### **ActiveX Properties and Methods for LabVIEW**

This book describes the LabVIEW properties and methods that third-party software can access through the ActiveX protocol.

(Windows) To view related topics, click the **Locate** button, shown at left, on the toolbar at the top of this window. The *LabVIEW Help* highlights this topic in the **Contents** tab so you can navigate the related topics.

# **Application Properties and Methods**

This book contains the property and method reference information for the Application class.

# **Application Methods (ActiveX)**

Installed With: Base Package

An Application object exports methods that affect LabVIEW.



**Note** Method parameters denoted by [] are optional for that method.

Refer to the <u>ActiveX Enumerations</u> for more information about the enumerations used in various methods.

Method	Description
BringToFront	On Windows and Mac OS, brings the application windows to the front. Details
BrowseDataSocket	Launches the National Instruments <b>DataSocket Browser</b> dialog box to establish a connection to a DataSocket item. Details
CreateLibrary	Creates a new LabVIEW project library. Details
GetVIReference	Places a VI in memory and returns the IDispatch pointer for the VI. <u>Details</u>
GetVIVersion	Returns the version of LabVIEW in which the VI was last saved. LabVIEW does not open the VI when retrieving the version information. If you specify a path to a file that is not a VI, LabVIEW returns error 6559. Details
LibraryGetFileLVVersion	Returns the version of LabVIEW in which the LabVIEW project library was created. Details
MassCompile	Loads and compiles all the VIs in a directory, including any VIs in subdirectories, for the specified application instance. Details
NewProject	Creates a new, empty LabVIEW project. You also can use the LabVIEW <b>Project Explorer</b> window to create a new project. <u>Details</u>
OpenLibrary	Opens the LabVIEW project library. <u>Details</u>
OpenProject	Loads a LabVIEW project from disk. If the project is already in memory, this method returns a reference to the existing project. You

	can use this reference with the Project properties. In LabVIEW, you also can select File»Open Project to navigate to and open a project. Details
Quit	Quits the application. Details

# **BringToFront Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

On Windows and Mac OS, brings the application windows to the front.

object.BringToFront()

none

#### **BrowseDataSocket Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Launches the National Instruments **DataSocket Browser** dialog box to

establish a connection to a DataSocket item.

object.BrowseDataSocket([prompt], [selectedURL])

#### **Parameters**

Name	Туре	Description
prompt	_	Specifies the title of the <b>DataSocket Browser</b> dialog box.
selectedURL		Returns the URL of the connection specified by the user in the <b>DataSocket Browser</b> dialog box.

none

# **CreateLibrary Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Creates a new LabVIEW project library.

object.CreateLibrary([createPalette])

#### **Parameters**

Name	Туре	Description
createPalette	Boolean	Specifies whether you want to set a default
		palette for all VIs the project library owns. The default is FALSE, which does not set a default palette.

Library\*

### **GetVIReference Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Places a VI in memory and returns the IDispatch pointer for the VI.

object.GetVIReference(viPath, [password], [resvForCall], [options])

#### **Parameters**

Name	Туре	Description
viPath	String	Absolute path to the VI placed in memory.
password	String	If the VI is password-protected, you must enter the password to make any edits.
resvForCall	reserve the VI of its pare the time VI. How the edit the VI of this refe	If TRUE, the VI is set to an execution state called reserved. You cannot edit a reserved VI because the VI can be called as a subVI at any time while its parent VI runs. Setting the VI to reserved at the time of the reference speeds up calls to the VI. However, this state does not support any of the edit mode properties or methods. To bring the VI out of the reserved execution state, close this reference and open a new reference by setting <b>resvForCall</b> to FALSE.
		If you are working with a reentrant VI, it is important to set this parameter to TRUE so that LabVIEW can call the VI efficiently.
options	long	A bit set that specifies how the VI reference is treated. <b>options</b> can be a combination of the following values. The default is $0x10$ .
		0x01 Record modifications. An asterisk (*) appears by the VI title to indicate that changes have been made using VI Server. The VI must be in edit mode for LabVIEW to record the modifications.
		Ox02 Open templates for editing. This option opens the original .vit file. If you do not select this option, LabVIEW opens a new instance of the template VI. Edits made to an instance do not affect the original .vit file.This option has no effect on nontemplate files.
		0x04 Prompt user to save changes when this VI reference closes if all the following

	<ul> <li>The referenced VI or its subVIs contain unsaved changes.</li> <li>There are no other open references to the referenced VI.</li> <li>The referenced VI is able to leave memory. A VI is able to leave memory, for example, if no other VIs call the VI, the front panel of the VI is closed, and the VI is not a member of an open project library, and so on.</li> </ul>
0x08	Prepare for reentrant run. Reserves the target VI so it cannot be edited and if the target VI is reentrant, allocates a dedicated parallel data space for this VI reference. If the target VI is not reentrant, this method returns an error. When you release the VI reference, LabVIEW unreserves the reentrant target VI and deallocates a parallel data space. Use this option with the Run method to run multiple instances of a reentrant VI simultaneously. If you target a reentrant VI and do not use this option, this method returns a reference to the VI without allocating a parallel data space for the VI reference. When you do not use this option, multiple calls to this method for a reentrant VI return references to the same VI with the same data space, and this method does not clone the VI.
0x10	Prompt user to find missing subVIs of the referenced VI.
0x20	Do not display the loading dialog box when searching for missing subVIs of the referenced VI.



**Note** This option does not affect whether LabVIEW prompts you to find the missing VIs or not.

VirtualInstrument \*

#### **GetVIVersion Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Returns the version of LabVIEW in which the VI was last saved. LabVIEW does not open the VI when retrieving the version information. If you specify a path to a file that is not a VI, LabVIEW returns error 6559.

object.GetVIVersion(viPath, versNum)

#### **Parameters**

Name	Туре	Description
viPath	String	Path to the VI whose version you want to get.
versNum	unsigned long	Returns the version number of the VI specified in <b>viPath</b> .

String

# **LibraryGetFileLVVersion Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Returns the version of LabVIEW in which the LabVIEW project library

was created.

object.LibraryGetFileLVVersion(libPath, [versNum])

#### **Parameters**

Name	Type	Description
libPath	path	Specifies the path to the library.
versNum	Long*	Returns the version number.

String

## **MassCompile Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Loads and compiles all the VIs in a directory, including any VIs in subdirectories, for the specified application instance.

This method is similar to the **Mass Compile** option on the **Mass Compile** dialog box.

object.MassCompile(directory, [logFile], [appendLog], [viCacheSize], [reloadLVSBs], [User Stopped])

#### **Parameters**

Name	Туре	Description
directory	String	Directory where the mass compile begins.
logFile	String	File path where the results of the mass compile are placed. The default is no log file.
appendLog	Boolean	Indicates whether results are appended to the log file. The default is FALSE.
viCacheSize	Long	Number of VIs allowed to be in memory during the mass compile. The default is none.
reloadLVSBs	Boolean	If TRUE, ignores CINs in VIs and the application searches for them. Use this parameter when a large number of CINs have been recompiled and need to be reloaded. The default is FALSE.
User Stopped		Indicates whether the user stopped the mass compile operation.

none

# **NewProject Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Creates a new, empty LabVIEW project. You also can use the LabVIEW

**Project Explorer** window to create a new project.

object.NewProject()

Project\*

# **OpenLibrary Method**

**Installed With:** Base Package

Class: Application Methods (ActiveX)

Opens the LabVIEW project library.

object.OpenLibrary(path)

#### **Parameters**

Name	Туре	Description
path	String	Specifies the path to the project library to open.

#### **Return Value**

Library\*

### **OpenProject Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Loads a LabVIEW project from disk. If the project is already in memory, this method returns a reference to the existing project. You can use this reference with the <u>Project</u> properties.

In LabVIEW, you also can select **File»Open Project** to navigate to and open a project.

object.OpenProject(path)

#### **Parameters**

Name	Туре	Description
path	String	Path to the project.

#### **Return Value**

Project\*

# **Quit Method**

Installed With: Base Package

Class: Application Methods (ActiveX)

Quits the application.

object.Quit()

#### **Return Value**

## **Application Properties (ActiveX)**

Installed With: Base Package

An Application object exports properties that affect LabVIEW.



**Note** Property parameters denoted by [] are optional for that property or method.

Refer to the <u>ActiveX Enumerations</u> for more information about the enumerations used in various properties.

Property	Description
AllProjects	Returns an array of references to each LabVIEW project in memory. You can use these references with the Project properties. Details
AllVIsInMemory	Returns a list of VIs in memory for the calling VI application instance. This property returns an error if you use it to return a list of remote VIs. Use the <a href="ExportedVIs">ExportedVIs</a> property to return a list of exported VIs. This property is read only. This property is available only in the local versions of LabVIEW. <a href="Details">Details</a>
AppKind	Indicates the LabVIEW instance the VI is running in, including invalid application types. Details
ApplicationDirectory	Absolute path to the directory where the application is located. This property is read only.  Details
AppName	Filename of the application. This property is read only. Details
AppTargetCPU	Indicates the target CPU of an

	application. This property is read only. Details
AppTargetOS	Indicates the operating system for which the application was built. This property is read only.  Details
AutomaticClose	Specifies whether the application closes when all the references to it are released. The default is TRUE. Details
CmdArgs	Returns an array of user-defined command-line arguments passed when LabVIEW launched. Details
DefaultDataLocation	Path to the LabVIEW Data directory. Details
ExportedVIs	Returns a list of exported VIs in memory. This property is read only. Details
Language	Indicates the language of the LabVIEW environment or standalone application as a string according to ISO 639. Values include en, de, fr, ja, ko, and zhon. Details
OSName	Name of the operating system where the application is actually running. This string value is the same as <a href="AppTargetOS">AppTargetOS</a> , except among versions of Windows—specifically, <a href="AppTargetOS">AppTargetOS</a> has one enumeration on Windows 2000/XP while <a href="OSName">OSName</a> returns the exact version of Windows your application is running on. This property is read only. <a href="Details">Details</a>

OSVersion	Version number of the actual operating system. This property is read only. <u>Details</u>
PrintDefaultPrinter	Gets or sets the name of the default printer in LabVIEW.  Details
PrintersAvailable	Returns an alphabetized array of printer names available on the computer. <u>Details</u>
PrintingColorDepth	If TRUE, LabVIEW sends color/grayscale output to the printer. If FALSE, LabVIEW sends monochrome output to the printer. Details
PrintMethod	Gets or sets how LabVIEW prints. Details
PrintSetupCustomConnector	Specifies whether to print the connector pane and icon of VIs when using the custom format with one of the print documentation VI methods.  Details
PrintSetupCustomControlDesc	Specifies whether to print descriptions of front panel controls when using the custom format with one of the print documentation VI methods.  Details
PrintSetupCustomControls	Specifies whether to print LabVIEW front panel control information when using the custom format with one of the print documentation VI methods. This property does not affect the image of the front panel, just the list of controls and indicators in

	the generated documentation.  Details
PrintSetupCustomControlTypes	Specifies whether to print data type information for LabVIEW front panel controls when using the custom format with one of the print documentation VI methods. Details
PrintSetupCustomDescription	Specifies whether to print VI descriptions when using the custom format with one of the print documentation VI methods.  Details
PrintSetupCustomDiagram	Specifies whether to print LabVIEW block diagrams when using the custom format with one of the print documentation VI methods. Details
PrintSetupCustomDiagramHidden	Specifies whether to print the hidden frames in LabVIEW Case and Stacked Sequence structures that might be present on the block diagram when using the custom format with one of the print documentation VI methods. Details
PrintSetupCustomDiagramRepeat	Specifies whether to print visible subdiagrams of each Case, Event, and Stacked Sequence structure in sequence with the non-visible subdiagrams when using the custom format with one of the print documentation VI methods. If you print the visible frames in sequence, those frames will print twice.  Details

Specifies whether to print the configuration information for any LabVIEW Express VIs on the block diagram when using the custom format with one of the print documentation VI methods. Details
Specifies whether to print the hierarchy of the VI in memory when using the custom format with one of the print documentation VI methods.  Details
Specifies whether to print the VI revision history information when using the custom format with one of the print documentation VI methods.  Details
Specifies whether to print the label and/or caption for each control when using the custom format with one of the print documentation VI methods.  Details
Specifies whether to print the LabVIEW front panel when using the custom format with one of the print documentation VI methods. Details
Specifies whether to print the LabVIEW front panel with a border when using the custom format with one of the print documentation VI methods.  Details

PrintSetupCustomSubVIs	Specifies whether to print a list of the LabVIEW subVIs and Express VIs including the icon, name, and path when using the custom format with one of the print documentation VI methods. If a polymorphic VI is a subVI, LabVIEW prints the instance used in the top-level VI including the icon, name, and path. If the top-level VI is polymorphic, LabVIEW prints a list of all the instances of the polymorphic VI including the icon, name, and path of each instance. Details
PrintSetupFileWrapText	Maximum number of characters on a single line in a file. Affects the PrintVIToHTML, PrintVIToRTF, and PrintVIToText methods. Set this property to 0 to print all the characters on one line so the text does not wrap. Details
PrintSetupJPEGQuality  DrintSetupJPEGQuality	Percentage value from 0 to 100 specifying the level of quality you want for the JPEG graphic in VIs printed either interactively or with the PrintVITOHTML method. The scale balances image quality and file size. A value in the 75-95 range produces a compressed file with a high-quality image, and a value below 50 produces a smaller file size with a low-quality image. The default is 80. Details
PrintSetupPNGCompressLevel	Number between –1 and 9 that

	specifies the level of compression you want for the PNG file for graphics in VIs printed either interactively or with the PrintVITOHTML method. The quality of the graphic is not affected by the compression, but the graphic file size and speed of compression are affected by this value. Valid values range from –1 to 9 and balance file compression with speed. O designates no compression. 1 designates the best speed with some compression. 9 designates the best compression, but slower speed. –1 (the default value) designates the best combination of good compression and speed. Details
RTHostConnected	If you are not targeted to a real-time (RT) platform, this property is always TRUE. If you are targeted to an RT-platform and the host is connected, this property is TRUE. When this property is FALSE, the host is not connected, requiring that the RT Module application must be able to handle all operations independent of the host. Use this property to determine when it is safe to display a dialog box that requires user interaction.  Details
SaveVersions	An array of the LabVIEW versions to which this version of

	LabVIEW can save. The array is in reverse chronological order, with the element at position zero in the array being the current version of LabVIEW. Each element of the array is a valid version to pass to VI server functions which save to previous versions. Details
ShowFPTipStrips	Writing the property: Setting this property to TRUE enables the display of tip strips on LabVIEW front panel controls. Setting the property to FALSE prevents them from being displayed. Reading the property: Specifies whether tip strips display when idling over controls on the front panel. Details
UserName	User name used to open the application. This property is read only. Details
Version	Version number of the application. This property is read only. Details
VIServerPort	Gets or sets the LabVIEW VI server port. LabVIEW returns an error if you attempt to access this property on a remote application. Details

# **AllProjects Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Returns an array of references to each LabVIEW project in memory. You can use these references with the <u>Project</u> properties.

object.AllProjects

# **Data Type**

Array of projects

#### **AllVIsInMemory Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Returns a list of VIs in memory for the calling VI application instance. This property returns an error if you use it to return a list of remote VIs. Use the <a href="ExportedVIs">ExportedVIs</a> property to return a list of exported VIs. This property is read only. This property is available only in the local versions of LabVIEW.



**Note** If you have multiple application instances open simultaneously, be sure to wire an application reference to the **reference** input. For example, LabVIEW opens a new application instance each time you create a LabVIEW project or a target for a LabVIEW project.

object.AllVIsInMemory

### **Data Type**

Array of strings

### **AppKind Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Indicates the LabVIEW instance the VI is running in, including invalid application types.

- 1. Development System, including the Base Package, Full Development System, or Professional Development System
- 2. Run-Time System
- 3. Student Edition
- 4. Embedded LabVIEW, including LabVIEW Real-Time
- 5. Evaluation
- 6. Custom

object.AppKind

### **Data Type**

<u>AppKindEnum</u>

# **ApplicationDirectory Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Absolute path to the directory where the application is located. This

property is read only.

object.ApplicationDirectory

## **Data Type**

String

# **AppName Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Filename of the application. This property is read only.

object.AppName

## **Data Type**

String

# **AppTargetCPU Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Indicates the target CPU of an application. This property is read only.

object.AppTargetCPU

<u>AppTargCPUEnum</u>

# **AppTargetOS Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Indicates the operating system for which the application was built. This

property is read only.

object.AppTargetOS

<u>AppTargOSEnum</u>

# **AutomaticClose Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether the application closes when all the references to it are

released. The default is TRUE.

object.AutomaticClose

Boolean

#### **CmdArgs Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Returns an array of user-defined command-line arguments passed when LabVIEW launched.

User-defined arguments start after two hyphens (--) surrounded by spaces in the command line. The first string in the array is the name of the executable launched. This property does not return the name of the VI launched or the LLB that contains the VI. If a user-defined command-line argument contains double quotation marks ("), this property returns the argument without the quotation marks.

If you use this property in a stand-alone application, you can pass all arguments as user-defined arguments so you do not need to enter the two hyphens before user-defined arguments in the command line.

This property is similar to the **Pass all command line arguments to application** option on the **Advanced** page of the LabVIEW **Application Properties** dialog box.

object.CmdArgs

Array of strings

#### **DefaultDataLocation Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Path to the LabVIEW Data directory.

You can use this directory to store the data files LabVIEW generates, such as .lvm or .txt files. When you install LabVIEW, the installer creates a LabVIEW Data subdirectory in the default file directory for the operating system to help you organize and locate the data files LabVIEW generates.

object.DefaultDataLocation

String

# **ExportedVIs Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Returns a list of exported VIs in memory. This property is read only.

object.ExportedVIs

Array of strings

### **Language Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Indicates the language of the LabVIEW environment or stand-alone application as a string according to ISO 639. Values include  $\rm en, de, fr, ja, ko, and zh-cn.$ 

object.Language

String

### **OSName Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Name of the operating system where the application is actually running. This string value is the same as <a href="AppTargetOS">AppTargetOS</a>, except among versions of Windows—specifically, <a href="AppTargetOS">AppTargetOS</a> has one enumeration on Windows 2000/XP while <a href="OSName">OSName</a> returns the exact version of Windows your application is running on. This property is read only.

object.OSName

string

# **OSVersion Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Version number of the actual operating system. This property is read

only.

object.OSVersion

string

# **PrintDefaultPrinter Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Gets or sets the name of the default printer in LabVIEW.

To set the default printer, enter the full path to the printer, such as

 $\mbox{myprinter}\ab01.$ 

object.PrintDefaultPrinter

String

# **Printers Available Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Returns an alphabetized array of printer names available on the

computer.

object.PrintersAvailable

Array of strings

# **PrintingColorDepth Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

If TRUE, LabVIEW sends color/grayscale output to the printer. If FALSE,

LabVIEW sends monochrome output to the printer.

object.PrintingColorDepth

Boolean

# **PrintMethod Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Gets or sets how LabVIEW prints.

object.PrintMethod

<u>PrintMethodsEnum</u>

### **PrintSetupCustomConnector Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the connector pane and icon of VIs when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Connector

# **PrintSetupCustomControlDesc Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print descriptions of front panel controls when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Control Desc

#### **PrintSetupCustomControls Property**

Installed With: Base Package

**Class**: Application Properties (ActiveX)

Specifies whether to print LabVIEW front panel control information when using the custom format with one of the print documentation <u>VI methods</u>. This property does not affect the image of the front panel, just the list of controls and indicators in the generated documentation.

object. Print Setup Custom Controls

# **PrintSetupCustomControlTypes Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print data type information for LabVIEW front panel controls when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Control Types

# **PrintSetupCustomDescription Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print VI descriptions when using the custom format

with one of the print documentation VI methods.

object. Print Setup Custom Description

# **PrintSetupCustomDiagram Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print LabVIEW block diagrams when using the custom format with one of the print documentation  $\frac{VI\ methods}{}$ .

object. Print Setup Custom Diagram

### **PrintSetupCustomDiagramHidden Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the hidden frames in LabVIEW Case and Stacked Sequence structures that might be present on the block diagram when using the custom format with one of the print documentation VI methods.

object. Print Setup Custom Diagram Hidden

#### **PrintSetupCustomDiagramRepeat Property**

Installed With: Base Package

**Class**: Application Properties (ActiveX)

Specifies whether to print visible subdiagrams of each Case, Event, and Stacked Sequence structure in sequence with the non-visible subdiagrams when using the custom format with one of the print documentation VI methods. If you print the visible frames in sequence, those frames will print twice.

object. Print Setup Custom Diagram Repeat

### **PrintSetupCustomExpressVIConfigInfo Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the configuration information for any LabVIEW Express VIs on the block diagram when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Express VI Config Info

# **PrintSetupCustomHierarchy Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the hierarchy of the VI in memory when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Hierarchy

# **PrintSetupCustomHistory Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the VI revision history information when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom History

# **PrintSetupCustomLabel Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the label and/or caption for each control when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Label

<u>PrintSetupCustomLabelEnum</u>

# **PrintSetupCustomPanel Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the LabVIEW front panel when using the custom format with one of the print documentation <u>VI methods</u>.

object.PrintSetupCustomPanel

Boolean

# **PrintSetupCustomPanelBorder Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print the LabVIEW front panel with a border when using the custom format with one of the print documentation <u>VI methods</u>.

object. Print Setup Custom Panel Border

Boolean

### **PrintSetupCustomSubVIs Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Specifies whether to print a list of the LabVIEW subVIs and Express VIs including the icon, name, and path when using the custom format with one of the print documentation <u>VI methods</u>. If a polymorphic VI is a subVI, LabVIEW prints the instance used in the top-level VI including the icon, name, and path. If the top-level VI is polymorphic, LabVIEW prints a list of all the instances of the polymorphic VI including the icon, name, and path of each instance.

object. Print Setup Custom Sub VIs

Boolean

# **PrintSetupFileWrapText Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Maximum number of characters on a single line in a file. Affects the <a href="PrintVIToHTML">PrintVIToRTF</a>, and <a href="PrintVIToText">PrintVIToText</a> methods. Set this property to 0 to print all the characters on one line so the text does not wrap.

object. Print Setup File Wrap Text

long

#### **PrintSetupJPEGQuality Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Percentage value from 0 to 100 specifying the level of quality you want for the JPEG graphic in VIs printed either interactively or with the <a href="PrintVIToHTML">PrintVIToHTML</a> method. The scale balances image quality and file size. A value in the 75-95 range produces a compressed file with a high-quality image, and a value below 50 produces a smaller file size with a low-quality image. The default is 80.

object.PrintSetupJPEGQuality

long

### **PrintSetupPNGCompressLevel Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Number between -1 and 9 that specifies the level of compression you want for the PNG file for graphics in VIs printed either interactively or with the <u>PrintVIToHTML</u> method. The quality of the graphic is not affected by the compression, but the graphic file size and speed of compression are affected by this value. Valid values range from -1 to 9 and balance file compression with speed. 0 designates no compression. 1 designates the best speed with some compression. 9 designates the best compression, but slower speed. -1 (the default value) designates the best combination of good compression and speed.

object.PrintSetupPNGCompressLevel

long

#### **RTHostConnected Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

If you are not targeted to a real-time (RT) platform, this property is always TRUE. If you are targeted to an RT-platform and the host is connected, this property is TRUE. When this property is FALSE, the host is not connected, requiring that the RT Module application must be able to handle all operations independent of the host. Use this property to determine when it is safe to display a dialog box that requires user interaction.

object.RTHostConnected

Boolean

#### **SaveVersions Property**

Installed With: Base Package

**Class**: Application Properties (ActiveX)

An array of the LabVIEW versions to which this version of LabVIEW can save. The array is in reverse chronological order, with the element at position zero in the array being the current version of LabVIEW. Each element of the array is a valid version to pass to VI server functions which save to previous versions.

object.SaveVersions

variant

#### **ShowFPTipStrips Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Writing the property: Setting this property to TRUE enables the display of tip strips on LabVIEW front panel controls. Setting the property to FALSE prevents them from being displayed. Reading the property: Specifies whether tip strips display when idling over controls on the front panel.

object.ShowFPTipStrips

Boolean

# **UserName Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

User name used to open the application. This property is read only.

object.UserName

string

# **Version Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Version number of the application. This property is read only.

object.Version

string

# **VIServerPort Property**

Installed With: Base Package

Class: Application Properties (ActiveX)

Gets or sets the LabVIEW VI server port. LabVIEW returns an error if you

attempt to access this property on a remote application.

object.VIServerPort

### **Data Type**

unsigned long

#### **ActiveX Enumerations**

The following enumerations are used in the various properties and methods.

**AppKindEnum** 

**AppTargCPUEnum** 

**AppTargOSEnum** 

**ExecStateEnum** 

**FPRunTimePosEnum** 

**FPStateEnum** 

**HTMLImageFormatEnum** 

**LibraryItemScopeEnum** 

**PageOrientationEnum** 

PrintFormatEnum

**PrintMarginsEnum** 

**PrintMethodsEnum** 

**VIExecSysEnum** 

**VIFPBehaviorEnum** 

**VILockStateEnum** 

**VIPriorityEnum** 

**VITypeEnum** 

# **AppKindEnum**

Indicates the type of application.

Name	Value	Description
eInvalidAppKind	0	Invalid application kind
eDevSysKind	1	LabVIEW Development System, including the Base Package, Full Development System, or Professional Development System
eRunTimeSysKind	2	Run-Time System
eStudEdKind	3	LabVIEW Student Edition
eEmbeddedKind	4	Embedded System, including LabVIEW Real- Time
eEvaluation	5	Evaluation version

## **AppTargCPUEnum**

Indicates the central processing unit (CPU) of the machine on which the application is running.

Name	Value	Description
eInvalidTargCPU	0	Invalid target CPU
eMotorola68kCPU	1	Motorola 68K
ePowerPCCPU	2	VxWorks PowerPC
eIntelx86CPU	3	Intel x86
eSPARCCPU	4	SPARC
ePARISCCPU	5	PA-RISC
eMIPSCPU	6	MIPS
eAlphaCPU	7	Alpha

## **AppTargOSEnum**

Indicates the type of operating system (OS) on which the application is running. Mac OS refers to Mac OS versions 8 and 9. Carbon refers to Mac OS X.

Name	Value	Description
eInvalidTargOS	0	Invalid Target OS
eMacOS	1	Mac OS
eWin31OS	2	Windows 3.1
eWin95NTOS	3	Windows 95/NT
eSolaris1OS	4	Solaris 1
eSolaris2OS	5	Solaris 2
eHPUXOS	6	HP-UX
ePowerMaxOS	7	PowerMax
eLinuxOS	8	Linux
elrixOS	9	Irix
eRhapsodyOS	10	Rhapsody
eBeOS	11	BeOS
eAIXOS	12	AIX
eOSF1OS	13	OSF1
eVxWorksOS	14	VxWorks
ePharlapOS	15	Phar Lap
eCarbonOS	16	Carbon
eRTXOS	17	RTX
eWin64OS	18	Windows x64
eLinux64OS	19	Linux x64

### **ExecStateEnum**

Indicates the execution state of a VI.

Name	Value	Description
eBad	0	VI has errors; it cannot run.
eldle	1	VI is not running, but the VI is in memory.
eRunTopLevel	2	VI is running as a top-level VI.
eRunning	3	VI is running as a subVI.

### **FPRunTimePosEnum**

Indicates the run-time LabVIEW front panel position.

Name	Value	Description
eRTPUnchanged	0	Unchanged
eRTPCentered	1	Centered
eRTPMaximized	2	Maximized
eRTPMinimized	3	Minimized
eRTPCustom	4	Custom

### **FPStateEnum**

Indicates how LabVIEW displays the front panel window.

Name	Value	Description
Invalid	0	If you read this property and an error occurs, the property returns this value. If you set the property to this value, the property returns an error.
Standard	1	The front panel window is open but is not minimized, maximized, or hidden.
Closed	2	The front panel window is not open. If you set the property to this value, the property returns an error.
Hidden	3	The front panel window is floating but is not visible because LabVIEW is not the active application. If you set the property to this value and close all references to the front panel, the window remains open but hidden. Because the VI is open, the <b>Getting Started</b> window does not appear. To solve this problem, open the VI from the operating system. For example, double-click the VI in Windows Explorer to open the VI.
Minimized	4	The front panel window is minimized.
Maximized	5	The front panel window is maximized.

## **HTMLImageFormatEnum**

Indicates the HTML image format to use for printing to an HTML file.

Name	Value	Description
ePNG	0	PNG format
eJPEG	1	JPEG format
eGIF	2	GIF format

# LibraryItemScopeEnum

Indicates the LibraryItem scope.

Name	Value	Description
eLibScopeInvalid	0	Invalid Scope
eLibScopePublic	1	Public
eLibScopePrivate	2	Private
eProtected	3	Protected

### **PrintFormatEnum**

Indicates the image format to use for printing.

Name	Value	Description
eCustom	0	Use the Application properties to specify the format.
eStandard	1	Prints the VI description, icon and connector pane, front panel, and block diagram.
eUsingPanel	2	Prints the VI description, front panel, and controls and indicators, including data types, names, and descriptions. LabVIEW prints the controls and indicators in tabbing order.
eUsingSubVI	3	Prints the VI description, icon and connector pane, and connected controls and indicators, including data types, names, and descriptions. LabVIEW prints the controls and indicators in tabbing order.
eComplete	4	Prints the VI description; icon and connector pane; front panel; controls and indicators, including data types, names, and descriptions; block diagram; a list of subVIs, including icons, names, and paths; revision history information; and the VI hierarchy.

# **PrintMarginsEnum**

Array of variants that include the following elements:

Element	Туре	Description
element 1	(float)	top
element 2	(float)	left
element 3	(float)	bottom
element 4	(float)	right
element 5	(byte)	measurement system (0=Inches, 1=cm)

## **PrintSetupCustomLabelEnum**

Indicates whether to print the label and/or caption for each control when using the custom format with one of the print documentation VI methods.

Name	Value	Description
Label	0	Prints only the label of the front panel control.
Caption	1	Prints only the caption of the front panel control.
Caption[Label]		Prints the caption and the label in square brackets of the front panel control.

## **VIExecSysEnum**

Indicates the execution system in which a VI runs.

Name	Value	Description
eESysInvalid	0	Invalid execution system
eESysUserInterface	1	User interface
eESysNormal	2	Standard
eESysInstrIO	3	Instrument I/O
eESysDAQ	4	Data acquisition (DAQ)
eESysOther1	5	Other 1
eESysOther2	6	Other 2
eESysSameAsCaller	7	Same as caller

### **VILockStateEnum**

Indicates the lock state of a VI.

Name	Value	Description
eInvalidLockState	0	Invalid VI lock state
eUnlockedState	1	Unlocked
eLockedNoPwdState	2	Locked, no password
ePwdProtectedState	3	Password-protected

# **VIPriorityEnum**

Indicates the execution priority of a VI.

Name	Value	Description
ePriInvalid	0	Invalid priority
ePriBackground	1	Background
ePriNormal	2	Normal
ePriAboveNormal	3	Above normal
ePriHigh	4	High
ePriCritical	5	Critical
ePriSubroutine	6	Subroutine

## **VITypeEnum**

Indicates the type of a VI.

Name	Value	Description
eInvalidVIType	0	Invalid VI type
eStandardVIType	1	Standard VI
eControlVIType	2	Control VI
eGlobalVIType	3	Global VI
ePolymorphicVIType	4	Polymorphic VI
eConfigurationVIType	5	Configuration VI
eSubSystemVIType	6	SubSystem
eFacadeVIType	7	Facade VI
eMethodVIType	8	Method VI

## **Project Properties and Methods**

This book contains the property and method reference information for the Project class.

# **Project Methods (ActiveX)**

Installed With: Base Package

Method	Description
Close	Closes a LabVIEW project and all references to all items in the project. This method also closes the LabVIEW <b>Project Explorer</b> window if it is open and closes any VIs that are open in any application instance owned by the project. In LabVIEW, you also can select <b>File»Close All</b> to close the project and the <b>Project Explorer</b> window. Details
CloseWindow	Closes only the LabVIEW <b>Project Explorer</b> window. This method does not close VI windows. Details
DeployItems	Deploys an array of items in the LabVIEW project.  Details
OpenWindow	Displays the LabVIEW <b>Project Explorer</b> window.  Details
Save	Saves the LabVIEW project to the path that the project was previously saved to or loaded from or to a specified path. Details
SaveForPrevious	Saves a copy of the LabVIEW project that is readable by LabVIEW version 8.0 and later. <u>Details</u>

#### **Close Method**

Installed With: Base Package

Class: Project Methods (ActiveX)

Closes a LabVIEW project and all references to all items in the project. This method also closes the LabVIEW **Project Explorer** window if it is open and closes any VIs that are open in any application instance owned by the project. In LabVIEW, you also can select **File»Close All** to close the project and the **Project Explorer** window.

### **Syntax**

object.Close()

### **Return Value**

none

### **CloseWindow Method**

Installed With: Base Package

Class: Project Methods (ActiveX)

Closes only the LabVIEW **Project Explorer** window. This method does

not close VI windows.

In LabVIEW, you also can select **File»Exit** to close the **Project Explorer** window and any VIs that are open.

### **Syntax**

object.CloseWindow()

### **Return Value**

none

## **DeployItems Method**

Installed With: Base Package

Class: Project Methods (ActiveX)

Deploys an array of items in the LabVIEW project.

### **Syntax**

object.DeployItems(items, silent)

#### **Parameters**

Name	Туре	Description
items	ProjectItem*	Specifies an array of references to items in the project.
silent		If TRUE, LabVIEW does not display any dialog boxes during the deploy operation.

### **Return Value**

none

## **OpenWindow Method**

Installed With: Base Package

Class: Project Methods (ActiveX)

Displays the LabVIEW **Project Explorer** window.

### **Syntax**

object.OpenWindow()

### **Return Value**

none

#### **Save Method**

Installed With: Base Package

Class: Project Methods (ActiveX)

Saves the LabVIEW project to the path that the project was previously saved to or loaded from or to a specified path.

If the project has not been saved and the value of the **path** parameter is <Not A Path> or **path** is not specified, this method returns an error.

### **Syntax**

object.Save(path)

#### **Parameters**

Name	Type	Description
path	String	Path to the project.

### **Return Value**

none

#### **SaveForPrevious Method**

Installed With: Base Package

Class: Project Methods (ActiveX)

Saves a copy of the LabVIEW project that is readable by LabVIEW

version 8.0 and later.

object.SaveForPrevious(Path)

#### **Parameters**

Name	Туре	Description
Path	Path	Path to where you want to save a copy of the
	control	project.

#### **Return Value**

none

# **Project Properties (ActiveX)**

Installed With: Base Package

In the following topics, an asterisk (\*) at the end of a data type name indicates that the property is a pointer.

Property	Description
ActiveItemsInTree	If the LabVIEW <b>Project Explorer</b> window has focus, returns an array of references to the selected project items in the <b>Project Explorer</b> window. If the <b>Project Explorer</b> window does not have focus, returns a reference to the project item associated with the VI window that has focus. You can use these references with the <u>ProjectItem</u> properties. <u>Details</u>
Application	Returns the application reference for the LabVIEW project. You can use this reference with the <a href="Application">Application</a> properties. <a href="Details">Details</a>
Description	Gets or sets the description of the LabVIEW project. Details
MyComputer	Returns a reference to <b>My Computer</b> in the <b>Project Explorer</b> window for the LabVIEW project. You can use this reference with the <u>TargetItem</u> properties. <u>Details</u>
Name	Gets the name of the LabVIEW project. If the project has been saved, this property returns the filename with the file extension. If the project has not been saved, this property returns untitled project x. Details
Path	Gets the path to the saved LabVIEW project on disk. This path includes the filename of the project. Details
Root	Returns a reference to the project root in the LabVIEW <b>Project Explorer</b> window. The project root is the top item that represents the LabVIEW project in the <b>Project Explorer</b> window. You can use this reference with the <b>ProjectItem</b> properties.

	<u>Details</u>
SelectedItemsInTree	Returns an array of references to the project items currently selected in the LabVIEW <b>Project Explorer</b> window. You can use these references with the <b>ProjectItem</b> properties. <b>Details</b>
Targets	Returns an array of references to all targets in the current LabVIEW project. You can use these references with the <u>TargetItem</u> properties. <u>Details</u>
WindowState	Returns the state of the LabVIEW <b>Project Explorer</b> window. Details

#### **ActiveItemsInTree Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

If the LabVIEW **Project Explorer** window has focus, returns an array of references to the selected project items in the **Project Explorer** window. If the **Project Explorer** window does not have focus, returns a reference to the project item associated with the VI window that has focus. You can use these references with the <u>ProjectItem</u> properties.

object.ActiveItemsInTree

1D array of ProjectItems

## **Application Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Returns the application reference for the LabVIEW project. You can use this reference with the <u>Application</u> properties.

If the project has multiple targets, each target is associated with a single application instance.

object.Application

Application\*

# **Description Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Gets or sets the description of the LabVIEW project.

object.Description

String

# **MyComputer Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Returns a reference to **My Computer** in the **Project Explorer** window for the LabVIEW project. You can use this reference with the <u>TargetItem</u> properties.

object.MyComputer

TargetItem\*

#### **Name Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Gets the name of the LabVIEW project. If the project has been saved, this property returns the filename with the file extension. If the project has not been saved, this property returns to 0 untitled project to 0.

object.Name

string

# **Path Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Gets the path to the saved LabVIEW project on disk. This path includes

the filename of the project.

object.Path

string

#### **Root Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Returns a reference to the project root in the LabVIEW **Project Explorer** window. The project root is the top item that represents the LabVIEW project in the **Project Explorer** window. You can use this reference with the <u>ProjectItem</u> properties.

object.Root

ProjectItem\*

#### **SelectedItemsInTree Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Returns an array of references to the project items currently selected in the LabVIEW **Project Explorer** window. You can use these references with the <u>ProjectItem</u> properties.

object.SelectedItemsInTree

1D array of ProjectItems

## **Targets Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Returns an array of references to all targets in the current LabVIEW project. You can use these references with the <u>TargetItem</u> properties.

object.Targets

1D array of TargetItems

# **WindowState Property**

Installed With: Base Package

Class: Project Properties (ActiveX)

Returns the state of the LabVIEW **Project Explorer** window.

#### **Values**

- 0 **eInvalidFPState**—If you read this property and an error occurs, the property returns this value.
- 1 **eVisible**—The **Project Explorer** window is open but is not minimized, maximized, or hidden.
- 2 eClosed—The Project Explorer window is not open.
- 3 **eHidden**—The **Project Explorer** window is floating but is not visible because LabVIEW is not the active application.
- 4 eMinimized—The Project Explorer window is minimized.
- 5 **eMaximized**—The **Project Explorer** window is maximized.

object.WindowState

## **Data Type**

**FPStateEnum** 

# **ProjectItem Properties and Methods**

This book contains the property and method reference information for the ProjectItem class.

# **ProjectItem Methods (ActiveX)**

Installed With: Base Package

Method	Description
AddFile	Adds a file at the specified path to the LabVIEW project with the referenced project item as the parent. Details
AddItem	Adds a new project item of the specified type to the LabVIEW project with the referenced item as the parent. Details
AddItemFromMemory	Adds an item that exists in memory to the LabVIEW project with the referenced project item as the parent. Details
Delete	Removes the referenced project item from the LabVIEW project. If the item has children, the children also are deleted. Details
DeleteTag	Removes the named tag from the LabVIEW project item. Details
GetAllDescendents	Returns LabVIEW project items that are descendents of the referenced item. If you specify a value for <b>type</b> , this method only returns project items of the specified type. Details
GetTag	Returns a tag value associated with the LabVIEW project item. Details
GetTagNames	Returns an array of tag names of all tags on the referenced LabVIEW project item. <u>Details</u>
GetXMLTag	Returns the value in an XML tag. <u>Details</u>
SetTag	Sets a tag value associated with the LabVIEW project. Details
SetXMLTag	Set the value of an XML tag associated with the LabVIEW project. Details

### **AddFile Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Adds a file at the specified path to the LabVIEW project with the

referenced project item as the parent.

object.AddFile(path)

#### **Parameters**

Name	Туре	Description	
path	String	Path to the file you want to add.	

ProjectItem\*

#### **AddItem Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Adds a new project item of the specified type to the LabVIEW project with the referenced item as the parent.

This method returns an error if the specified type cannot be created under the current item. The method also returns an error when you try to add a shared variable to a library that is not opened in a project.

object.AddItem(name, path, type)

#### **Parameters**

Name	Туре	Description			
name	String	Name of the project item you want to add.			
path	String	Path to the project item you want to add.			
type	String	Specifies the project item type. The following values are the most common <b>type</b> values:			
		• VI			
		• Folder			
		• Library			
		• XControl			
		• EXE			
		• DLL			
		Source Distribution			
		• Installer			
		• Zip File			
		Hyperlink			
		Some of the values listed above are supported only on certain LabVIEW Development Systems. Refer to <a href="mailto:ni.com/labview">ni.com/labview</a> for more information about LabVIEW Development Systems.			
		To determine the value of <b>type</b> for a project item that is not listed above, use the <u>TypeString</u> property to programmatically return the type of the item. For crosslanguage reasons, this string is always in English.			

ProjectItem\*

# **AddItemFromMemory Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Adds an item that exists in memory to the LabVIEW project with the

referenced project item as the parent.

object.AddItemFromMemory(name)

#### **Parameters**

Name	Type	Description
name	•	Name that exists in memory, such as foo.vi, of the project item you want to add.

ProjectItem\*

### **Delete Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Removes the referenced project item from the LabVIEW project. If the

item has children, the children also are deleted.

object.Delete()

none

# **DeleteTag Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Removes the named tag from the LabVIEW project item.

object.DeleteTag(tagName)

#### **Parameters**

Name	Type	Description
tagName	String	Name of the tag. Use the <u>GetTagNames</u> method to
		retrieve the tag names.

Boolean

#### **GetAllDescendents Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Returns LabVIEW project items that are descendents of the referenced item. If you specify a value for **type**, this method only returns project items of the specified type.

This method returns a flat list that includes all descendents unless you use the **type**, **excludeDependencies**, **excludeBuilds**, and **excludeTargets** parameters to limit what the method returns.

object.GetAllDescendents(type, [excludeDependencies], [excludeBuilds], [excludeTargets])

#### **Parameters**

Name	Туре	Description
type	String	Specifies the project item type. The following values are the most common type values:  • VI • Folder • Library • XControl • EXE • DLL • Source Distribution • Installer • Zip File • Hyperlink Some of the values listed above are supported only on certain LabVIEW Development Systems. Refer to ni.com/labview for more information about LabVIEW Development Systems.  To determine the value of type for a project item that is not listed above, use the TypeString property to programmatically return the type of the item. For cross-language reasons, this string is always in English.
excludeDependencies	Boolean	If TRUE, does not return  Dependencies or any items under  Dependencies. The default is FALSE.
excludeBuilds	Boolean	If TRUE, does not return <b>Build Specifications</b> or any items under <b>Build Specifications</b> . The default is FALSE.

excludeTargets	If TRUE, does not return any targets or any items under targets. The default
	is FALSE.

1D array of ProjectItems

# **GetTag Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Returns a tag value associated with the LabVIEW project item.

object.GetTag(tagName, value, [isPersistent])

#### **Parameters**

Name	Туре	Description
tagName	String	Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
value	variant	Specifies the tag value as a variant.
isPersistent		Returns TRUE if LabVIEW saves the tag information when you save the LabVIEW project. Returns FALSE otherwise.

Boolean

# **GetTagNames Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Returns an array of tag names of all tags on the referenced LabVIEW

project item.

object.GetTagNames()

Array of strings

# **GetXMLTag Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Returns the value in an XML tag.

Use the **SetXMLTag** method to set the value in an XML tag.

object.GetXMLTag(tagName, [value], [isPersistent])

#### **Parameters**

Name	Туре	Description
tagName	_	Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
value	String*	Returns the value of the XML tag as a string.
isPersistent		Returns TRUE if LabVIEW saves the tag information when you save the LabVIEW project. Returns FALSE otherwise.

#### **Return Value**

Boolean

# **SetTag Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Sets a tag value associated with the LabVIEW project.

object.SetTag(tagName, value, [isPersistent])

#### **Parameters**

Name	Туре	Description
tagName	String	Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
value	variant	Specifies the tag value as a variant.
isPersistent	Boolean	If TRUE, LabVIEW saves the tag information when you save the LabVIEW project. The default is FALSE.

#### **Return Value**

none

# **SetXMLTag Method**

Installed With: Base Package

Class: ProjectItem Methods (ActiveX)

Set the value of an XML tag associated with the LabVIEW project.

object.SetXMLTag(tagName, value, [isPersistent])

#### **Parameters**

Name	Туре	Description
tagName	String	Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
value	variant	Specifies the tag value as a variant.
isPersistent	Boolean	If TRUE, LabVIEW saves the tag information when you save the LabVIEW project. The default is FALSE.

#### **Return Value**

none

# **ProjectItem Properties (ActiveX)**

Installed With: Base Package

In the following topics, an asterisk (\*) at the end of a data type name indicates that the property is a pointer.

Property	Description
Children	Returns an array of references to all children of the referenced LabVIEW project item. You can use these references with the <a href="ProjectItem">ProjectItem</a> properties. <a href="Details">Details</a>
ContainsConflicts	Returns TRUE if this item or items beneath this item in the project tree are in conflict. <u>Details</u>
DisplayName	Gets, as a string, the name of the LabVIEW project item as the name appears in the LabVIEW <b>Project Explorer</b> window. <u>Details</u>
FindCallers	Returns an array of references to all project items that reference the project item you select. Details
FindConflicts	Returns an array of references to all project items that are in conflict with the project item you select. Details
FindItemsWithNoCallers	Returns an array of references to all top-level project items that call no other project items.  Details
FindSubVIs	Returns an array of references to all project items that are subVIs of the item you select.  Details
Icon	Gets the icon of the project item in the LabVIEW <b>Project Explorer</b> window using the <b>Icon</b> cluster, which limits size and color depth. Details
InConflict	Returns TRUE if the referenced LabVIEW project item conflicts with another item with a different path under the same target. Details
ItemID	Gets the item ID of the referenced LabVIEW

Gets the type of the LabVIEW project item as it exists in the project library as a GUID string. Use the <a href="LibraryItemTypeString">LibraryItemTypeString</a> property to return the type as a human-readable string.
<u>Details</u>
Returns the type of the LabVIEW project item as it exists in the project library as a string. Use the <u>LibraryItemType</u> property to return the type as a GUID. <u>Details</u>
Gets the name of the referenced LabVIEW project item as it is saved in the file. Details
Returns a reference to the parent of the referenced LabVIEW project item. You can use this reference with the <a href="ProjectItem">ProjectItem</a> properties. <a href="Details">Details</a>
Gets the path of the referenced LabVIEW project item. <u>Details</u>
Returns a reference to the LabVIEW project to which the referenced project item belongs. You can use this reference with the <a href="Project">Project</a> properties. <a href="Details">Details</a>
Returns a reference to the target that contains the referenced LabVIEW project item. You can use this reference with the <u>TargetItem</u> properties. <u>Details</u>
Gets the type of the LabVIEW project item as a GUID string. Use the TypeString property to return the type as a string. Details
Gets the type of the LabVIEW project item as a string. Use the <a href="TypeGUID">TypeGUID</a> property to return the type as a GUID. <a href="Details">Details</a>

VI	If the LabVIEW project item refers to a VI, this property returns a reference to the VI. Otherwise, this property returns NULL. <u>Details</u>
VINoLoad	If the project item refers to a VI in memory, this property returns a reference to the VI. If the project item refers to a VI not in memory or to something that is not a VI, this property returns Not a Refnum. You can use this reference with the VI properties. Details
VisibleInTree	If TRUE, indicates that the LabVIEW <b>Project Explorer</b> window is visible and ancestors of the LabVIEW project item are expanded in the <b>Project Explorer</b> window up to the root of the project item. <b>Details</b>

## **Children Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns an array of references to all children of the referenced LabVIEW project item. You can use these references with the <a href="ProjectItem">ProjectItem</a> properties.

object.Children

1D array of ProjectItems

## **ContainsConflicts Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns TRUE if this item or items beneath this item in the project tree

are in conflict.

object.ContainsConflicts

Boolean

## **DisplayName Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets, as a string, the name of the LabVIEW project item as the name

appears in the LabVIEW **Project Explorer** window.

object.DisplayName

string

## **FindCallers Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns an array of references to all project items that reference the

project item you select.

object.FindCallers

## **FindConflicts Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns an array of references to all project items that are in conflict with

the project item you select.

object.FindConflicts

1D array of ProjectItems

## **FindItemsWithNoCallers Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns an array of references to all top-level project items that call no

other project items.

object. Find Items With No Callers

1D array of ProjectItems

## **FindSubVIs Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns an array of references to all project items that are subVIs of the

item you select.

object.FindSubVIs

1D array of ProjectItems

### **Icon Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the icon of the project item in the LabVIEW **Project Explorer** window using the **Icon** cluster, which limits size and color depth.

object.Icon

ImageData

# **InConflict Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns TRUE if the referenced LabVIEW project item conflicts with

another item with a different path under the same target.

object.InConflict

Boolean

#### **ItemID Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the item ID of the referenced LabVIEW project item as a string. You can use this item ID to uniquely identify the referenced item even if the path, name, or location in the LabVIEW **Project Explorer** window changes.

object.ItemID

string

# **LibraryItemType Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the type of the LabVIEW project item as it exists in the project library as a GUID string. Use the <u>LibraryItemTypeString</u> property to return the type as a human-readable string.

If the item is not part of a library, this property returns an error.

object.LibraryItemType

string

# **LibraryItemTypeString Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns the type of the LabVIEW project item as it exists in the project library as a string. Use the <u>LibraryItemType</u> property to return the type as a GUID.

This property is valid only if the LabVIEW project item exists in a project library.

object.LibraryItemTypeString

string

# **Name Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the name of the referenced LabVIEW project item as it is saved in the file.

Use the <u>DisplayName</u> property to get the name as it appears in the LabVIEW **Project Explorer** window.

object.Name

string

# **Parent Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns a reference to the parent of the referenced LabVIEW project

item. You can use this reference with the **ProjectItem** properties.

object.Parent

ProjectItem\*

# **Path Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the path of the referenced LabVIEW project item.

object.Path

string

# **Project Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns a reference to the LabVIEW project to which the referenced project item belongs. You can use this reference with the <a href="Project">Project</a>

properties.

object.Project

Project\*

# **Target Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Returns a reference to the target that contains the referenced LabVIEW project item. You can use this reference with the <u>TargetItem</u> properties.

object.Target

TargetItem\*

# **TypeGUID Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the type of the LabVIEW project item as a GUID string. Use the

TypeString property to return the type as a string.

object.TypeGUID

string

# **TypeString Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

Gets the type of the LabVIEW project item as a string. Use the <a href="TypeGUID"><u>TypeGUID</u></a>

property to return the type as a GUID.

object.TypeString

string

#### **VI Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

If the LabVIEW project item refers to a VI, this property returns a reference to the VI. Otherwise, this property returns NULL.



**Note** This property does not record changes to the VI. If you use this reference for editing the VI you will not be prompted to save when the reference closes.

object.VI

## **Data Type**

VirtualInstrument \*

### **VINoLoad Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

If the project item refers to a VI in memory, this property returns a reference to the VI. If the project item refers to a VI not in memory or to something that is not a VI, this property returns Not a Refnum. You can use this reference with the  $\underline{VI}$  properties.

object.VINoLoad

## **Data Type**

VIObjRef

## **VisibleInTree Property**

Installed With: Base Package

Class: ProjectItem Properties (ActiveX)

If TRUE, indicates that the LabVIEW **Project Explorer** window is visible and ancestors of the LabVIEW project item are expanded in the **Project Explorer** window up to the root of the project item.

object.VisibleInTree

## **Data Type**

Boolean

# **Library Properties and Methods**

This book contains the property and method reference information for the Library class.

# **Library Methods (ActiveX)**

Installed With: Base Package

Inherits from **ProjectItem**.

Use these methods to configure LabVIEW project libraries.

Method	Description
DeleteLibTag	Deletes a tag from the LabVIEW project library. <u>Details</u>
DisconnectFromLibrary	Disconnects a LabVIEW project library that is a sublibrary of another project library from the owning project library. <a href="Details">Details</a>
GetLibTag	Retrieves a tag set on the LabVIEW project library. Details
GetLockState	Returns the lock state of the LabVIEW project library and indicates whether the password for the project library is in the password cache.  Details
GetSourceScope	Retrieves the access scope for an item that the LabVIEW project library owns, such as public or private access. Details
GetTagNames	Retrieves a list of names for all tags set on the LabVIEW project library. Details
HiliteInProjectWindow	Displays this library in a <b>Project Explorer</b> window. If the library is already in a <b>Project Explorer</b> window, this method displays the <b>Project Explorer</b> window and sets focus to the library item in the window. If the library is already in a stand-alone project library window for which the library is the root item, this method displays the stand-alone project library window. If the library is not in any window, this method opens a stand-alone project library window containing the library. <b>Details</b>
Save	Saves the LabVIEW project library file. You also can right-click the project library in the LabVIEW

	<b>Project Explorer</b> window and select <b>Save As</b> from the shortcut menu to save the project library file. Details
SaveCopy	Saves a copy of the LabVIEW project library file.  Details
SaveForPrevious	Saves a copy of the LabVIEW project library that is readable by LabVIEW version 8.0 and later. Details
SetLibTag	Sets a tag on the LabVIEW project library.  Details
SetLockState	Sets the level of editing permission for a LabVIEW project library, including the password. Details
SetSourceScope	Sets the access scope of an item that the LabVIEW project library owns, such as public or private access. <u>Details</u>

# **DeleteLibTag Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Deletes a tag from the LabVIEW project library.

object.DeleteLibTag(tagName, [itemRef])

#### **Parameters**

Name	Туре	Description
tagName	_	Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
itemRef	ProjectItem*	Reference to the Project Library Item.

Boolean

# **DisconnectFromLibrary Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Disconnects a LabVIEW project library that is a sublibrary of another

project library from the owning project library.

object.DisconnectFromLibrary()

none

# **GetLibTag Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Retrieves a tag set on the LabVIEW project library.

object.GetLibTag(tagName, [value], [isPersistent], [itemRef])

#### **Parameters**

Name	Туре	Description
tagName	_	Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
value	variant	Returns the value of the tag.
isPersistent		Returns TRUE if LabVIEW saves the tag information when you save the LabVIEW project. Returns FALSE otherwise.
itemRef	ProjectItem*	Reference to the Project Library Item.

Boolean

#### **GetLockState Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Returns the lock state of the LabVIEW project library and indicates whether the password for the project library is in the password cache.

This method is similar to the **Protection** box on the **General Settings** page of the **Project Library Properties** dialog box.

object.GetLockState([pwdInCache])

#### **Parameters**

Name	Туре	Description
•		Indicates whether the password is currently in the password cache.

<u>VILockStateEnum</u>

# **GetSourceScope Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Retrieves the access scope for an item that the LabVIEW project library

owns, such as public or private access.

object.GetSourceScope(sourceRef, [inherited])

#### **Parameters**

Name	Туре	Description
sourceRef	,	Reference to the item in the project library for which you want to retrieve scope.
inherited		Indicates whether the specified item is inherited.

<u>LibraryItemScopeEnum</u>

# **GetTagNames Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Retrieves a list of names for all tags set on the LabVIEW project library.

object.GetTagNames([itemRef])

#### **Parameters**

Name	Туре	Description
itemRef	ProjectItem*	Reference to the Project Library Item.

Array of strings

### HiliteInProjectWindow Method

Installed With: Base Package

Class: Library Methods (ActiveX)

Displays this library in a **Project Explorer** window. If the library is already in a **Project Explorer** window, this method displays the **Project Explorer** window and sets focus to the library item in the window. If the library is already in a stand-alone project library window for which the library is the root item, this method displays the stand-alone project library window. If the library is not in any window, this method opens a stand-alone project library window containing the library.

object.HiliteInProjectWindow()

none

#### **Save Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Saves the LabVIEW project library file. You also can right-click the project library in the LabVIEW **Project Explorer** window and select **Save As** from the shortcut menu to save the project library file.

object.Save([path])

#### **Parameters**

Name	Туре	Description
path	String	Path to the LabVIEW project library file.

### **Return Value**

none

# **SaveCopy Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Saves a copy of the LabVIEW project library file.

object.SaveCopy(destDir, [newName], [newLibPath])

#### **Parameters**

Name	Туре	Description
destDir	_	Specifies the destination directory for the copy of the project library.
newName	String	Specifies the name of the copy of the project library.
newLibPath	String	Returns the path to the copy of the project library.

### **Return Value**

none

### **SaveForPrevious Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Saves a copy of the LabVIEW project library that is readable by LabVIEW

version 8.0 and later.

object.SaveForPrevious(Path)

#### **Parameters**

Name	Туре	Description
Path	Path	Path to the directory where you want to save the
	control	LabVIEW project library.

### **Return Value**

none

# **SetLibTag Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Sets a tag on the LabVIEW project library.

object.SetLibTag(tagName, [value], [isPersistent], [itemRef])

#### **Parameters**

Name	Туре	Description
tagName		Name of the tag. Use the <u>GetTagNames</u> method to retrieve the tag names.
value	variant	Specifies the value of the tag.
isPersistent		Returns TRUE if LabVIEW saves the tag information when you save the LabVIEW project. Returns FALSE otherwise.
itemRef	ProjectItem*	Reference to the Project Library Item.

### **Return Value**

none

#### SetLockState Method

Installed With: Base Package

Class: Library Methods (ActiveX)

Sets the level of editing permission for a LabVIEW project library,

including the password.

This method is similar to the **Protection** box on the **General Settings** 

page of the LabVIEW **Project Library Properties** dialog box.

object.SetLockState(lockState, [Interactive], [Password], [putInCache])

#### **Parameters**

Name	Туре	Description
lockState	<u>VILockStateEnum</u>	Lock state.
Interactive	Boolean	Specifies whether to display a dialog box that prompts you to change the lock state. The default is FALSE.
Password	String	If <b>lockState</b> is Password-protected, <b>Password</b> is the new password. If you are changing <b>lockState</b> from Password-protected, <b>Password</b> is the old password. The default is an empty string.
putInCache	Boolean	If <b>lockState</b> is Password-protected, <b>putinCache</b> specifies whether to place the new password in the cache. The default is FALSE.

### **Return Value**

none

## **SetSourceScope Method**

Installed With: Base Package

Class: Library Methods (ActiveX)

Sets the access scope of an item that the LabVIEW project library owns, such as public or private access.

This method is similar to the **Access Scope** box on the **Item Settings** page of the LabVIEW **Project Library Properties** dialog box.

object.SetSourceScope(sourceRef, [scope])

#### **Parameters**

Name Type Descr	ription
	ence to the item in the project for which you want to set
scope LibraryItemScopeEnum Specification.	

### **Return Value**

none

# **Library Properties (ActiveX)**

Installed With: Base Package

Inherits from **ProjectItem**.

Use these properties to configure LabVIEW project libraries.

Property	Description
AlarmsEventsDBComputer	Specifies the name of the computer where the alarms and events database resides. <u>Details</u>
AlarmsEventsDBName	Specifies the name of the database where the LabVIEW Shared Variable Engine logs alarms and events.  Details
AlarmsEventsDBPath	Specifies the file path to the database where the LabVIEW Shared Variable Engine logs alarms and events.  Details
AlarmsEventsEnable	Enables alarms and events logging capability if set to TRUE. <u>Details</u>
AlarmsEventsUseDataLoggingDB	Sets the alarms and events database to be the same as the database that logs data. If set to TRUE, LabVIEW ignores properties set on the AlarmsEvents database. Details
DataLoggingDBComputer	Specifies the name of the computer where the database resides. Details
DataLoggingDBName	Specifies the name of the database where the LabVIEW Shared Variable Engine logs data. Details
DataLoggingDBPath	Specifies the file path to the database where the LabVIEW Shared Variable Engine logs data. Details
DataLoggingEnable	Enables data logging capability if set to TRUE. <u>Details</u>

DataLoggingLifespan	Sets the lifespan (in days) of the logged data in the database. If value is 0, then data is always available, otherwise the database can destroy data after the number of specified days. Details
Description	Specifies the description of the LabVIEW project library for the <b>Context Help</b> window. <u>Details</u>
HelpDocumentPath	Path or symbolic path to an HTML file (.htm or .html) or compiled help file (.chm or .hlp) to which the LabVIEW project library is linked. Details
HelpDocumentTag	Index keyword or HTML filename for a topic in the compiled help file to which the LabVIEW project library is linked. To link to a bookmark within an HTML file, add # followed by the name of the bookmark to the end of the filename. Details
LocalName	The localized name of the item.  Details
Version	The version number of the LabVIEW project library. Details

# **AlarmsEventsDBComputer Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Specifies the name of the computer where the alarms and events

database resides.

object. A larms Events DB Computer

# **Data Type**

string

# **AlarmsEventsDBName Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Specifies the name of the database where the LabVIEW Shared Variable

Engine logs alarms and events.

object.AlarmsEventsDBName

# **Data Type**

string

# **AlarmsEventsDBPath Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Specifies the file path to the database where the LabVIEW Shared

Variable Engine logs alarms and events.

object.AlarmsEventsDBPath

# **Data Type**

string

# **AlarmsEventsEnable Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Enables alarms and events logging capability if set to TRUE.

object.AlarmsEventsEnable

Boolean

# **AlarmsEventsUseDataLoggingDB Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Sets the alarms and events database to be the same as the database that logs data. If set to TRUE, LabVIEW ignores properties set on the AlarmsEvents database.

object. A larms Events Use Data Logging DB

Boolean

# **DataLoggingDBComputer Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Specifies the name of the computer where the database resides.

object.DataLoggingDBComputer

string

# **DataLoggingDBName Property**

Installed With: Base Package

Class: <u>Library Properties (ActiveX)</u>

Specifies the name of the database where the LabVIEW Shared Variable

Engine logs data.

object.DataLoggingDBName

string

# **DataLoggingDBPath Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Specifies the file path to the database where the LabVIEW Shared

Variable Engine logs data.

object.DataLoggingDBPath

string

# **DataLoggingEnable Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Enables data logging capability if set to TRUE.

object.DataLoggingEnable

Boolean

#### **DataLoggingLifespan Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Sets the lifespan (in days) of the logged data in the database. If value is 0, then data is always available, otherwise the database can destroy data after the number of specified days.

The database only discards data if new data is logged. This property affects all data from the LabVIEW project library stored in the database, not just the most recent data. For example, if you set this property to four days and you have three days of data, the database does not destroy any data. However, if you set this same property to two days from four days and you have three days of data, the database can destroy any data collected more than two days ago.

object.DataLoggingLifespan

Long

### **Description Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Specifies the description of the LabVIEW project library for the **Context** 

**Help** window.

If you want to display a carriage return in the LabVIEW **Context Help** window, you must separate paragraphs with two carriage returns.

object.Description

string

#### **HelpDocumentPath Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

Path or symbolic path to an HTML file (.htm or .html) or compiled help file (.chm or .hlp) to which the LabVIEW project library is linked.

If the path is to a compiled help file, use the <u>HelpDocumentTag</u> property to determine the specific topic in that help file.

This property is similar to the **Help path** text box on the **Documentation** page of the LabVIEW **Project Library Properties** dialog box.

object.HelpDocumentPath

string

#### **HelpDocumentTag Property**

Installed With: Base Package

Class: <u>Library Properties (ActiveX)</u>

Index keyword or HTML filename for a topic in the compiled help file to which the LabVIEW project library is linked. To link to a bookmark within an HTML file, add # followed by the name of the bookmark to the end of the filename.

Use this property only when <u>HelpDocumentPath</u> is a path to a compiled help file (.chm or .hlp).

For .chm files, this property can be an HTML filename or index keyword. For .hlp files, this property can be an index keyword.

This property is similar to the **Help tag** text box on the **Documentation** page of the LabVIEW **Project Library Properties** dialog box.

object.HelpDocumentTag

string

# **LocalName Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

The localized name of the item.

This property is similar to the **Localized Name** text box on the **Documentation** page of the LabVIEW **Project Library Properties** dialog box.

object.LocalName

string

# **Version Property**

Installed With: Base Package

Class: Library Properties (ActiveX)

The version number of the LabVIEW project library.

This property is similar to the **Version Number** component on the **General Settings** page of the LabVIEW **Project Library Properties** dialog box.

object.Version

LVLibraryVersion

# **XInterfaceLibrary Properties and Methods**

This book contains the property and method reference information for the XInterfaceLibrary class.

# XInterfaceLibrary Methods (ActiveX)

Installed With: Base Package

Inherits from **ProjectItem**»Library.

Method	Description
AddAbility	Creates a new ability of the specified type. <u>Details</u>
AddMethod	Creates a new method. <u>Details</u>
NewProperty	Creates a new property folder and returns a reference to it. Details

# **AddAbility Method**

Installed With: Base Package

Class: XInterfaceLibrary Methods (ActiveX)

Creates a new ability of the specified type.

object.AddAbility(name, [viPath], [folder])

#### **Parameters**

Name	Туре	Description
name	String	Name for the ability you want to create.
viPath		If you specify <b>viPath</b> , LabVIEW uses an existing VI for the ability. If you do not specify <b>viPath</b> , LabVIEW creates a new VI for the ability.
folder	ProjectItem*	Creates the ability in the specified folder.

### **Return Value**

ProjectItem\*

### **AddMethod Method**

Installed With: Base Package

Class: XInterfaceLibrary Methods (ActiveX)

Creates a new method.

object.AddMethod([viPath], [folder])

#### **Parameters**

Name	Туре	Description
viPath		If you specify <b>viPath</b> , LabVIEW uses an existing VI for the method. If you do not specify <b>viPath</b> , LabVIEW creates a new VI for the method.
folder	ProjectItem*	Creates the method in the specified folder.

### **Return Value**

ProjectItem\*

## **NewProperty Method**

Installed With: Base Package

Class: XInterfaceLibrary Methods (ActiveX)

Creates a new property folder and returns a reference to it.

object.NewProperty(name, folder)

#### **Parameters**

Name	Туре	Description
name	String	Name of the new property folder to create.
folder	ProjectItem*	Folder in which to create the new property folder.

### **Return Value**

PropertyFolder \*

## **XInterfaceLibrary Properties (ActiveX)**

Installed With: Base Package

Inherits from **ProjectItem**»Library.

No properties are specific to this class.

## **PropertyFolder Properties and Methods**

This book contains the property and method reference information for the PropertyFolder class.

# **PropertyFolder Methods (ActiveX)**

Installed With: Base Package

Inherits from **ProjectItem**.

Method	Description
	Creates a new read or write property VI if you do not set <b>viPath</b> . If you set <b>viPath</b> , LabVIEW uses an existing VI
	as a property VI. <u>Details</u>

## **AddPropertyVI Method**

Installed With: Base Package

Class: PropertyFolder Methods (ActiveX)

Creates a new read or write property VI if you do not set **viPath**. If you

set viPath, LabVIEW uses an existing VI as a property VI.

object.AddPropertyVI([read], [viPath])

#### **Parameters**

Name	Туре	Description
read		If TRUE, creates a read property VI. If FALSE, creates a write property VI. The default is TRUE.
viPath		Path to the property VI. This parameter is optional. If you specify a path, LabVIEW uses an existing VI as a property VI.

### **Return Value**

ProjectItem\*

## **XPropertyFolder Properties and Methods**

This book contains the property and method reference information for the XPropertyFolder class.

# **XPropertyFolder Properties (ActiveX)**

Installed With: Base Package

Inherits from <u>ProjectItem</u>»<u>PropertyFolder</u>.

Property	Description
Description	Sets or gets the description of the property folder. <u>Details</u>
HelpTag	Sets or gets the help tag of the property folder. <u>Details</u>
Identifier	Sets or gets the property identifier of the property folder.  Details
LongName	Sets or gets the long name of the property folder. <u>Details</u>

# **Description Property**

Installed With: Base Package

Class: XPropertyFolder Properties (ActiveX)

Sets or gets the description of the property folder.

object.Description

## **Data Type**

string

## **HelpTag Property**

Installed With: Base Package

Class: XPropertyFolder Properties (ActiveX)

Sets or gets the help tag of the property folder.

object.HelpTag

## **Data Type**

string

# **Identifier Property**

Installed With: Base Package

Class: XPropertyFolder Properties (ActiveX)

Sets or gets the property identifier of the property folder.

object.Identifier

## **Data Type**

string

# **LongName Property**

Installed With: Base Package

Class: XPropertyFolder Properties (ActiveX)

Sets or gets the long name of the property folder.

object.LongName

## **Data Type**

string

# **XPropertyFolder Methods (ActiveX)**

Installed With: Base Package

Inherits from <u>ProjectItem</u>»<u>PropertyFolder</u>.

No methods are specific to this class.

## **TargetItem Properties and Methods**

This book contains the property and method reference information for the TargetItem class.

# **TargetItem Methods (ActiveX)**

Installed With: Base Package

Inherits from **ProjectItem**.

Method	Description
·	Refreshes <b>Dependencies</b> under this target in the LabVIEW <b>Project Explorer</b> window. You also can use the <b>Project Explorer</b> window to refresh dependencies. <b>Details</b>

## **RefreshDependencies Method**

Installed With: Base Package

Class: TargetItem Methods (ActiveX)

Refreshes **Dependencies** under this target in the LabVIEW **Project Explorer** window. You also can use the **Project Explorer** window to refresh dependencies.

object.RefreshDependencies()

#### **Return Value**

none

## **TargetItem Properties (ActiveX)**

Installed With: Base Package

Inherits from **ProjectItem**.

In the following topics, an asterisk (\*) at the end of a data type name indicates that the property is a pointer.

Property	Description
BuildSpecifications	Returns a reference to <b>Build Specifications</b> under the current target in the LabVIEW <b>Project Explorer</b> window. You can use this reference with the <a href="ProjectItem">ProjectItem</a> properties. <a href="Details">Details</a>
Dependencies	Returns a reference to <b>Dependencies</b> under the current target in the LabVIEW <b>Project Explorer</b> window. You can use this reference with the <a href="ProjectItem">ProjectItem</a> properties. <a href="Details">Details</a>

## **BuildSpecifications Property**

Installed With: Base Package

Class: TargetItem Properties (ActiveX)

Returns a reference to **Build Specifications** under the current target in the LabVIEW **Project Explorer** window. You can use this reference with the <u>ProjectItem</u> properties.

object.BuildSpecifications

## **Data Type**

ProjectItem\*

## **Dependencies Property**

Installed With: Base Package

Class: TargetItem Properties (ActiveX)

Returns a reference to **Dependencies** under the current target in the LabVIEW **Project Explorer** window. You can use this reference with the <u>ProjectItem</u> properties.

object.Dependencies

## **Data Type**

ProjectItem\*

# **VI Properties and Methods**

This book contains the property and method reference information for the VI class.

## VI Methods (ActiveX)

Installed With: Base Package

A VirtualInstrument object exports methods that affect a given VI. A VirtualInstrument object is instantiated by invoking the <u>GetVIReference</u> method on the Application object.



**Note** Methods parameters denoted by [] are optional for that method.

Refer to <u>ActiveX Enumerations</u> for more information about enumerations used in various properties.

Method	Description
Abort	Aborts the VI execution. <u>Details</u>
Call	Calls the VI as a subVI. <u>Details</u>
Call2	Calls the VI as a subVI. Optionally, you can open the front panel of the VI and close it after the call is over. You also can suspend the VI when called. Details
CenterFrontPanel	Specifies whether to automatically center the LabVIEW front panel window on the computer screen. <u>Details</u>
CloseFrontPanel	Closes the LabVIEW front panel window. <u>Details</u>
DisconnectFromLibrary	Disconnects the VI from the owning LabVIEW project library. <u>Details</u>
ExportVIStrings	Exports the following strings about VI and LabVIEW front panel objects to a tagged text file: VI name and description, object caption labels, object free labels, default data (string, table, path, and array default data), private data (listbox item names, table row and column headers, graph plot names, graph cursor names, graph annotation names, and tab control page captions), and polymorphic VI data (instance

	names in the polymorphic VI and selector shortcut menus). <u>Details</u>
FPGetRuntimePos	Returns the position of the LabVIEW window at run time. <u>Details</u>
FPRunTimePosRunCentered	Sets a VI to center its front panel every time the VI runs. <u>Details</u>
FPRunTimePosRunCustom	Sets a custom position for the VI to move its front panel to every time the VI runs. <u>Details</u>
FPRunTimePosRunMax	Sets a VI to maximize its front panel every time the VI runs. <u>Details</u>
FPRunTimePosRunMin	Sets a VI to minimize its front panel every time the VI runs. Details
FPRunTimePosRunUnchanged	Sets the VI to not move its front panel when the VI runs. <u>Details</u>
GenerateCCodeVariant	Contains the code generation attributes (name, value pairs) that determine how the LabVIEW C Code Generator generates the C code from the block diagram. Any attributes you do not set use the default behavior. Details
GetControlValue	Gets the value of the named LabVIEW control or indicator as variant data. Use the LabVIEW Variant to Data function to convert the data to another LabVIEW data type. Details
GetLockState	Returns the lock state of the VI and indicates whether the password for the VI is in the password cache. Details
ImportVIStrings	Imports the following strings about VI and LabVIEW front panel objects from a tagged text file: VI name and description, object caption labels, object free labels, default data (string, table, path, and array default data), private data (listbox item names, table row and

	column headers, graph plot names, graph cursor names, graph annotation names, and tab control page captions), and polymorphic VI data (instance names in the polymorphic VI and selector shortcut menus). Details
MakeCurValueDefault	Changes the defaults of all controls on the LabVIEW front panel to be the current values. This method is available only in edit mode. <u>Details</u>
OpenFrontPanel	Opens the LabVIEW front panel window. If the front panel is already open, this method changes the state of the front panel window to the state you wire to this method. Use the CloseFrontPanel method to close the front panel window. You also can use the FPState property to set the state of a front panel window that is already open. Details
PrintPanel	Prints just the LabVIEW front panel to the current printer. You cannot use this method to print a block diagram, list of controls, or polymorphic VI front panel.  Details
PrintVIToHTML	Saves the VI information to an HTML file and saves the graphics in external files. Details
PrintVIToPrinter	Prints the VI information to a printer. If you use this method in a stand-alone application or DLL, LabVIEW prints only the front panel. Details
PrintVIToRTF	Saves the VI information to an RTF file.  Details
PrintVIToText	Saves the VI information to a text file. You cannot save the icon, connector

	pane, front panel, block diagram, subVI icons, and VI hierarchy to text. <u>Details</u>
ReinitializeAllToDefault	Changes the current values of all controls on the LabVIEW front panel to their defaults. Details
Revert	Discards changes and reloads a VI from disk. <u>Details</u>
Run	Starts the VI execution, similar to the Run button in LabVIEW. This method is different than calling a VI because it uses the current values of all front panel controls for execution rather than using data passed in through parameters. This method also ignores the ShowFPOnCall property of a VI and the CloseFPAfterCall property. You cannot use this method to run a VI that is already reserved for execution by another VI. Details
SaveForPrevious	Saves a copy of the VI that is readable by LabVIEW version 8.0 and later.  Details
SaveInstrument	Saves a VI and synchronizes the VI with the edited version in other application instances before saving.  Details
SaveRunTimeMenu	Saves the LabVIEW run-time menu to a file specified by <b>filePath</b> . This method works only when the VI is running. It saves only menu items with valid tags. <a href="Details">Details</a>
SetControlValue	Sets the value of a named LabVIEW control or indicator. <u>Details</u>
SetLockState	Sets the lock state of a VI. If <b>Interactive</b> is FALSE (default), you can use <b>Password</b> to unlock a password-

	protected VI or set the password of an unprotected VI. If <b>Interactive</b> is TRUE, LabVIEW ignores <b>Password</b> and displays a dialog box that prompts you to change the lock state. <u>Details</u>
SetVIIcon	Sets the image of a VI icon from a file.  Details

#### **Abort Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Aborts the VI execution.

This method is similar to the **Abort Execution** button on the LabVIEW

toolbar.

object.Abort()

#### **Return Value**

Boolean

### **Call Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Calls the VI as a subVI.

object.Call([paramNames], [paramVals])

#### **Parameters**

Name	Туре	Description
paramNames	array of strings by ref	Names of the LabVIEW front panel objects that act as input and output parameters to the call.
paramVals	_	Input values for the input parameters and return values from the output parameters in the order in which the names were specified in <b>paramNames</b> . Designates the front panel object values for the input parameters. The values of parameters can be numeric, Boolean, string, or array. A cluster in LabVIEW is represented as an array of variants in ActiveX.

#### **Return Value**

none

#### **Call2 Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Calls the VI as a subVI. Optionally, you can open the front panel of the VI and close it after the call is over. You also can suspend the VI when called.

object.Call2([paramNames], [paramVals], [openFP], [closeFPAfterCall], [suspendOnCall])

#### **Parameters**

Name	Туре	Description
paramNames	array of strings by ref	Names of the LabVIEW front panel objects that act as input and output parameters to the call.
paramVals	array of variants by ref	Input values for the input parameters and return values from the output parameters in the order in which the names were specified in <b>paramNames</b> . Designates the front panel object values for the input parameters. The values of parameters can be numeric, Boolean, string, or array. A cluster in LabVIEW is represented as an array of variants in ActiveX.
openFP	Boolean	Open front panel of the VI
closeFPAfterCall	Boolean	Close front panel of the VI after the call if it was not already open.
suspendOnCall	Boolean	Suspend the VI (Pause the execution) on call.

#### **Return Value**

none

### **CenterFrontPanel Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Specifies whether to automatically center the LabVIEW front panel

window on the computer screen.

object.CenterFrontPanel()

#### **Return Value**

none

#### **CloseFrontPanel Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Closes the LabVIEW front panel window.

Use the **OpenFrontPanel** method to open a front panel window.

object.CloseFrontPanel()

#### **Return Value**

none

## **DisconnectFromLibrary Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Disconnects the VI from the owning LabVIEW project library.

object.DisconnectFromLibrary()

# **ExportVIStrings Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Exports the following strings about VI and LabVIEW front panel objects to a tagged text file: VI name and description, object caption labels, object free labels, default data (string, table, path, and array default data), private data (listbox item names, table row and column headers, graph plot names, graph cursor names, graph annotation names, and tab control page captions), and polymorphic VI data (instance names in the polymorphic VI and selector shortcut menus).

object.ExportVIStrings(stringFile, [interactive], [logFile], [captions], [diagram])

#### **Parameters**

Name	Туре	Description
stringFile	String	Path of the VI strings file including the filename. If you do not enter a filename, set <b>interactive</b> to TRUE so the user can set the VI strings filename.
interactive	Boolean	Specifies whether to display the file dialog box to select the name of the VI strings file. The default value is FALSE.
logFile	String	Path of the log file created to list errors that occurred while exporting VI strings to a tagged text file. The default is no logging.
captions	Boolean	Specifies whether control captions should be automatically created. The default value is FALSE.
diagram	Boolean	Specifies whether to export block diagram strings. The default value is FALSE.

### **FPGetRuntimePos Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Returns the position of the LabVIEW window at run time.

For single-pane front panels, minimum size refers to the content area of that pane, not including the scroll bars. For multi-pane front panels, minimum size refers to the entire front panel, including any visible scroll bars.

object.FPGetRuntimePos([type], [position], [size], [monitor], [useCurPos], [useCurSize])

#### **Parameters**

Name	Туре	Description
type	FPRuntimePos	Returns whether the front panel is unchanged, centered, maximized, minimized, or has a custom setting.
position	16-bit integer	A cluster of top and left pairs that indicate the front panel window position.
size	16-bit integer	A cluster of height and width pairs that indicate the size in pixels of the front panel.
monitor	short	Specifies the monitor on which the front panel window appears, if you have multiple monitors. The value 0 is the primary monitor and is the default value. If you specify a value other than 0, the VI runs on that monitor. If you specify a monitor number greater than the number of monitors on the system, or if you specify a negative number, LabVIEW returns an error. The monitor number corresponds to the video card you plug the monitor in to.
useCurPos	Boolean	Returns the value of the <b>Use Current Position</b> checkbox in the <b>Window Run- Time Position</b> dialog box in LabVIEW.
useCurSize	Boolean	Returns the value of the <b>Use Current Size</b> checkbox in the <b>Window Run-Time Position</b> dialog box in LabVIEW.

**FPRunTimePosEnum** 

#### FPRunTimePosRunCentered Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets a VI to center its front panel every time the VI runs.

For single-pane front panels, minimum size refers to the content area of that pane, not including the scroll bars. For multi-pane front panels, minimum size refers to the entire front panel, including any visible scroll bars.

object.FPRunTimePosRunCentered([monitor], [size])

#### **Parameters**

Name	Туре	Description
monitor	short	Specifies the monitor on which the front panel window appears, if you have multiple monitors. The value 0 is the primary monitor and is the default value. If you specify a value other than 0, the VI runs on that monitor. If you specify a monitor number greater than the number of monitors on the system, or if you specify a negative number, LabVIEW returns an error. The monitor number corresponds to the video card you plug the monitor in to.
size		A cluster of height and width pairs that indicate the size in pixels of the front panel.

#### FPRunTimePosRunCustom Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets a custom position for the VI to move its front panel to every time the

VI runs.

For single-pane front panels, minimum size refers to the content area of that pane, not including the scroll bars. For multi-pane front panels, minimum size refers to the entire front panel, including any visible scroll bars.

object.FPRunTimePosRunCustom([position], [size])

### **Parameters**

Name	Туре	Description
position		A cluster of top and left pairs that indicate the front panel window position.
		A cluster of height and width pairs that indicate the size in pixels of the front panel.

# FPRunTimePosRunMax Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets a VI to maximize its front panel every time the VI runs.

object.FPRunTimePosRunMax([monitor])

#### **Parameters**

Name	Туре	Description
monitor	short	Specifies the monitor on which the front panel window appears, if you have multiple monitors. The value 0 is the primary monitor and is the default value. If you specify a value other than 0, the VI runs on that monitor. If you specify a monitor number greater than the number of monitors on the system, or if you specify a negative number, LabVIEW returns an error. The monitor number corresponds to the video card you plug the monitor in to.

# FPRunTimePosRunMin Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets a VI to minimize its front panel every time the VI runs.

object.FPRunTimePosRunMin([monitor])

### **Parameters**

Name	Туре	Description
monitor		Specifies the monitor on which the front panel window appears, if you have multiple monitors. The value 0 is the primary monitor and is the default value. If you specify a value other than 0, the VI runs on that monitor. If you specify a monitor number greater than the number of monitors on the system, or if you specify a negative number, LabVIEW returns an error. The monitor number corresponds to the video card you plug the monitor in to.

# FPRunTimePosRunUnchanged Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets the VI to not move its front panel when the VI runs.

object. FPRunTimePosRunUnchanged ()

### **GenerateCCodeVariant Method**

**Installed With:** Microprocessor SDK

Class: VI Methods (ActiveX)

Contains the code generation attributes (name, value pairs) that determine how the LabVIEW C Code Generator generates the C code from the block diagram. Any attributes you do not set use the default behavior.

object.GenerateCCodeVariant()

#### **GetControlValue Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Gets the value of the named LabVIEW control or indicator as variant data. Use the LabVIEW Variant to Data function to convert the data to another LabVIEW data type.

The first time you call this method on a VI whose front panel is not open, this method returns the default values of the control or indicator rather than the actual values. Thereafter, it returns the actual value.

For optimization purposes, LabVIEW does not keep track of data values on controls and indicators until LabVIEW determines you want them, that is, until you call this method or display the front panel. When you display the front panel, LabVIEW begins keeping track of the values.

object.GetControlValue(controlName)

#### **Parameters**

Name	Туре	Description
controlName	String	Name of the control or indicator whose value you
		want.

variant

# **GetLockState Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Returns the lock state of the VI and indicates whether the password for

the VI is in the password cache.

object.GetLockState([pwdInCache])

Name	Туре	Description
•		Indicates whether the password is currently in the password cache.

<u>VILockStateEnum</u>

## **GetPanelImage Method**

Installed With: Base Package

Class: VI Methods

Returns an image of the LabVIEW front panel.

If a front panel is not visible, LabVIEW does not update the values in the objects on the front panel. If you call a VI whose front panel is not visible and you use the Get Panel Image method to create an image of the front panel, the image does not reflect any value changes that occurred when you ran the VI. If you want the image to reflect value changes, make sure the front panel is open before any values change.

object.GetPanelImage([visibleonly], [imgdepth], [img], [colors], [bounds])

Name	Туре	Description
visibleonly	Boolean	Sets whether the panel image includes just the objects in the visible area of the front panel or all the front panel objects. The default is TRUE.
imgdepth	Long	Color depth (number of supported colors) for the graphics. The values are 2 (black and white), 16, 256 and 2 <sup>24</sup> (true color).
img	1D array of bytes by ref	Returns the image bitmap as an array of bytes.
colors	1D array of unsigned long by ref	Returns the color table as an array of longs.
bounds	1D array of shorts (four elements) by ref	Returns the bounding rectangle of the image as top, left, bottom, and right values.

## ImportVIStrings Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Imports the following strings about VI and LabVIEW front panel objects from a tagged text file: VI name and description, object caption labels, object free labels, default data (string, table, path, and array default data), private data (listbox item names, table row and column headers, graph plot names, graph cursor names, graph annotation names, and tab control page captions), and polymorphic VI data (instance names in the polymorphic VI and selector shortcut menus).

object.ImportVIStrings(stringFile, [interactive], [logFile])

Name	Туре	Description
stringFile	String	Path of the VI strings file including the filename. If you do not enter a filename, set <b>interactive</b> to TRUE so the user can set the VI strings filename.
interactive	Boolean	If TRUE, the user can see the file dialog box to select the name of the VI strings file. The default is FALSE.
logFile	String	Path of the log file created to list errors that occurred while importing VI strings from a tagged text file. The default is no logging.

### MakeCurValueDefault Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Changes the defaults of all controls on the LabVIEW front panel to be the

current values. This method is available only in edit mode.

object.MakeCurValueDefault()

## **OpenFrontPanel Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Opens the LabVIEW front panel window. If the front panel is already open, this method changes the state of the front panel window to the state you wire to this method. Use the <a href="CloseFrontPanel">CloseFrontPanel</a> method to close the front panel window.

You also can use the <u>FPState</u> property to set the state of a front panel window that is already open.

object.OpenFrontPanel([activate], [state])

Name	Туре	Description
activate		If TRUE (default), the front panel window opens as the active window.
state	<u>FPStateEnum</u>	Sets the state in which to open the front panel window.

### **PrintPanel Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Prints just the LabVIEW front panel to the current printer. You cannot use this method to print a block diagram, list of controls, or polymorphic VI

front panel.

object.PrintPanel([entirePanel])

Name	Туре	Description
entirePanel		Indicates whether the entire front panel is printed. If FALSE (default), only the currently visible portion of the front panel is printed.

## **PrintVIToHTML Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Saves the VI information to an HTML file and saves the graphics in

external files.

object.PrintVIToHTML(htmlFilePath, [append], [format], [imageFormat], [imageDepth], [imageDirectory])

Name	Туре	Description
htmlFilePath	String	Path to the HTML file in which you want to save the VI information. The directory in which you want to save the file must already exist, and you must wire a full path including the HTML filename.
append	Boolean	Specifies whether the new information will be appended to an existing file. The default is FALSE.
format	<u>PrintFormatEnum</u>	Specifies which VI information to print and the format of the printout. You can select from eCustom, eStandard, eUsingPanel, eUsingSubVI, or eComplete.
imageFormat	<u>HTMLImageFormatEnum</u>	Format of the graphic files. Select from ePNG or eJPEG.
imageDepth	Long	Sets the color depth, or number of supported colors, of the image: 1 (1-bit, black and white), 4 (4-bit, 16 colors), 8 (8-bit, 256 colors), or 24 (24-bit, true color). The default is 8.
imageDirectory	String	Path to the directory in which you want to save the graphic files. The directory in which you want to save the files must already exist. If not specified, LabVIEW

saves the image files in the same directory as the HTML
or RTF file.

## **PrintVIToPrinter Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Prints the VI information to a printer. If you use this method in a stand-

alone application or DLL, LabVIEW prints only the front panel.

object.PrintVIToPrinter([format], [scalePanel], [scaleDiagram], [pageHeaders], [pageBreaks], [sectionHeaders])

Name	Туре	Description
format	PrintFormatEnum	Specifies which VI information to print and the format of the printout. You can select from eCustom, eStandard, eUsingPanel, eUsingSubVI, or eComplete.
scalePanel	Boolean	Specifies whether the front panel will be scaled to fit the page. The default is TRUE.
scaleDiagram	Boolean	Specifies whether the block diagram will be scaled to fit the page. The default is TRUE.
pageHeaders	Boolean	Specifies whether page headers (which include the page number, VI name, and last modification data) will be printed. The default is TRUE.
pageBreaks	Boolean	Specifies whether page breaks will be inserted between the following sections: connector icon and description, front panel, list of front panel control details, block diagram, block diagram details, VI hierarchy, and list of subVIs. The default is FALSE.
sectionHeaders	Boolean	Specifies whether to print headers for each section listed above. If <b>format</b> is <b>eCustom</b> , <b>eStandard</b> , or <b>eComplete</b> , the default is TRUE. Otherwise, the default is FALSE.

## **PrintVIToRTF Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Saves the VI information to an RTF file.

object.PrintVIToRTF(rtfFilePath, [append], [format], [imageFormat], [imageDepth], [imageDirectory], [helpFormat])

Name	Туре	Description
rtfFilePath	String	Path to the RTF file in which you want to save the VI information. The directory in which you want to save the file must already exist, and you must specify a full path including the RTF filename.
append	Boolean	Specifies whether the new information will be appended to an existing file. The default is FALSE.
format	PrintFormatEnum	Specifies which VI information to print and the format of the printout. You can select from eCustom, eStandard, eUsingPanel, eUsingSubVI, or eComplete.
imageFormat	Long	Format of the graphics file. The default is 0, which corresponds to the BMP format. BMP is the only format supported.
imageDepth	Long	Sets the color depth, or number of supported colors, of the image: 1 (1-bit, black and white), 4 (4-bit, 16 colors), 8 (8-bit, 256 colors), or 24 (24-bit, true color). The default is 8.
imageDirectory	String	Path to the directory in which you want to save the graphic files. The directory in which you want to save the files must already exist. If not specified, LabVIEW saves the image files in the same directory as the HTML or RTF file.
helpFormat	Boolean	Specifies whether to save the graphics externally and place references in the RTF file. Set this

parameter to TRUE if you are creating a help file. The default is FALSE.
I ALSE.

### **PrintVIToText Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Saves the VI information to a text file. You cannot save the icon, connector pane, front panel, block diagram, subVI icons, and VI hierarchy to text.

object.PrintVIToText(textFilePath, [append], [format])

#### **Parameters**

Name	Туре	Description
textFilePath	String	Path to the text file in which you want to save the VI information. The directory in which you want to save the file must already exist, and you must specify a full path including the text filename.
append	Boolean	Specifies whether the new information will be appended to an existing file. The default is FALSE.
format		Specifies which VI information to print and the format of the printout. You can select from eCustom, eStandard, eUsingPanel, eUsingSubVI, or eComplete.
		Note The eCustom, eStandard, eUsingPanel, and eUsingSubVI parameters print only connector pane controls.

### ReinitializeAllToDefault Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Changes the current values of all controls on the LabVIEW front panel to

their defaults.

object. Reinitialize All To Default ()

## **Revert Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Discards changes and reloads a VI from disk.

object.Revert()

#### **Run Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Starts the VI execution, similar to the **Run** button in LabVIEW. This method is different than calling a VI because it uses the current values of all front panel controls for execution rather than using data passed in through parameters. This method also ignores the <a href="ShowFPOnCall">ShowFPOnCall</a> property of a VI and the <a href="CloseFPAfterCall">CloseFPAfterCall</a> property. You cannot use this method to run a VI that is already reserved for execution by another VI.

object.Run([async])

#### **Parameters**

Name	Туре	Description
async	Boolean	If TRUE, you do not need to wait for the VI to finish
		running. The default is FALSE.

### **SaveForPrevious Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Saves a copy of the VI that is readable by LabVIEW version 8.0 and later.

object.SaveForPrevious(viPath, [warnings])

#### **Parameters**

Name	Туре	Description
viPath	String	Path to where you want to save the VI.
warnings	String	Returns any warnings that occurred while saving the VI.

### **SaveInstrument Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Saves a VI and synchronizes the VI with the edited version in other

application instances before saving.

object.SaveInstrument([viPath], [saveACopy], [withoutDiagram])

#### **Parameters**

Name	Туре	Description
viPath	String	Path of the VI file. The default is the current location.
saveACopy	Boolean	Indicates whether to save a copy of the VI to another file.
withoutDiagram	Boolean	Indicates whether to save the VI without the block diagram.

### SaveRunTimeMenu Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Saves the LabVIEW run-time menu to a file specified by **filePath**. This method works only when the VI is running. It saves only menu items with

valid tags.

object.SaveRunTimeMenu(filePath)

#### **Parameters**

Name	Туре	Description
filePath	String	Path to where you want to save the run-time menu.

### **SetControlValue Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets the value of a named LabVIEW control or indicator.

object.SetControlValue(controlName, value)

#### **Parameters**

Name	Туре	Description
controlName	String	Name of the front panel control.
value		New value of the front panel control. This value can be numeric, Boolean, string, or array. A cluster in LabVIEW is represented as an array of variants in ActiveX.

#### SetLockState Method

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets the lock state of a VI. If **Interactive** is FALSE (default), you can use **Password** to unlock a password-protected VI or set the password of an unprotected VI. If **Interactive** is TRUE, LabVIEW ignores **Password** and displays a dialog box that prompts you to change the lock state.

object.SetLockState(lockState, [Interactive], [Password], [putInCache])

#### **Parameters**

Name	Туре	Description
lockState	<u>VILockStateEnum</u>	Lock state.
Interactive	Boolean	Specifies whether to display a dialog box that prompts you to change the lock state. The default is FALSE.
Password	String	If <b>lockState</b> is Password-protected, <b>Password</b> is the new password. If you are changing <b>lockState</b> from Password-protected, <b>Password</b> is the old password. The default is an empty string.
putInCache	Boolean	If <b>lockState</b> is Password-protected, <b>putinCache</b> specifies whether to place the new password in the cache. The default is FALSE.

### **SetVIIcon Method**

Installed With: Base Package

Class: VI Methods (ActiveX)

Sets the image of a VI icon from a file.

object.SetVIIcon(imageFile)

#### **Parameters**

Name	Туре	Description
imageFile	String	Path to the image you want to use for the VI icon.

#### **Return Value**

none

#### **VI Properties (ActiveX)**

Installed With: Base Package

A VirtualInstrument object exports properties that affect a given VI. A VirtualInstrument object is instantiated by invoking the <u>GetVIReference</u> method on the Application object.

In the following topics, an asterisk (\*) at the end of a data type name indicates that the property is a pointer. Property parameters denoted by [] are optional for that property.

Refer to <u>ActiveX Enumerations</u> for more information about enumerations used in various properties.

Property	Description
AllowDebugging	If TRUE, you can use debugging tools on the VI. Set this property to FALSE to reduce memory requirements and to improve performance slightly for the VI. Details
BDModificationBitSet	Indicates whether changes were made to the block diagram since the VI was saved or opened, depending on which was last. If the value is zero, no changes were made. If the value is nonzero, changes were made. This property is read only. Details
BDSize	Size of the LabVIEW block diagram in bytes. This property is read only.  Details
Callees	Returns a list of all the subVIs or callees in a VI. This property excludes any subVIs or callees from the callees list if they are located in an unused case of a case structure. This property is read only. Details
Callers	Returns a list of all the loaded VIs that call the referenced VI. This property is read only. Details

CloneName	Name of the clone VI. Returns an error if the VI is not a clone. <u>Details</u>
CloseFPAfterCall	Indicates whether to close the LabVIEW front panel after the VI runs.  Details
CodeSize	Amount of memory used for VI code in bytes. This property is read only.  Details
DataSize	Amount of memory allocated for data in bytes. You cannot read this property if the VI is running. This property is read only. Details
Description	Description of the VI that appears in the LabVIEW <b>Context Help</b> window when you move the cursor over the VI icon and in VI documentation you generate. <u>Details</u>
EditMode	If TRUE, the VI opens in edit mode. If FALSE, the VI opens in run mode, and the title bar, menu bar, and toolbar do not appear. Details
ExecPriority	Indicates the priority of the VI when it runs in parallel with other tasks.  Details
ExecState	Indicates the execution state of the VI. <u>Details</u>
ExpandWhenDroppedAsSubVI	Expands to show terminals when dropped as a subVI. <u>Details</u>
FPAllowRTPopup	Indicates whether to display shortcut menus for LabVIEW front panel objects while the VI runs. If you do not display default run-time shortcut menus, you can continue to include customized shortcut menus. Details
FPBehavior	Sets the behavior of the LabVIEW

	front panel window. Valid values include 0 (Invalid), 1 (Default), 2 (Floating), 3 (Floating/Auto-Hide), and 4 (Modal). Details
FPHiliteReturnButton	Indicates whether to highlight Boolean controls that have a shortcut key of <enter>. Details</enter>
FPKeepWinProps	Indicates whether the LabVIEW front panel window adjusts its size in proportion with a change in monitor resolution. The window changes size so it covers the same percentage of the screen that it covered at its original resolution. Details
FPMinimizable	Indicates whether the user can minimize the LabVIEW front panel window while the VI runs. <u>Details</u>
FPModificationBitSet	Indicates whether changes were made to the LabVIEW front panel since the VI was saved. If the value is zero, no changes where made. If the value is nonzero, changes were made. Details
FPMonitor	The monitor on which the LabVIEW front panel window appears, if you have multiple monitors. The value 0 is the primary monitor. Details
FPResizable	Indicates whether the user can resize the LabVIEW front panel window while the VI runs. Details
FPRunTransparently	Sets the VI to run transparently.  Details
FPShowMenuBar	Indicates whether to display the menu bar on the LabVIEW front panel while the VI runs. <u>Details</u>
FPSize	Size of the LabVIEW front panel in bytes. This property is read only.

	<u>Details</u>
FPState	Current state of the LabVIEW front panel window. Valid values include 0 (Invalid), 1 (Standard), 2 (Closed), 3 (Hidden), 4 (Minimized), and 5 (Maximized). A state of Standard or Maximized indicates that the front panel window is visible to the user. If you attempt to set this property for a front panel window that is not open, the property returns an error. Details
FPTitleBarVisible	Indicates whether to display a title bar on the LabVIEW front panel while the VI runs. Details
FPTransparency	Sets the window transparency level of the VI. The level of transparency is a percentage where 0 is opaque and 100 is invisible. This property returns an error if you specify a value outside the range of 0 to 100. If you set <a href="FPRunTransparently">FPRunTransparently</a> to False, changing this property has no effect. <a href="Details">Details</a>
FPWinBounds	The four elements in the cluster are the top, left, bottom, and right values of the LabVIEW front panel window, which includes the interior region, scroll bars, title bar, menu bar, and toolbar. They are in global screen coordinates, that is, the numbers refer to coordinates within a computer monitor's screen (rather than an open window). You can set this property only for VIs with open front panels. If you do not want the VI for which you want to set this property to appear to users, use the Hidden value of the

	FPStateEnum data type for the FPState property to hide the front panel of the VI. Details
FPWinClosable	Indicates whether the close button in the LabVIEW title bar is disabled and the <b>Close</b> item in the LabVIEW <b>File</b> menu is disabled. Details
FPWinCustomTitle	Indicates whether the VI has a custom title string. Write FALSE to remove the custom title string. Details
FPWinIsFrontMost	Writing TRUE to this value brings the front panel to the front. This property applies only in the application instance of the calling VI. Writing FALSE to this value has no effect. If you read this property, it indicates whether the front panel window is the front window (ignoring floating windows). Details
FPWinOpen	National Instruments recommends you use the OpenFrontPanel method instead of this property. Details
FPWinPanelBounds	The four elements in the cluster are the top, left, bottom, and right values of the interior portion of the LabVIEW front panel, not including the scroll bars, title bar, menu bar, and toolbar. The cluster elements are in global screen coordinates, that is, the numbers refer to coordinates within a computer monitor's screen (rather than an open window). You can set this property only for VIs with open front panels. If you do not want the VI for which you want to set this property to appear to users, use the Hidden value of the FPStateEnum data type for the FPState property to hide the front

	panel of the VI. <u>Details</u>
FPWinTitle	String that appears in the LabVIEW title bar. This string does not have to match the VI filename. Details
HelpDocumentPath	Path to an HTML file (.htm or .html) or compiled help file (.chm or .hlp) to which the VI is linked. If the path is to a compiled help file, use the HelpDocumentTag property to determine the specific topic in that help file. Details
HelpDocumentTag	Index keyword or HTML filename for a topic in the compiled help file to which the VI is linked. To link to a bookmark within an HTML file, add # followed by the name of the bookmark to the end of the filename. Use this property only when HelpDocumentPath is a path to a compiled help file (.chm or .hlp). For .chm files, this property can be an HTML filename or index keyword. For .hlp files, this property can be an index keyword. Details
HistAddCommentsAtSave	Indicates whether to add a comment to the VI revision history every time the VI is saved. Details
HistoryText	Returns all the text that was added to the VI revision history. This property is read only. Details
HistPromptAtClose	Indicates whether to prompt for a VI revision history comment when the VI closes. <u>Details</u>
HistPromptForCommentsAtSave	Indicates whether to prompt for a VI revision history comment when the VI is saved. <u>Details</u>

HistRecordAppComments	Indicates whether to add comments to the VI revision history when certain events occur, such as conversion to a new version of LabVIEW, subVI changes, and changes to the name or path of the VI. Details
HistUseDefaults	Indicates whether to use the global default history or to use the values entered in other history properties.  Details
IsCloneVI	Returns TRUE if the VI is a clone VI.  Details
IsProbe	Returns TRUE if the VI is running as a probe and the probe VI is open.  Details
IsReentrant	Indicates whether a VI can be reentrant. Details
Library	Returns the name of the LabVIEW project library, XControl, or LabVIEW class that owns the VI. If no library, XControl, or class owns the VI, the property returns NULL. Details
LogAtFinish	Indicates whether to write LabVIEW front panel values to a datalog file after the VI runs. Details
LogFilePath	Path of the datalog file in which LabVIEW front panel data and a time stamp are written. Details
Name	Name of the VI file. You can write this property only if the VI has not been saved to disk. If a LabVIEW project library owns the VI, this property returns the qualified name of the VI, which includes the project library filename. Details
OwningApp	Returns a reference to the Application

	that owns this VI. Close this reference when you are finished using it. <u>Details</u>
Path	Path to the VI file. This property is read only. Details
PreferredExecSystem	Indicates the execution system in which the VI runs. <u>Details</u>
PrintHeaderDatePrint	If TRUE, LabVIEW includes the date printed in the headers for the VI. Use the PrintingHeaders property to set whether LabVIEW prints the headers for the VI. Details
PrintHeaderModifyDate	If TRUE, LabVIEW includes the last modified date in the headers for the VI. Use the PrintingHeaders property to set whether LabVIEW prints the headers for the VI. Details
PrintHeaderPageNumber	If TRUE, LabVIEW includes the page number in the headers for the VI. Use the <u>PrintingHeaders</u> property to set whether LabVIEW prints the headers for the VI. <u>Details</u>
PrintHeaderVIIcon	If TRUE, LabVIEW includes the VI icon in the headers for the VI. Use the PrintingHeaders property to set whether LabVIEW prints the headers for the VI. Details
PrintHeaderVIName	If TRUE, LabVIEW includes the VI name in the headers for the VI. Use the PrintingHeaders property to set whether LabVIEW prints the headers for the VI. Details
PrintingBDScaling	If TRUE, LabVIEW scales the block diagram to fit on the printed page.  Details
PrintingFPScaling	If TRUE, LabVIEW scales the front

	panel to fit on the printed page. <u>Details</u>
PrintingHeaders	If TRUE, LabVIEW prints headers for the VI. Use the PrintHeader properties in this class to customize the contents of the headers. <u>Details</u>
PrintingHeaderVIPath	If TRUE, LabVIEW includes the VI path in the headers for the VI. Use the PrintingHeaders property to set whether LabVIEW prints the headers for the VI. Details
PrintingOrientation	Gets or sets the page orientation to use when printing the VI. Details
PrintLogFileAtFinish	Indicates whether to print the LabVIEW front panel after the VI runs.  Details
PrintMargins	Gets or sets the page margins to use when printing the VI in inches or centimeters. <u>Details</u>
ReentrancyType	Indicates the type of reentrancy LabVIEW uses for a reentrant VI. This property is similar to options on the Execution Properties page of the VI Properties dialog box. Details
RevisionNumber	Current revision number of the VI.  Details
RunOnOpen	Indicates whether to run the VI when it opens. <u>Details</u>
RunTimeMenuPath	When read, this property returns the run-time menu path of the VI. When written, this property updates the runtime menu path of the VI. If the VI is running when you write this property, it updates the menu with data from the new path. <a href="Details">Details</a>
ShowFPOnCall	Indicates whether to show the

	LabVIEW front panel when the VI is called. Details
ShowFPOnLoad	Indicates whether to show the LabVIEW front panel when the VI is loaded. Details
SuspendOnCall	Indicates whether the VI should be suspended when it is called as a subVI. Use this property carefully with reentrant VIs. Details
TBShowAbortButton	Indicates whether to display the <b>Abort Execution</b> button on the LabVIEW toolbar while the VI runs. Details
TBShowFreeRunButton	Indicates whether to display the <b>Run Continuously</b> button on the LabVIEW toolbar while the VI runs. Details
TBShowRunButton	Indicates whether to display the <b>Run</b> button on the LabVIEW toolbar while the VI runs. Details
TBVisible	Indicates whether to display the LabVIEW toolbar while the VI runs.  Details
VIModificationBitSet2	Indicates whether changes were made to the VI since the VI was last saved. If the value is zero, no changes have been made. If the value is nonzero, changes have been made. This property is read only. Details
VIType	Indicates the type of the VI. <u>Details</u>

#### **AllowDebugging Property**

Installed With: Base Package Class: VI Properties (ActiveX)

If TRUE, you can use debugging tools on the VI. Set this property to FALSE to reduce memory requirements and to improve performance slightly for the VI.

You cannot debug a reentrant VI.



**Note** Writing this property causes the VI to be recompiled. Writing TRUE causes extra code and data to be generated for the VI. This code and data enables LabVIEW to set breakpoints and single-step through the VI. Writing FALSE removes this code and data, reducing the total size of the VI.

object.AllowDebugging

Boolean

#### **BDModificationBitSet Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether changes were made to the block diagram since the VI was saved or opened, depending on which was last. If the value is zero, no changes were made. If the value is nonzero, changes were made.

This property is read only.

object.BDModificationBitSet

long

## **BDSize Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Size of the LabVIEW block diagram in bytes. This property is read only.

object.BDSize

long

#### **Callees Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Returns a list of all the subVIs or callees in a VI. This property excludes any subVIs or callees from the callees list if they are located in an unused case of a case structure. This property is read only.

object.Callees

Array of strings

## **Callers Property**

Installed With: Base Package

Class: VI Properties (ActiveX)

Returns a list of all the loaded VIs that call the referenced VI. This

property is read only.

object.Callers

Array of strings

# **CloneName Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Name of the clone VI. Returns an error if the VI is not a clone.

object.CloneName

string

### **CloseFPAfterCall Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to close the LabVIEW front panel after the VI runs.

object.CloseFPAfterCall

Boolean

## **CodeSize Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Amount of memory used for VI code in bytes. This property is read only.

object.CodeSize

long

## **DataSize Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Amount of memory allocated for data in bytes. You cannot read this

property if the VI is running. This property is read only.

object.DataSize

long

### **Description Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Description of the VI that appears in the LabVIEW **Context Help** window when you move the cursor over the VI icon and in VI documentation you generate.

You can format the text in the description to appear bold in the **Context Help** window.

object.Description

String

### **EditMode Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, the VI opens in edit mode. If FALSE, the VI opens in run mode, and the title bar, menu bar, and toolbar do not appear.

To write this property, you must call it before you open the front panel of the VI.

object.EditMode

Boolean

# **ExecPriority Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates the priority of the VI when it runs in parallel with other tasks.

object.ExecPriority

**VIPriorityEnum** 

# **ExecState Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Indicates the execution state of the VI.

object.ExecState

**ExecStateEnum** 

# **ExpandWhenDroppedAsSubVI Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Expands to show terminals when dropped as a subVI.

object. Expand When Dropped As Sub VI

Boolean

### **FPAllowRTPopup Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display shortcut menus for LabVIEW front panel objects while the VI runs. If you do not display default run-time shortcut menus, you can continue to include customized shortcut menus.

object.FPAllowRTPopup

Boolean

## **FPBehavior Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Sets the behavior of the LabVIEW front panel window. Valid values include 0 (Invalid), 1 (Default), 2 (Floating), 3 (Floating/Auto-Hide), and 4 (Modal).

object.FPBehavior

**VIFPBehaviorEnum** 

## **FPHiliteReturnButton Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Indicates whether to highlight Boolean controls that have a shortcut key

of <Enter>.

object.FPHiliteReturnButton

Boolean

### **FPKeepWinProps Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether the LabVIEW front panel window adjusts its size in proportion with a change in monitor resolution. The window changes size so it covers the same percentage of the screen that it covered at its original resolution.

object.FPKeepWinProps

Boolean

## **FPMinimizable Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Indicates whether the user can minimize the LabVIEW front panel window while the VI runs.

If you load the VI in a subpanel control, this property is read only and always returns a value of FALSE.

This property is similar to the **Allow user to minimize window** checkbox in the LabVIEW **Customize Window Appearance** dialog box.

object.FPMinimizable

Boolean

# **FPModificationBitSet Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Indicates whether changes were made to the LabVIEW front panel since the VI was saved. If the value is zero, no changes where made. If the value is nonzero, changes were made.

object.FPModificationBitSet

long

# **FPMonitor Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

The monitor on which the LabVIEW front panel window appears, if you

have multiple monitors. The value 0 is the primary monitor.

object.FPMonitor

short

#### **FPResizable Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Indicates whether the user can resize the LabVIEW front panel window while the VI runs.

If you load the VI in a subpanel control, this property is read only and always returns a value of FALSE.

This property is similar to the **Allow user to resize window** checkbox in the LabVIEW **Customize Window Appearance** dialog box.

object.FPResizable

Boolean

# **FPRunTransparently Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Sets the VI to run transparently.

object.FPRunTransparently

Boolean

### **FPShowMenuBar Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display the menu bar on the LabVIEW front panel

while the VI runs.

If you load the VI in a subpanel control, this property is read only and always returns a value of FALSE.

object.FPShowMenuBar

Boolean

### **FPSize Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Size of the LabVIEW front panel in bytes. This property is read only.

This property is similar to the **Front Panel Objects** option on the **Memory Usage** page of the LabVIEW **VI Properties** dialog box.

object.FPSize

long

#### **FPState Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Current state of the LabVIEW front panel window. Valid values include 0 (Invalid), 1 (Standard), 2 (Closed), 3 (Hidden), 4 (Minimized), and 5 (Maximized).

A state of Standard or Maximized indicates that the front panel window is visible to the user.

If you attempt to set this property for a front panel window that is not open, the property returns an error.

object.FPState

**FPStateEnum** 

#### **FPTitleBarVisible Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display a title bar on the LabVIEW front panel while the VI runs.

If you load the VI in a subpanel control, this property is read only and always returns a value of FALSE.

This property is similar to the **Window has title bar** option on the LabVIEW **Customize Window Appearance** dialog box.

object.FPTitleBarVisible

Boolean

#### **FPTransparency Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Sets the window transparency level of the VI. The level of transparency is a percentage where 0 is opaque and 100 is invisible. This property returns an error if you specify a value outside the range of 0 to 100. If you set <a href="FPRunTransparently">FPRunTransparently</a> to False, changing this property has no effect.

object.FPTransparency

char

#### **FPWinBounds Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

The four elements in the cluster are the top, left, bottom, and right values of the LabVIEW front panel window, which includes the interior region, scroll bars, title bar, menu bar, and toolbar. They are in global screen coordinates, that is, the numbers refer to coordinates within a computer monitor's screen (rather than an open window).

You can set this property only for VIs with open front panels. If you do not want the VI for which you want to set this property to appear to users, use the Hidden value of the <a href="FPStateEnum">FPStateEnum</a> data type for the <a href="FPStateEnum">FPState</a> property to hide the front panel of the VI.

When you read this property, LabVIEW returns the window bounds of the VI. If the front panel of the VI is not open, LabVIEW returns the window bounds of the VI in the position it was last saved.

If you load the VI in a subpanel control, this property is read only.

object.FPWinBounds

Array of short variants (four elements)

### **FPWinClosable Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Indicates whether the close button in the LabVIEW title bar is disabled and the **Close** item in the LabVIEW **File** menu is disabled.

This property prevents users from closing a VI during execution.

If you load the VI in a subpanel control, this property is read only and always returns a value of FALSE.

object.FPWinClosable

Boolean

# **FPWinCustomTitle Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether the VI has a custom title string. Write FALSE to remove

the custom title string.

object.FPWinCustomTitle

Boolean

#### **FPWinIsFrontMost Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Writing TRUE to this value brings the front panel to the front. This property applies only in the application instance of the calling VI. Writing FALSE to this value has no effect. If you read this property, it indicates whether the front panel window is the front window (ignoring floating windows).

object.FPWinIsFrontMost

Boolean

#### **FPWinOpen Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

National Instruments recommends you use the OpenFrontPanel method instead of this property.

Reading this property indicates whether the window currently is open. In certain situations, the value returned by this property might not be accurate. For example, if the window is hidden and you read this property, the property returns a value of TRUE. However, the window is not visible to the user.

Use the OpenFrontPanel method to set the state of the window when you display it. Use the FPState property for more accurate information about the current state of the window. Use the CloseFrontPanel method to close the window.

If you set this property to TRUE, LabVIEW displays the front panel window. If FALSE, LabVIEW closes the window. If no other references to the VI are open, LabVIEW removes the VI from memory. If you set this property to FALSE in a stand-alone application, the application stops.

If you load the VI in a subpanel control, this property is read only and always returns a value of TRUE.

If you set the boundaries of a front panel to be outside the boundaries of the screen, LabVIEW ignores those settings when you open the front panel. LabVIEW displays a portion of the window title bar so you can drag the window to another location on the screen. Use the FPWinPanelBounds property to set the bounds of the front panel window.

object.FPWinOpen

none

#### **FPWinPanelBounds Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

The four elements in the cluster are the top, left, bottom, and right values of the interior portion of the LabVIEW front panel, not including the scroll bars, title bar, menu bar, and toolbar. The cluster elements are in global screen coordinates, that is, the numbers refer to coordinates within a computer monitor's screen (rather than an open window).

You can set this property only for VIs with open front panels. If you do not want the VI for which you want to set this property to appear to users, use the Hidden value of the <a href="FPStateEnum">FPStateEnum</a> data type for the <a href="FPStateEnum">FPState</a> property to hide the front panel of the VI.

When you read this property, LabVIEW returns the front panel bounds of the VI. If the front panel of the VI is not open, LabVIEW returns the front panel bounds of the VI in the position it was last saved.

If you load the VI in a subpanel control, this property is read only.

object.FPWinPanelBounds

1D array of shorts (four elements) by ref

## **FPWinTitle Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

String that appears in the LabVIEW title bar. This string does not have to match the VI filename.

This property is similar to the **Window title** option on the **Window Appearance** page of the LabVIEW **VI Properties** dialog box.

object.FPWinTitle

String

#### **HelpDocumentPath Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Path to an HTML file (.htm or .html) or compiled help file (.chm or .hlp) to which the VI is linked. If the path is to a compiled help file, use the <a href="HelpDocumentTag">HelpDocumentTag</a> property to determine the specific topic in that help file.

This property is similar to the **Help Path** text box on the **Documentation** page of the LabVIEW **VI Properties** dialog box.

object.HelpDocumentPath

string

#### **HelpDocumentTag Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Index keyword or HTML filename for a topic in the compiled help file to which the VI is linked. To link to a bookmark within an HTML file, add # followed by the name of the bookmark to the end of the filename. Use this property only when <a href="HelpDocumentPath">HelpDocumentPath</a> is a path to a compiled help file (.chm or .hlp).

For .chm files, this property can be an HTML filename or index keyword. For .hlp files, this property can be an index keyword.

This property is similar to the **Help Tag** text box on the **Documentation** page of the LabVIEW **VI Properties** dialog box.

object.HelpDocumentTag

string

## **HistAddCommentsAtSave Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to add a comment to the VI revision history every time the VI is saved.

This property is similar to the options available in the LabVIEW **History** window.

object. Hist Add Comments At Save

Boolean

# **HistoryText Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Returns all the text that was added to the VI revision history. This

property is read only.

object.HistoryText

string

# **HistPromptAtClose Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to prompt for a VI revision history comment when the VI closes.

This property is similar to the options available in the LabVIEW **History** window.

object.HistPromptAtClose

Boolean

## **HistPromptForCommentsAtSave Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to prompt for a VI revision history comment when the

VI is saved.

This property is similar to the options available in the LabVIEW **History** window.

object. Hist Prompt For Comments At Save

Boolean

#### **HistRecordAppComments Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Indicates whether to add comments to the VI revision history when certain events occur, such as conversion to a new version of LabVIEW, subVI changes, and changes to the name or path of the VI.

This property is similar to the options available in the LabVIEW **Revision History Properties** page of the **VI Properties** dialog box or the **Revision History Options** page of the **Options** dialog box.

object. Hist Record App Comments

Boolean

#### **HistUseDefaults Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to use the global default history or to use the values entered in other history properties.

This property is similar to the options available in the **Revision History** page of the LabVIEW **VI Properties** dialog box. You can specify the global default history in the LabVIEW **VI Properties** dialog box.

object.HistUseDefaults

Boolean

# **IsCloneVI Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Returns TRUE if the VI is a clone VI.

object.IsCloneVI

Boolean

# **IsProbe Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Returns TRUE if the VI is running as a probe and the probe VI is open.

object.IsProbe

Boolean

# **IsReentrant Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Indicates whether a VI can be reentrant.

object.IsReentrant

Boolean

# **Library Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Returns the name of the LabVIEW project library, XControl, or LabVIEW class that owns the VI. If no library, XControl, or class owns the VI, the property returns NULL.

object.Library

Library\*

## **LogAtFinish Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Indicates whether to write LabVIEW front panel values to a datalog file after the VI runs.



**Note** If you set LogAtFinish without having set the LogFilePath property, the next time this VI finishes executing, a dialog box appears prompting you to specify the file path.

object.LogAtFinish

Boolean

#### **LogFilePath Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Path of the datalog file in which LabVIEW front panel data and a time stamp are written.



**Note** Use this property in conjunction with the <u>LogAtFinish</u> property, which specifies whether to log the data. If you set the LogAtFinish property without also setting this property, a dialog box appears at run time, prompting you to specify a file path.

object.LogFilePath

string

#### **Name Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Name of the VI file. You can write this property only if the VI has not been saved to disk. If a LabVIEW project library owns the VI, this property returns the qualified name of the VI, which includes the project library filename.

object.Name

string

# **OwningApp Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Returns a reference to the Application that owns this VI. Close this

reference when you are finished using it.

object.OwningApp

Application\*

# **Path Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Path to the VI file. This property is read only.

object.Path

string

# **PreferredExecSystem Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Indicates the execution system in which the VI runs.

object. Preferred Exec System

VIExecSysEnum

# **PrintHeaderDatePrint Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, LabVIEW includes the date printed in the headers for the VI. Use the <u>PrintingHeaders</u> property to set whether LabVIEW prints the

headers for the VI.

object.PrintHeaderDatePrint

Boolean

# **PrintHeaderModifyDate Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

If TRUE, LabVIEW includes the last modified date in the headers for the VI. Use the <u>PrintingHeaders</u> property to set whether LabVIEW prints the headers for the VI.

object. Print Header Modify Date

Boolean

# **PrintHeaderPageNumber Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

If TRUE, LabVIEW includes the page number in the headers for the VI. Use the <u>PrintingHeaders</u> property to set whether LabVIEW prints the headers for the VI.

object. Print Header Page Number

Boolean

## **PrintHeaderVIIcon Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, LabVIEW includes the VI icon in the headers for the VI. Use the <a href="PrintingHeaders">PrintingHeaders</a> property to set whether LabVIEW prints the headers for the VI.

the VI.

object.PrintHeaderVIIcon

## **PrintHeaderVIName Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, LabVIEW includes the VI name in the headers for the VI. Use the <u>PrintingHeaders</u> property to set whether LabVIEW prints the headers

for the VI.

object.PrintHeaderVIName

# **PrintingBDScaling Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, LabVIEW scales the block diagram to fit on the printed page.

This property is similar to the **Scale printed block diagram to fit page** checkbox on the **Print Options** page of the LabVIEW **VI Properties** dialog box.

object.PrintingBDScaling

### **PrintingFPScaling Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, LabVIEW scales the front panel to fit on the printed page.

This property is similar to the **Scale printed front panel to fit page** checkbox on the **Print Options** page of the LabVIEW **VI Properties** 

dialog box.

object.PrintingFPScaling

### **PrintingHeaders Property**

Installed With: Base Package Class: VI Properties (ActiveX)

If TRUE, LabVIEW prints headers for the VI. Use the PrintHeader properties in this class to customize the contents of the headers.

This property is similar to the **Print header (name, date, page number)** checkbox on the **Print Options** page of the LabVIEW **VI Properties** dialog box.

object.PrintingHeaders

### **PrintingHeaderVIPath Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

If TRUE, LabVIEW includes the VI path in the headers for the VI. Use the <u>PrintingHeaders</u> property to set whether LabVIEW prints the headers for the VI.

object.PrintingHeaderVIPath

# **PrintingOrientation Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Gets or sets the page orientation to use when printing the VI.

object.PrintingOrientation

<u>PageOrientationEnum</u>

# **PrintLogFileAtFinish Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to print the LabVIEW front panel after the VI runs.

object.PrintLogFileAtFinish

### **PrintMargins Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Gets or sets the page margins to use when printing the VI in inches or centimeters.

This property is similar to the **Use custom page margins** checkbox on the **Print Options** page of the LabVIEW **VI Properties** dialog box.

object.PrintMargins

**PrintMarginsEnum** 

#### ReentrancyType Property

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates the type of reentrancy LabVIEW uses for a reentrant VI. This property is similar to options on the **Execution Properties** page of the **VI Properties** dialog box.

You can select from the following options:

- Share clones between instances—Allocates clones of the reentrant VI on-demand for each call to the reentrant VI. Do not select this option if a clone VI must preserve state information across calls or if the VI is part of an application that needs to minimize jitter.
- Preallocate clone for each instance—Allocates clones of the reentrant VI for each call to the reentrant VI. This option is recommended for VIs that are to run with low jitter on LabVIEW Real-Time.

object.ReentrancyType

kReentrancyTypeTD

#### **RevisionNumber Property**

Installed With: Base Package Class: VI Properties (ActiveX)

Current revision number of the VI.

When writing this property, you can write only a value greater than the current revision number. If you attempt to write a revision number whose value is less than the current revision number, the property returns an error.

This property is similar to the options available in the **Revision History** page of the LabVIEW **VI Properties** dialog box.

object.RevisionNumber

long

# **RunOnOpen Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to run the VI when it opens.

object.RunOnOpen

#### **RunTimeMenuPath Property**

Installed With: Base Package Class: VI Properties (ActiveX)

When read, this property returns the run-time menu path of the VI. When written, this property updates the run-time menu path of the VI. If the VI is running when you write this property, it updates the menu with data from the new path.

Use this property to programmatically specify the path for a run-time menu (.rtm) file. This is useful if you are developing multilingual applications and you want to switch menus for each language programmatically.

object.RunTimeMenuPath

string

# **ShowFPOnCall Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to show the LabVIEW front panel when the VI is called.

object.ShowFPOnCall

# **ShowFPOnLoad Property**

Installed With: Base Package

Class: VI Properties (ActiveX)

Indicates whether to show the LabVIEW front panel when the VI is

loaded.

object.ShowFPOnLoad

# **SuspendOnCall Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether the VI should be suspended when it is called as a

subVI. Use this property carefully with reentrant VIs.

object.SuspendOnCall

#### **TBShowAbortButton Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display the **Abort Execution** button on the LabVIEW toolbar while the VI runs.

This property is similar to the **Show Abort button** option on the LabVIEW **Customize Window Appearance** dialog box.

object.TBShowAbortButton

# **TBShowFreeRunButton Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display the Run Continuously button on the

LabVIEW toolbar while the VI runs.

object. TB Show Free Run Button

# **TBShowRunButton Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display the **Run** button on the LabVIEW toolbar

while the VI runs.

object.TBShowRunButton

# **TBVisible Property**

Installed With: Base Package
Class: VI Properties (ActiveX)

Indicates whether to display the LabVIEW toolbar while the VI runs.

object.TBVisible

# **VIModificationBitSet2 Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Indicates whether changes were made to the VI since the VI was last saved. If the value is zero, no changes have been made. If the value is nonzero, changes have been made. This property is read only.

object.VIModificationBitSet2

hyper

# **VIType Property**

**Installed With:** Base Package

Class: VI Properties (ActiveX)

Indicates the type of the VI.

object.VIType

**VITypeEnum** 

# **PageOrientationEnum**

Indicates the page orientation when LabVIEW prints the VI.

Name	Value	Description
Portrait	0	Prints the report so the short edge of the paper is the top of the page.
Landscape	1	Prints the report so the long edge of the paper is the top of the page.
Rotated Portrait	2	Rotates the paper 90 degrees counterclockwise and prints the report using a portrait orientation.
Rotated Landscape	3	Rotates the paper 90 degrees counterclockwise and prints the report using a landscape orientation.

# **PrintMethodsEnum**

Indicates how LabVIEW prints the data.

Name	Value	Description
Standard	0	Translates the VI print data (front panel, block diagram, icon, and so on) and sends it to the printer using the standard operating system drawing commands. You must use this option if the printer does not have PostScript support or if you want the printer driver to handle the PostScript translation instead of LabVIEW.
PostScript	1	Translates the VI print data in PostScript (.ps) format and sends it to the printer as PostScript commands. Do not use this option if the printer or printer driver does not support PostScript printing. PostScript printouts reproduce the image of the screen, including patterns, line styles, and fonts, more accurately.
Bitmap	2	Creates a bitmap, draws all data for that page into the bitmap, and sends the bitmap to the printer.

#### **VIFPBehaviorEnum**

Indicates how LabVIEW displays the front panel window.

Name	Value	Description
Invalid	0	If you read this property and an error occurs, the property returns this value. If you set the property to this value, the property returns an error.
Default	1	The front panel window is not floating or modal.
Floating	2	The front panel window stays on top of all other non-modal LabVIEW windows. The <b>Just-In-Time Advice</b> window is an example of a floating window.
Floating/Auto- Hide	3	The front panel window is floating and when you switch from LabVIEW to another application, the window does not appear.
Modal	4	The front panel window stays on top of all other LabVIEW windows until you close the window or open another modal window.