Agilent IntuiLink PSA and ESA Automation

There are two automation objects in IntuiLink PSA and ESA automation:

- ESAEZ Automation Server Object
- PSA/ESA Utilities Object

PSA/ESA Utilities Object

Use the PSA/ESA Utilities Object to:

- Work with the spectrum analyzer connection to the PC.
- Work with the spectrum analyzer descriptions and identity.

ESAEZ Automation Server Object

Use this Automation Server for a smaller file size and faster performance. Recommended for:

- Visual Basic
- Visual C++

ESAEZ Automation Server Object

Properties   Methods   Events

The **ESAEZ Automation Server** allows you to communicate with Agilent PSA and ESA E/L/EMC-series spectrum analyzers using Visual Basic.

With the **ESAEZ Automation Server**, you can:

- control the instrument.
- download data.
- download a bitmap of the screen image.

Recommended for:

- Visual Basic
- Visual C++

Name:	ESA Automation Server
Description:	AgtESALib
File Name:	AgtESA.DLL

Syntax

ESAEZ

Remarks

- Before you can use the **ESAEZ** object in your application, you must reference the ESA Automation Server.
- To distribute applications you create with the **ESAEZ** object, you must install and register it on the user's computer.

Utilities Object

See Also   Properties   Methods   Applies To

Use the Utilities Object to:

- Work with the spectrum analyzer connection to the PC.
- Work with the spectrum analyzer descriptions and identity.

Objects

Properties

CenterFrequency Property

See Also   Applies to

Gets/sets the center frequency of the spectrum analyzer.

Syntax
object.**CenterFrequency** [= *value*]

Data Type

Double

Settings

value Sets the center frequency value in Hz unless otherwise specified. Factory default values and valid values are dependent on spectrum analyzer models, as shown below:

Model	Valid Range of Values	Factory Preset Value
HP ESA E4401B, E4411B, E7401A	-80 MHz to 1.58 GHz	750 MHz
HP ESA E4402B, E4403B, E7402A, E7403A	-80 MHz to 3.10 GHz	1.5 GHz
HP ESA E4404B, E7404A	-80 MHz to 6.78 GHz	3.35 GHz
HP ESA E4405B, E7405A	-80 MHz to 13.3 GHz	6.6 GHz
HP ESA E4407B, E4408B	-80 MHz to 27.0 GHz	13.25 GHz
HP PSA E4440A	-99.999995 MHz to 27 GHz	13.255 GHz

ComponentDescription Property

See Also   Applies to

Returns a description of the automation server. Read-Only.

Syntax
object.**ComponentDescription**

Data Type
String

ComponentManufacturer Property

See Also   Applies to

Returns the Component Manufacturer/Developer name. Read-Only.

Syntax
object.**ComponentManufacturer**

Data Type
String

ComponentProgID Property

See Also   Applies to

Returns the component Program ID of the automation server. Read-Only.

Syntax
object.**ComponentProgID**

Data Type
String

Remarks

- The program ID is placed in the system registry as a cross-reference to the GUID.

ComponentVersion Property

See Also   Applies to

Returns the version of the automation server. Read-Only.

Syntax
object.**ComponentVersion**

Data Type
String

ConnectionName Property

See Also   Applies to

Get the instrument's symbolic connection address/name. Read-Only.

Syntax
object.**ConnectionName**

Data Type

String

DetectDeviceErrors Property

See Also   Applies to

Enables/disables the device errors debugging function.

Syntax
object.**DetectDeviceErrors** [= {True | False}]

Data Type
Boolean

Settings

{True | False}
Default = FALSE

Remarks

- When DeviceDetectErrors is set to TRUE, the driver polls the instrument after any property or method that uses I/O to see if an error occurred. If an error is detected, the property or method will return an appropriate error.
- When DeviceDetectErrrors is FALSE, the driver will not check for instrument errors. If an I/O operation results in an instrument error, it will go undetected until either:

  DeviceDetectErrors is set to TRUE, or

The QueryInstrumentError method is used.

InputAttenuation Property

See Also   Applies to

Gets/sets the input attenuator. This value is set at the auto value if input attenuation is set to ‘On’. If input attenuation is set to ‘Off’, the value may be set manually.

Syntax
object.**InputAttentuation** [= *value*]

Data Type

Double

Settings

value Sets the input attenuator value in dB. Factory default value is 10 dB. Valid values are dependent on spectrum analyzer models, as shown below:

Model	Valid Range of Values
HP ESA E4401B, E4411B, E7401A	0 to 60 dB
HP ESA E4402B, E4403B, E7402A, E7403A	0 to 75 dB
HP ESA E4404B, E7404A	0 to 75 dB
HP ESA E4405B, E7405A	0 to 75 dB
HP ESA E4407B, E4408B	0 to 65 dB
HP PSA E4440A	0 to 70 dB

InstanceName Property

See Also   Applies to

Gets/sets the name of this instance of the automation server.

Syntax
object.**InstanceName** [= *value*]

Data Type

String

Settings

value As String may be set to any arbitrary string.

Remarks

- The instance name is used for logging and other features that require object identification.

InstrumentFirmwareVersion Property

See Also   Applies to

Returns the instrument's version. Read-Only.

Syntax
object.**InstrumentFirmwareVersion**

Data Type

String

Remarks

- The instrument version is returned as a part of the instrument's response to the IEEE 488.2 *IDN? Query.

InstrumentManufacturer Property

See Also   Applies to

Returns the name of the instrument's manufacturer. Read-Only.

Syntax
object.**InstrumentManufacturer**

Data Type

String

InstrumentModel Property

See Also   Applies to

Returns the instrument's model number. Read-Only

Syntax
object.**InstrumentModel**

Data Type

String

Remarks

- This property returns the same string as the SupportedModels property. For example: "4411B".

InstrumentSerialNumber Property

See Also   Applies to

Returns the instrument's serial number. This may not be supported by all instruments. Read-Only.

Syntax
object.**InstrumentSerialNumber**

Data Type

String

IO Property

See Also   Applies to

Returns an object that provides an interface to the underlying I/O automation server. Read-Only.

Syntax
object.**IO**

Remarks

- Returns the IO object from the automation server.

LogInterface Property

See Also   Applies to

Enables/disables the error logging facility.

Syntax
object.**LogInterface** [= {True | False}]

Data Type

Boolean

Settings

True enables error logging, False disables error logging. Default = False.

Remarks

- When enabled, the WriteLog Events are sent to clients registered In Visual Basic (using the WithEvents option).

RangeChecking Property

See Also   Applies to

Enables/disables input value range checking.

Syntax
object.**RangeChecking** [= {True | False }]

Data Type

Boolean

Settings

True enables range checking, False disables range checking.

Default = True

Remarks

- Use this property for program development and debugging.
- When enabled, values are checked for applicability to the spectrum analyzer model. Invalid ranges are reported before the I/O operation executes.
- The range checking operation slows program execution. For fastest program execution set RangeChecking to False, DetectDeviceErrors to False and LogInterface to False.

ReferenceLevel Property

See Also   Applies to

Gets/sets the amplitude value of the reference level for the Y-axis.

Syntax
object.**ReferenceLevel** [= *value*]

Data Type

Double

Settings

value   Sets the value in the current active units. The factory preset value is 0 dBm. The valid values for the ESA models are –149.9 to 50 dBm with zero reference level offset. The valid values for the PSA models are -170 dBm to 30 dBm.

Remarks

- The input attenuator setting may be affected. The minimum displayed value of reference level is –327.6, and the maximum displayed value is 327.6.

ResolutionBandwidth Property

See Also   Applies to

Gets/sets the resolution bandwidth.

Syntax
object.**ResolutionBandwidth** [= *value*]

Data Type

Double

Settings

value Sets the resolution bandwidth in Hz, unless otherwise specified. The factory preset is 3 MHz. Valid values for the ESA models are:

- 10 Hz to 5 MHz with Option 1DR, narrow resolution bandwidth
- 1 kHz to 5 MHz without Option 1DR

Valid values for PSA models are:

- 3 MHz to 8 MHz

ScalePerDivision Property

See Also   Applies to

Gets/sets the per-division display scaling for the Y-axis.

Syntax
object.**ScalePerDivision** [= *value*]

Data Type

Double

Settings

value Sets the per-division display scaling in dB. The factory preset is 10 dB. Valid values are 0.01 dB to 20.0 dB.

ScaleType Property

See Also   Applies to

Gets the vertical graticule divisions as log or linear units. Read-Only.

Syntax
object.**ScaleType**

Data Type

String

Remarks

- By default, the units are logarithmic.

SpanFrequency Property

See Also   Applies to

Gets/sets the frequency span.

Syntax
object.**SpanFrequency** [= *value*]

Data Type

Double

Settings

value Sets the center frequency value in Hz unless otherwise specified. Factory default values and valid values are dependent on spectrum analyzer models, as shown below:

Model	Valid Range of Values	Factory Preset Value
HP ESA E4401B, E4411B, E7401A	0 Hz to 1.58 GHz	1.5 GHz
HP ESA E4402B, E4403B, E7402A, E7403A	0 Hz to 3.10 GHz	3.0 GHz
HP ESA E4404B, E7404A	0 Hz to 6.78 GHz	6.7 GHz
HP ESA E4405B, E7405A	0 Hz to 13.3 GHz	13.2 GHz
HP ESA E4407B, E4408B	0 Hz to 27.0 GHz	26.5 GHz
HP PSA E4440A	0 Hz to 26.5 GHz	26.49 GHz

Remarks

- Setting the span to 0 Hz puts the analyzer into zero span.

StartFrequency Property

See Also   Applies to

Gets/sets the start frequency.

Syntax
object.**StartFrequency** [= *value*]

Data Type

Double

Settings

value Sets the start frequency value in Hz. The factory default value is 0 Hz for ESA models and 10.0 MHz for PSA models. Valid values are dependent on spectrum analyzer models, as shown below:

Model	Valid Range of Values
HP ESA E4401B, E4411B, E7401A	-80 MHz to 1.58 GHz
HP ESA E4402B, E4403B, E7402A, E7403A	-80 MHz to 3.10 GHz
HP ESA E4404B, E7404A	-80 MHz to 6.78 GHz
HP ESA E4405B, E7405A	-80 MHz to 13.3 GHz
HP ESA E4407B, E4408B	-80 MHz to 27.0 GHz
HP PSA E4440A	-100 MHz to 27 GHz

StopFrequency Property

See Also   Applies to

Gets/sets the stop frequency.

Syntax
object.**StopFrequency** [= *value*]

Data Type

Double

Settings

value Sets the stop frequency value in Hz. The factory default values and valid values are dependent on spectrum analyzer models, as shown below:

Model	**Valid Range of Values**	**Factory Preset Value**
HP ESA E4401B, E4411B, E7401A	-80 MHz to 1.58 GHz	1.5 GHz
HP ESA E4402B, E4403B, E7402A, E7403A	-80 MHz to 3.10 GHz	3.0 GHz
HP ESA E4404B, E7404A	-80 MHz to 6.78 GHz	6.7 GHz
HP ESA E4405B, E7405A	-80 MHz to 13.3 GHz	13.2 GHz
HP ESA E4407B, E4408B	-80 MHz to 27 GHz	26.5 GHz
HP PSA E4440A	-99.99999 MHz to 27GHz	26.5 GHz

SupportedModels Property

See Also   Applies to

Returns a list of supported spectrum analyzer models. Read-only.

Syntax
object.**SupportedModels**

Data Type

String

SweepMode Property

See Also   Applies to

Gets the sweep mode for the selected trace. Read-Only.

Syntax
object.**SweepMode**

Data Type

String

Settings

Returns the value AgtESA_SweepMode_Continuous if the analyzer is in continuous sweeping mode.

Returns the value AgtESA_SweepMode_Single when there is only a single sweep.

SweepTime Property

See Also   Applies to

Gets/sets the time in which the instrument sweeps the display.

Syntax
object.**SweepTime** [= *value*]

Data Type

String

Settings

value Sets the time, in seconds, in which the instrument sweeps the display. For ESA models, valid values are:

5 microseconds to 2000 seconds without Option AYX, fast digitized time domain sweeps

20 microseconds to 2000 seconds with Option AYX, in zero span only

For PSA models, valid values are:

1 microsecond minimum; 2 kiloseconds maximum

Remarks

- A span value of 0 Hz causes the analyzer to enter zero span mode. In zero span the X-axis represents time rather than frequency. In this mode, the sweep time may be set to faster values when Option AYX (fast digitized time domain sweeps) is installed.

Timeout Property

See Also   Applies to

Gets/sets the automation server's I/O timeout value in milliseconds.

Syntax
object.**Timeout** [= *value*]

Data Type

Long

Settings

value As Long sets the timeout in milliseconds. For example, setting *value* to 1000 sets a 1-second timeout value. Default = 5000.

Remarks

- For example, use the following statement to set a 5 second timeout.

```
AgtESAScope1.Timeout = 5000
```

Utilities Property

See Also   Applies to

Returns a UtilitiesESA object that provides an interface to a set of instrument utility functions. Read-Only.

Syntax
object.**Utilities**

VideoBandwidth Property

See Also   Applies to

Gets/sets the video bandwidth.

Syntax
object.**VideoBandwidth** [= *value*]

Data Type

Double

Settings

value Sets the video bandwidth in Hz, unless otherwise specified. The factory preset is 3 MHz. Valid values for ESA models are 1 Hz to 3 MHz. Valid values for PSA models are 1 Hz to 50 MHz. This range is dependent on the setting of:

`[:SENSe]:BANDwidth|BWIDth [RESolution]`

Methods

Connect Method

See Also   Applies to

Connects the automation server to an instrument at the specified address or symbolic name.

Syntax
object.**Connect** (connectionname, [IOProgID])

Settings

connectionname As String is the symbolic name of the connection.

IOProgID As String is an optional I/O address of the connection.

Data Type

String

Remarks

- GPIB

  Use a string in this form:

  GPIB*m* ::*n*

  where *m* is the board number, and *n* is the instrument GPIB address (for example, "GPIB0::22").

  Alternately use a VISA address ("GPIB::22::INSTR"). The I/O operations do not require that VISA be installed on the PC.

- RS-232

  Use a string in this form:

  COM*m* ::*parametername*=*nn*

  where *m* is the RS-232 port and *parametername* is one of the parameters described below. A comma separates multiple parameter names. Any, all, or no parameters may be used. If a parameter is missing, the default value is used.

Baud=*nnnn*	where *nnnn* are the digits of the baud rate. Default = 9600
Handshake=*s*	where *s* is none, xon_xoff, or dtr_dsr. Default = xon_xoff

Examples

```
"COM1::Baud=9600"

"COM2::Baud=2400,Handshake=xon_xoff"
```

ClearDevice Method

See Also   Applies to

Performs a device clear of the instrument.

Syntax
object.**ClearDevice**

ClearStatus Method

See Also   Applies to

Clears the instrument's status registers.

Syntax
object.**ClearStatus**

Close Method

See Also   Applies to

Closes the communication connection with the instrument.

Syntax
object.**Close**

Remarks

- If the session is not open or has already been closed, the Close method does nothing.

DisplayMessage Method

See Also   Applies to

Writes the specified message string to the instrument's display.

Syntax
object.**DisplayMessage** (*message*)

Settings

message As String is an arbitrary string.

Remarks

- The instrument limits the display to 44 characters.

Enter Method

See Also   Applies to

Reads data from the instrument as a string or number.

Syntax
object.**Enter** (*result* [, *format*])

Remarks

- Use this method to enter instrument data. Use the Output method to send instrument commands.
- For most applications that return a string or a number, no format argument is needed. These examples show the most common usage. *ESA* is an object in the Applies to list.

Returning a string

```
Dim reply As String
ESA.Enter reply
```

Returning a number

```
Dim reading As Double
ESA.Enter reading
```

Settings

result returns the data. The optional Format parameter determines how the data is returned.

format As String determines the format of the returned data.
Default = K (Freefield entry)

format is optional, or can contain one or two format identifiers separated by commas. The order of the format identifiers is ignored.

The format characters are:

"K" for Freefield entry,

"#" for don't flush buffer entry, and

IEEE 488.2 block formatted data. (See description below.)

Do not use the Freefield character and the IEEE 488.2 block characters in the same format string.

Format Identifiers

"K" -- Freefield

The data is interpreted based upon the data type of the result parameter. Use the data type best suited for the data returned. K is the default if no Format string is given.

result **data type**	**Description**
String	Characters are placed in the string. Carriage-return not immediately followed by line-feed is entered into the string. Entry to a string terminates on CR/LF, LF, or a character received with EOI true.
String()	Same as string, but parses the received characters at any comma. The entry terminates as in String or when the array is full.
Numeric	Returns the first number of the ASCII data returned from the instrument. Leading non-numeric characters are ignored. All blanks are ignored. Trailing non-numeric characters and characters sent with EOI true are delimiters. Numeric characters include digits, decimal point, +, -, e, and E when their order is meaningful. Valid data types are Byte, Long, Integer, Double, and Single.
Numeric()	Same as Numeric, but parses the ASCII string from the instrument and fills the array. The entry terminates when the array is full or at the end of data.
Variant	Same as string.
Variant ()	Same as string except that array is filled until end on CR/LF, LF, or a character is received with EOI true.

"#" -- Don't flush buffer

Saves the remaining data in the buffer after completion of Enter method. When the instrument returns several numbers as one ASCII string, you can retain any remaining data in the buffer by using this format character when reading with an Enter method. In the following example, the instrument returns two data points. The first line reads the first data point, and the second line reads the second data point after which the data in the buffer is discarded. The variables *reading1* and *reading2* are declared as double.

```
Spectrum Analyzer.Enter reading1, "K,#"
Spectrum Analyzer.Enter reading2, "K"
```

IEEE 488 block data

Using the Enter command with a format statement you can read IEEE 488.2 block data. This is a standard format used by some instruments to return large amounts of data in a binary form.

Setting	Description
I1	Integer, 1 byte
I2BE	Integer, 2 bytes, Big Endian
I2LE	Integer, 2 bytes, Little Endian
I4BE	Integer, 4 bytes, Big Endian
I4LE	Integer, 4 bytes, Little Endian
R4BE	Real, 4 bytes, Big Endian
R4LE	Real, 4 bytes, Little Endian
R8BE	Real, 8 bytes, Big Endian
R8LE	Real, 8 bytes, Little Endian

GetInstrumentDateTime Method

See Also   Applies to

Gets the date/time from the instrument. Read-only.

Syntax

object.**GetInstrumentDateTime**

Data Type

Variant(Date)

GetScreenImage Method

See Also   Applies to

Returns a bitmap image of what is currently displayed on the spectrum analyzer's display screen.

Syntax

object.**GetScreenImage** [(*format*)]

Data Type

Picture

Settings

format As Agt_ImageFormat defines the format for the image. The format can be set to BMP,GIF, TIF, JPG, WMF or EMF.

Remarks

- You can use the GetScreenImage method to insert the screen image in a control (such as Image or PictureBox) or you can save the screen image in a variable declared as a Picture.
- This method extends the timeout value in effect to allow for execution and then restores the timeout value to the one set by the Timeout property.
- This code inserts the screen image in an image control.

Set Image1.Picture = AgtESA1.GetScreenImage

- This code will set the variable screenShot to the screen image and then insert it in an Image control.

```
Dim screenShot As Picture
Set screenShot = AgtESA1.GetScreenImage
Set Image1.Picture = screenShot
```

GetSweepNumPoints Method

See Also   Applies to

Gets the number of points in the sweep from the instrument. Read-only.

Syntax
object.**GetSweepNumPoints**

Data Type

Long

Remarks

- This feature is available only in ESA L-series firmware with revision A.04.00 or greater.

GetTrace Method

See Also   Applies to

Gets the number of points in the specified trace.

Syntax
object.**GetTrace** (channel)

where channel is a value of 1, 2, or 3.

Data Type

Variant array of longs. The number of points returned will be the same as the current setting on the spectrum analyzer, a number between 101 and 8192.

Options Method

See Also   Applies to

Returns a variant containing a list of the options installed in the currently connected instrument.

Syntax
object.**Options**

Output Method

See Also   Applies to

Sends a string to the instrument.

Syntax
object.**Output** (*string*)

Settings

string As String.

Remarks

- Use this command to send instrument commands. Use the Enter method to read the reply from the instrument.

  For example, this code requests a peak-to-peak voltage measurement

  Spectrum Analyzer is an object in the Applies to list.

```
Spectrum Analyzer.Output "Measure:VPP?"
```

Preset Method

See Also   Applies to

Performs a preset of the instrument.

Syntax
object.**Preset**

QueryInstrumentError Method

See Also   Applies to

Returns a number and a string that report the most recent error in the instrument's error queue.

Syntax
object.**QueryInstrumentError** (errornumber, errordescription)

Data Type

errornumber As Long returns the instrument error number.

errordescription As String returns the instrument error string.

Remarks

- The instrument errors are reported as a signed number and, on some instruments, a descriptive string. Refer to the instrument's documentation for a complete list of error codes.
- Instruments with no errors in the error queue return 0,"No Error".
- Errors are reported on a first-in, first-out basis.
- You can empty the error queue without reading all the errors using the ClearStatus method.

ReadStateData Method

See Also   Applies to

Reads the current system setup data from the instrument. The system setup data is encoded in the binary IEEE 488.2 definite block format. The data within the definite block is encoded in the instrument's internal format.

Syntax
object.**ReadStateData**

Data Type

Variant

Remarks

- This method extends the timeout value in effect to allow for execution and then restores the timeout value to the one set by the Timeout property.

RecallState Method

See Also   Applies to

Recalls the instrument state from its specified internal register and updates the current instrument state.

Syntax
object.**RecallState** (*statenumber*)

Settings

statenumber As Long is the state register to recall.

Reset Method

See Also   Applies to

Performs a reset of the instrument.

Syntax
object.**Reset**

SaveScreenImage Method

See Also   Applies to

Saves a bitmap image of what is currently displayed on the spectrum analyzer's display screen to the specified file.

Syntax

object.**SaveScreenImage** (*filename*, [*format*])

Settings

filename As String contains a valid filename and path.

format As Agt_ImageFormat sets the type of file to save (BMP, GIF, WMF, EMF, TIF, JPG).

Remarks

- This method extends the timeout value in effect to allow for execution and then restores the timeout value to that set by the Timeout property.

SaveState Method

See Also   Applies to

Saves the current instrument state to its specified internal register.

Syntax
object.**SaveState** (statenumber)
where *statenumber* is from 2 to 30.

Settings

statenumber As long is the internal register number to use.

SelfTest Method

See Also   Applies to

Performs a self-test on the connected instrument and returns results of the test.

Syntax

object.**SelfTest** (testresult, resultmessage)

Data Types

testresult As Long returns a number indicating the self-test status.

resultmessage As String returns a string indicating the self-test result.

Remarks

- The instrument returns a "0" and an empty string if the self-test passes.
- If the self-test has a failure, the instrument returns a value and a string indicating the nature of the failing test. Some instruments return only the *testresult* value and *resultmessage* is set to null. Refer to the instrument documentation for results returned by the *TST command.
- This method extends the timeout value in effect to allow for execution and then restores the timeout value to the one set by the Timeout property.

StatusBits Method

See Also   Applies to

Returns the contents of the 'Status Events' and 'Service Request' registers of the instrument. This is a destructive read operation. Status Event register is returned in lower word. Service Request register is returned in upper word.

Syntax
object.**StatusBits**

Data Type

Long

TraceMode Method

See Also   Applies to

Gets the display mode for the selected trace. Read-Only.

Syntax
object.**TraceMode** (TRACe # |) WRITe | MAXHold | MINHold | VIEW | BLANk|)

Data Type

String

Settings

Returns the display mode for the selected trace, one of WRIT, MAXH, MINH, VIEW, or BLAN.

- Write puts the trace in the normal mode, updating the data.
- Maximum hold displays the highest measured trace value for all the data that has been measured since the function was turned on.
- Minimum hold displays the lowest measured trace value for all the data that has been measured since the function was turned on.
- View turns on the trace data so that it can be viewed on the display.
- Blank turns off the trace data so that it is not viewed on the display.

VerifyDevice Method

See Also   Applies to

Verifies, by returning a Boolean operator, that the connected instrument is compatible with the specified parameters.

Syntax

object.**VerifyDevice** (*model*, [*version*], [*serialnumber*])

Data Type

Boolean

Settings

model As String is the only required parameter.

version As String is an optional parameter.

serialnumber As String is an optional parameter.

WriteStateData Method

See Also   Applies to

Writes system setup data in the specified buffer to the instrument and updates current system setup. See the ReadStateData method.

Syntax

object.**WriteStateData** (*statedata*)

where *statenumber* is 2 to 30.

Settings

statedata As Variant is a buffer containing the state data.

Remarks

- This method extends the timeout value in effect to allow for execution and then restores the timeout value to the one set by the Timeout property.

Events

WriteLog Event

Applies to

This event is sent from the ILog interface. The ILog interface sends diagnostic trace messages for every method of the automation server.

Syntax

object.**WriteLog** (*source*, *logging*, *logmessage*)

Settings

source As String is the source of the event. *Source* should be either the InstanceName if not null or the class name parsed from the ComponentProgID.

logging As Log Type sets the type of logging as Error, Trace, Warning, or information event.

logmessage As String contains the message to be logged.

Remarks

- The I/O and device servers let the client handle actually writing to a log or trace file, and clients are free to handle the messages however they see fit.

Constants

Agt_ImageFormat Constants

Used With

Value	Constant
0	Agt_ImageFormat_BMP
2	Agt_ImageFormat_GIF
3	Agt_ImageFormat_WMF
4	Agt_ImageFormat_EMF
5	Agt_ImageFormat_TIF
6	Agt_ImageFormat_JPG

Technical Support

Technical Support

Agilent provides programming samples for illustration purposes only, without warranty either expressed or implied, including, but not limited to, the implied warranties of merchantability and/or fitness for a particular purpose.

This help file assumes that you are familiar with the programming language being demonstrated and the tools used to create and debug procedures. Agilent support engineers can help answer questions relating to the functionality of the software components provided by Agilent, but they will not modify these samples to provide added functionality or construct procedures to meet your specific needs.

To contact Agilent for technical assistance, refer to the support section in the README.TXT file located in the directory where you installed IntuiLink. By default, IntuiLink is installed in the following directory:

...\Program Files\Agilent\IntuiLink\ESA

If you have limited programming experience, please contact the manufacturer of your development language for further information and assistance.