing the data about the specified property |

Get detailed information about a property.

4.9 Trigger / Strobe Methods

4.9.1 GetTrigger

| Argument | Type | Description |
|----------|---------------------------------|---|
| pStruct | [out, retval] TriggerStruct* | A TriggerStruct containing the trigger data |

Get trigger status.

4.9.2 SetTrigger

| Argument | Type | Description |
|----------|---------------------|---|
| pStruct | [in] TriggerStruct* | A TriggerStruct containing the trigger data to be set |

Set trigger status.

4.9.3 SetTriggerBroadcast

| Argument | Type | Description |
|----------|---------------------|---|
| pStruct | [in] TriggerStruct* | A TriggerStruct containing the trigger data to be set |

Set and broadcast trigger status.

4.9.4 GetStrobe

| Argument | Type | Description |
|----------|-----------------------------|---|
| lMode | [in] LONG | The strobe source to retrieve |
| pStruct | [out, retval] StrobeStruct* | A StrobeStruct containing the strobe data |

Get strobe status.

4.9.5 SetStrobe

| Argument | Type | Description |
|----------|--------------------|---|
| lMode | [in] LONG | The strobe source to set |
| pStruct | [in] StrobeStruct* | A StrobeStruct containing the strobe data to be set |

Set strobe status.

4.9.6 SetStrobeBroadcast

| Argument | Type | Description |
|----------|--------------------|---|
| lMode | [in] LONG | The strobe source to set |
| pStruct | [in] StrobeStruct* | A StrobeStruct containing the strobe data to be set |

Set and broadcast strobe broadcast.

4.10 Memory Channel Methods

4.10.1 GetNumMemoryChannels

| Argument | Type | Description |
|--------------|---------------------|---|
| pNumChannels | [out, retval] LONG* | The number of available memory channels |

Get the number of memory channels available.

4.10.2 GetCurrentMemoryChannel

| Argument | Type | Description |
|-----------------|---------------------|----------------------------|
| pCurrentChannel | [out, retval] LONG* | The current memory channel |

Get the number of the memory channel currently being used.

4.10.3 SaveToMemoryChannel

| Argument | Type | Description |
|-----------------|------------|-------------------------------|
| lCurrentChannel | [in] LONG* | The memory channel to save to |

Save current settings to the specified memory channel.

4.10.4 RestoreFromMemoryChannel

| Argument | Type | Description |
|----------|------------|---------------------------------|
| lChannel | [in] LONG* | The memory channel to load from |

Restore settings from the specified memory channel.

4.11 Register Methods

4.11.1 ReadRegister

| Argument | Type | Description |
|-----------|------------|---------------------------|
| lRegister | [in] LONG | The register to retrieve |
| pIValue | [in] LONG* | The value of the register |

Get the register value at the specified offset.

4.11.2 WriteRegister

| Argument | Type | Description |
|-----------|-----------|---------------------------------|
| lRegister | [in] LONG | The register to retrieve to set |
| lValue | [in] LONG | The value of the register |

Set the register value at the specified offset.

4.11.3 ReadRegisterBlock

| Argument | Type | Description |
|---------------|------------|---|
| sAddrHigh | [in] LONG | The top 16 bits of the 48-bit absolute address to read |
| lAddrLow | [in] LONG | The bottom 32 bits of the 48-bit absolute address to read |
| pBuffer | [in] LONG* | The buffer to receive the data |
| lBufferLength | [in] LONG | The length of the data buffer, in quadlets |

Get the register block at the specified offset.

4.11.4 WriteRegisterBlock

| Argument | Type | Description |
|---------------|------------|--|
| sAddrHigh | [in] LONG | The top 16 bits of the 48-bit absolute address to write |
| lAddrLow | [in] LONG | The bottom 32 bits of the 48-bit absolute address to write |
| pBuffer | [in] LONG* | The buffer of data to be written |
| lBufferLength | [in] LONG | The length of the data buffer, in quadlets |

Set the register block at the specified offset.

4.12 Drawing Methods

All coordinates used for drawing are relative to the image origin before any auto-resizing (if applicable) has taken place.

4.12.1 DrawEllipse

| Argument | Type | Description |
|----------|------------|---------------------------------|
| X1 | [in] SHORT | X coordinate of the start point |
| Y1 | [in] SHORT | Y coordinate of the start point |
| X2 | [in] SHORT | X coordinate of the end point |
| Y2 | [in] SHORT | Y coordinate of the end point |
| sWidth | [in] SHORT | Width of the line |
| redVal | [in] SHORT | Red value |
| greenVal | [in] SHORT | Green value |
| blueVal | [in] SHORT | Blue value |

Draw an ellipse bounded by the coordinates (X1, Y1) and (X2, Y2).

4.12.2 DrawLine

| Argument | Type | Description |
|----------|------------|---------------------------------|
| X1 | [in] SHORT | X coordinate of the start point |
| Y1 | [in] SHORT | Y coordinate of the start point |
| X2 | [in] SHORT | X coordinate of the end point |
| Y2 | [in] SHORT | Y coordinate of the end point |
| sWidth | [in] SHORT | Width of the line |
| redVal | [in] SHORT | Red value |
| greenVal | [in] SHORT | Green value |
| blueVal | [in] SHORT | Blue value |

Draw a line from the coordinates (X1, Y1) to (X2, Y2).

4.12.3 DrawPixel

| Argument | Type | Description |
|----------|------------|---------------------------|
| XPos | [in] SHORT | X coordinate of the pixel |
| YPos | [in] SHORT | Y coordinate of the pixel |
| redVal | [in] SHORT | Red value |

| | | |
|----------|------------|-------------|
| greenVal | [in] SHORT | Green value |
| blueVal | [in] SHORT | Blue value |

Draw a pixel at the coordinates (XPos, YPos) with the specified RGB value. This does not overwrite the actual image data.

4.12.4 DrawRectangle

| Argument | Type | Description |
|----------|------------|---------------------------------|
| X1 | [in] SHORT | X coordinate of the start point |
| Y1 | [in] SHORT | Y coordinate of the start point |
| X2 | [in] SHORT | X coordinate of the end point |
| Y2 | [in] SHORT | Y coordinate of the end point |
| sWidth | [in] SHORT | Width of the line |
| redVal | [in] SHORT | Red value |
| greenVal | [in] SHORT | Green value |
| blueVal | [in] SHORT | Blue value |

Draw a rectangle bounded by the coordinates (x1, y1) and (x2, y2).

4.12.5 DrawText

| Argument | Type | Description |
|----------|------------|---------------------------------|
| XPos | [in] SHORT | X coordinate of the start point |
| YPos | [in] SHORT | Y coordinate of the start point |
| bstrText | [in] BSTR | The text to be written |
| redVal | [in] SHORT | Red value |
| greenVal | [in] SHORT | Green value |
| blueVal | [in] SHORT | Blue value |

Draw the specified text at the coordinates (XPos, YPos).

5 Events

5.1 General Events

5.1.1 CameraArrival

A new camera has arrived on the bus. Depending on the bus topology, the index used to refer to cameras may have changed. It is highly recommended that `GetCameraList()` be called to refresh the camera list.

5.1.2 CameraRemoval

A camera has been removed from the bus. Depending on the bus topology, the index used to refer to cameras may have changed. It is highly recommended that `GetCameraList()` be called to refresh the camera list.

5.1.3 BusReset

A bus reset has occurred.

5.1.4 Image

A new image has been received from the camera. `GetImageInformation()` can be called when this message is received in order to obtain more information about the image.

6 Additional Resources

For more detailed information regarding advanced camera functionality such as trigger modes, please see the documentation included in your FlyCapture install. These files can be typically be found in C:\Program Files\Point Grey Research\PGR FlyCapture\doc.

6.1 Getting Started Manual

A *Getting Started Manual* provides a broad overview of the general capabilities of a specific camera. It contains basic information such as physical dimensions and general camera features, as well as instructions on installation of the camera.

6.2 Technical Reference Manual

A *Technical Reference Manual* provides in-depth information regarding a specific camera. It contains full specifications for a camera, including advanced camera-specific features such as frame buffers, HDR modes and trigger modes.

6.3 Point Grey Digital Camera Register Reference

The *Point Grey Digital Camera Register Reference* provides detailed information about the various features, formats and control parameters implemented by each PGR IEEE-1394 camera. It should be used in conjunction with the camera specific *Technical Reference Manual* or *Getting Started Manual* for a full understanding of a specific camera system.

7 Technical Support Resources

Point Grey Research Inc. endeavors to provide the highest level of technical support possible to our customers. Most support resources can be accessed through the Product Support section of our website: www.ptgrey.com/support.

Creating a Customer Login Account

The first step in accessing our technical support resources is to obtain a Customer Login Account. This requires a valid name, e-mail address, and camera serial number. To apply for a Customer Login Account go to www.ptgrey.com/support/downloads/.

Knowledge Base

Our on-line knowledge base at www.ptgrey.com/support/kb/ contains answers to some of the most common support questions. It is constantly updated, expanded, and refined to ensure that our customers have access to the latest information.

Product Downloads

Customers with a Customer Login Account can access the latest software and firmware for their cameras from our downloads site at www.ptgrey.com/support/downloads. We encourage our customers to keep their software and firmware up-to-date by downloading and installing the latest versions.

Contacting Technical Support

Before contacting Technical Support, have you:

1. *Read the product documentation and user manual?*
2. *Searched the Knowledge Base?*
3. *Downloaded and installed the latest version of software and/or firmware?*

If you have done all the above and still can't find an answer to your question, contact our Technical Support team at www.ptgrey.com/support/contact/.

8 Contacting Point Grey Research Inc.

For any questions, concerns or comments please contact us via the following methods:

Email: For all general questions about Point Grey Research please contact us at info@ptgrey.com.

For technical support (existing customers only) contact us at <http://www.ptgrey.com/support/contact/>.

Knowledge Base: Find answers to commonly asked questions in our knowledge base at <http://www.ptgrey.com/support/kb/>.

Downloads: Users can download the latest manuals and software from <http://www.ptgrey.com/support/downloads/>

| | | |
|---------------------|--|---|
| Main Office: | Mailing Address: Point Grey Research, Inc. 12051 Riverside Way Richmond, BC, Canada V6W 1K7 | Tel: +1 (604) 242-9937 Toll-free (North America only): +1 (866) 765-0827 Fax: +1 (604) 242-9938 sales@ptgrey.com |
|---------------------|--|---|

Distributors

| | |
|-----|---|
| USA | Tel: +1 (866) 765-0827 na-sales@ptgrey.com |
|-----|---|

| | | |
|------------------|---|---|
| Europe Israel | Mailing Address: Point Grey Research GmbH Schwieberdinger Strasse 60 71636 Ludwigsburg Germany | Tel: +49 7141 488817-0 Fax: +49 7141 488817-99 eu-sales@ptgrey.com |
|------------------|---|---|

| | |
|-------|---|
| Japan | ViewPLUS Inc. (http://www.viewplus.co.jp/) |
|-------|---|

| | |
|-------|--|
| Korea | Cylod Co. Ltd. (http://www.cylod.com) |
|-------|--|

| | |
|-------|--|
| China | LUSTER LightVision Tech. Co., Ltd (www.lusterlighttech.com) |
|-------|--|

| | |
|-----------------------------------|--|
| Singapore Malaysia Thailand | Voltrium Systems Pte Ltd. (www.voltrium.com.sg) |
|-----------------------------------|--|

| | |
|--------|---|
| Taiwan | Apo Star Co., Ltd. (www.apostar.com.tw) |
|--------|---|