



Iris AO, Inc.

Iris AO PTT Deformable Mirror System Python Package

Version 1.3 rev B



1.1 Python API

The Python API is distributed as a Python package. It contains the functions required to control the mirrors. It is recommended to use Anaconda to use and run this package. This package has been tested on Python 2.7 with Anaconda2 64-bit on Linux Ubuntu 15.04 and Windows 7 & 8 OS.

On Linux, the Iris AO dll library can be used only in super user mode. Start Python using the command `sudo ~/anaconda/bin/python2.7`

The following code example demonstrates how to set the position of one segment of the mirror

```
# Import control functions from the package
import IrisAO_PythonAPI as IrisAO_API
try:
    # Connect to the specified mirror device
    mirror = IrisAO_API.MirrorConnect('FSC37-02-01-0916', '13130002',
False)

    # Set the position of Segment 1 to 0.1 um, 0.2 mrad, 0.3 mrad
    IrisAO_API.SetMirrorPosition(mirror, 1, (0.1, 0.2, 0.3))

    # Send the settings to the mirror
    IrisAO_API.MirrorCommand(mirror, IrisAO_API.MirrorSendSettings)

    # Release the specified mirror device connection
    IrisAO_API.MirrorRelease(mirror)
except Exception as e:
    print e
```

1.1.1 MirrorConnect

This function provides support for connecting to the specified mirror device. Mirror connections are required to access the associated mirror level functions.

Syntax	mirrorHandle = MirrorConnect (mirrorNumber, driverIDNumber, HardwareDisabled)	
Parameter	Type	Description
mirrorNumber	str	Specifies a mirror serial number string to load the matching mirror configuration file. Only the serial number is required as the library loads the associated '.mcf' files with the matching serial number.
driverIDNumber	str	Specifies a driver box serial number string to load the matching driver box configuration file. Only the serial number is required as the library loads the associated '.dcf' files with the matching serial number.
HardwareDisabled	bool	Specifies a flag to disable the hardware. Disabling the hardware is often useful for debugging purposes when you do not wish to send live values to the mirror.
mirrorHandle	long	The return value is a handle to the mirror connection; otherwise an exception is thrown to indicate an error has occurred.

Syntax	mirrorHandle = MirrorConnect (mirrorNumber, driverIDNumber, HardwareDisabled)	
Parameter	Type	Description
Exceptions	RuntimeError	Exception raised if the files are not readable, or if the driver is unable to connect to the mirror.

The following code example shows how to properly connect and release mirror connections, and handles the possible exceptions. All mirror connections should be properly released before shutting down the application.

```
# Connect to a mirror: get a mirror handle
try:
    mirror = IrisAO_API.MirrorConnect(mirror_num, driver_num, disableHW)
except Exception as e:
    print e
    sys.exit("There was a problem connecting to the mirror, exiting")

try:
    IrisAO_API.MirrorRelease(mirror)
except Exception as e:
    print e
    print "There was a problem releasing the connection with the mirror"
```

1.1.2 SetMirrorPosition

This function provides support getting the current position for the specified mirror segment number, or a list of segment numbers. Positions are only valid following a call to MirrorCommand using the command parameter MirrorSendSetting.

For one segment:

Syntax	SetMirrorPosition(MirrorHandle, Segment, PTTposition)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
Segment	int	Specifies the segment to be controlled
PTTPosition	tuple (float,float,float)	Specifies the PTT position for the segment. Units are (um, mrad, mrad)
Exceptions	RuntimeError	Exception raised if the mirror handle is not correct

For a list of segments:

Syntax	SetMirrorPosition(MirrorHandle, segmentList, PTTPositions)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
SegmentList	list of n int	Specifies the segments to be controlled
PTTPositions	list of n tuples(float,float,float)	Specifies the PTT position for each segment in the segmentList. Units are (um, mrad,mrad)

Syntax	SetMirrorPosition(MirrorHandle, segmentList, PTTPositions)	
Parameters	Type	Description
Exceptions	RuntimeError	Exception raised if the mirror handle is not correct

1.1.3 MirrorCommand

This function can be used with two different arguments: one to flatten the mirror, the other one to send the position to the mirror.

Syntax	MirrorCommand (MirrorHandle,command)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
command	long	Two values are supported: <ul style="list-style-type: none"> - IrisAO_PythonAPI.MirrorSendSettings: send the mirror positions on all segments to the connected mirror - IrisAO_PythonAPI.MirrorInitSettings: Flattens the mirror positions on all segments
Exceptions	RuntimeError	Exception raised if the mirror handle or the command is not correct

The following code example demonstrates how to flatten the mirror segments:

```
import IrisAO_PythonAPI as IrisAO_API
try:
    # Connect to the specified mirror device
    mirror = IrisAO_API.MirrorConnect('FSC37-02-01-0916', '13130002',
False)

    # Flatten the mirror
    IrisAO_API.MirrorCommand(mirror, IrisAO_API.MirrorInitSettings)

    # Release the specified mirror device connection
    IrisAO_API.MirrorRelease(mirror)
except Exception as e:
    print e
```

1.1.4 GetMirrorPosition

This function provides support for getting the current position for the specified mirror segment number, or for a list of segments.

For one segment:

Syntax	Position, Lockedflag, ReachableFlag = GetMirrorPosition (MirrorHandle, Segment)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
Segment	int	Specifies the segment whose position is to be read

Syntax	Position, Lockedflag, ReachableFlag = GetMirrorPosition (MirrorHandle, Segment)	
Parameters	Type	Description
Position	Tuple (float,float,float)	Current values of the segment position (z,xgrad,ygrad). Units are in microns, mrad, mrad for the reachable PTT values.*
LockedFlag	bool	Specifies if a segment is locked. The segment is locked when true.
ReachableFlag	bool	Specifies if a segment position is reachable. The segment is reachable when true.
Exceptions	RuntimeError	Exception raised if the mirror handle is not correct

For a list of segments:

Syntax	Positions, Lockedflags, ReachableFlags = GetMirrorPosition (MirrorHandle, SegmentList)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
SegmentList	List of n int	Specifies the segments whose position are to be read
Positions	List of n Tuples (float,float,float)	List of the current values of the segment positions. (z,xgrad,ygrad). Units are in microns, mrad, mrad for the reachable PTT values.*
LockedFlags	List of n bool	Specifies if a segment is locked. The segment is locked when true.
ReachableFlags	List of n bool	Specifies if a segment position is reachable. The segment is reachable when true.
Exceptions	RuntimeError	Exception raised if the mirror handle is not correct

* Small numerical rounding errors may occur between Python and the underlying IrisAO.Devices.dll functions. When they occur, the errors are in the 10^{-6} range for the values reported in microns, mrad, mrad. For example, setting a piston value of 0 microns may return a reachable value in the range of 10^{-6} microns.

1.1.5 SetModalPosition

The **SetModalPosition** tool sets the mirror shape based on modal coefficient values. The modal coefficient values (or the amplitude of the Zernike modes) are defined in the mirror configuration file. It can set the values of one coefficient or of a list of coefficients.

For one coefficient:

Syntax	SetModalPosition (MirrorHandle,CoefficientValueCouple)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.

Syntax	SetModalPosition (MirrorHandle,CoefficientValueCouple)	
Parameters	Type	Description
CoefficientValueCouple	Tuple (int,float)	The first element of the tuple is the coefficient number. The default set is defined as Zernike modes from zero through twenty The second element of the tuple is the corresponding magnitude.Units in microns <i>rms</i> .
Exceptions	RuntimeError	Exception raised if the mirror handle is not correct

For a list of coefficients:

Syntax	SetModalPosition (MirrorHandle,CoefficientValueCouples)	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
CoefficientValueCouples	List of Tuples (int,float)	List of (coefficient,value) couples. The first element of the tuple is the coefficient number. The default set is defined as Zernike modes from zero through twenty The second element of the tuple is the corresponding magnitude.Units in microns <i>rms</i> .
Exceptions	RnuntimeError	Exception raised if the mirror handle is not correct

1.1.6 MirrorRelease

This function provides support for releasing the supplied mirror connection.

Syntax	Release = MirrorRelease (MirrorHandle);	
Parameters	Type	Description
MirrorHandle	long	Specifies a mirror connection handle used to associate the current mirror connection.
release	long	The return value is 0 if the mirror connection was successfully released, otherwise the return value is the supplied mirror handle.
Exceptions	RuntimeError	Exception raised if the mirror handle is not correct

All mirror connections should be properly released before shutting down the application.