



Description: Information on how building applications with the new Dockable Symbol Toolbar, Smart Symbols, and Edit Connections can make your life a breeze.

OS Requirement: Windows XP/2000/NT/95/98/Me.

General Requirement: Genesis32

The Dockable Symbol Toolbar provides a way for users to quickly build applications in GraphWorX32. It has all of the same functionality as the original Symbol Library (standalone EXE), plus many exciting new features.

A Smart Symbol is a special kind of symbol that has built-in dynamics, which can be attached to any combination of OPC tags, local aliases, expressions, or global aliases. Smart Symbols can be anything from a simple switch that turns green on true and red on false, to a complicated panel full of switches, dials, and sensors representing a control device on your manufacturing floor.

Coupling Smart Symbols with the Edit Connections dialog can significantly decrease your development time, and this application note will show you how. First, a brief introduction to the features of the Dockable Symbol Toolbar itself:

Features carried over from Standalone Symbol Library:

- Drag & drop support
- Cut, copy, paste, rename, and delete symbols
- Category password protection
- Multiple size icons to choose from
- Customizable root directory
- Hierarchical directory structure

New Features

Floating vs. Dockable

The fact that the Symbol Library is now a toolbar means that it can either be docked or floating just like any other standard Windows toolbar. You can set the position of the toolbar in one of two ways: through the GraphWorX32 menu, or by dragging the toolbar around the screen. The menu option is located on the "Compatibility" tab under "Format", "Application Preferences".

The 5 positions are:

- Floating
- Docked to left
- Docked to top
- Docked to right
- Docked to bottom

Note that although the toolbar mode is now the default, you can still run the Symbol Library as a separate EXE if you wish to do so. (Specify "Standalone" in the Application Preferences menu)

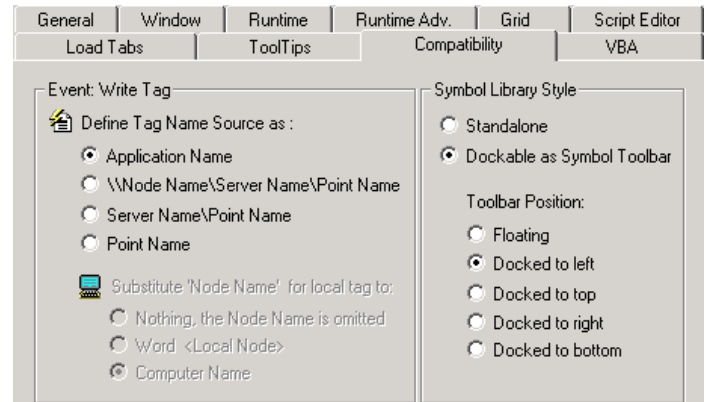


Fig. 1: Symbol Library Style Preferences

Category Navigation

Navigation within this sleek toolbar is extremely simple. There is a drop-down combo box where you can expand and collapse subdirectories using the + and - keys, then pick a new category with a single click. You can also navigate from within the list view. Every category and directory has an up-arrow folder icon, which will bring you to the parent directory when you double-click on it. This icon is also located next to the combo box for single-click navigation.

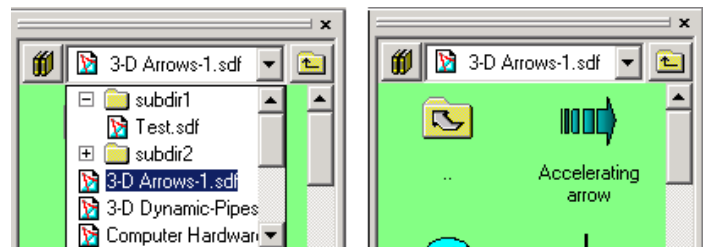


Fig. 2: Methods of Symbol Toolbar Navigation

Smart Symbols

The easiest way to understand Smart Symbols is to use a simple example. So, go into the "Switches & Sliders.sdf" category in your "2_Misc" symbol toolbar and drag the "UpDown1" symbol onto your GraphWorX32 display. This is a simple Symbol. Go into runtime to see the behavior of this Smart Symbol.

If you double-click on the switch, the property inspector will tell you that it is a "Symbol", and you can see that it has a Digital Selector and a Pick action associated with it. It is not clear what those actions are connected to or how they will behave however. Click on "OK" to close the property inspector.



Fig. 3: Light Switch Example



Edit Connections

Now right click on the switch and select **Edit Connections...**
This will pop up the dialog box shown in Figure 4.

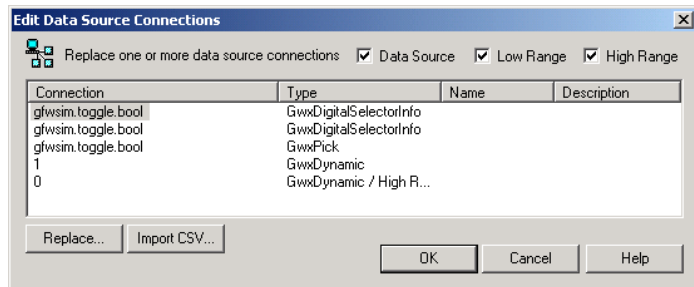


Fig. 4: Edit Connections Dialog Box

Each connection represents a dynamic or part of a dynamic action, such as Flash, Hide, Animate, Rotate, Size, or Pick. The tags or expressions that are connected to each dynamic control each action. Notice that the light switch in our example is currently controlled by simulation variables, but in a real world application, you would want to connect it to some electrical output values to actually monitor when a light switch is on or off. Edit Connections allows you to do precisely that by changing what each dynamic action is pointing to.

For demonstration purposes, let's replace the current connections with a different simulation variable:

1. Highlight the first connection, gwfsim.toggle.bool
2. Click on **Replace...**

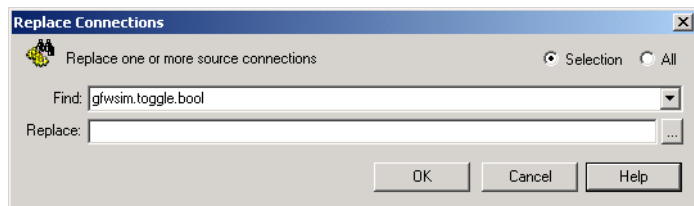


Fig. 5: Replace Connections Dialog Box

3. Click on the ... button, and open the **Unified Browser**. A new window will open up and you can choose the location of the tag you want to associate with this symbol.
4. Browse to the following tag:
ICONICS.Simulator.1\SimulatePLC.BOOL.In1
(See Figure 6 for its location), and then click "OK".
5. To ensure that our changes are consistent across the symbol, select the radio button **All** in the upper right corner of the **Replace Connections** dialog box.

Note: You can also replace connections one-by-one, if you would like to use different tags for each dynamic.

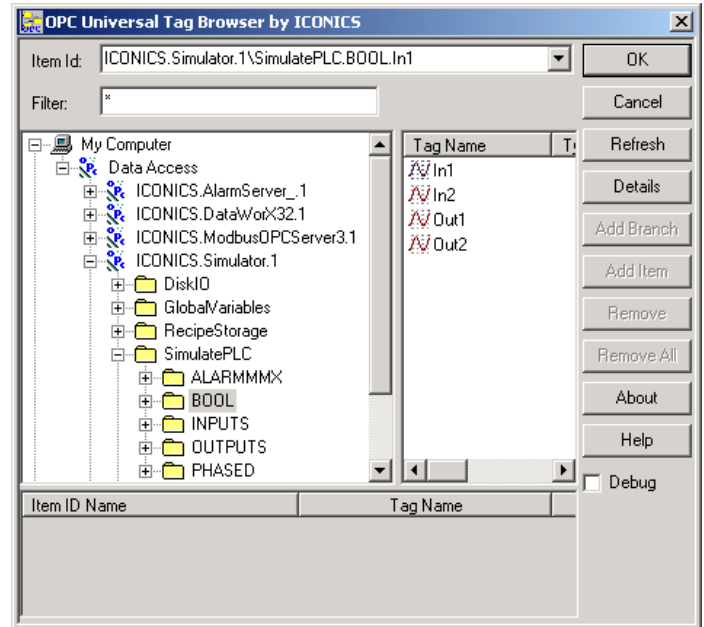


Fig. 6: ICONICS.Simulator Tag Location

6. Click "OK". Your Edit Connections box should now look like this:

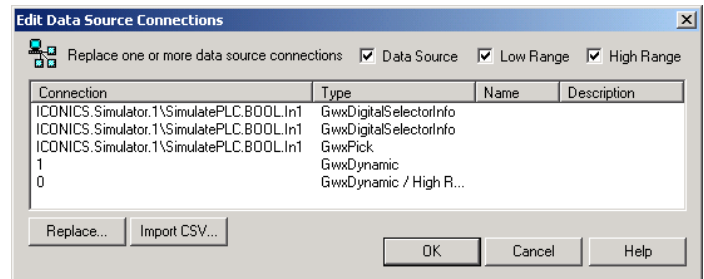


Fig. 7: Edit Connections Box With Replaced Tags

7. Click on "OK" to close the Edit Connections box.
8. Go into runtime once more, and note the difference in the switch's behavior.

And remember, you can edit connections to any OPC tag, expression, local alias, local variable, simulation variable, or global alias: it's that easy!