IJYMonoReqd

Overview

This interface deals with operations specific to Monochromators.

Move to Grating

Function: HRESULT MovetoGrating([in]double GratingDensity)
Description: Moves turret to a position that holds a grating with specified density.

NOTE: Recommended to use MoveToTurret instead. Grating density not guaranteed to be non-ambiguous

Parameters:
GratingDensity is in Grooves/mm.
See Also: MovetoTurret, GetCurrentGrating.

Get Current Grating

Function: HRESULT GetCurrentGrating([out] double* pGratingDensity, [out] VARIANT* pGratings);
Description: Retrieves the current grating density and a list of all grating densities.
Parameters:
Value pGratingDensity is in Grooves/mm.
Variant pGratings is a safearray of type double.
See Also: MovetoGrating.

Move to Wavelength

Function: HRESULT MovetoWavelength([in]double newWavelength);
Description: Moves wavelength to specified value. If a move requires backlash correction, the backlash move is completed by the IsBusy method. The backlash-needed condition sets the busy status of the mono to TRUE. It is completed when IsBusy is called and returns FALSE.
Parameters:
Wavelength units can be set with IJYSystemReqd::SetDefaultUnits. The units are initialized to Nanometers in the component.
Example:

```
'Scan the mono from 500 to 600 nm with steps of 10
myMono.SetDefaultUnits( jyutWavelength, jyuNanometers )
for ( monoPos = 500 to 600 step 10 )
  myMono.MoveToWavelength( monoPos )
  while ( myMono.IsBusy() )
  '   do nothing
  wend
  '  do something, like take data
Next monoPos
```

See Also: GetCurrentWavelength, Calibrate.

Get Current Wavelength

Function: HRESULT GetCurrentWavelength([out, retval] double* pWavelength);
Description: Retrieves the current wavelength value. If the mono is moving, it returns the wavelength value at the moment of executing this method. To read final wavelength, repeatedly call method IsBusy until it returns FALSE.
Parameters:
Wavelength units are set with IJYSystemReqd::SetDefaultUnits. The units are initialized to Nanometers in the component.
Example:

```
` Watch the mono position during a long move
myMono.MoveToWavelength( 0 )
` Wait until we get to start
while ( myMono.IsBuxy() )
wend
dim longMove as double
longMove = 1000
myMono.MoveToWavelength( longMove )
while( monoPosRead < longMove )
  txtDisplay.Text = myMono.GetCurrentWavelength
wend
```

See Also: MovetoWavelength, Calibrate.

Calibrate

Function: HRESULT Calibrate([in]double actualWavelength);
Description: Sets the current wavelength to the specified value. The setting persists until mono is reinitialized. his command is used to correct the actual position of the monochromator. For non-autocalibrating monochromators, a first pass calibration should be done when the instrument is powered on.
Parameters:

actualWavelength is taken as wavelength value in BASE UNITS and for BASE GRATING. This value is usually read from the front of the physical monochromator.

NOTE: Beware that it is not a value in default units like most other parameters used in monochromator methods! Changing this parameter without a complete understanding can result in damaging hardware.

See Also: MovetoWavelength, GetCurrentWavelength.

Move to Slit Width

Function: HRESULT MovetoSlitWidth([in]enum SlitLocation sl, [in]double newWidth);
Description: Moves a slit to a specified width.
Parameters:

Slit location can be the enumerated values
Front_Entrance = 0,
Side_Entrance =1,
Front_Exit =2,
Side_Exit =3,
First_Intermediate =4,
Second_Intermediate =5
Slit width is in units set by method IJYSystemReqd::SetDefaultUnits. The units are initialized to millimeters in the component.

See Also: GetCurrentSlitWidth, CalibrateSlitWidth, SetDefaultUnits, GetDefaultUnits.

Get Current Slit Width

Function: HRESULT GetCurrentSlitWidth([in]enum SlitLocation sl, [out, retval] double* p Retrieves Val);
Description:the current slit width value. If the slit is in motion, the value retrieved is the width at the moment of executing this method. If the move requires slit backlash, the busy status will be set to TRUE and the backlash move is completed when IsBusy returns FALSE.
Parameters:

Slit location can be the enumerated values
Front_Entrance = 0,
Side_Entrance =1,
Front_Exit =2,
Side_Exit =3,
First_Intermediate =4,

Second_Intermediate =5
Slit width is in units set by method IJYSystemReqd::SetDefaultUnits. The units are initialized to millimeters in the component.
See Also: MovetoSlitWidth, CalibrateSlitWidth.

Calibrate Slit Width

Function: HRESULT CalibrateSlitWidth([in]enum SlitLocation slitNumber);
Description: Sets the specified slit width to be zero.
Parameters:
Slit location can be the enumerated values
Front_Entrance = 0,
Side_Entrance =1,
Front_Exit =2,
Side_Exit =3,
First_Intermediate =4,
Second_Intermediate =5
See Also: MovetoSlitWidth, GetCurrentSlitWidth.

Is Busy

Function: HRESULT IsBusy([out, retval]VARIANT_BOOL* BusyStatus);
Description: Checks the busy status of the monochromator.
Parameters:
BusyStatus is TRUE if the grating or any accessories are still moving, or a backlash correction is needed for wavelength or slit. BusyStatus if FALSE when no motor is moving and no backlash flagged.
See Also: IsReady

Move to Mirror Position

Function: HRESULT MovetoMirrorPosition([in]enum MirrorLocation MirrorNumber, [in]enum MirrorLocation NewMirrorPosition);
Description: Moves the specified mirror to the specified position.
Parameters:
MirrorNumber can be one of the following values
EntranceMirror = 0
ExitMirror =1
NewMirrorPosition can be one of the following values
Front =2,
Side =3
See Also: GetCurrentMirrorPosition

Get Current Mirror Position

Function: HRESULT GetCurrentMirrorPosition([in]enum MirrorLocation MirrorNumber, [out, retval]enum MirrorLocation* pMirrorPostion);
Description: Retrieves the position of the specified mirror.
Parameters:
MirrorNumber can be one of the following values
EntranceMirror = 0
ExitMirror =1
pMirrorPosition can be a reference to one of the following values
Front =2,
Side =3
See Also: MovetoMirrorPosition

Open Shutter

Function: HRESULT OpenShutter();

Description: Opens the "active" shutter. If the Entrance Mirror is in the Side (Lateral) position, the "active" shutter is the Side Entrance Shutter. On the other hand, if the Entrance Mirror is in the Frunt (Axial) position, or if there is no Entrance Mirror, the Front Entrance shutter in "active".
Parameters:
See Also: CloseShutter, GetCurrentShutterPosition.

Close Shutter

Function: HRESULT CloseShutter();
Description: Closes the "active" shutter. Refer to the OpenShutter method for more information.
Parameters:
See Also: OpenShutter, GetCurrentShutterPosition.

Get Current Shutter Position

Function: HRESULT GetCurrentShutterPosition([out, retval]enum OpenOrClose* pShutterPosition);
Description:
Parameters:
pShutterPosition refers to one of the following values:
ShutterClose = 0,
ShutterOpen = 1
See Also: OpenShutter, CloseShutter, GetCurrentShutterPosition.

Stop

Function: HRESULT Stop();
Description: Stops the monochromator or slits.
Parameters:
See Also:

Is Ready

Function: HRESULT IsReady([out, retval] VARIANT_BOOL* ReadyStatus);
Description: Retrieves status of the monochromator's controller. Sending commands to a controller while the controller-issued moves are still in process may result in controller's command error. IsReady and IsBusy may return independ values in the case of multiplex controllers. For example, a DataScan controller may be used to controller two monochromators, Mono1 and Mono2. In the case that Mono1 is moving and Mono2 is not moving, the following values will be returned:
IsBusy IsReady
Mono1 TRUE FALSE
Mono2 FALSE FALSE
Parameters:
The returned ReadyStatus is TRUE if all the moves issued by the controller are completed. The returned ReadyStatus is FALSE if all the moves issues by the controller are not completed.
See Also: IsBusy

Is Sub Item Installed

Function: HRESULT IsSubItemInstalled([in] enum MonoSubItemType type, [out, retval] VARIANT_BOOL* installed);
Description: Retrieves the configuration of the monochromator regarding installation of difference items. Configuration information can be set using IJYSystemReqd::Configure.
Parameters:
The following values can be used for type:
Slit_Front_Entrance =0,

Slit_Side_Entrance =1,
Slit_Front_Exit =2,
Slit_Side_Exit =3,
Slit_First_Intermediate =4,
Slit_Second_Intermediate =5,
Mirror_Entrance =6,
Mirror_Exit =7,
MonoDrive =8,
Turret =9,
Front_Shutter =10,
Side_Shutter =11

See Also: IJYSystemReqd:Configure.

Get Current Grating With Details

Function: HRESULT GetCurrentGratingWithDetails([out] double* pGratingDensity, [out] VARIANT* pGratings, [out]VARIANT* pBlaze, [out]VARIANT* pDescription);
Description: Retrieves current grating density, and lists of grating densities, Blaze information, and Descriptions.
Parameters:

pGratingDensity is density in Grooves/mm.
Variant pGratings is a safearray of type double.
Variant pBlaze is a safearray of type BSTR.
Variant pDescription is a safearray of type BSTR.

See Also:

Move to Turret

Function: HRESULT MovetoTurret([in] int turretNumberZeroBased);
Description: Moves to a turret position as specified.
Parameters:

turretNumberZeroBased is zero-based turret position.

See Also: MovetoGrating, GetCurrentTurret

Get Current Turret

Function: HRESULT GetCurrentTurret([out, retval] int* turretNumberZeroBased);
Description: Retrieves turret position.
Parameters:

The returned value turretNumberZeroBased points to zero-based turret position.

See Also:

Is Target Within Limits

Function: HRESULT IsTargetWithinLimits([in]enum jyMonoMoveType what, [in] double where, [out] VARIANT* withinLimits, [out] double* minForCurrentGrating, [out] double* minForCurrentGrating, [in, optional] VARIANT whichSlit);
Description: Checks whether the target position is within the allowed physical range of the hardware. It also returns the high and low limits that can be used on user interface. The low and high limits are given in the current units and ,for the case of wavelength, for the current grating.
Parameters:

Value "what" can have one of the following values
jyMonoMoveTypeWavelength,
jyMonoMoveTypeSlit
Value "where" is the target position of type double in default units.
Value WithinLimits returns VARIANT_BOOL.
Value minForCurrentGrating is the low limit for the current grating in current units
Value maxForCurrentGrating is the high limit for the current grating in current units
Value whichSlit specifies the slit when checking slit ranges.

See Also:

Set JY Logger Properties

Function: HRESULT SetJYLoggerProperties([in]enum logLevel newLevel ,[in, optional] BSTR newFileName ,[in, optional] BSTR newFilePath ,[in, optional] int newMaxBackupFiles,[in, optional] int newMaxFileSize);
Description: For Internal Use Only

Get Grating Motor Speeds/Set Grating Motor Speeds

Function: HRESULT GetGratingMotorSpeeds([out]double* pFrequencyMin, [out]double* pFrequencyMax, [out]double* pRampTime);
HRESULT SetGratingMotorSpeeds([in]double FrequencyMin, [in]double FrequencyMax, [in] double RampTime);
Description: For Internal Use Only

Slit Motor Speed

Function: HRESULT SlitMotorSpeed([in]enum SlitLocation sl, [out, retval]double* pFrequency);
HRESULT SlitMotorSpeed([in]enum SlitLocation sl, [in] double Frequency);
Description: For Internal Use Only