CMOS Camera Specific Options:

This documents outlines the different camera specific options available on the following CMOS cameras:

1) Atik Horizon / Atik Horizon 2
2) ACIS range of camera

Most features apply to both cameras, but a note will be made if cameras do not have the feature.

The following features can be set using the Camera Specific Options:

1) Gain / Offset
2) Pad Data
3) Even Illumination
4) Fast Mode

Please note: Gain / Offset for these cameras is only available via Camera Specific Options. The ArtemisSetGain /ArtemisGetGain will not work for these cameras.

Camera Specific Options Functions:

In order to check / update the settings, you will need the following functions (described below):

1) ArtemisHasCameraSpecificOption
2) ArtemisCameraSpecificOptionGetData
3) ArtemisCameraSpecificOptionSetData

All functions require a handle to the camera (which is returned from the ArtemisConnect method) as well as an ID which defines the option you are using. The IDs are as follows:

```
ID_GOPresetMode             = 1;
ID_GOPresetLow              = 2;
ID_GOPresetMed              = 3;
ID_GOPresetHigh             = 4;
ID_GOCustomGain             = 5;
ID_GOCustomOffset           = 6;

ID_EvenIllumination          = 12;
ID_PadData                   = 13;
ID_ExposureSpeed             = 14;
ID_BitSendMode               = 15;

ID_FX3Version  = 200;
ID_FPGAVersion = 201;
```

These will be described below.

ArtemisHasCameraSpecificOption

```
bool    ArtemisHasCameraSpecificOption(ArtemisHandle handle,  unsigned short id);
```

The ArtemisHasCameraSpecificOption is used to check whether the camera has the requested option. This function can be called on all cameras (including non-CMOS cameras).

The function will return false if:

1) The option isn't available on the camera (Including if the ID doesn't exist)
2) The camera is not found

Otherwise it will return true.

ArtemisCameraSpecificOptionGetData

```
int     ArtemisCameraSpecificOptionGetData(ArtemisHandle handle, unsigned short id,
unsigned char * data, int dataLength, int &actualLength);
```

This function is used to get the current value of a given option. The function will return 'ARTEMIS_INVALID_PARAM' if the option isn't available on the camera. Otherwise it will return 'ARTEMIS_OK'.

To call this method, you need to supply an 'unsigned char *' which will function will populate with the result. You also need to pass the length of that array (This is to avoid any memory issue created by attempting to write to an array which is too small). The function also returns an 'actualLength' param which says how many bytes the resulting data actually was. In general, actualLength and dataLength should be the same. This length depends on the data being returned.

Return Type	Length	Notes

Bool	1	
Unsigned short	2	
GainOffsetPreset	5	Used for Low,Med,High Presets. The bytes are as follows: 0: hasPreset (bool) (should always be true, used in factory) 1-2: Gain Value (unsigned short) 3-4: OffsetValue (unsigned short) You should not attempt to change these values
Unsigned short range	Get = 6 Set = 2	Used for Custom Gain / Offset To Get the current values (bytes): 0-1: min (unsigned short) 2-3: max (unsigned short) 4-5: current (unsigned short) To set the current value (bytes) 0-1: value (unsigned short)
Version	6	To get the version (bytes): 0-1: Major (unsigned short) 2-3: Minor (unsigned short) 4-5: Patch (unsigned short) You cannot set this function

ArtemisCameraSpecificOptionSetData

```
int      ArtemisCameraSpecificOptionSetData(ArtemisHandle handle, unsigned short id,
unsigned char * data, int dataLength);
```

This function is used to set the current value of a given option. The function will return 'ARTEMIS_INVALID_PARAM' if the option isn't available on the camera. Otherwise it will return 'ARTEMIS_OK'.

You will need to pass the value in via the 'data' param. You also need to supply the data length (See table above).

Version:

The CMOS cameras contain two parts:

1) FX3
2) FPGA

They both have different version numbers (they usually won't match). The FPGA and FX3 firmware is always uploaded by the dll, so version numbers will change depending on the dll version.

To get the version number:

ID = `ID_FX3Version or ID_FPGAVersion`

`Data (6 bytes)`

`0-1` `Major`

2-3 Minor

4-5 Patch

Gain / Offset:

The camera has three gain / offset presets: Low, Medium, and High. These are calibrated in the factory and shouldn't be changed. You can also set the gain/offset into 'Custom' mode, which allows you to set the gain/offset to any value within range:

To Set the Preset mode, you need to call 'ArtemisCameraSpecificOptionSetData' with the following

ID = `ID_GOPresetMode`

`Data (unsigned short):`

1) Custom = 0
2) Low = 1
3) Med = 2
4) High = 3

You can find out the current mode, you can call 'ArtemisCameraSpecificOptionGetData' with the following:

ID = `ID_GOPresetMode`

`Data (unsigned short):`

0-1 Mode

To get the current custom offset / gain values:

ID = `ID_GOCustomGain or ID_GOCustomOffset`

`Data (6 bytes)`

`0-2` `Min`

2-3 Max

4-5 Current

To set the custom offset / gain:

ID = `ID_GOCustomGain or ID_GOCustomOffset`

Data (unsigned short)

0-1 Value

The custom gain/offset range is:

Camera Type	**Gain**	**Offset**
Horizon / Horizon 2	0 to 60 Note: 0 to 30 is the 'useful' range	0 to 511
ACIS	0 to 24	0 to 4095

Note: Any changed to these options will be saved on the camera and persist even after a power cycle.

Pad Data:

The CMOS sensor we use have a 12-bit output, however, the SDK produces 16-bit images. The pad data option controls whether the images use the upper 12 bits or the lower 12 bits. With 'PadData = true' images will be produced using the upper 12 bits (I.e. all values will be in the range 0-35535, but all values will be a multiple of 16). With 'PadData = False' the lower 12 bits will be used (i.e. all values will be 0-4095).

To get the current value:

ID = `ID_PadData`

`Data (bool)`

`To set the current value`

ID = `ID_PadData`

`Data (bool)`

Even Illumination:

This function is only available on the Horizon / Horizon 2. And only in power save mode.

In general you would have this function switch on.

With even illumination on, the image will clear the sensor before taking an image. This crates an image which is more uniformly exposed, but with slightly higher read noise.

With even illumination off, the image will not be cleared before exposing, the read noise is lower, but you will see a slight gradient across the image.

To get the current value:

ID = `ID_EvenIllumination`

`Data (bool)`

`To set the current value`

ID = `ID_ EvenIllumination`

`Data (bool)`

Fast Mode:

This function is only available on the Atik Horizon 2 and ACIS cameras. (Not Atik Horizon)

Exposure Speed:

There are three exposure speeds available:

1) Power Save: Returns the camera to power save mode between exposures. This is the slowest exposure mode, but has the lowest read noise.
2) Normal Mode: Doesn't return to standby mode between exposures but waits for the user to start the next image. This is faster than power save mode, but slightly noisier.
3) Fast Mode: In this mode, the camera can produce a continuous stream of images. This is the fastest mode of operation. (See 'Fast Mode' below)

Bit Send Mode

There are also two different bit send modes:

1) 16-bit: This is the default setting. Data from the camera is sent as 16bit. This is a slower transfer rate, but no processing is required by the PC
2) 12-bit: Data is sent as 12-bits. Data transfer from the camera is faster, but some processing is required by the PC.

In general, the data transfer from the camera to the PC is the slow part. Setting the bit send mode to 12-bits should thus increase the frame rate, however, this causes the PC to do more work, which could potentially slow other parts of the program down. It is up to the user to determine which mode works best for them.

Fast Mode:

To maximise the frame rate, you need to use the 'ArtemisStartFastExposure' method:

```
BOOL ArtemisStartFastExposure(ArtemisHandle handle, int ms)
```

The camera must be in Fast mode for this function to succeed. Images will not be returned in the usual way, but instead, you will need to supply a function callback, which will be called as each image arrives:

```
typedef void(*FastModeCallback)(ArtemisHandle handle, int x, int y, int w, int h, int binx, int biny, void * imageBuffer);

Bool ArtemisSetFastCallback(ArtemisHandle handle, FastModeCallback callback)
```

This function will set a function callback which will be called everytime an image arrived. This will be called in the download thread, so it is important that you handle this properly. It is advised that you copy this image into your applications thread as quickly as possible, because the download thread will not continue onto the next image until this function has returned.

Note: Once started, any changes to the exposure (such as exposure time, binning, subframe, gain/offset) will be ignore. It is not possible to change these settings and run the camera at high frame rates. If you want to change the settings, you need to call 'ArtemisStopExposure' first.

In summary: In order to run the camera at the highest frame rate, you need to do the following:

1) Set Expsoure Speed to 'Fast'

2) Set ‘Bit Send Mode’ accordingly
3) Call ‘ArtemisStartFastExposure’ and listen for images on the ‘ArtemisSetFastCallback’.