



GraphWorX64 Scripting

Local and Global Aliases



APPLICATION NOTE

September 2013

Description: Guide to accessing local and global aliases in scripts.

OS Requirement: Vista x64/ Windows 7 x64/ Windows 8 x64/ Windows Server 2008 x64/ Windows Server 2008 R2 x64/ Windows Server 2012.

General Requirement: Knowledge of Global and Local Aliasing, GraphWorX64 and JScript.NET basics.

Introduction

GraphWorX64 gives its users the opportunity to use JScript.NET scripting to perform a large variety of tasks. Skillful designers can enhance their operator displays with custom functionalities that are not included in the very extensive range of functions and features built into GraphWorX64.

In general, ICONICS recommends the use of scripting only when it is really necessary and the required function cannot be fulfilled by other means, such as standard GraphWorX64 features, Commanding, or independent custom components. A few reasons that other methods should be considered before using scripting in your display include:

- Scripting is not supported in Silverlight or HTML5 (AnyGlass) displays.
- Functions and methods in the scripting library may change over time and it may make future upgrades difficult.
- The GraphWorX64 Search and Replace feature does not search through the scripts.
- Scripts are stored in the display, not individual objects, so objects that require scripts cannot be stored in the Symbol Library.
- If there are multiple designers maintaining a project, scripts are more difficult to learn than built-in functionality.

Despite these drawbacks, scripting is an extremely powerful tool that can save time in creating exactly the functionality you need.

In this application note we will show you how to read and set local and global aliases within your scripts.

Local Aliases

The "GetAliasValue" and "SetAliasValue" methods can be used to read or write to local aliases. These methods are called from the ThisDocument object. Here is an example:

```
ThisDocument.SetAliasValue("MyAlias", "NewValue");  
MessageBox.Show(ThisDocument.GetAliasValue("MyAlias"));
```

You can also use these methods to get and set aliases with respect to individual objects. This is useful when you have the same alias set to different values on different objects. Below is an example of how to get the local alias associated with a particular label:

```
var obj = ThisDocument.GetElementByName("MyLabel");  
var aliasValue : String  
aliasValue = ThisDocument.GetLocalAliases(obj)["MyAlias"];  
MessageBox.Show(aliasValue);
```

Global Aliases

The "GetGlobalAliases" "SetGlobalAliases" methods can be used to read or write to global aliases. These methods are called from the ThisWindow object.

The string returned by GetGlobalAliases and used as the parameter for SetGlobalAliases needs to be in a specific format. This string can define the values for multiple aliases and themes at once, and can set the scope for those aliases. For best results, it's recommended that you add a pick action to your screen with a "Set Global Aliases" action, then configure it to set some of the aliases or themes you're interested in. Take a look at the string used in the GlobalAliases property of the button. This is the same format required by the GetGlobalAliases and SetGlobalAliases methods.

Here is an example of how to set the global alias Buildings theme to "BuildingB":

```
ThisWindow.SetGlobalAliases("Buildings=BuildingB;");
```

Load Local Aliases to Global Aliases

Here is an example you can use to load local alias values into global aliases:

```
var ServerAlias : String =  
    ThisDocument.GetAliasValue('Server');  
var DeviceAlias : String =  
    ThisDocument.GetAliasValue('Device');  
var GlobalAlias : String = '#Server=' + ServerAlias +  
    ';#Device=' + DeviceAlias + ';';  
  
ThisWindow.SetGlobalAliases(GlobalAlias);
```