
SUMMARY

This Tech Note will demonstrate some techniques that can be used to modify and improve existing ArcestrA symbols. This is not intended to be an exhaustive list but rather a series of examples of things that can be done to pre-built Wonderware symbols that will improve their usability and give some insight into how ArcestrA symbols function.

SITUATION

Frequently Asked Questions:

Who would want to do this?

Anyone who is using existing ArcestrA graphic symbols but isn't entirely happy with some aspects of them but doesn't know how to modify them.

In what versions of Application Server can this be done?

The screenshots for this TechTip are taken from Application Server 17.3.1. However, most of these techniques should work (possibly with some cosmetic differences) on System Platform 2012 or later.

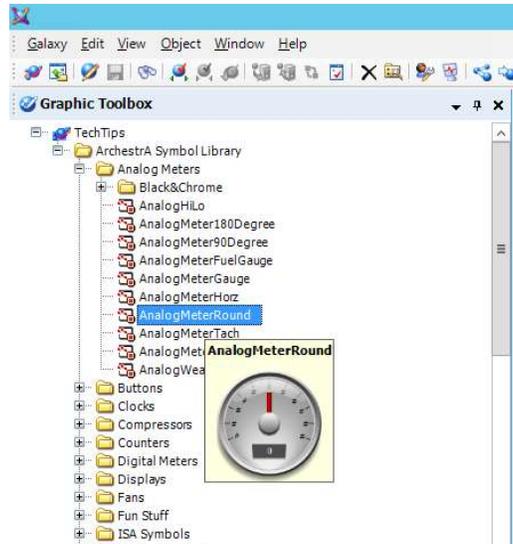
Is any special knowledge required before starting these exercises?

Anyone with a basic knowledge of Wonderware Application Server should be able to follow this TechTip.

Modifying existing ArchestrA symbols.

Scenario: There is an existing ArchestrA system you want to use but it isn't quite what you want and wish you could modify it.

In the following examples, the “AnalogMeterRound” symbol will be used.

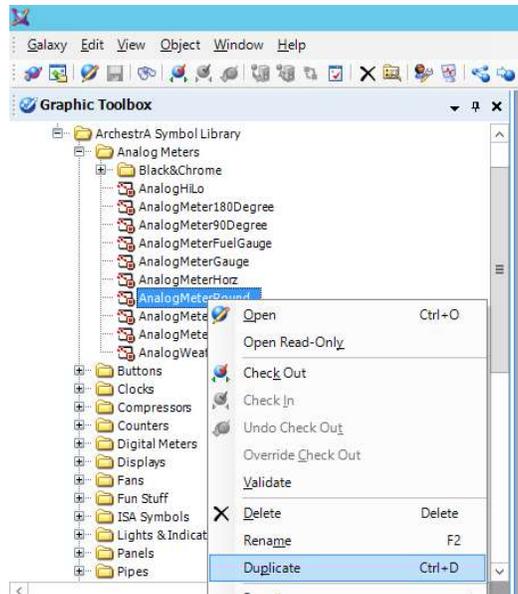


However, the techniques shown in this document are applicable to many of the other pre-built objects.

In the case of the AnalogMeterRound, four changes will be made:

- A. The color of the needle will be changed from red to blue.
- B. The translucent, reflective component at the top of the dial meant to simulate reflection on a glass face will be removed making it easier to read.
- C. The value display will be modified to show floating-point values rather than integers as the object does by default
- D. A display for engineering-units will be added along with the corresponding custom property.

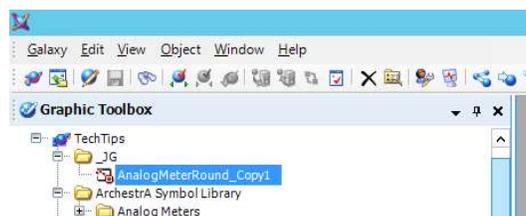
1. Duplicate the symbol to be modified by right-clicking on it and choosing the “Duplicate” option.



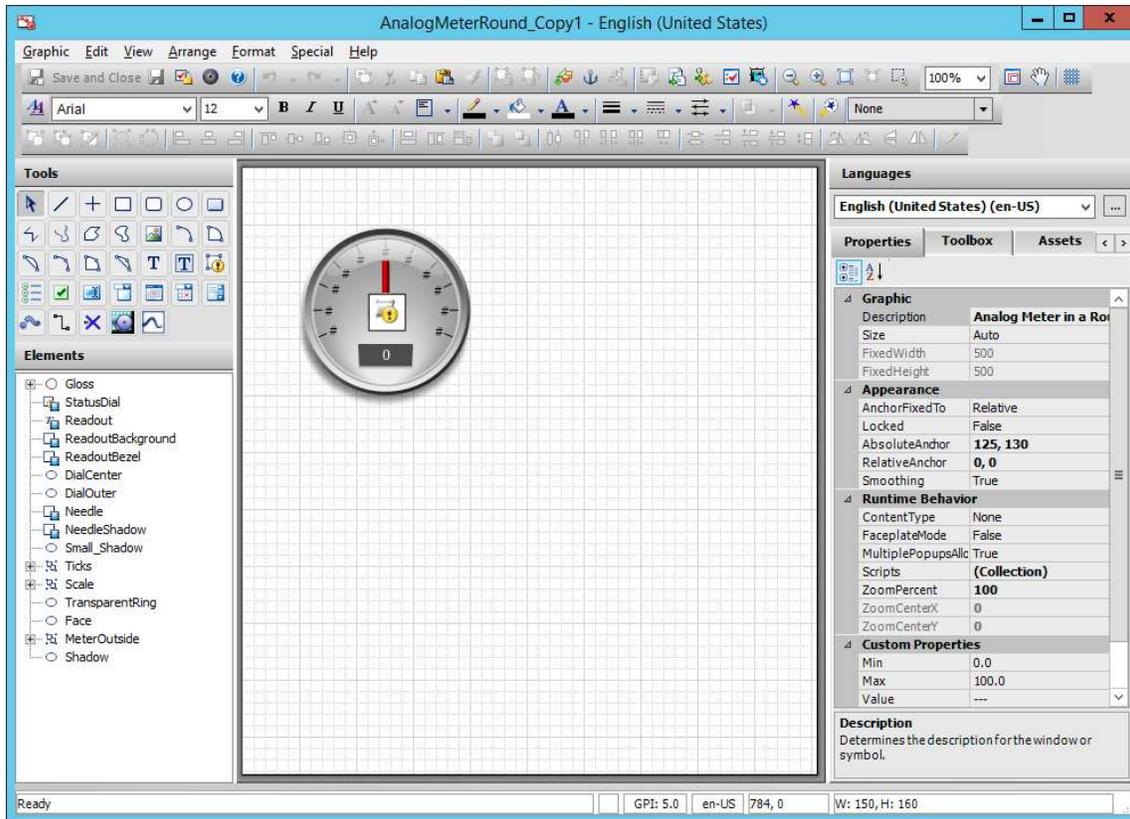
While it is technically possible to edit the original copy of the symbol, this is a potentially dangerous practice that might create difficulties in the future. The best practice is to make a duplicate of the original object and then modify that.

Another good practice is to create a separate “Graphic Toolset” by right-clicking on the galaxy name in the Graphic Toolbox and selecting “New > Graphic Toolset”.

In the example below, the duplicate object has been dragged to a new graphic toolset called “_JG”. The underscore character is used to place it alphabetically at the top. The rest of the name can be whatever you like.

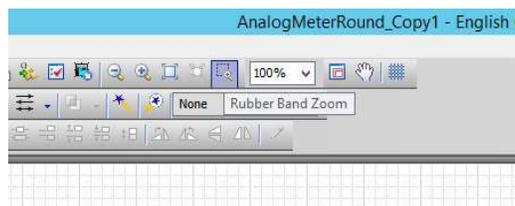


2. Double-click on the duplicate to open the ArchestrA graphic editor.

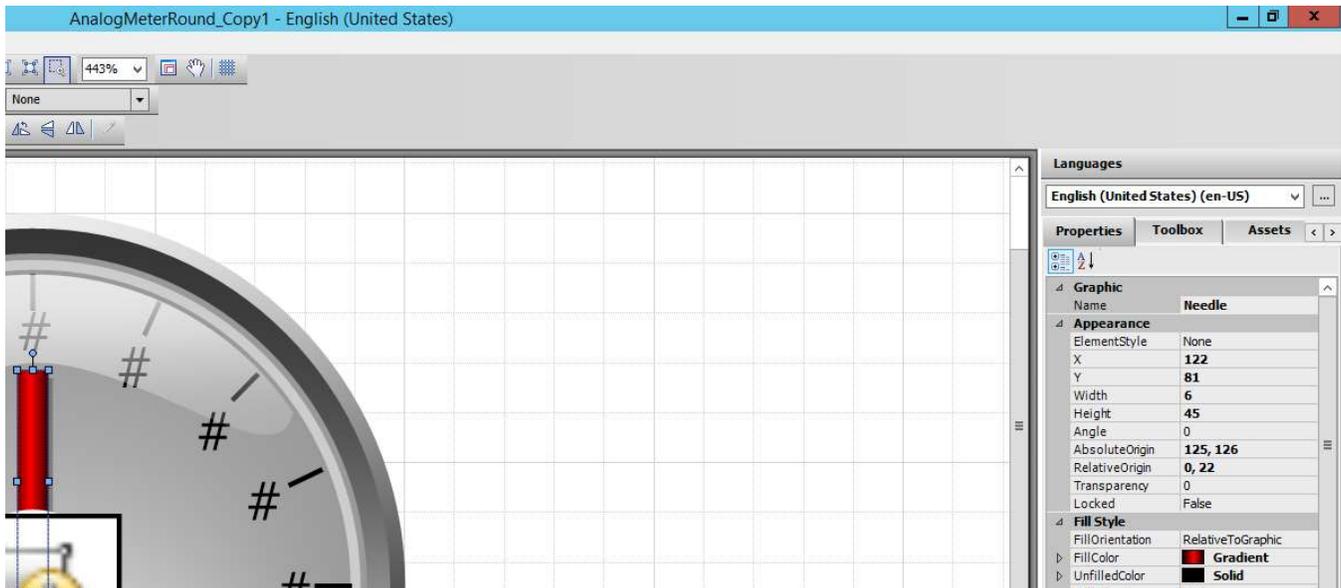


3. Maximize the editor and locate the “Tools” menu in the upper left, the “Elements” window below that and the “Properties” tab on the right edge of the screen. These will all be used in the upcoming steps.

You may also wish to zoom in on the object so it is more clearly visible in the editor. To do this click the “Rubber Band Zoom” button to the left of the zoom percentage display on the upper row of icons.

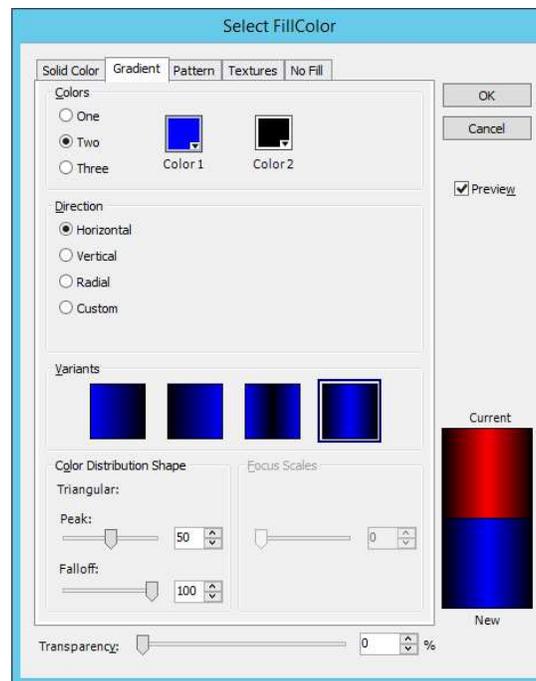


4. The first of the four modifications will be to change the color of the indicator from red to blue (or some other neutral color of your choice). Fortunately, the original creator of this object clearly labeled the various components as seen in the “Elements” list on the left edge of the editor. This makes it easy to locate the “Needle” element and left-click on it to select it. Notice that, when you do, the properties of the needle are displayed in the “Properties” list on the right side of the screen.



In order to change the color, left-click on the “FillColor” attribute that is currently set to a red gradient and then click on the button with three dots to bring up the color editor.

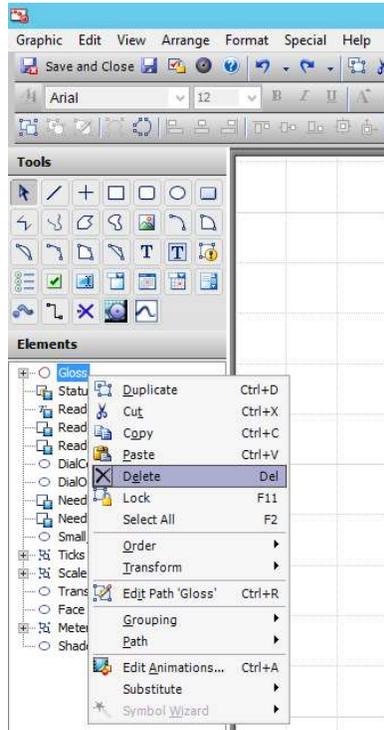
In this example, you will retain the gradient fill (as opposed to changing it to a Solid Color, Pattern or Textures) and change “Color1” from red to blue as shown below.



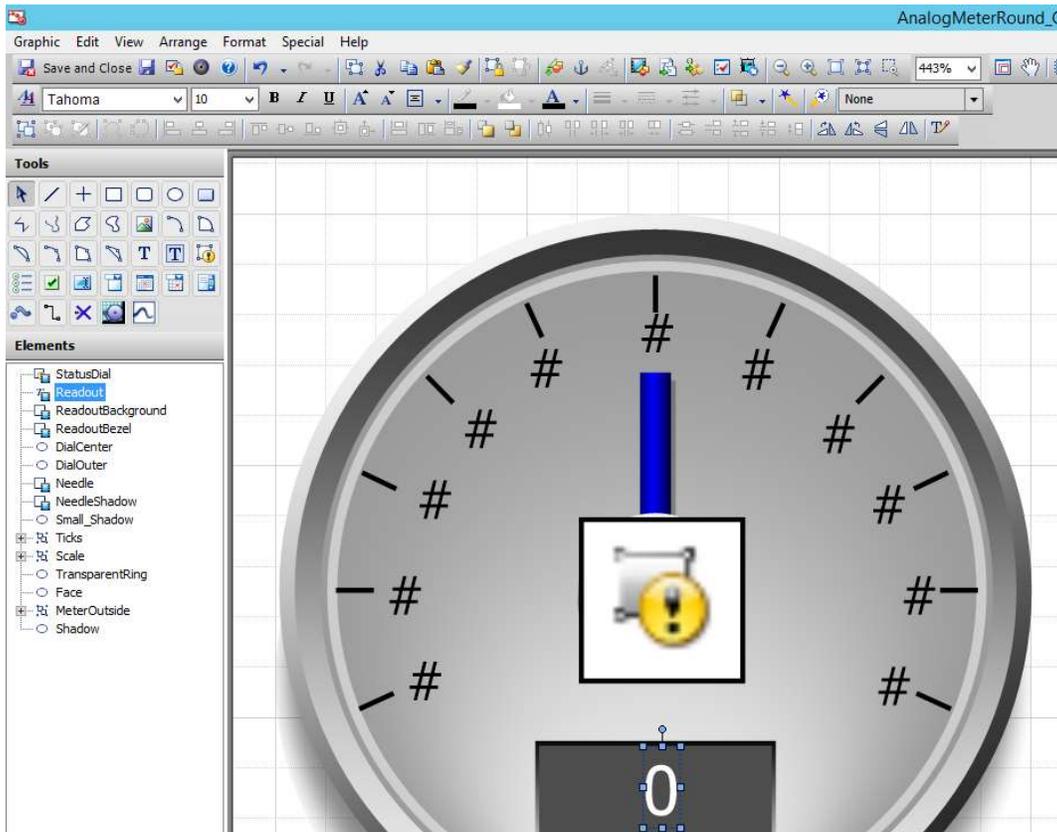
Then click “Ok” to close the color editor and note that the meter needle has changed color from red to blue.

Note: there is also an element called “NeedleShadow” but as this is colored grey, it isn’t necessary to change it. If you wanted to create a flatter appearance for the meter needle, you could delete the “NeedleShadow” element entirely.

5. The second of the four steps is to remove the “gloss” element at the top of the meter. To do this, right-click on the “Gloss” element and select “Delete”.



6. The third of the four steps is to increase the precision of the display so it can show more than just whole numbers. In order to do this, left-click on “Readout” in the elements list. Note that both the element on the list as well as the actual “0” object in the editor are selected.

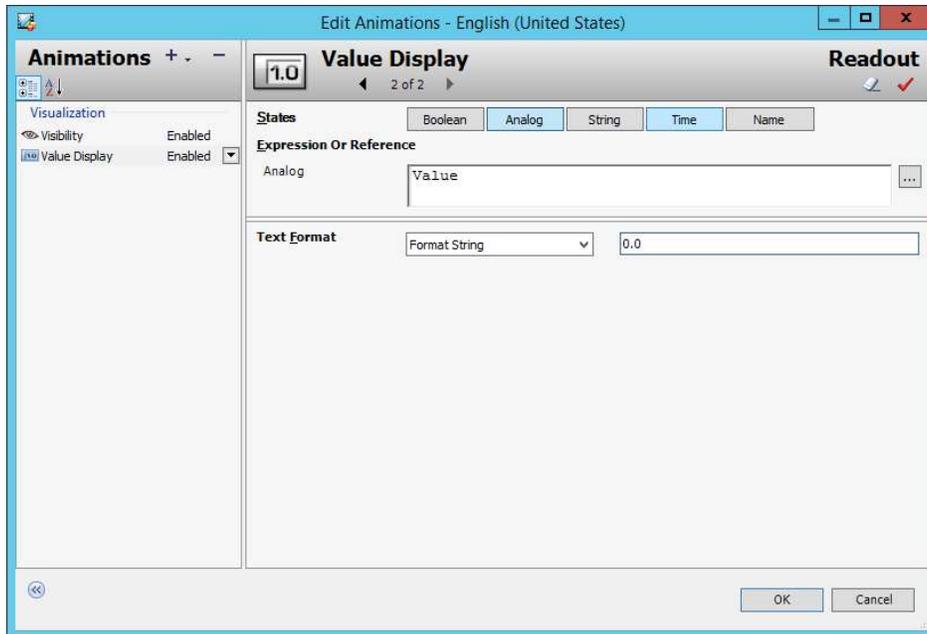


Also note that there is a small overlay on the element list that is meant to represent a 35mm film-frame. Further note that some of the elements (e.g. “DialCenter” and “DialOuter”) don’t have them while others (e.g. “Readout” and “Needle”) do. The film-frame is present when the element has one or more animations assigned to it.

In order to change how many decimal places of precision are displayed, the animation for “Readout” must be edited. To edit the animation, either double-click on the element on the element list or right-click on it and select “Edit Animations”.

When the “Edit Animations” window opens, note that there are two animations currently assigned to this element – a “Visibility” animation in which the visibility of the element is controlled via a custom property called “DisplayDigital” and a “Value Display” which actually controls the display of the value. Note that the “Text Format” is currently set to “0”. This tells the display to round whatever value is contained in the “Value” custom property to the nearest whole number when displaying it.

In order to display one value to the right of the decimal, change the “0” to “0.0”.



Click on “Ok” to accept the change and close the editor.

Note that the “Readout” element still contains a “0”. This will not prevent floating-point values from being displayed.

7. The last of the steps is to add the ability to display engineering units to the object. This change will require several different modifications.

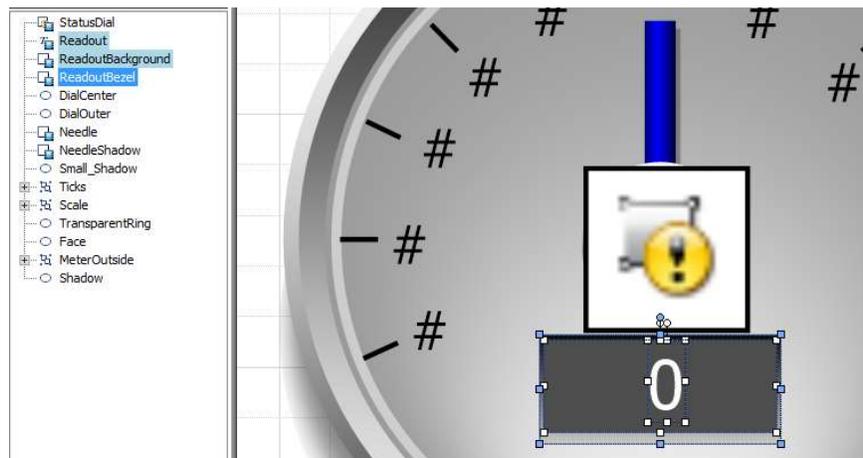
First, the existing value display objects must be moved slightly upwards to make room for the new display.

Second, a new element will have to be created in order to display engineering units.

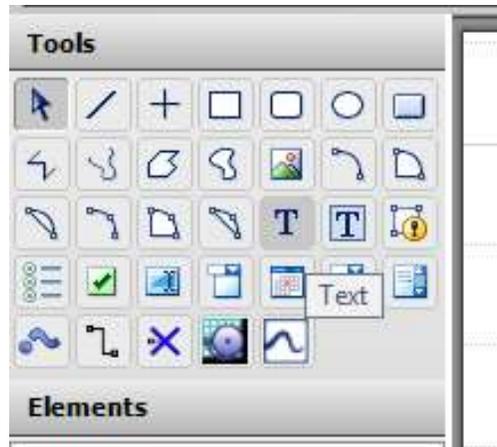
Third, a new custom property will need to be created in order to allow engineering units to be passed to the symbol.

To move the “Readout”, “ReadoutBackground” and “ReadoutBezel” elements upwards, select them by left-clicking on “Readout” and then shift-left-clicking on “ReadoutBezel”. This will select all three components.

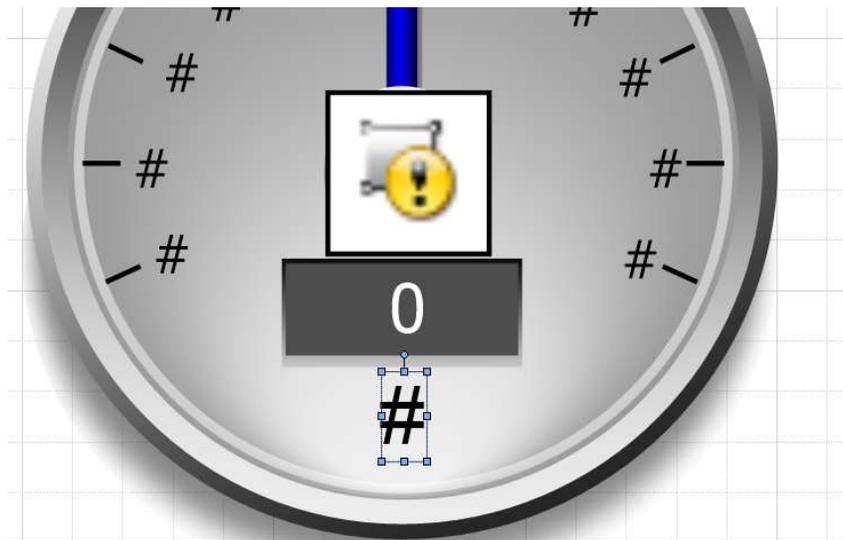
Then, use the arrow key to shift them upwards until the top edge of the readout is just touching the bottom of the square that is the “StatusDial”.



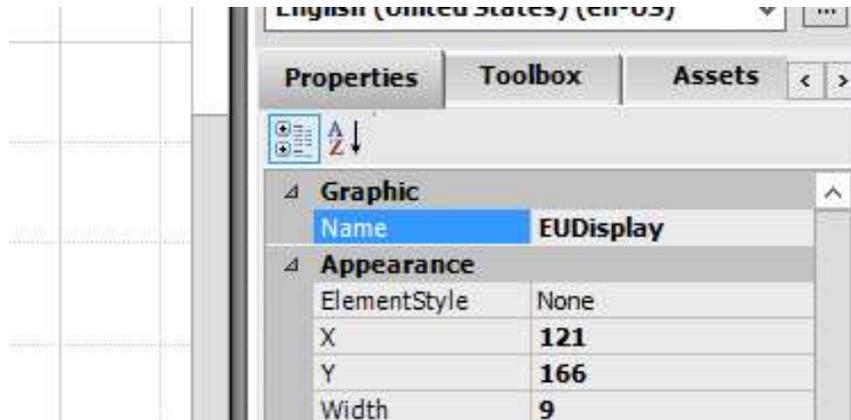
To create the new element, use the “Text” tool...



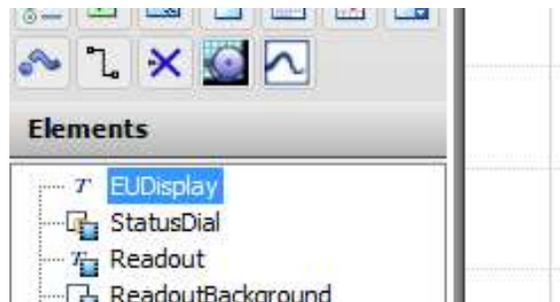
...and type a single “#” beneath the existing “0”. Once it has been drawn, drag it with the mouse or use the arrow keys to place it centered with respect to the “0” like this:



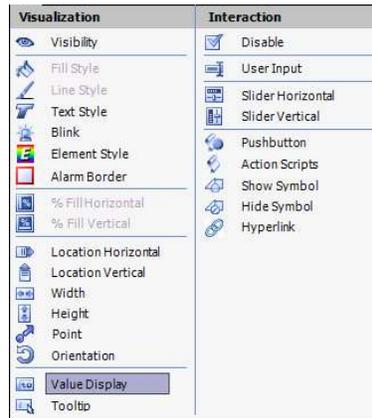
On the “Properties” tab, to the right of the editor, change the “Name” to “EUDisplay” (with no quotes and no spaces).



When you do that, note that the name on the elements list has also changed to match.

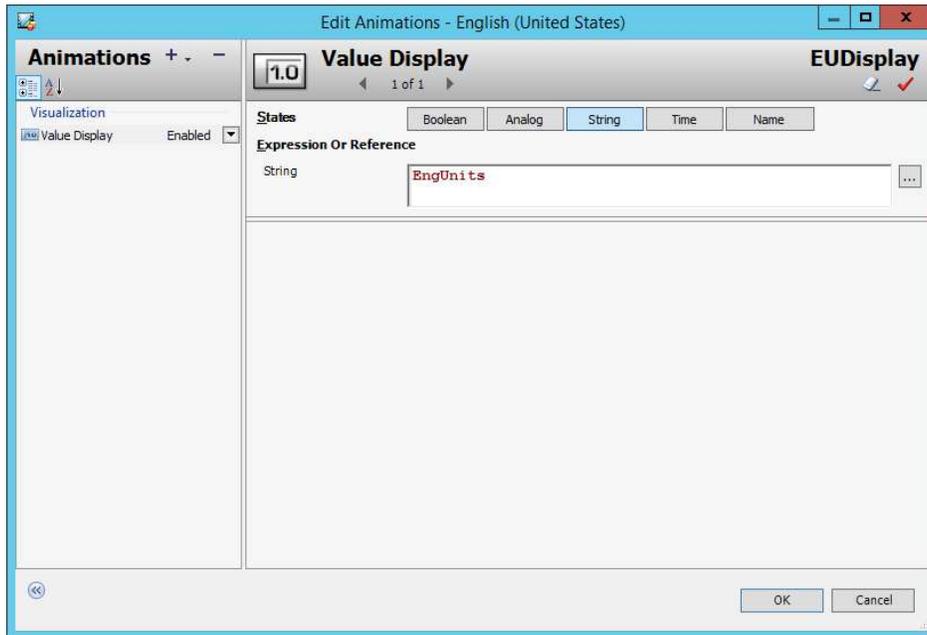


Next, the animation to display the engineering units value must be created. To add an animation link, double-click on the new “EUDisplay” entry on the elements list. In the “Edit Animations” window, click on the “+” to add a new animation – choose “Value Display” ...



...then press the “String” button.

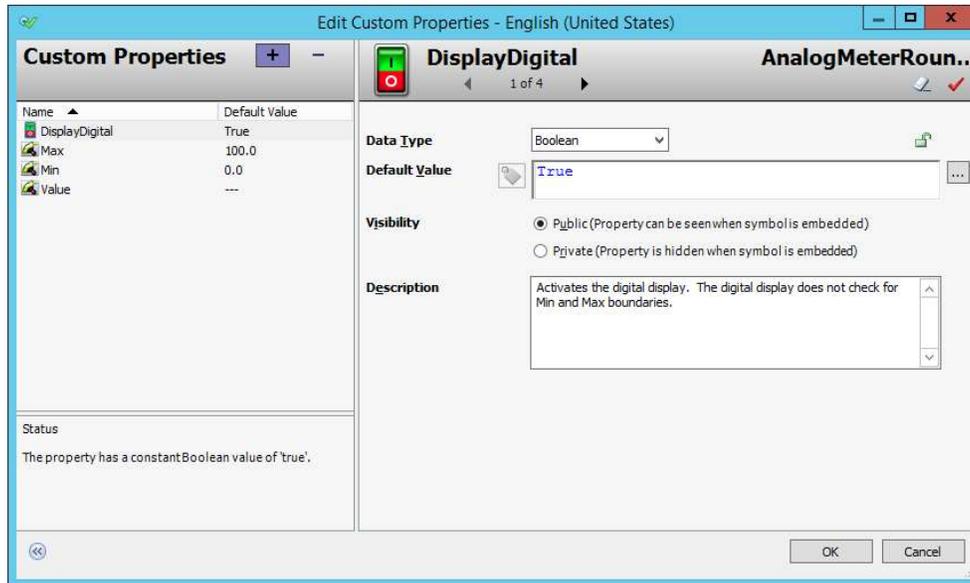
In the “Expression or Reference” field, type “EngUnits” (with no quotes and no spaces). This is a reference to a custom property that has not yet been created. However, it is useful to know that references to custom properties can be created before the actual custom property.



Click on “OK” to close the editor and accept the changes.

Note that if you tried to save the work you’ve done this far, you would get a warning that the reference to EngUnits could not be resolved. This warning will vanish when the custom property is created.

To create the custom property, right-click on the grid background of the editor and select “Custom Properties”. Note that this symbol already has four:

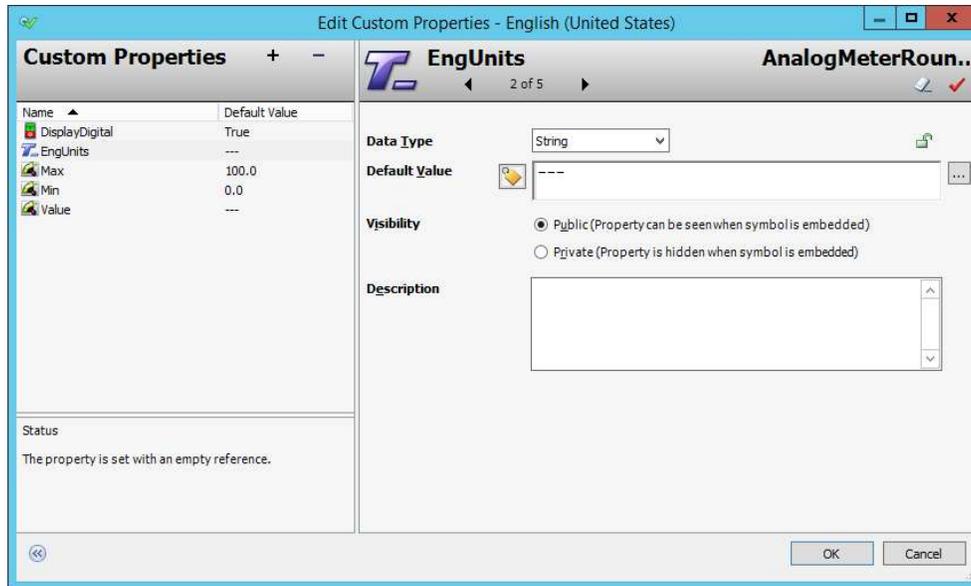


To create a new one, click on the “+” and name it “EngUnits”. Note that the name must correspond to the name used in animation links and not the name used to the element to display the value.

Set the “Data Type” to “String” and the “Default Value” to “---”(three hyphens with no spaces and no quotes).

Next, note the small button to the left of the default value. If you rest your mouse pointer on it, the popup help window should read “Click to switch to ‘Expression or Reference’ mode”. If you click on the button, it will toggle between a  and a tag icon . In this case, it must be set to tag so that the value of the tag that is eventually tied to this graphic can be passed to it.

When you're finished, the window should look like this:



Click on "OK" to close the editor. The modifications are now complete. Click "Save and Close" in the upper-left corner of the editor and you can use the custom object wherever you wish.

Version 1.0 – J. Gwynne – 3/26/20