



GENESIS32 - Dynamically Creating Global Aliases



APPLICATIONS NOTE

October 2013

Description: Guide to dynamically creating Global Aliases.

OS Requirement: Windows XP / Windows Vista / Windows 7 / Windows 8 / Windows Server 2003 / Windows Server 2008 / Windows Server 2008 R2 / Windows Server 2012

General Requirement: General Knowledge of ICONICS GENESIS32 applications.

Introduction

Aliases are a way of creating reusable displays for different sets of tags. The advantage of using aliases is that you could create one display and use it to display different information depending on the need.

Global Aliasing (GAS) uses a central database-driven engine to perform text substitution. During runtime you select a theme that represents the set of values you would like to use. Based on how they are configured and where you chose to use them in your displays the aliases will resolve into parts of OPC tags, whole OPC tags, or bits of description appropriate for the chosen theme.

In some cases you may need more values than the ones you specified in the database, or you may need whole new aliases. If you don't want to change the existing GAS configuration, you can dynamically change alias values, or even create new ones.

The Global Aliasing Engine (GASengine.exe) will start automatically as soon as a global alias is used in runtime. This engine must be running for any aliases to function (dynamic or preconfigured).

NOTE: All aliases and values you create on the fly with this app note will no longer exist once the engine stops. For more permanent changes use the Global Aliasing Configurator, located in the ICONICS Tools start menu.

Dynamic Aliases with a Pick Action

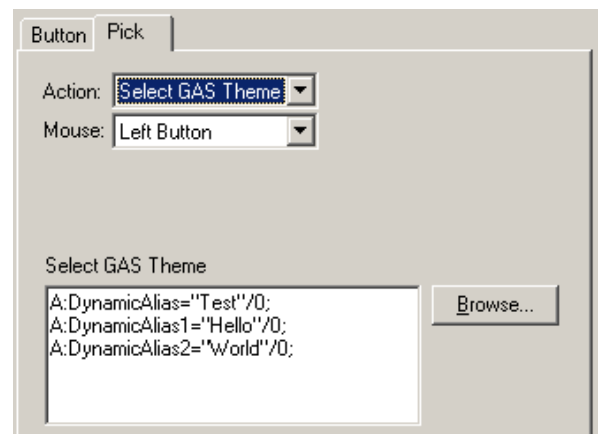
These steps will show you how to create dynamic global aliases. For the standard use and configuration of global aliases, please refer to the application note entitled, *GENESIS32 - Global Aliasing Quick Start*.

1. Open a GraphWorX32 display by going to Start → Programs → ICONICS GENESIS32 → GraphWorX32 → GraphWorX32
2. Insert a Text Object and type "<#DynamicAlias#>" (without quotes) as the text of the object.
3. Insert a Button Object and pick "Select GAS Theme" from the "Actions" drop down list.

4. Instead of browsing for a theme, simply type in your new global alias definition as shown below. This alias does not need to exist in your Global Aliasing Configuration.

A: AliasName=Value/Level

5. In the above example, replace *AliasName* with the name of the alias ("DynamicAlias" for our example) and *Value* with its new value. The "/Level" syntax allows you to specify the scope on which to set the alias. For more information about scopes in global aliasing, see the help file. In most cases level 0 (machine level) can be used.
6. If you want to set more than one alias on this pick action, put the next alias on a new line and separate them with semi-colons, as shown in Figure 1.



7. Figure 1 - Creating New Alias Values

8. Go into Runtime and click the button. You should see your alias be replaced by its new value.

Dynamic Aliases with Scripting

To perform the same action with VBA or VBScript you can use the "ThisDisplay.SetGASThemes" function. It takes a single string parameter which is the same list of aliases you would type into the "Select GAS Theme" pick action and returns true or false based on whether the action succeeded or not.

For example, use this code to set three new aliases (note that this code must be all on one line to be properly formatted):

```
ThisDisplay.SetGASThemes "A:DynamicAlias='Test'/'0;  
A:DynamicAlias1='Hello'/'0;A:DynamicAlias2='World'/'0;"
```