

1

VWorks ActiveX control

This section contains the following topics:

- “About the VWorks ActiveX control” on page 2
- “Methods” on page 4
- “Events” on page 16
- “Enumerated types” on page 20

About the VWorks ActiveX control

What is the VWorks ActiveX control

The VWorks ActiveX control is the software component that allows the VWorks software to interact with a third-party lab automation system.

How the VWorks ActiveX control is used

In an Agilent Technologies automation system, the VWorks software runs in standalone mode, and the ActiveX control is not used. The operator uses the VWorks software, which is already configured to control the devices in the system. However, some integrations, such as those with LIMS, require that a third-party application control the VWorks software. The VWorks ActiveX control enables third-party applications to interface with the VWorks software. Through the ActiveX control, the third-party application can create an instance of the VWorks software, schedule protocol runs, and display various VWorks dialog boxes.

Each ActiveX control consists of a collection of the following:

- *Methods*. Functions that can be called to invoke individual operations
- *Properties*. Attributes or features of the ActiveX control
- *Events*. Notifications that methods have completed or resulted in errors

To ensure proper integration, you must know the available methods and properties for the ActiveX control.

The following diagram illustrates the use of the VWorks ActiveX control in a lab automation system environment. Actions you perform are conducted through ActiveX methods. System responses are relayed back through ActiveX events.

Note: Although the VWorks ActiveX control generates events, the third-party application must implement handlers for them.



Integrating the VWorks ActiveX control

When integrating the VWorks ActiveX control in a third-party lab software:

- 1 Install the VWorks ActiveX control. To install the VWorks ActiveX control:

- a** Insert the VWorks software CD into the controlling computer CD-ROM drive.
 - b** In the CD folder, double-click VWorks Installer.exe.
 - c** Follow the directions in the installation wizard window.
- 2** Open the Command Prompt window and type VWorks/register to register the application program interface.
- 3** Refer to the description of the Methods and Properties in this section.

Related information

For information about...	See...
VWorks ActiveX methods	“Methods” on page 4
VWorks ActiveX events	“Events” on page 16
VWorks ActiveX enumerated types	“Enumerated types” on page 20

Methods

AbortProtocol

Description

Aborts the protocol run that is in progress.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode = oVWorks4COM.AbortProtocol();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode = oVWorks4COM.AbortProtocol()
```

CloseProtocol

Description

Closes the specified protocol file.

Parameters

Name	Type	Description
protocol	BSTR	The protocol file path.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20. <i>Note:</i> returnCode is RETURN_SUCCESS if the file closed successfully.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode = oVWorksCOM.CloseProtocol("myprotocol.pro");
```

Visual Basic .NET example

```
Dim vwRetCode As VWorks4Lib.V11ReturnCode
vwRetCode = oVWorksCOM.CloseProtocol("myprotocol.pro")
```

CompileProtocol

Description

Compiles the protocol and is used with the LogMessage event.

Parameters

Name	Type	Description
protocol	BSTR	The protocol file path.
errorCount	*LONG	The number of errors found.
warningCount	*LONG	The number of warnings found.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See "Enumerated types" on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
LONG errCount, wrnCount;
retCode = oVWorks4COM.CompileProtocol ("c:\\myprotocol.pro",
&errCount, &wrnCount);
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
Dim errCount, wrnCount as Long
retCode = oVWorks4COM.CompileProtocol ("c:\\myprotocol.pro",
errCount, wrnCount)
```

EnumerateUsers

Description

Returns the list of users with VWorks accounts.

Parameters

None.

Returns

Name	Type	Description
user	*VARIANT	The user name.
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual Basic .NET example

```
Dim oEnumerateUsers As Object = Nothing
Dim retCode As VWorks4Lib.V11ReturnCode
Dim sUsers As String = ""

retCode = oVWorksCOM.EnumerateUsers(oEnumerateUsers)

If Not (oEnumerateUsers Is Nothing) Then
    Dim i As Integer
    For i = 0 To oEnumerateUsers.GetLength(0) - 1
        sUsers = sUsers & oEnumerateUsers(i) & " , "
    Next
End If
```

GetSimulationMode

Description

Gets the simulation mode state.

Parameters

None.

Returns

Name	Type	Description
mode	VARIANT_BOOL	The value that indicates the simulation state: • True = The simulation mode is on. • False = The simulation mode is off.

Visual C++ example

```
VARIANT_BOOL bSimMode;  
bSimMode= oVWorksCOM.GetSimulationMode();
```

Visual Basic .NET example

```
Dim bSimMode as Boolean  
bSimMode= oVWorksCOM.GetSimulationMode()
```

GetTipStates

Description

Gets the state of the tipboxes in a protocol for automated tip tracking.

Parameters

Name	Type	Description
protocol	BSTR	The protocol file path.

Returns

Name	Type	Description
TipStateXML	BSTR	The current status of the tipboxes.
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual Basic .NET example

```
Dim TipStateXML As String = ""  
Dim retCode As VWorks4Lib.V11ReturnCode  
retCode = oVWorksCOM.GetTipStates("c:\myprotocol.pro",  
TipStateXML)
```

LoadProtocol

Description

Loads the protocol for a run.

Parameters

Name	Type	Description
protocol	BSTR	The protocol file path.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode=oVWorks4COM.CompileProtocol ("c:\\myprotocol.pro");
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode=oVWorks4COM.CompileProtocol ("c:\\myprotocol.pro")
```

LoadRunsetFile

Description

Loads the runset file.

Parameters

Name	Type	Description
runset	BSTR	The runset file path.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode=oVWorks4COM.LoadRunsetFile ("c:\\myrunset.rst");
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode=oVWorks4COM.LoadRunsetFile ("c:\\myrunset.rst")
```

Login

Description

Logs into VWorks software using the provided user name and password.

Parameters

Name	Type	Description
userName	BSTR	The user name.
password	BSTR	The password.

Returns

Name	Type	Description
loginResult	V11LoginResult	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11LoginResult retCode;
loginResult= oVWorksCOM.Login("user1","mypassword!");
```

Visual Basic .NET example

```
Dim loginResult as VWorks4Lib.V11LoginResult
loginResult= oVWorksCOM.Login("user1","mypassword!")
```

Logout

Description

Logs out the current user session.

Parameters

None.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode = oVWorksCOM.Logout();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode = oVWorksCOM.Logout()
```

PauseProtocol

Description

Pauses the protocol run that is in progress. The tasks that are in progress will be finished. No new tasks will be started.

Parameters

None.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode = oVWorksCOM.PauseProtocol();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode = oVWorksCOM.PauseProtocol()
```

ReinitializeDevices

Description

Reinitializes devices.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode=oVWorks4COM.ReinitializeDevices ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode=oVWorks4COM. ReinitializeDevices ()
```

ResumeProtocol

Description

Resumes the protocol run.

Parameters

None.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode = oVWorksCOM.ResumeProtocol ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode = oVWorksCOM.ResumeProtocol ()
```

RunProtocol

Description

Starts the protocol run.

Parameters

Name	Type	Description
protocol	BSTR	The protocol file path.
runCount	LONG	The number of times to run the protocol.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode=oVWorks4COM.RunProtocol ("c:\\myprotocol.pro",2);
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode=oVWorks4COM.RunProtocol ("c:\\myprotocol.pro",2)
```

SetSimulationMode

Description

Sets the simulation mode.

Parameters

Name	Type	Description
mode	VARIANT_BOOL	The value that sets the simulation state: - True = Turns on the simulation mode. - False = Turns off the simulation mode.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20. <i>Note:</i> returnCode is always RETURN_SUCCESS.

Visual C++ example

```
oVWorksCOM.SetSimulationMode(VARIANT_TRUE);  
oVWorksCOM.SetSimulationMode(VARIANT_FALSE);
```

Visual Basic .NET example

```
oVWorksCOM.SetSimulationMode(True)  
oVWorksCOM.SetSimulationMode(False)
```

ShowDiagsDialog

Description

Displays the device diagnostics dialog box.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode=oVWorks4COM.ShowDiagsDialog ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode=oVWorks4COM. ShowDiagsDialog ()
```

ShowLoginDialog

Description

Displays the User Authentication (or login) dialog box.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode=oVWorks4COM.ShowLoginDialog ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode=oVWorks4COM. ShowLoginDialog ()
```

ShowManageUserDialog

Description

Displays the User Management dialog box.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode=oVWorks4COM.ShowManageUserDialog ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode=oVWorks4COM. ShowManageUserDialog ()
```

ShowOptionsDialog

Description

Displays the Options dialog box.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode=oVWorks4COM.ShowOptionsDialog ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode=oVWorks4COM. ShowOptionsDialog ()
```

ShowPlateGroupEditorDialog

Description

Displays the Plate Group Editor dialog box.

Parameters

None

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;  
retCode=oVWorks4COM.ShowPlateGroupEditorDialog ();
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode  
retCode=oVWorks4COM. ShowPlateGroupEditorDialog ()
```

ShowTipStateEditor

Description

Displays the Tip State Editor dialog box.

Parameters

Name	Type	Description
Protocol	BSTR	The protocol file path.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode=oVWorks4COM.ShowTipStateEditor
("c:\\myprotocol.pro");
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode=oVWorks4COM. ShowTipStateEditor
("c:\\myprotocol.pro")
```

ShowVWorks

Description

Displays or hides the VWorks software window.

Parameters

Name	Type	Description
showOrHide	VARIANT_BOOL	The value that displays or hides the window: - TRUE = Display the window. - FALSE = Hide the window.

Returns

Name	Type	Description
returnCode	V11ReturnCode	See “Enumerated types” on page 20.

Visual C++ example

```
VWorks4Lib.V11ReturnCode retCode;
retCode=oVWorks4COM.ShowVWorks (VARIANT_TRUE);
```

Visual Basic .NET example

```
Dim retCode as VWorks4Lib.V11ReturnCode
retCode=oVWorks4COM. ShowVWorks (True)
```

Related information

For information about...	See...
VWorks ActiveX control	“About the VWorks ActiveX control” on page 2
VWorks ActiveX events	“Events” on page 16
VWorks ActiveX enumerated types	“Enumerated types” on page 20

Events

LogMessage

Description

The LogMessage event occurs every time a message is posted to the log.

Parameters

Name	Type	Description
session	LONG	The session ID.
logClass	LONG	The type of log event: Error, Warning, or Event.
timeStamp	BSTR	The time at which the error occurred.
device	BSTR	The device name. An empty string is permitted.
location	BSTR	The location. An empty string is permitted.
process	BSTR	The process plate name. An empty string is permitted.
task	BSTR	The task name. An empty string is permitted.
fileName	BSTR	The protocol file or device file name. An empty string is permitted.
message	BSTR	The the error message text.

Returns

None.

MessageBoxAction

Description

The MessageBoxAction occurs when a user response is required.

Parameters

Name	Type	Description
caption	BSTR	The title of the message.
message	BSTR	The message body.
session	LONG	The session ID.

Name	Type	Description
type	LONG	The type of message: 0 = MB_OK 1 = MB_OKCANCEL 2 = MB_ABORTRETRYIGNORE 3 = MB_YESNOCANCEL 4 = MB_YESNO 5 = MB_RETRYCANCEL

Returns

Name	Type	Description
actionToTake	LONG*	The value that indicates the action to take: 1 = OK 2 = CANCEL 3 = ABORT 4 = RETRY 5 = IGNORE 6 = YES 7 = NO

ProtocolComplete

Description

The ProtocolComplete event occurs after startup, cleanup, and main protocols are finished.

Parameters

Name	Type	Description
session	LONG	The session ID.
protocol	BSTR	The protocol file path.
protocol_type	BSTR	The type of protocol: Startup, Main, or Cleanup.

Returns

None.

ProtocolAborted

Description

The ProtocolAborted event occurs when the operator or automation client aborts the protocol run.

Parameters

Name	Type	Description
session	LONG	The session ID.
protocol	BSTR	The protocol file path.
protocol_type	BSTR	The type of protocol: Startup, Main, or Cleanup.

Returns

None.

RecoverableError

Description

The RecoverableError event occurs whenever an error is displayed and expects the operator to abort, retry, or ignore the error.

Parameters

Name	Type	Description
session	LONG	The session ID.
device	BSTR	The device name. An empty string is permitted.
location	BSTR	The location. An empty string is permitted.
description	BSTR	The description of the error.

Returns

Name	Type	Description
actionToTake	*LONG	The value that indicates the action to take: 0 = Abort 1 = Retry 2 = Ignore

Name	Type	Description
vworksHandlesError	*VARIANT_BOOL	Value values are: - TRUE = Allows the VWorks software to handle the error. the VWorks software will not display the error message. - FALSE = Prevents the VWorks software from handling the error.

UnrecoverableError

Description

The UnrecoverableError event occurs when an error is displayed and does not expect the operator to respond with a decision.

Parameters

Name	Type	Description
session	LONG	The session ID.
description	BSTR	The description of the error.

Returns

None.

UserMessage

Description

The UserMessage event occurs when a User Message task occurs.

Parameters

Name	Type	Description
session	LONG	User message tasks can prompt user for data entry.
caption	BSTR	
message	BSTR	
wantsData	VARIANT_BOOL	

Returns

Name	Type	Description
userData	*BSTR	Allows user to enter data if wantsData = True.

Related information

For information about...	See...
VWorks ActiveX control	“About the VWorks ActiveX control” on page 2
VWorks ActiveX methods	“Methods” on page 4
VWorks ActiveX enumerated types	“Enumerated types” on page 20

Enumerated types

V11ReturnCode

Description

Indicates the method call status.

Constants

Name	Value	Description
RETURN_SUCCESS	0	The method was called successfully.
RETURN_BAD_ARGS	1	The method contains bad arguments.
RETURN_FAIL	2	The method call failed.

V11LoginResult

Description

Indicates the login status.

Constants

Name	Value	Description
LOGIN_SUCCESS	0	The login was successful.
LOGIN_FAIL	1	The login failed.
LOGIN_DISABLED	2	The login was disabled.
LOGIN_EXPIRED	3	The login period passed.
LOGIN_TOO_MANY_FAILED_ATTEMPTS	4	Too many login attempts were made and failed.

Name	Value	Description
LOGIN_NOT_ENOUGH_AUT HORIZATION	5	Higher access priveleges are required to perform the requested action.

Related information

For information about...	See...
VWorks ActiveX control	“About the VWorks ActiveX control” on page 2
VWorks ActiveX methods	“Methods” on page 4
VWorks ActiveX events	“Events” on page 16

