

ASTRA Automation API Interface

The ASTRA Automation API exposes a COM interface for automation. The COM interface to ASTRA allows one to create experiments, set sample parameters, and control ASTRA data collection from another program. In this document, the various methods and events that make up the interface are detailed. We recommend reading “*ASTRA Automation API - Getting Started*” before diving in this manual.

Contents

Introduction	4
Parameters and Return Values	4
Units.....	4
Experiments	4
.NET Framework.....	4
Structures.....	5
SampleInfo	5
LogonResult.....	5
UvChannelDetails.....	5
UvDeviceDetails	5
BaselinePoint.....	6
BaselineType (enum)	6
BaselineDetails.....	6
PeakRange	7
PeakRanges	7
ProcedureDetails.....	7
Properties.....	8
InstrumentsDetected.....	8
Application Methods	9
SetAutomationIdentity	9
GetAutomationUid.....	9
GetAutomationClientInfo	9
GetAutomationClientProcessId.....	9
GetVersion.....	10
Show	10
GetWindowHandle	10
IsEmbedded.....	10
RequestQuit	10
File Methods	11
GetExperimentTemplates.....	11
OpenExperiment.....	11
RunExperiment	11
SaveExperiment	11
SaveExperimentWithDescription	11
CloseExperiment	12
Experiment Methods.....	13
NewExperimentFromTemplate.....	13
GetExperimentName.....	13
GetExperimentDescription.....	13
SetExperimentDescription.....	13
GetCollectionDuration.....	13

SetCollectionDuration.....	14
ValidateExperiment	14
UseInstrumentCalibrationConstant	14
StartCollection.....	14
StopCollection.....	15
Pump Methods	16
GetPumpFlowRate.....	16
SetPumpFlowRate.....	16
Injector Methods	17
GetInjectedVolume.....	17
SetInjectedVolume	17
Sample Methods.....	18
GetSample.....	18
SetSample.....	18
GetSampleName.....	18
SetSampleName.....	18
GetSampleDescription.....	18
SetSampleDescription.....	19
GetSampleDndc	19
SetSampleDndc	19
GetSampleA2	19
SetSampleA2	19
GetSampleUvExtinction.....	20
SetSampleUvExtinction.....	20
GetSampleConcentration	20
SetSampleConcentration.....	20
Result Methods.....	21
GetBaselines.....	21
UpdateBaselines	21
GetPeakRanges	21
AddPeakRange	21
UpdatePeakRange	21
RemovePeakRange	22
GetResults	22
SaveResults.....	22
GetDataSet	22
SaveDataSet	22
GetProcedureData	23
GetProcedureDetails	23
Miscellaneous	24
GetResultsSnapshot (obsolete).....	24
GetIsExperimentRunning.....	24
ValidateLogon	24
GetDataDatabaseDirectory	24
HasVisionUv.....	25
SetupVisionUv	25
PushVisionUvData	25
HasCollectedData.....	25
Events.....	26
ExperimentClosed.....	26
ExperimentReady (Obsolete if using ASTRA 8.1.1 or later)	26
ExperimentRead (Only available with ASTRA 8.1.1 or later)	26
ExperimentRun (Only available with ASTRA 8.1.1 or later)	26

ExperimentWrite (Only available with ASTRA 8.1.1 or later)	27
PreparingForCollection.....	27
WaitingForAutoInject	27
CollectionStarted	28
CollectionAborted.....	28
CollectionFinished.....	28
InstrumentDetectionCompleted.....	28
Appendix A: ASTRA Error Codes.....	29

Introduction

Parameters and Return Values

Parameters are marked as either [in] or [out] to signify whether they are intended to be used as function inputs or outputs. Parameters marked [out, retval] will contain the *return value* of a function returning S_OK.

The IAstra interface will return S_OK to indicate success and an ASTRA-specific error code in the case of common failures. E_FAIL is returned in exceptional cases, or when no suitable error code is defined. See *Appendix A* for a list of errors.

Units

The units for physical quantities are different from the units shown in ASTRA. For ease of reference, they are listed here and when different the Automation units is highlighted in light gray:

Physical quantity	ASTRA units	Automation units
Flow rate	mL/min	mL/min
Duration	min, mL, sec	min
Injected volume	µL	mL
dn/dc	mL/g	mL/g
A2	mol mL/g ²	mol mL/g ²
UV Ext. Coef.	mL/(mg cm)	mL/(g cm)
Concentration	mg/mL	g/mL

Experiments

Most functions will operate on an ASTRA experiment. Experiments are identified by their ID and this ID is usually the first argument of the function. A new experiment ID is created whenever you open or create an experiment. The ID is no longer usable after closing an experiment and passing an invalid ID to a function will result in an error.

.NET Framework

A .NET/COM interop assembly is provided with the installation package. In this interop assembly, if a function has an argument marked [out, retval], the signature in .NET will transform this parameter into the function's return value.

All SAFEARRAY(X) are converted into a .NET array of type X. The type long is converted into a .NET System.Int32, double into System.Double, BSTR as a System.String and BOOL as a System.Boolean.

Structures

SampleInfo

Data structure used for getting and setting sample information in calls to [GetSample](#) and [SetSample](#).

Field Name	Description
<i>name</i>	Name of the sample
<i>description</i>	Description of the sample
<i>dndc</i>	dn/dc value in mL/g
<i>a2</i>	Second virial coefficient (mol mL/g ²)
<i>uvExtinction</i>	UV extinction coefficient (mL/(g cm))
<i>Concentration</i>	Concentration value (g/mL)

LogonResult

Data structure used for getting logon result in calls to [ValidateLogon](#).

Field Name	Description
<i>isValid</i>	Boolean value to indicate whether the logon credential is valid
<i>errorMessage</i>	Error message during validation, empty if no error occurred
<i>errorDetails</i>	Detailed error message

UvChannelDetails

Data structure contains the UV Channel detail information, included in the [UvDeviceDetails](#) data structure, used for setup Vision UV.

Field Name	Description
<i>useChannel</i>	Indicate whether the channel is used (boolean)
<i>waveLength</i>	The wavelength setting of this UV channel in nm
<i>bandwidth</i>	The bandwidth setting of this UV channel in nm
<i>useReference</i>	Indicate whether the reference channel is used (boolean)
<i>refWaveLength</i>	The reference wavelength setting of this UV channel in nm
<i>refBandwidth</i>	The reference bandwidth setting of this UV channel in nm

UvDeviceDetails

Data structure contains the UV device detail information, used for setup Vision UV with call to [SetupVisionUv](#).

Field Name	Description

<i>deviceName</i>	Name of the UV Device
<i>deviceModel</i>	Model of the UV Device
<i>supportsPeakWidth</i>	Indicate whether this device supports “ <i>peakWidth</i> ”
<i>peakWidth</i>	The peak width of this device (string)
<i>supportsSlitWidth</i>	Indicate whether this device supports “ <i>slitWidth</i> ”
<i>slitWidth</i>	The slit width of this device (string)
<i>supportsRequireLampUV</i>	Indicates whether this device supports “ <i>requireLampUV</i> ”
<i>requireLampUV</i>	Does this device require a UV Lamp to run this method
<i>supportsRequireLampVis</i>	Indicates whether this device supports “ <i>requireLampVis</i> ”
<i>requireLampVis</i>	Does this device require a Visible Lamp to run this method
<i>uvChannels</i>	The Array of UvChannelDetails of the UV channels in this UV device

BaselinePoint

Data structure contains the x, y coordinates for a point, used to represent a baseline.

Field Name	Description
<i>x</i>	X-coordinate of a baseline extrema
<i>y</i>	Y-coordinate of a baseline extrema

BaselineType (enum)

Enumerated type used to represent the type of a baseline.

Field Name	Description
<i>eBT_None</i>	The baseline has no type associated
<i>eBT_Manual</i>	The baseline is manually set
<i>eBT_SnapY</i>	The baseline is manually set for the x-abscissa but automatically using the corresponding y-abscissa value
<i>eBT_Automatic</i>	The baseline is automatically set

BaselineDetails

Data structure contains baseline details, used for retrieving and updating baselines from an experiment, through calls to `GetBaselines` and `UpdateBaselines`.

Field Name	Description
<i>seriesName</i>	The name of the baseline series
<i>type</i>	The type of the baseline (BaselineType)
<i>start</i>	The starting point of the baseline (BaselinePoint)

<i>end</i>	The ending point of the baseline (BaselinePoint)
------------	--

PeakRange

Data structure contains peak range information, used for updating peak ranges from an experiment, through call to [UpdatePeakRange](#).

Field Name	Description
<i>number</i>	The number associated with the peak range
<i>start</i>	The starting x-axis value of the peak range
<i>end</i>	The ending x-axis value of the peak range

PeakRanges

Data structure contains peak ranges details.

Field Name	Description
<i>size</i>	The number of peak ranges in this structure
<i>peaks</i>	An array of PeakRange object

ProcedureDetails

Data structure contains procedure details, used for retrieving procedure data sets from an experiment, through calls to `GetProcedureDetails` and `UpdateProcedureData`.

Field Name	Description
<i>name</i>	The name of the procedure
<i>objectID</i>	The ID of the procedure

Properties

InstrumentsDetected

Determine if instrument detection has taken place. Instruments are detected during startup, and when a user refreshes instruments from the System->Instruments dialog.

Name	Type	Description
<i>result</i> [out, retval]	BOOL	TRUE if instrument detection completed, otherwise FALSE.

NOTE: *The result of this property does not indicate which instruments (if any) were detected, only that detection took place. Clients should call [ValidateExperiment](#) to ensure that needed instruments are present.*

Application Methods

SetAutomationIdentity

It is required that before executing any operation on the COM interface, the COM client identifies itself to ASTRA. Without this identification, all functions operating on ASTRA except [GetAutomationUid](#), [GetAutomationClientInfo](#), [GetAutomationClientId](#) will fail with E_REQUEST_OUT_OF_SEQUENCE.

Name	Type	Description
<i>entityName</i> [in]	BSTR	Name of client
<i>entityVersion</i> [in]	BSTR	Version of client
<i>pid</i> [in]	Long	Process ID of client
<i>entityGuid</i> [in]	BSTR	Unique identifier of client as a string UID
<i>enabled</i> [in]	BOOL	Should always be set to TRUE
<i>trustedEntityPassword</i> [in]	SAFEARRAY(BYTE)	Ignored

GetAutomationUid

Get the unique identifier of the client identified via [SetAutomationIdentity](#).

Name	Type	Description
<i>guid</i> [out, retval]	BSTR *	Unique identifier of the client

GetAutomationClientInfo

Get the information regarding the client identified via [SetAutomationIdentity](#).

Name	Type	Description
<i>info</i> [out, retval]	BSTR *	String representing the identity of the client

GetAutomationClientId

Get the process identifier of the client identified via [SetAutomationIdentity](#). Can be used to check if a client is still using the ASTRA SDK.

Name	Type	Description
<i>pid</i> [out, retval]	long *	The process identifier of the last client

GetVersion

Get the version of Astra.

Name	Type	Description
<i>version</i> [out, retval]	BSTR *	The version of ASTRA

Show

Show/hide the Astra application interface.

Name	Type	Description
<i>show</i> [in]	BOOL	TRUE to display Astra interface, FALSE to hide interface

GetWindowHandle

Get the ASTRA Window Handle.

Name	Type	Description
<i>windowHandle</i> [out, retval]	ULONGLONG *	Pointer value of the main ASTRA window handle

IsEmbedded

Check if ASTRA was started via COM or as a standalone application.

Name	Type	Description
<i>result</i> [out, retval]	BOOL	TRUE if ASTRA was started via a COM client, FALSE otherwise

RequestQuit

Prior to stopping all interaction with ASTRA, you need to call this method for a proper shutdown of ASTRA. This function has no argument.

File Methods

GetExperimentTemplates

Get all available methods from the ASTRA system database as a list of filenames.

Name	Type	Description
<i>templates</i> [out, retval]	SAFEARRAY(BSTR)*	Array of experiment method filenames

OpenExperiment

Open an experiment given the file name, get the experiment Id when it is opened.

Name	Type	Description
<i>fileName</i> [in]	BSTR	The file location of the experiment
<i>experimentId</i> [out, retval]	long *	The ID of the opened experiment

RunExperiment

Run the given experiment. If some parameters have changed since the last run of the experiment, new results will be calculated.

Name	Type	Description
<i>experimentId</i> [out, retval]	long *	The ID of the opened experiment

SaveExperiment

Save the given experiment to a file.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment to save
<i>filename</i> [in]	BSTR	The path of the file to save the experiment as

SaveExperimentWithDescription

Save the given experiment with a description.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment t
<i>filename</i> [in]	BSTR	The path of the file to save the experiment as
<i>description</i> [in]	BSTR	The description to attach with the experiment

CloseExperiment

Close the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment to close

Experiment Methods

NewExperimentFromTemplate

Create a new experiment from the method with the given ID value.

Name	Type	Description
sourceTemplate [in]	BSTR	Full path to the method to use. See GetExperimentTemplates to retrieve list of available methods
experimentId [out, retval]	Long *	The ID of the created experiment

GetExperimentName

Get the name for the given experiment. The name is read-only and can only be changed because of calling [SaveExperiment](#).

Name	Type	Description
experimentId [in]	long	The ID of the experiment
name [out, retval]	BSTR *	The experiment name

GetExperimentDescription

Get the description for the given experiment. In ASTRA the description is in the Experiment Configuration screen.

Name	Type	Description
experimentId [in]	long	The ID of the experiment
description [out, retval]	BSTR *	The experiment's description

SetExperimentDescription

Set the description for the given experiment. In ASTRA the description is in the Experiment Configuration screen.

Name	Type	Description
experimentId [in]	long	The ID of the experiment
description [in]	BSTR	The experiment's description

GetCollectionDuration

Get the data collection duration for the given experiment. Time value is in minutes.

Name	Type	Description

<i>experimentId</i> [in]	long	The ID of the experiment
<i>time</i> [out, retval]	double *	Collection duration in minutes

SetCollectionDuration

Set the data collection duration for the given experiment. Can also be used to extend the duration of a collection that has already begun, but care must be taken to ensure that the new collection time is not shorter than the current duration. Time value is in minutes.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>time</i> [in]	double	Collection duration in minutes

ValidateExperiment

Validate the given experiment prior to performing data collection.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>error</i> [out]	BSTR *	Contains experiment validation failures and/or warnings.
<i>result</i> [out, retval]	BOOL *	TRUE when validation is successful, FALSE otherwise

NOTE: If experiment validates with warnings, result will be TRUE and the error will contain warning text.

UseInstrumentCalibrationConstant

Use the calibration constant retrieved from the physical instrument instead of the value specified in the experiment configuration instrument profile if the two values are different.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>state</i> [in]	BOOL	True to use the instrument's calibration constant, false to use the profile's constant

NOTE: The default behavior of ASTRA is to use the value from the instrument.

StartCollection

Starts collecting data for a given experiment. The wait for auto-inject flag is always set for collections started through the ASTRA COM interface. A [WaitingForAutoInject](#) event will be triggered once the collection is ready to begin, followed by a [CollectionStarted](#) event once data is received.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment

StopCollection

Stops collecting data for a given experiment. A [CollectionAborted](#) event will be triggered once the collection has stopped.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment

Pump Methods

GetPumpFlowRate

Get the flow rate of the pump for the given experiment. Flow rate is in mL/min.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>flowRate</i> [out, retval]	double *	Pump flow rate in mL/min

SetPumpFlowRate

Set the flow rate of the pump for the given experiment. Flow rate is in mL/min.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>flowRate</i> [in]	double	Pump flow rate in mL/min

Injector Methods

GetInjectedVolume

Get the injected volume of the injector for the given experiment. Volume is in mL.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>InjectedVolume</i> [out, retval]	double *	Injected volume in mL

SetInjectedVolume

Set the injected volume of the injector for the given experiment. Volume is in mL.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>InjectedVolume</i> [in]	double	Injected volume in mL

Sample Methods

GetSample

Get the injected sample information for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>sample</i> [out, retval]	SampleInfo *	Structure containing sample information

SetSample

Set the injected sample information for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>sample</i> [in]	SampleInfo *	Structure containing sample information

GetSampleName

Get the injected sample name for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>name</i> [out, retval]	BSTR *	The sample name

SetSampleName

Set the injected sample name for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>name</i> [inl]	BSTR	The sample name

GetSampleDescription

Get the injected sample description for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>description</i> [out, retval]	BSTR *	The sample description

SetSampleDescription

Set the injected sample description for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>description</i> [in]	BSTR	The sample description

GetSampleDndc

Get the injected sample Dndc value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>dndc</i> [out, retval]	double *	The dn/dc value in mL/g

SetSampleDndc

Set the injected sample Dndc value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>dndc</i> [in]	double	The dn/dc value in mL/g

GetSampleA2

Get the injected sample a2 value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>a2</i> [out, retval]	double *	The A2 value in mol mL/g ²

SetSampleA2

Set the injected sample a2 value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>a2</i> [in]	double	The A2 value in mol mL/g ²

GetSampleUvExtinction

Get the injected sample UV extinction value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>uvExtinction</i> [out, retval]	double *	The UV extinction value in mL/(g cm)

SetSampleUvExtinction

Set the injected sample UV extinction value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>uvExtinction</i> [in]	double	The UV extinction value in mL/(g cm)

GetSampleConcentration

Get the injected sample concentration value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>concentration</i> [out, retval]	double *	The concentration value in g/mL

SetSampleConcentration

Set the injected sample concentration value for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>concentration</i> [in]	double	The concentration value in g/mL

Result Methods

GetBaselines

Get the baseline information for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	SAFEARRAY(BaselineDetails) *	Array of BaselineDetails object

UpdateBaselines

Update the baseline information for the given experiment, existing baselines will be overwritten by the given baselines.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>baselines</i> [in]	SAFEARRAY(BaselineDetails)	Array of BaselineDetails object

GetPeakRanges

Get the peak range information for the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	SAFEARRAY(PeakRange) *	Array of PeakRange object

AddPeakRange

Add a peak range to the given experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>start</i> [in]	double	The starting/left point of the range on x-axis
<i>end</i> [in]	Double	The ending/right point of the range on x-axis

UpdatePeakRange

Update a peak range in the given experiment that has the same peak number.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>peak</i> [in]	PeakRange *	The PeakRange object contains the updated peak information

RemovePeakRange

Remove a peak range in the given experiment that has the same peak number.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>peakNumber</i> [in]	long	The peak number of the peak range to be removed

GetResults

Get experiment results as a XML string, given the ID for the experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	BSTR *	Experiment results as a single XML string

SaveResults

Save experiment results in XML format to a text file given the ID for the experiment and the full path of the target file.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>fileName</i> [in]	BSTR	Full path where results will be saved as a XML file

GetDataSet

Get data set information given the ID for the experiment and the dataset definition name.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>definitionName</i> [in]	BSTR	The name of the dataset definition to use
<i>result</i> [out, retval]	BSTR *	Dataset information as a single string, values delimited by comma

SaveDataSet

Save dataset information to a CSV file given the ID for the experiment, the dataset definition name, and the full path of the target file.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>definitionName</i> [in]	BSTR	The name of the dataset definition to use
<i>fileName</i> [in]	BSTR	Full path where dataset will be saved

GetProcedureData

Get data set information given the ID for the experiment and the ID for a procedure. The procedure ID can be retrieved using GetProcedureDetails.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>procedureId</i> [in]	BSTR	The name of the dataset definition to use
<i>result</i> [out, retval]	BSTR *	Dataset information as a single string, values delimited by comma

GetProcedureDetails

Get a list of all procedures in an experiment and the corresponding procedure IDs, given an experiment ID.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	SafeArray(ProcedureDetails)*	A list of ProcedureDetails structures, each containing a procedure name and ID

Miscellaneous

RunExperiment

Run the experiment to generate dataset information. When the run is complete an [ExperimentRun](#) event is triggered.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment

GetResultsSnapshot (obsolete)

Get the results snapshot of an experiment, given an experiment Id.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>xmlResultsSnapshot</i> [out, retval]	BSTR *	Results of the experiment

GetIsExperimentRunning

Get the running state of the experiment.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	BOOL *	TRUE if experiment is running, FALSE otherwise

ValidateLogon

Logon to Astra (when Security Pack is enabled) given the user ID, password, and domain. This method will return the result of the logon attempt.

Name	Type	Description
<i>userId</i> [in]	BSTR	User ID
<i>password</i> [in]	BSTR	Password
<i>domain</i> [in]	BSTR	Domain
<i>result</i> [out, retval]	LogonResult *	A LogonResult structure

GetDataDatabaseDirectory

Get a list of unique sub-folders directory given the root directory path of the Security Pack database. It will only be available when Security Pack is enabled.

Name	Type	Description
<i>rootPath</i> [in]	BSTR	Base folder used to retrieve unique sub-folders
<i>directory</i> [out, retval]	SAFEARRAY(BSTR) *	An array of unique directory folders string

HasVisionUv

Check if the experiment's configuration contains a Vision UV Instrument Profile.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	BOOL *	TRUE if a VISION UV instrument profile is present in the experiment's configuration, FALSE otherwise

SetupVisionUv

Setup the existing UV profile of the ASTRA experiment with the given UV instrument details.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>details</i> [in]	UvDeviceDetails *	A UvDeviceDetails structures with all the information related to the UV instrument

PushVisionUvData

Push new UV data received during collection.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>channelCount</i> [in]	long	Number of Channels sent in the data stream
<i>data</i> [in]	SAFEARRAY(double)	Data array as a sequence of doubles, every sequence being of "channelCount + 1" in length. The first double being the time when data was collected, the remaining doubles for the data for each collected channel

HasCollectedData

Check if the experiment had some data collected.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment
<i>result</i> [out, retval]	BOOL *	TRUE if experiment has some collected data, FALSE otherwise

Events

Most operations in ASTRA are asynchronous and the proper way to know they completed is to use an event notification system. Examples of asynchronous operations are loading/saving an ASTRA data file, starting a collection, ...

Except [InstrumentDetectionCompleted](#), all events take the experiment ID of the experiment that triggered the notification.

It is required to wait for the [InstrumentDetectionCompleted](#) event before starting any collections and it is recommended to wait for this event after creating the Astra class instance.

ExperimentClosed

This event is triggered once an experiment has been closed. A closed experiment can no longer be read or receive commands, and its ID should no longer be used in calls to ASTRA.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

ExperimentReady (Obsolete if using ASTRA 8.1.1 or later)

Note: If targeting ASTRA 8.1.1 or later, do not use this event, use any of the following instead: [ExperimentRead](#), [ExperimentWrite](#), [ExperimentRun](#).

This event is triggered once an experiment has finished an operation and is in the ready state (it is ready to be read and/or receive commands). This event is triggered when ASTRA finishes creating an experiment from method, after an experiment's save operation has completed, and after a collection has finished (immediately following a [CollectionFinished](#) event).

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

ExperimentRead (Only available with ASTRA 8.1.1 or later)

This event is triggered once an experiment has been read from disk after calling [OpenExperiment](#).

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

ExperimentRun (Only available with ASTRA 8.1.1 or later)

This event is triggered once an experiment has been run. A run occurs whenever a change made to the experiment will require some recalculation. You will always get a run event:

- When calling [OpenExperiment](#) or [RunExperiment](#).
- At the end of a collection, after the [CollectionFinished](#) event is triggered.
- When changing the configuration post collection, i.e., calling:
 - [SetSample](#)
 - [SetSampleDndc](#)
 - [SetSampleA2](#)
 - [SetSampleUvExtinction](#)
 - [SetSampleConcentration](#)
 - [SetPumpFlowRate](#)
 - [SetInjectedVolume](#)
 - [SetupVisionUv](#)

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

ExperimentWrite (Only available with ASTRA 8.1.1 or later)

This event is triggered once an experiment has been saved to disk after calling [SaveExperiment](#) or [SaveExperimentWithDescription](#).

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

PreparingForCollection

This event is triggered when the experiment collection has been initiated, but data has not yet been received (pending an auto-inject signal).

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

WaitingForAutoInject

This event is triggered when the experiment collection is ready and has started waiting for the auto-inject signal from the instrument.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

CollectionStarted

This event is triggered when the auto-inject signal has been processed, and collection data is being received.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

CollectionAborted

This event is triggered when the data collection is aborted by the user, either through the ASTRA UI or the COM API.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

CollectionFinished

This event is triggered when data collection has completed. Once complete, the experiment may be saved.

Name	Type	Description
<i>experimentId</i> [in]	long	The ID of the experiment triggering the event

InstrumentDetectionCompleted

This event is triggered when instrument detection completes. Instruments are detected during startup, and when a user refreshes instruments from the System->Instruments dialog. This event has no argument.

Appendix A: ASTRA Error Codes

The Astra COM interface makes use of the following ASTRA-specific error codes, which are exported for convenience in the type library as an enumeration (see table below). It is recommended to use the GetErrorMessage function to get a string description of one of the errors below.

Error	Description	Code
E_EXP_TMPLNOTFOUND	Method not found.	0x80040201
E_EXP_BADHANDLE	Invalid experiment ID.	0x80040202
E_EXP_INVALID	Experiment validation failure.	0x80040203
E_EXP_RUNNING	Cannot modify running experiment.	0x80040204
E_EXP_FLOWMODEONLY	Operation requires flow mode experiment.	0x80040205
E_EXP_NOCONFIG	Experiment configuration not found.	0x80040206
E_EXP_NOPUMP	Pump not found in configuration.	0x80040207
E_EXP_NOINJECTOR	Injector not found in configuration.	0x80040208
E_EXP_NOSAMPLE	Injected sample not found in configuration.	0x80040209
E_SYS_INSTRUMENTS	Instrument hardware detection not finished.	0x8004020A
E_SYS_ACCESSDENIED	Insufficient privileges to perform operation.	0x8004020B
E_LIC_DISABLED	Missing license feature activation key.	0x8004020C
E_FILE_CORRUPT	Cannot read file. File is either from a new version of ASTRA or corrupt.	0x8004020D
E_DB_NOT_MIGRATED	ASTRA system database is currently migrating. Wait a few moments and try again.	0x8004020E
E_UV_NOT_DETECTED	No VISION UV instrument was found in configuration.	0x8004020F
E_UV_INVALID_CONFIG	Invalid configuration provided for the VISION UV.	0x80040210
E_UV_INVALID_DATA	UV data does not match expected input.	0x80040211
E_FILE_NAME_EXISTS	An attempt was made to save and the file name already existed.	0x80040212
E_FILE_CHECKED_OUT	This file is locked for editing by another user.	0x80040213
E_FILE_SAVE_FAILED	Unexpected file save failure.	0x80040214
E_EXP_NO_COLLECTED_DATA	Experiment did not collect any data.	0x80040215
E_EXP_NO_RESULTS	Failed to extract results from experiment.	0x80040216
E_EXP_RESULTS_SAVE_FAILED	Failed to save results.	0x80040217
E_EXP_NO_DATASET	Cannot find dataset from experiment.	0x80040218
E_EXP_DATASET_SAVE_FAILED	Failed to save dataset.	0x80040219
E_EXP_RUN_EXPERIMENT_FAILED	Failed to run experiment.	0x8004021A
E_EXP_AUTOFINDBASELINES_FAILED	Failed to autofind baselines.	0x8004021B
E_EXP_AUTOFINDEPOCHS_FAILED	Failed to autofind peaks.	0x8004021C
E_ASTRA_NOT_SHOWN	Failed to show ASTRA main window.	0x8004021D
E_REQUEST_OUT_OF_SEQUENCE	Before using this ASTRA functionality, you need to call SetAutomationIdentity .	0x8004021E
E_EXP_FAILED_TO_OPEN	Failed to allocate experiment's slot before opening it.	0x8004021F

E_UNEXPECTED_ASTRA_ERROR	ASTRA failed for an unknown reason. Check the ASTRA log file for more details.	0x80040220
E_EXP_CONFIGURATION_UPDATE_FAILED	Failed to update configuration with new parameters.	0x80040221
E_NULL_ARGUMENT	Got a null argument when a non-null one was expected.	0x80040222
E_SYS_LOAD_METHODS_FAILED	Failed to load methods from the ASTRA system database.	0x80040223
E_EXP_CREATION_FAILED	Failed to create a new experiment from a method.	0x80040224
E_EXP_GET_INFO_FAILED	Failed to retrieve some experiment details (such as name, pump flow, injected volume, ...).	0x80040225
E_EXP_BASIC_COLLECTION_NOT_FOUND	Could not find the Basic Collection Procedure in experiment.	0x80040226
E_EXP_CANNOT_VIEW_BASIC_COLLECTION	Could not view the Basic Collection Procedure.	0x80040227
E_EXP_PROCEDURE_UPDATE_FAILED	Could not update procedure to force waiting on auto-inject signal.	0x80040228
E_EXP_STOP_COLLECTION_FAILED	Failed to stop the collection.	0x80040229
E_NOT_ENOUGH_MEMORY	Not enough memory to complete current operation.	0x8004022A
E_ASTRA_ALREADY_IN_USE	ASTRA is already in use by another client. Please close the other client or ASTRA and restart this ASTRA client.	0x8004022B
E_SIZE_MISMATCH	Source and destination size does not match.	0x8004022C