Camera Interface Standard 1.0
Programmer's Reference

Please conduct discussions regarding
this interface and its usage on the
ASCOM-Talk List.
Click here to join ASCOM-Talk

Introduction

This document describes the interface used by low-level telescope "camera" components as part of the Astronomy Common Object Model (ASCOM). Components that implement this interface can provide way for programs to control various CCD cameras via a standard set of properties and methods.

This specification covers a simple, low-level camera control interface for taking exposures, reading out the image, and accessing common functions such as controlling the cooler. It does not provide for accessories such as filter wheels. Those sorts of devices have their own interfaces. The characteristics of this interface comply with the ASCOM Quality Guidelines, assuring consistent behavior and compatibility with the widest possible variety of Windows Automation clients.

For a summary of the changes and additions, please refer to the Release Notes.

Implementation and Conventions

All of the methods and properties specified must be present in any conforming driver's interface. However, some methods and properties may not actually be implemented; they will raise a "not implemented" error when called or accessed.

Usage

To control a particular camera type, a program would create an instance of the driver for that camera type, then use the standard properties and methods described in this document to effect control of that camera. Thus any program or script that uses the standard driver interface automatically gains access to any camera type for which a driver exists. For more information on ASCOM, see the ASCOM Home Page.

Release Notes

3-July-2007

Removed return value from StartExposure method; was a documentation error. (D. George)

3-July-2007

Standard document transferred to HTMLHelp. Minor edits. (D. George)

18-Aug-2006

Standards adopted December 2005, first driver developed and available (Finger Lakes Instruments). Revision to remove "proposed" language and reorganize into two clearly separate interface sections. Corrected spelling of ElectronsPerADU, added missing IsPulseGuiding property, changed BSTR to String.

Camera Object

Remarks

Use this interface to perform basic operations on a CCD camera. It is designed to be used by client applications that wish to provide camera control capabilities that are device-independent. By using this interface, your application will be freed from dealing with the details of the low-level protocols in use by various cameras.

Discovery API

Not all of the features in this interface are required to be supported by all drivers and camera. This permits support for a wide range of camera types without the evils of least-common-denominator designs. Camera includes properties that permit discovery of supported features.

Available features can be determined by reading one of the Can___ properties, all of which must be supported. These Can___ properties are intended to declare only that the camera supports the particular capability. They are not intended to reflect the current state of the camera. Furthermore, connecting to the camera may be required in order to determine the capabilities to reflect in the Can___ properties (an error will be raised if a Can___ property is not available). Once connected, however, the Can___ properties must not change.

Concurrency

This interface does not provide for concurrency control. In the interest of keeping things simple at this level, the architecture assumes that client applications will refrain from performing conflicting operations.

BinX

Property

Camera.BinX (Short)

Syntax

Camera.BinX [= Short]

Exceptions

Must throw an exception for illegal binning values

Remarks

Sets the binning factor for the X axis. Also returns the current value. Should default to 1 when the camera link is established. Note: driver does not check for compatible subframe values when this value is set; rather they are checked upon StartExposure.

BinY

Property

Camera.BinY (Short)

Syntax

Camera.BinY [= Short]

Exceptions

Must throw an exception for illegal binning values

Remarks

Sets the binning factor for the Y axis Also returns the current value. Should default to 1 when the camera link is established. Note: driver does not check for compatible subframe values when this value is set; rather they are checked upon StartExposure.

CameraState

Property

Camera.CameraState (read only, enumeration)

Syntax

Camera.CameraState

Exceptions

Must return an exception if the camera status is unavailable.

Remarks

Returns one of the following status information:

Value	State	Meaning
0	CameraIdle	At idle state, available to start exposure
1	CameraWaiting	Exposure started but waiting (for shutter, trigger, filter wheel, etc.)
2	CameraExposing	Exposure currently in progress
3	CameraReading	Camera array is being read out (digitized)
4	CameraDownload	Downloading data to PC
5	CameraError	Camera error condition serious enough to preventfurther operations (link fail, etc.)

CameraXSize

Property

Camera.CameraXSize (read only, Long)

Syntax

Camera.CameraXSize

Exceptions

Must throw exception if the value is not known

Remarks

Returns the width of the camera sensor in unbinned pixels.

CameraYSize

Property

Camera.CameraYSize (read only, Long)

Syntax

Camera.CameraYSize

Exceptions

Must throw exception if the value is not known

Remarks

Returns the height of the camera sensor in unbinned pixels.

CanAbortExposure

Property

Camera.CanAbortExposure (read only, Boolean)

Syntax

Camera.CanAbortExposure

Exceptions

None

Remarks

Returns True if the camera can abort exposures; False if not.

CanAsymmetricBin

Property

Camera.CanAsymmetricBin (read only, Boolean)

Syntax

Camera.CanAsymmetricBin

Exceptions

Must throw exception if the value is not known (n.b. normally only occurs if no link established and camera must be queried)

Remarks

If True, the camera can have different binning on the X and Y axes, as determined by BinX and BinY. If False, the binning must be equal on the X and Y axes.

CanGetCoolerPower

Property

Camera.CanGetCoolerPower (read only, Boolean)

Syntax

Camera.CanGetCoolerPower

Exceptions

Must throw exception if the value is not known (n.b. normally only occurs if no link established and camera must be queried)

Remarks

If True, the camera can return the cooler power level. If False, this information is not available. The cooler power level is normally regulated internally to the camera, based on the temperature setpoint.

CanPulseGuide

Property

Camera.CanPulseGuide (read only, Boolean)

Syntax

Camera.CanPulseGuide

Exceptions

None

Remarks

Returns True if the camera can send autoguider pulses to the telescope mount; False if not. (Note: this does not provide any indication of whether the autoguider cable is actually connected.)

CanSetCCDTemperature

Property

Camera.CanSetCCDTemperature (read only, Boolean)

Syntax

Camera.CanSetCCDTemperature

Exceptions

None

Remarks

If True, the camera's cooler setpoint can be adjusted. If False, the camera either uses open-loop cooling or does not have the ability to adjust temperature from software, and setting the TemperatureSetpoint property has no effect.

CanStopExposure

Syntax

Camera.CanStopExposure (read only, Boolean)

Syntax

Camera.CanStopExposure

Exceptions

Must throw exception if not supported. Must throw exception if an error condition such as link failure is present

Remarks

Some cameras support StopExposure, which allows the exposure to be terminated before the exposure timer completes, but will still read out the image. Returns True if StopExposure is available, False if not.

CCDTemperature

Property

Camera.CCDTemperature (read only, Double)

Syntax

Camera.CCDTemperature

Exceptions

Must throw exception if data unavailable.

Remarks

Returns the current CCD temperature in degrees Celsius. Only valid if CanControlTemperature is True.

Connected

Property

Camera.Connected (Boolean)

Syntax

Camera.Connected [= Boolean]

Exceptions

Must throw exception if unsuccessful.

Remarks

Controls the link between the driver and the camera. Set True to enable the link. Set False to disable the link (this does not switch off the cooler). You can also read the property to check whether it is connected.

CoolerOn

Property

Camera.CoolerOn (Boolean)

Syntax

Camera.CoolerOn [= Boolean]

Exceptions

Must throw exception if not supported. Must throw exception if an error condition such as link failure is present

Remarks

Turns on and off the camera cooler, and returns the current on/off state. Warning: turning the cooler off when the cooler is operating at high delta-T (typically -20C below ambient) may result in thermal shock. Repeated thermal shock may lead to damage to the sensor or cooler stack. Please consult the documentation supplied with the camera for further information.

CoolerPower

Property

Camera.CoolerPower (read only, Double)

Syntax

Camera.CoolerPower

Exceptions

Must throw exception if not supported. Must throw exception if an error condition such as link failure is present

Remarks

Returns the present cooler power level, in percent. Returns zero if CoolerOn is False.

Description

Property

Camera.Description (read only, String)

Syntax

Camera.Description

Exceptions

Must throw exception if description unavailable

Remarks

Returns a description of the camera model, such as manufacturer and model number. Any ASCII characters may be used. The string shall not exceed 68 characters (for compatibility with FITS headers).

ElectronsPerADU

Property

Camera.ElectronsPerADU (read only, Double)

Syntax

Camera.ElectronsPerADU

Exceptions

Must throw exception if data unavailable.

Remarks

Returns the gain of the camera in photoelectrons per A/D unit. (Some cameras have multiple gain modes; these should be selected via the SetupDialog and thus are static during a session.)

FullWellCapacity

Property

Camera.FullWellCapacity (read only, Double)

Syntax

Camera.FullWellCapacity

Exceptions

Must throw exception if data unavailable.

Remarks

Reports the full well capacity of the camera in electrons, at the current camera settings (binning, SetupDialog settings, etc.)

HasShutter

Property

Camera.HasShutter (read only, Boolean)

Syntax

Camera.HasShutter

Exceptions

None

Remarks

If True, the camera has a mechanical shutter. If False, the camera does not have a shutter. If there is no shutter, the StartExposure command will ignore the Light parameter.

HeatSinkTemperature

Property

Camera.HeatSinkTemperature (read only, Double)

Syntax

Camera.HeatSinkTemperature

Exceptions

Must throw exception if data unavailable.

Remarks

Returns the current heat sink temperature (called "ambient temperature" by some manufacturers) in degrees Celsius. Only valid if CanControlTemperature is True.

ImageArray

Property

Camera.ImageArray (read only, Long)

Syntax

Camera.ImageArray

Exceptions

Must throw exception if data unavailable.

Remarks

Returns a safearray of Long of size NumX * NumY containing the pixel values from the last exposure. The application must inspect the Safearray parameters to determine the dimensions. Note: if NumX or NumY is changed after a call to StartExposure it will have no effect on the size of this array. This is the preferred method for programs (not scripts) to download iamges since it requires much less memory.

For color or multispectral cameras, will produce an array of NumX * NumY * NumPlanes. If the application cannot handle multispectral images, it should use just the first plane.

ImageArrayVariant

Property

Camera.ImageArrayVariant (read only, Variant)

Syntax

Camera.ImageArrayVariant

Exceptions

Must throw exception if data unavailable.

Remarks

Returns a safearray of Variant of size NumX * NumY containing the pixel values from the last exposure. The application must inspect the Safearray parameters to determine the dimensions. Note: if NumX or NumY is changed after a call to StartExposure it will have no effect on the size of this array. This property should only be used from scripts due to the extremely high memory utilization on large image arrays (26 bytes per pixel). Pixels values should be in Short, Long, or Double format.

For color or multispectral cameras, will produce an array of NumX * NumY * NumPlanes. If the application cannot handle multispectral images, it should use just the first plane.

ImageReady

Property

Camera.ImageReady (read only, Boolean)

Syntax

Camera.ImageReady

Exceptions

Must throw exception if hardware or communications link error has occurred.

Remarks

If True, there is an image from the camera available. If False, no image is available and attempts to use the ImageArray method will produce an exception.

IsPulseGuiding

Property

Camera.IsPulseGuiding (read only, Boolean)

Syntax

Camera.IsPulseGuiding

Exceptions

Must throw exception if hardware or communications link error has occurred.

Remarks

If True, pulse guiding is in progress. Required if the PulseGuide() method (which is non-blocking) is implemented. See the PulseGuide() method.

LastError

Property

Camera.LastError (read only, String)

Syntax

Camera.LastError

Exceptions

Must throw exception if no error condition.

Remarks

Reports the last error condition reported by the camera hardware or communications link. The string may contain a text message or simply an error code. The error value is cleared the next time any method is called.

LastExposureDuration

Property

Camera.LastExposureDuration (read only, Double)

Syntax

Camera.LastExposureDuration

Exceptions

Must throw exception if not supported or no exposure has been taken

Remarks

Reports the actual exposure duration in seconds (i.e. shutter open time). This may differ from the exposure time requested due to shutter latency, camera timing precision, etc.

LastExposureStartTime

Property

Camera.LastStartTime (read only, String)

Syntax

Camera.LastStartTime

Exceptions

Must throw exception if not supported or no exposure has been taken

Remarks

Reports the actual exposure start in the FITS-standard CCYY-MM-DDThh:mm:ss[.sss...] format.

MaxADU

Property

Camera.MaxADU (read only, Long)

Syntax

Camera.MaxADU

Exceptions

Must throw exception if data unavailable.

Remarks

Reports the maximum ADU value the camera can produce.

MaxBinX

Property

Camera.MaxBinX (read only, Short)

Syntax

Camera.MaxBinX

Exceptions

Must throw exception if data unavailable.

Remarks

If AsymmetricBinning = False, returns the maximum allowed binning factor. If AsymmetricBinning = True, returns the maximum allowed binning factor for the X axis.

MaxBinY

Property

Camera.MaxBinY (read only, Short)

Syntax

Camera.MaxBinY

Exceptions

Must throw exception if data unavailable.

Remarks

If AsymmetricBinning = False, equals MaxBinX. If AsymmetricBinning = True, returns the maximum allowed binning factor for the Y axis.

NumX

Property

Camera.NumX (Long)

Syntax

Camera.NumX [= Long]

Exceptions

None

Remarks

Sets the subframe width. Also returns the current value. If binning is active, value is in binned pixels. No error check is performed when the value is set. Should default to CameraXSize.

NumY

Property

Camera.NumY (Long)

Syntax

Camera.NumY [= Long]

Exceptions

None

Remarks

Sets the subframe height. Also returns the current value. If binning is active, value is in binned pixels. No error check is performed when the value is set. Should default to CameraYSize.

PixelSizeX

Property

Camera.PixelSizeX(read only, Double)

Syntax

Camera.PixelSizeX

Exceptions

Must throw exception if data unavailable.

Remarks

Returns the width of the sensor chip pixels in microns, as provided by the camera driver.

PixelSizeY

Property

Camera.PixelSizeY(read only, Double)

Syntax

Camera.PixelSizeY

Exceptions

Must throw exception if data unavailable.

Remarks

Returns the height of the sensor chip pixels in microns, as provided by the camera driver.

SetCCDTemperature

Property

Camera.SetCCDTemperature (Double)

Syntax

Camera.SetCCDTemperature [= Double]

Exceptions

Must throw exception if command not successful.
Must throw exception if CanSetCCDTemperature is False.

Remarks

Sets the camera cooler setpoint in degrees Celsius, and returns the current setpoint.

Note: camera hardware and/or driver should perform cooler ramping, to prevent thermal shock and potential damage to the CCD array or cooler stack.

StartX

Property

Camera.StartX (Long)

Syntax

Camera.StartX [= Long]

Exceptions

None

Remarks

Sets the subframe start position for the X axis (0 based). Also returns the current value. If binning is active, value is in binned pixels.

StartY

Property

Camera.StartY (Long)

Syntax

Camera.StartY [= Long]

Exceptions

None

Remarks

Sets the subframe start position for the Y axis (0 based). Also returns the current value. If binning is active, value is in binned pixels.

AbortExposure

Syntax

Camera.AbortExposure()

Parameters

None.

Returns

Nothing.

Exceptions

Must throw exception if camera is not idle and abort is unsuccessful (or not possible, e.g. during download).
Must throw exception if hardware or communications error occurs.
Must NOT throw an exception if the camera is already idle.

Remarks

Aborts the current exposure, if any, and returns the camera to Idle state.

PulseGuide

Syntax

Camera.PulseGuide(Direction, Duration)

Parameters

GuideDirections Direction - direction in which the guide-rate motion is to be made
Long Duration - Duration of the guide-rate motion (milliseconds)

Returns

Nothing.

Exceptions

Must throw exception if PulseGuide command is unsupported or the command is unsuccessful.

Remarks

This method may return immediately after the move has started, in which case back-to-back dual axis pulse-guiding can be supported. Use the IsPulseGuiding property to detect when all moves have completed.

Symbolic Constants

The (symbolic) values for GuideDirections are:

Value	Constant	Description
0	guideNorth	North (+ declination/elevation)
1	guideSouth	South (- declination/elevation)
2	guideEast	East (+ right ascension/azimuth)
3	guideWest	West (+ right ascension/azimuth)

Note: directions are nominal and may depend on exact mount wiring. guideNorth must be opposite guideSouth, and guideEast must be opposite guideWest.

SetupDialog

Syntax

Camera.SetupDialog

Parameters

None.

Returns

Nothing.

Exceptions

Must throw an exception if Setup dialog is unavailable.

Remarks

Launches a configuration dialog box for the driver. The call will not return until the user clicks OK or cancel manually.

StartExposure

Syntax

Camera.StartExposure (Duration, Light)

Parameters

Double Duration - Duration of exposure in seconds
Boolean Light - True for light frame, False for dark frame (ignored if no shutter)

Returns

Nothing.

Exceptions

Must throw exception if NumX, NumY, XBin, YBin, StartX, StartY, or Duration parameters are invalid.
Must throw exception if CanAsymmetricBin is False and BinX <> BinY
Must throw exception if the exposure cannot be started for any reason, such as a hardware or communications error

Remarks

Starts an exposure. Use ImageReady to check when the exposure is complete.

StopExposure

Syntax

Camera.StopExposure()

Parameters

None.

Returns

Nothing.

Exceptions

Must throw an exception if CanStopExposure is False
Must throw an exception if no exposure is in progress
Must throw an exception if the camera or link has an error condition
Must throw an exception if for any reason no image readout will be available.

Remarks

Stops the current exposure, if any. If an exposure is in progress, the readout process is initiated. Ignored if readout is already in process.