Unified Data Manager
Global Expressions

APPLICATION NOTE

September 2013

Description: Guide configuring Global Expressions in the Unified Data Manager.
OS Requirement: Vista x64/ Windows 7 x64/ Windows 8 x64/ Windows Server 2008 x64/ Windows Server 2008 R2 x64/ Windows Server 2012.
General Requirement: Installation of GENESIS64 v10.8, familiarity with the Expression Editor.

Introduction

The Unified Data Manager (UDM) provides a centralized management of commonly used expressions, groups, value sets, registers, and event triggers across the GENESIS64 suite. This document will help you understand global expressions from the configuration point of view and will demonstrate how they are used.

You can create expressions in the UDM and use them in other GENESIS64 applications. Each global expression is implemented as a single expression data tag, which recalculates on input values or on trigger data tags. Client applications can access these expression data tags as read/write data tags. Expressions can also accept parameters to provide further functionality.

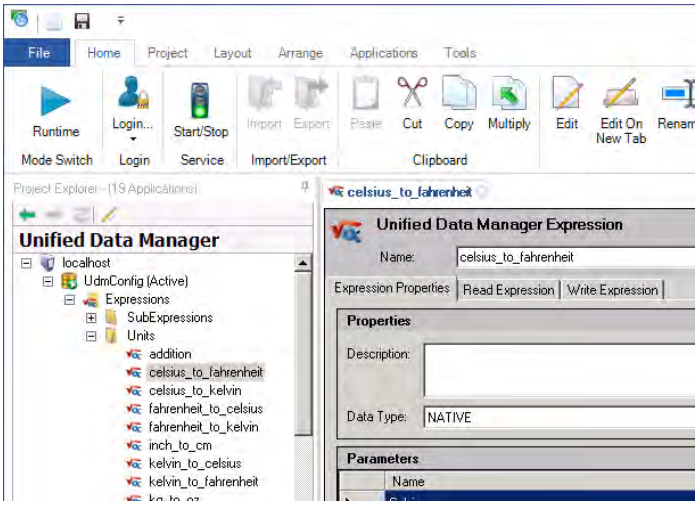

Figure 1 - Unified Data Manager with an Expression Configuration

Creating a New Expression

1. Launch Workbench by going to **Start → Programs → ICONICS → Workbench**, then go to the Unified Data Manager provider.
2. If desired, create a new configuration database by right clicking on your server at the root of the tree and select **New Database**.
3. Follow the Create Database Wizard to create a new configuration database. You can check or uncheck "Include Sample Data" checkbox.
4. You should now have a new configuration database. Right click it and select **Make Active**.
5. In the tree-control in the left-hand pane, expand your configuration, right-click on Expressions and select **+ Expression**.

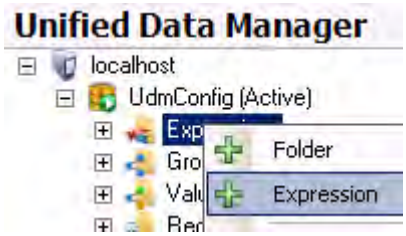

Figure 2 - Adding a new expression

6. Enter **TimesTen** in the Name field of the expression.

Simple Read-Only Expression

First, we will create a simple expression that multiplies a value by ten.

1. Once you have created the expression, select the **Read Expression** tab and click **Expression Editor** button.
2. Open the Expression Editor and enter the following expression. You can use the "Variables" button to browse for the Simulations → Float → Ramp → FastL000 tag, then multiply it by 10:

{{@sim64:Float.Ramp(1,-50.0,50.0,0).Value}} * 10

3. Click **OK** to the expression editor, then click **Apply** to save your changes.

Using the Expression

1. Launch GraphWorX64 by going to **Start → Programs → ICONICS → GENESIS64 → GraphWorX64 → GraphWorX64.**
2. Insert a Process Point and open the Data Browser.
3. Navigate to **OPC Classic Servers → My Server → Global Expressions** and select the expression you just created.

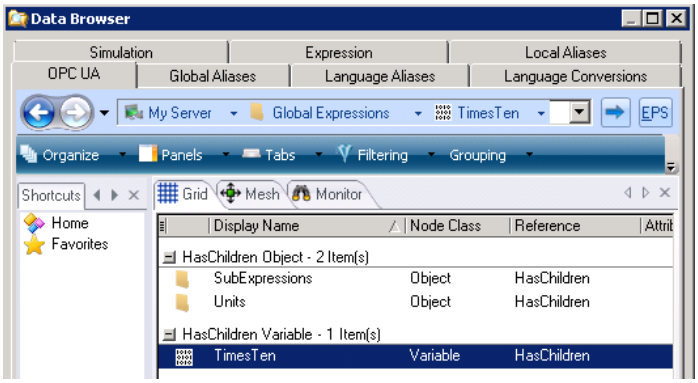

Figure 3 - Selecting an Expression from Data Browser

4. Enter runtime and you will see results of the read expression we configured.

Using Parameters

Multiplying a single simulation tag by 10 isn't very useful, so let's change the expression to use a parameter. By using a parameter, we can use one single expression to multiply any value by 10.

1. In the Unified Data Manager provider of Workbench, create a new expression just like your **TimesTen** expression, but name it **TimesTenParameter**.
2. Go to the **Expression Properties** tab.
3. Add a parameter named "tag". Select **String** for the type and leave the rest of the values at their defaults. Hit **Apply**.
4. Go to the **Read Expression** tab. Replace the bit inside the curly brackets with "<<tag>>" (no quotes). The resulting expression should look like:

x={{<<tag>>}} * 10

5. Click **Apply**.

Using the Expression

1. In GraphWorX64, add a new process point and browse for your **TimesTenParameter** expression again.
2. Select the **TimesTenParameter** expression in the browser, but do not double-click it, and do not click OK yet. Click in the address bar of the data browser, and you should see your parameter in the string, as in Figure 4.

   NOTE: If you do not see the "tag" parameter, click the refresh button in the Data Browser and select the tag again.

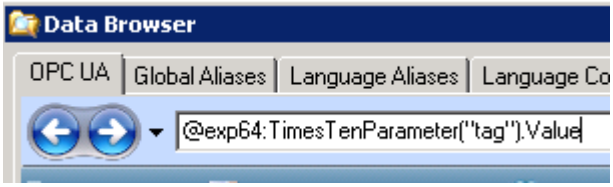

Figure 4 - Expression with parameter

3. Replace the "tag" value with the name of some DA point. It should look something like:

@exp64:TimesTenParameter("@ICONICS.AlarmSvr_.1\Area_Active_Count.Value").Value

4. Go into runtime. You should see the value of the tag you put in place of the parameter, multiplied by ten.
5. You can add multiple copies of this same expression in different process points on your display, specifying a different tag in each one, and each tag will be multiplied by ten.

Write Expression

Now, what if we want to write back to our expression and have the value we write multiplied by ten? This can be used for unit conversion, where the operator wants to see the value in one unit (inches, for example) but the PLC measures the value in another unit (centimeters). The expression will take the value from the PLC, convert it for the operator to read, then take the value the operator writes and convert it back before it writes it to the PLC.

1. Create a new expression identical to **TimesTenParameter**, but name it **TimesTenWrite**.
2. Go to the **Write Expression** tab and check the **Enable Write Expression** check box.
3. Click the **Expression Editor** button.
4. In the Expression Editor, type the following expression:

{{input}} / 10

   {{input}} is a keyword to represent the value written by the operator. This expression will take that input and divide it by 10, which is the opposite of our read expression. Click **OK** to the expression editor.
5. In the Output Tag section, type **<<tag>>**. The value calculated by the write expression will be written to the Output Tag, which in this case is our "tag" parameter.

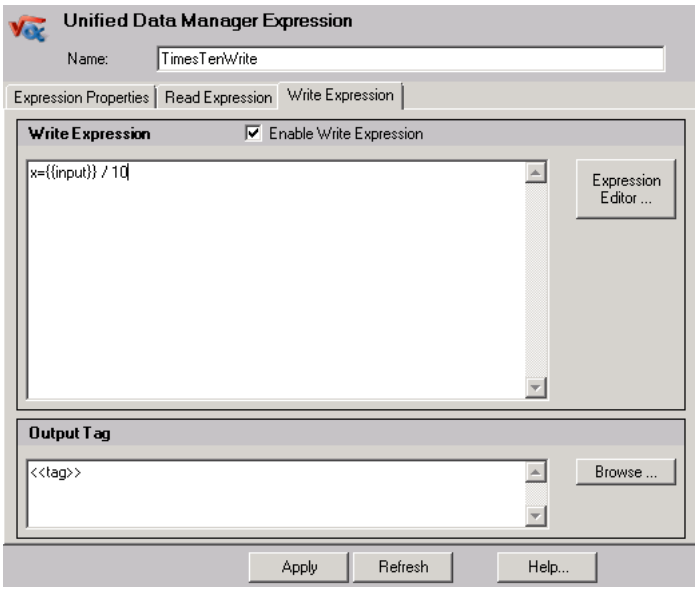

Figure 5 – Write Expression

6. Click **Apply** when done.

NOTE: You do not have to use a parameter here. You can browse for a specific point if you want. Keep in mind that the **Output Tag** can be only a **single point.** A single expression cannot write to more than one tag.

Using the Expression

This tag can be used exactly like **TimesTenParameter** in our previous example. Make sure to choose a writeable tag for your tag parameter value, and that you set the process point to be Data Entry. In runtime, you should be able to write to the expression and see the value you wrote divided by ten get written to your parameter tag.

We have just gone through some of the basic steps in utilizing global expression. For more information on using global expressions in the Unified Data Manager, please refer to the Help documentation.