



# GraphWorX64 Scripting Quick Start



APPLICATION NOTE

September 2013

**Description:** This Application Note will introduce GENESIS64 Scripting and explain how to perform some basic operations.

**OS Requirement:** Windows 8 x64 (Professional or Enterprise Edition) /Windows 7 x64 (Professional or Enterprise Edition) /Windows Server 2012 x64 /Windows Server 2008 R2 x64 /Windows Vista x64 SP2 (Business, Ultimate, or Enterprise Edition) /Windows Server 2008 x64

**General Requirement:** GENESIS64 installed; basic knowledge of GraphWorX64.

## Introduction

GraphWorX64 contains a powerful scripting engine which utilizes the JScript.NET language. This language combines the ease of use and versatility of a scripting language with the power of Microsoft .NET Framework.

It is possible to edit your scripts directly from inside GraphWorX64 and have the code run on specific events, like a button press or when the display has finished loading.

Some of the tasks you can achieve by using GraphWorX64 Scripting are:

- Create new elements in the display
- Alter the appearance of the display
- Add objects to the 3D view
- Rotate a 3D views
- Create pick actions on the fly
- Interact with other controls (AlarmWorX Viewer, EarthWorX Viewer, TrendWorX Viewer, etc)

**NOTE:** Changes done to the display through scripting while in Runtime mode will persist when you switch the display back to Configuration mode.

## Standard or Advanced Mode

Before we begin scripting, be sure you have set your Application Mode to Standard or Advanced. You will not be able to access the Events section if you are in Basic Mode.

To do this, go to the View menu, pick the Application Mode dropdown, and select Standard Mode or Advanced Mode.

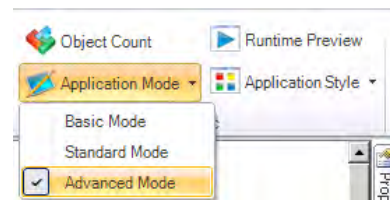


Figure 1 - Application Mode

## Creating a Script on a Button

This example will show you how to create a simple script that is executed on a button press.

1. Start GraphWorX64 and create a new blank display.
2. Add a button the display.
3. In the Dynamics window, change the Command from Load Display to Run Script.

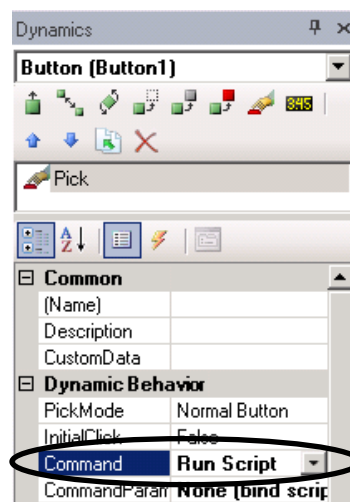


Figure 2 - Run Script Command

4. Clicking the lightning bolt icon in the Dynamics window to go to the Events section.

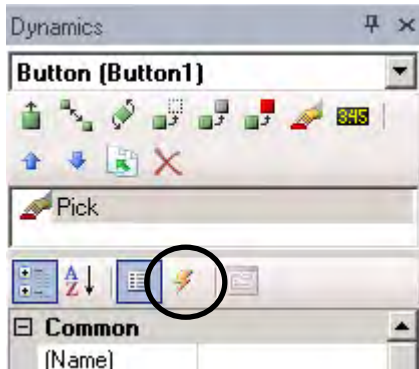


Figure 3 - Events

- Double-click in the empty area to the right of the RunScript event and click OK to the Script Options dialog that appears. This will create a new script function that will be executed when this button is clicked. It will also open the Script Editor.

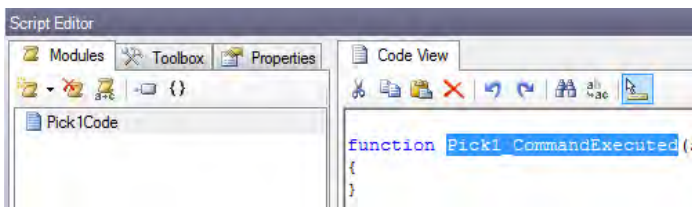


Figure 4 - Script Editor

- Between the curly brackets, type this line of code:  

```
MessageBox.Show("Hello World!");
```
- Put the display into runtime and click your button. A "Hello World!" message box will appear.

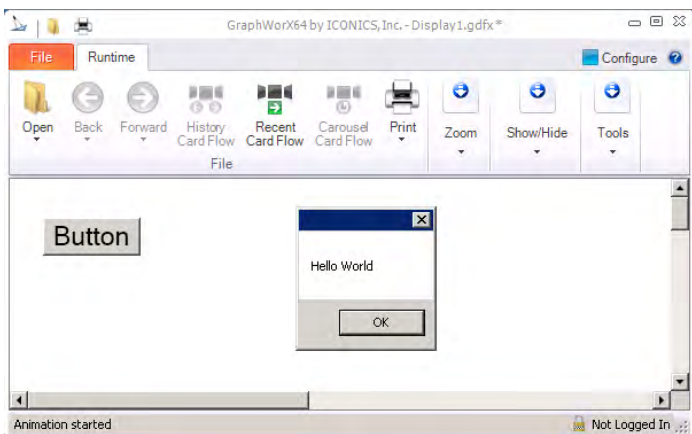


Figure 5 - Scripting Running

## Creating a Script on an Event

This example will show you how to create a simple script that is executed when the GraphWorX64 display loads. This example uses the LoadCompleted event, but a GraphWorX64 display may execute many other events that you can use to execute your code.

- Make sure you are in Configure Mode. Right-click on an empty space in your GraphWorX64 display and select Object Properties. This will open the properties window for the GraphWorX64 display itself.
- Click on the lightning bolt icon to go to the Events tab.
- Double-click in the blank area on the right of the LoadCompleted event. This will create a new script function that will be executed when this display has finished loading. It will also open the Script Editor.
- Between the curly brackets, type this line of code:

```
MessageBox.Show("Done Loading!");
```

- Save the display, close it, and open it again. As soon as the page has finished loading you should see your message box.

## Running a Script in Configure Mode

Occasionally you will want to run a script in configure mode. This can be useful for scripts that are designed to build a display.

- Add a rectangle to your display.
- Right-click on the rectangle and select "Group". This will convert the rectangle to a symbol.
- Right-click on the rectangle again and select "Convert to Smart Symbol". Click "OK" to the two dialogs that appear. Our rectangle is now a smart symbol.
- With the rectangle smart symbol selected, click on the lightning bolt icon in the Properties window to go to the Events tab.
- Double-click in the blank area on the right of the DesignModeDoubleClick event. This will create a new script function that will be executed when the smart symbol has been double-clicked. It will also open the Script Editor.

**NOTE:** The DesignModeDoubleClick event is only available for smart symbols.

- Between the curly brackets, type this line of code:

```
MessageBox.Show("Hello World!");
```



7. Save the display. Without going into runtime, double-click on the rectangle smart symbol. You should see your message box.

### Reading a Value from an OPC Tag

One of the most common things to do in a GraphWorX64 script is to read from or write to an OPC tag. This example will show you a simple way to read from a tag.

1. Add a process point to your display and connect it to an OPC tag or simulator variable. In this example we will use the localsim:sine simulator variable.
2. In the (Name) field of the process point, type "PPT1". Your process point's Dynamics panel should now look like the figure below.

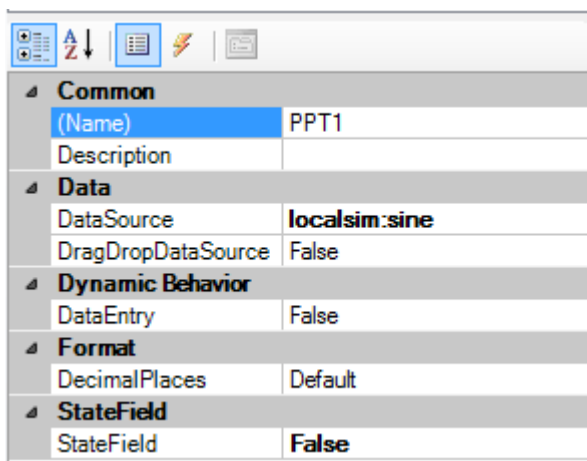


Figure 6 - Process Point Dynamics panel

3. Add a button to your display and configure it to run a script. (For more details, see the section above entitled "Creating a Script on a Button".)
4. In your button's function, add these lines:

```
var ppt =
ThisConfiguration.GetDynamicObjectByName("PPT1");
var pptVal = ppt.GetValueOfPrimaryDataSource();
MessageBox.Show(pptVal);
```

5. Go into runtime and click your button. A message box will appear with the value of the process point.

**NOTE:** Using a process point like this is the simplest way to manipulate an OPC value but it requires that a process point containing that OPC tag be on your display somewhere. If you do not want your users to interact with the process point object

you can hide it by adding a Hide dynamic to it, putting it on a hidden layer, or putting it behind another object.

### Writing a Value to an OPC Tag

This example will show you a simple way to write to an OPC tag.

1. Add a process point to your display and connect it to an OPC tag or simulator variable. For this example we will create a new simulator variable called Var1.
2. In the (Name) field of the process point, type "PPT2".
3. Add a button to your display and configure it to run a script. (For more details, see the section above entitled "Creating a Script on a Button".)
4. In your button's function, add these lines:

```
var ppt =
ThisConfiguration.GetDynamicObjectByName("PPT2");
ppt.WriteValueToPrimaryDataSource(5);
```

6. Go into runtime and click your button. Your process point's value will change to 5.

**NOTE:** As in the previous example, you can hide the process point by adding a Hide dynamic to it, putting it on a hidden layer, or putting it behind another object, but it must be on the display somewhere for the script to work.

**NOTE:** There is more than one way to read and write to OPC tags. See the application note entitled, *GraphWorX64 Scripting – Introduction to FrameWorX Client Wrapper* for another method that may work better for your application.

### Opening the Script Editor Window

If for some reason, the Script Editor does not appear or if you had accidentally closed it, you can always bring it up by selecting the "View" tab in the ribbon bar, and check the Scripts checkbox. You may also press Alt+F11 to show or hide it.

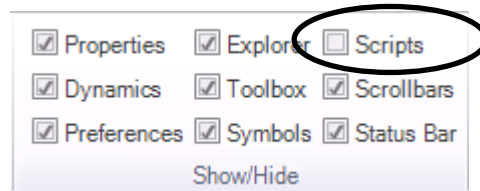


Figure 7 - Script Editor Visible Option



# GraphWorX64 Scripting Quick Start



APPLICATION NOTE

September 2013

## Conclusion

This was just a very brief introduction to scripting in GraphWorX64. Many more complex scripts can be written by using .NET Framework libraries and the GENESIS64 Scripting API.

A complete reference to the GENESIS64 API can be found in your Start Menu in the Programs group under ICONICS → GENESIS64 → Help. The file is called "GENESIS64 API Reference Help".

ICONICS Customer Connection Portal also hosts a library of automation examples that are very handy when you start scripting.