GraphWorX32 – Getting Started with Scripting

APPLICATIONS NOTE

October 2013

Description: Guide to scripting in GraphWorX32.
OS Requirement: Win 2000, XP Pro, Server 2003, Vista, Server 2008, Windows 7
General Requirement: Familiarity with GraphWorX32 and scripting in general.

Introduction

Automation can be a powerful tool to expand the abilities of your application beyond the standard functions of GraphWorX32. You can find examples of scripting within GraphWorX32 in the **C:\Program Files\ICONICS\GENESIS32\Examples\GraphWorX32\OLE Automation** directory in a standard GENESIS32 installation. This document will introduce you to the basics of scripting in GraphWorX32 and some commonly used automation calls.

Runtime and Configuration Scripts

In GraphWorX32, you can execute scripts both from runtime and from configure mode. Runtime scripts are the most common. To create a runtime script, follow the steps below.

1. Add a button to a new GraphWorX32 display.
2. Select "Run Script" as the Action.
3. Hit the "Create" button, type "Runtime" into the Script Name field, then hit OK.
4. Between the "Sub Runtime" line and the "End Sub" line, type this simple script:

```
MsgBox "Hello World"
```

5. Close the Microsoft Visual Basic window and put the display into runtime. Click the button to see your message box.

Configuration mode scripts are often used to create wizards that can build displays automatically. There are examples of such wizards in the "3_ScriptWizards" symbol library. (To open the symbol library from GraphWorX32, go to View → Toggle Symbol Toolbar.) You can drag these symbols onto your display and double-click them in configuration mode to create a variety of objects.

To create a configuration mode script, follow the steps below.

1. Add a rectangle to your display.
2. Right-click on the rectangle and select "Create VBA Script".
3. In the Script Name field, type "Configure", then hit OK.
4. Between the "Sub Configure" line and the "End Sub" line, type this simple script:

```
MsgBox "Hello World"
```

5. Close the Microsoft Visual Basic window and double-click on the rectangle object while still in configure mode. Your message will appear.

NOTE: The rest of the examples in this application note will be runtime scripts.

Reading from and Writing to a Process Point

The simplest way to read from or write to an OPC tag is to use automation for the GraphWorX32 Process Point object. This method lets GraphWorX32 do all of the work actually getting the value of a tag. This script will simply manipulate the value that GraphWorX32 has already received from the OPC server.

1. We need to add a process point to our display before we write any scripts. In GraphWorX32, go to Dynamics → Intrinsics → Process Point… or click on the button in the dynamics toolbar. Click on your display to add the process point.

   NOTE: The process point needs to exist on the display for this script to work, but it needs not to be visible to your users. You can put the point on a hidden layer, behind another object, or even use a hide dynamic if desired.

2. Click on the "Data Tags" button and select the OPC tag that you would like to manipulate with your script. When you are done configuring your process point, click on "OK".

   NOTE: You can use local variables too!

3. In your script, use the "GetPointObjectFromName" method to create a link to the process point we just created. The path to your tag in the script must match the path to your tag in the process point exactly when in runtime. In most cases, it's easiest to copy the text from the Data Source field of the process point and paste it into your script. Here is an example:

```
Set ppt =
ThisDisplay.GetPointObjectFromName("ICONICS.Simula
tor.1\SimulatePLC.OUTPUTS.FLOAT")
```

   NOTE: The code above must be all on one line to be properly formatted. If you are using aliases, you cannot simply copy and paste. If you have aliases in your process point, you must resolve those aliases within the script as well, or the script's path will not match the process point's path.

4. Now in your script "ppt.Value" will equal the value of the process point on your display. It is writeable, so you may set "ppt.Value" equal to some number if your script

requires it.
Here is the example extended:

```
Dim x as Integer
Dim y as Integer

Set ppt =
ThisDisplay.GetPointObjectFromName("ICONICS.Simula
tor.1\SimulatePLC.OUTPUTS.FLOAT")

x = ppt.Value       'store the value
x = 10 + x          'modify the value
ppt.Value = x       'write the value back
```

How to Use Automation with an ActiveX

There are many useful automation methods included with ActiveX controls, but how to create an object from an ActiveX to use such methods is not always obvious. Here is the proper way to create a link to your ActiveX objects in your script.

NOTE: This method will work for both, ICONICS ActiveX controls and third-party ActiveXs.

1. Add the desired ActiveX to your display.
2. In configure mode, right-click on the ActiveX and select "Application Property Inspector".

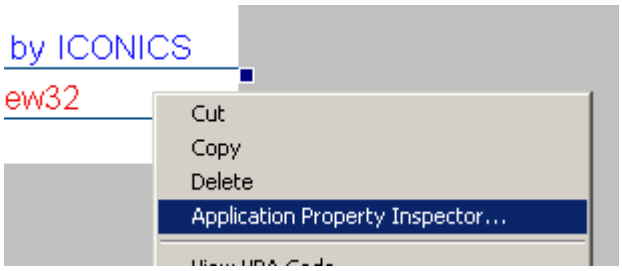

Figure 1 - Application Property Inspector

3. In the Object Name field, give your ActiveX a name. We will use this name in our script to get this ActiveX specifically.

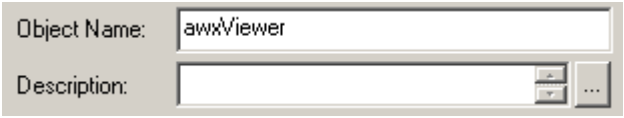

Figure 2 - Object Name

4. Now we can use the "GetVisibleObjectFromName" and "GetOleObject" methods to get the OLE object from the display. In the example below, any automation methods for the ActiveX can be called from oleObject.

```
Set oleObject = ThisDisplay.GetVisibleObjectFromName(
"awxViewer").GetOleObject
```

NOTE: The code above must be all on one line to be properly formatted.

5. There are many helpful examples of scripting using ActiveX automation in the Symbols Toolbar. Open the symbols toolbar by going to View → Toggle Symbol Toolbar or clicking on the button in the Draw toolbar. The ActiveX scripts can be found in 3_ScriptWizards/VB Script Wizards.sdf. Drag a wizard button onto you display and double-click it to generate the example scripts.

NOTE: For more information on the Symbols Toolbar, please refer to the application notes *GraphWorX32 – Using the Symbols Library* and *GraphWorX32 – Script Wizard Help*.

Discovering Automation Methods

A valuable resource for scripting is the Object Browser feature of the Visual Basic Editor. Whether you are using VBScript or VBA, the Object Browser can be used to look up all the available methods for a given object.

1. Open the Visual Basic Editor from GraphWorX32 by going to Tools → Macros → Visual Basic Editor or hitting Alt-F11.
2. Open the Object Browser by going to View → Object Browser, hitting F2, or clicking on the button in the toolbar.
3. From here you can browse or search all of the classes and methods available to you in your current libraries. To limit your search to a specific library, select it from the top drop-down list. If a library you want to look at is not listed, you can add it by going to Tools → References. The libraries for an ActiveX will be added automatically if the ActiveX is added to your display.

NOTE: Most, but not all automation methods listed in the Object Browser are also available in VBScript.

4. Another useful tool for finding automation methods is the *GENESIS32 - OLE Automation References* document, located on your installation disc or in the Documentation section of our website.
5. You can also find some helpful tips in the *GenBroker – OLE Automation Support* Application Note, also located on your CD or in the Application Notes section of our website.