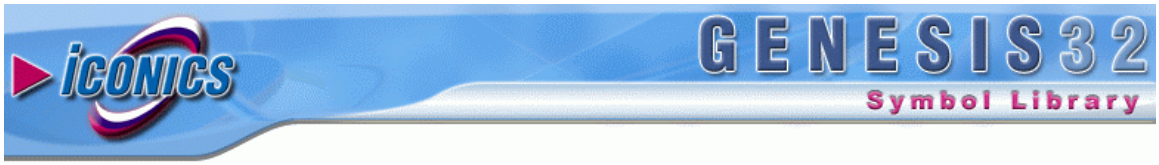


ICONICS, Inc.
100 Foxborough Blvd.
Foxborough, MA 02035
Phone: (508) 543-8600
Fax: (508) 543-1503
Worldwide Support Centers
www.iconics.com

© ICONICS, Inc. All rights reserved.
Specifications are subject to change without notice. GENESIS, GENESIS32, Pocket GENESIS, BizViz and their respective modules, OPC-To-The-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc. Other product and company names mentioned herein may be trademarks of their respective owners.

Introduction

You can use the Symbol Library to create and edit graphic objects as symbols and to build and maintain symbol categories. A symbol is a group of objects that is treated as a single object. These symbols can be user-created or loaded from the symbol library categories delivered with GraphWorX32.

GraphWorX32 supports the use of Smart symbols in displays. Smart symbols contain data-point connections. If you load a Smart symbol into your display in animation mode, the symbol begins displaying data immediately based on that data-point connection.

There is no limit to the number of Symbol Library category files (.sdf) you can create, or to the number of symbols that can exist within each file.

The main purpose of the Symbol Library is to preview and import previously stored GraphWorX32 symbols back into GraphWorX32. Additional functionality allows you to manage the content of the Symbol Library by:

- Creating, renaming, and deleting categories and directories.
- Storing, renaming, and deleting images in a category.

Symbol Library Design

There are two different modes for the Symbol Library:

- Standard stand-alone Symbol Library
- Dockable symbol toolbar in GraphWorX32

To launch the Symbol Library from GraphWorX32, click the **Symbols** button on the **Draw** toolbar in GraphWorX32.

To launch the stand-alone Symbol Library from the Windows **Start** menu, select **Programs > ICONICS GENESIS32 > GraphWorX32 > Symbol Library.**

Note: When you open the stand-alone Symbol Library, a splash screen image appears. The following command lines allow you to start the Symbol Library without showing the splash screen:

SymbolLibrary.exe -NoSplash

or

SymbolLibrary.exe /NoSplash

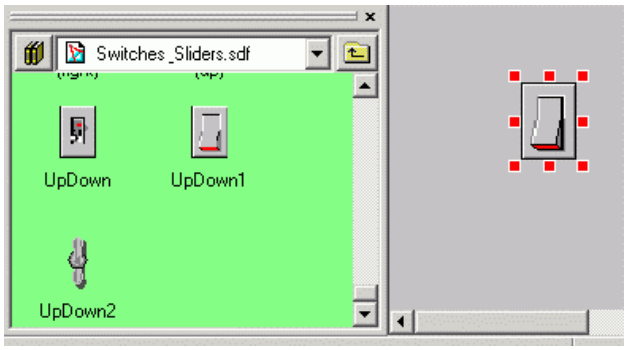
For example:

1. From the Windows **Start** menu select **Run.**
2. Type in "Command".
3. Then at the C:\ prompt type "SymbolLibrary -nosplash".

Smart Symbols

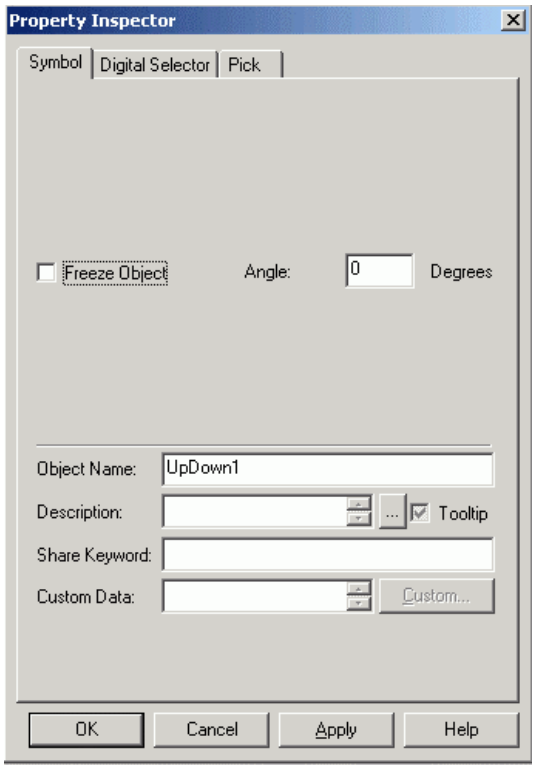
The easiest way to understand Smart Symbols is to use a simple example:

1. Go into the “Switches & Sliders” category in the Symbol Library and drag the “UpDown1” symbol onto your GraphWorX32 display, as shown in the figure below. This is a simple switch. Go into runtime mode to see the behavior of this Smart Symbol.



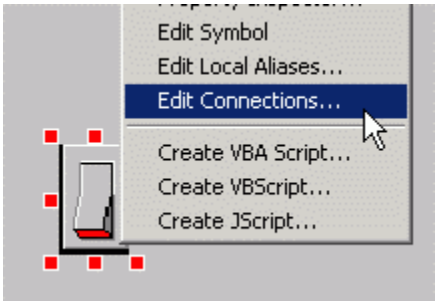
Switch Symbol in GraphWorX32 Display

2. If you double-click on the light switch, the **Property Inspector** will tell you that it is a “Symbol,” and you can see that it also has a **Digital Selector** and a **Pick** action associated with it, as shown in the figure below. But, it is not clear what those actions are connected to or how they will behave. Click **OK** to close the property inspector.



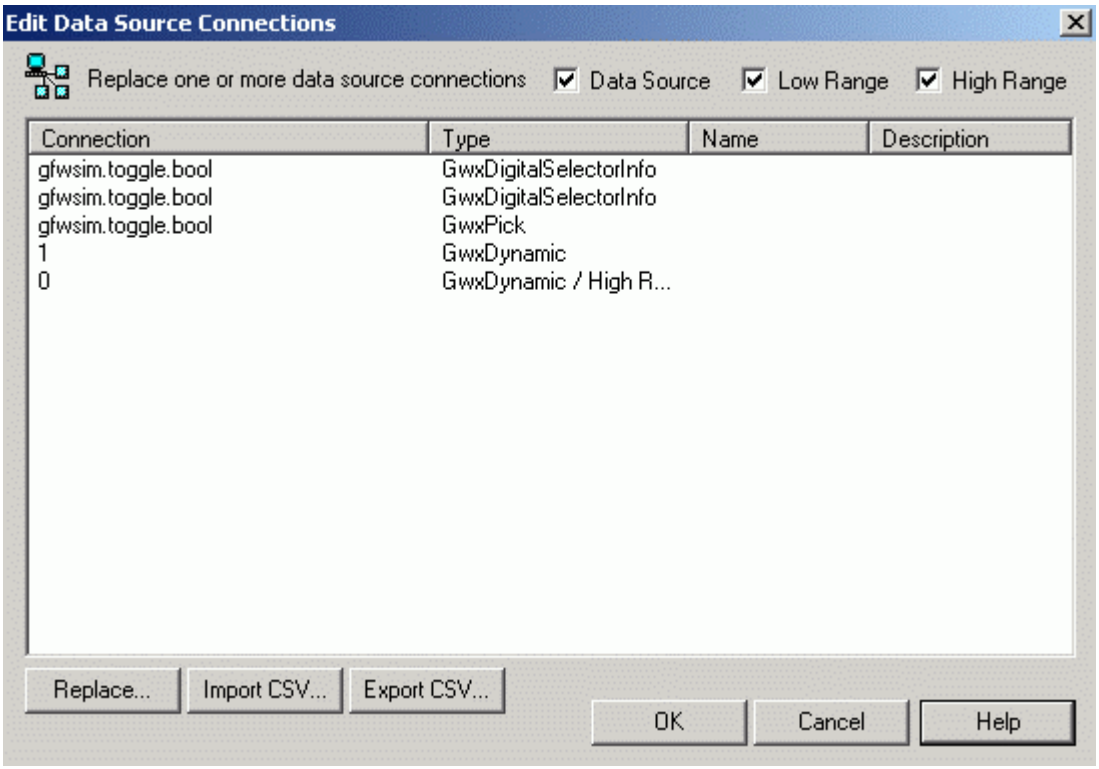
Property Inspector for Switch

3. Right-click on the switch and select **Edit Connections** from the pop-up menu, as shown in the figure below.



Editing Data Connections

4. This opens the **Edit Data Source Connections** dialog box, as shown in the figure below. 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. The **Edit Data Source Connections** allows you to do precisely that by changing what each dynamic action is pointing to.

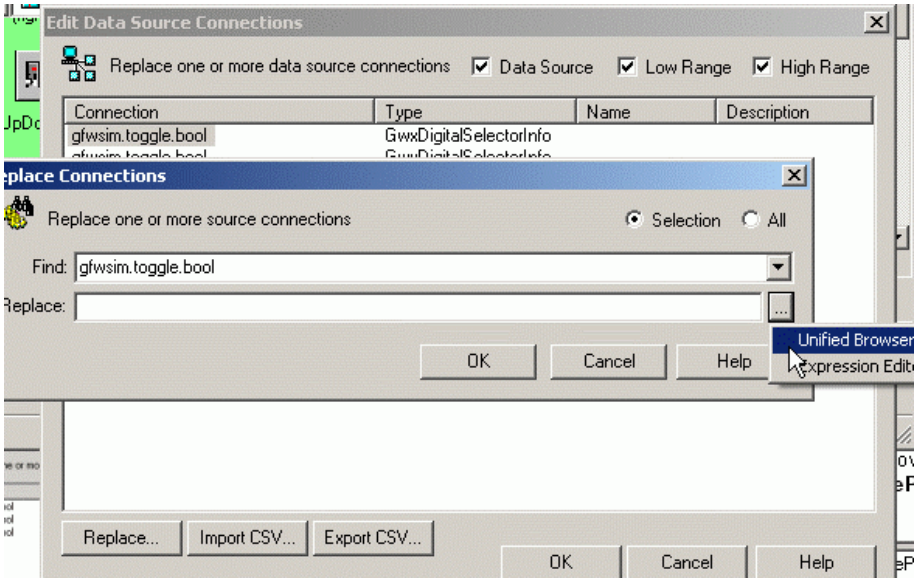


Edit Data Source Connections Dialog Box

5. For demonstration purposes, let's replace the current connections with a different simulation variable by clicking on the first connection: **gwfsim.toggle.bool.**

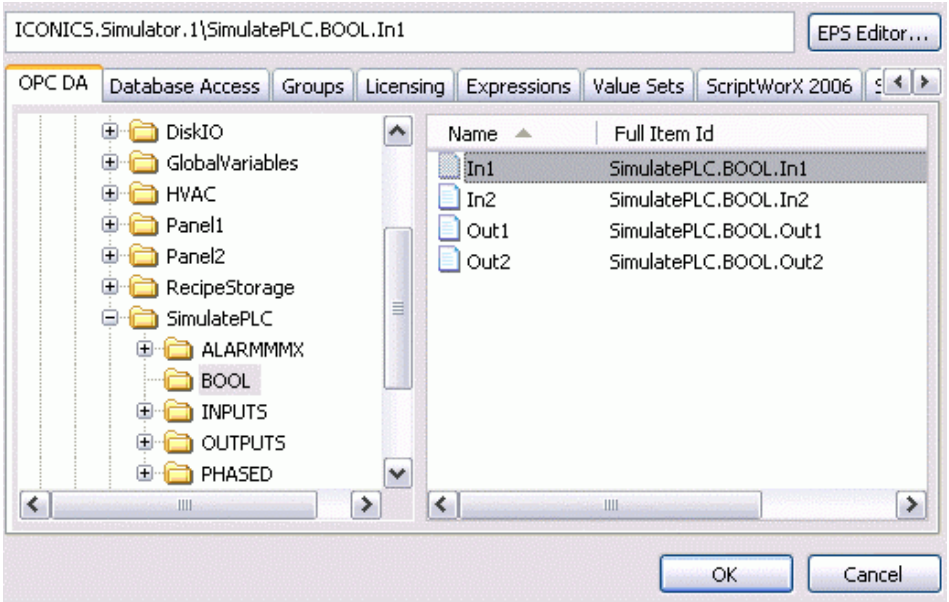
6. This opens the **Replace Connections** dialog box, as shown in the figure below. In the **Replace** field, click the **...** button and select **Unified Browser** from the pop-up menu.

   Note: In the Unified Data Browser, you can also edit connections to any OPC tag, expression, local alias, local variable, simulation variable, or global alias.



Replace Connections Dialog Box

7. This opens the Unified Data Browser. In the **OPC DA** tab, browse to the following tag: **ICONICS.Simulator.1\SimulatePLC.BOOL.In1,** and then click **OK,** as shown in the figure below.



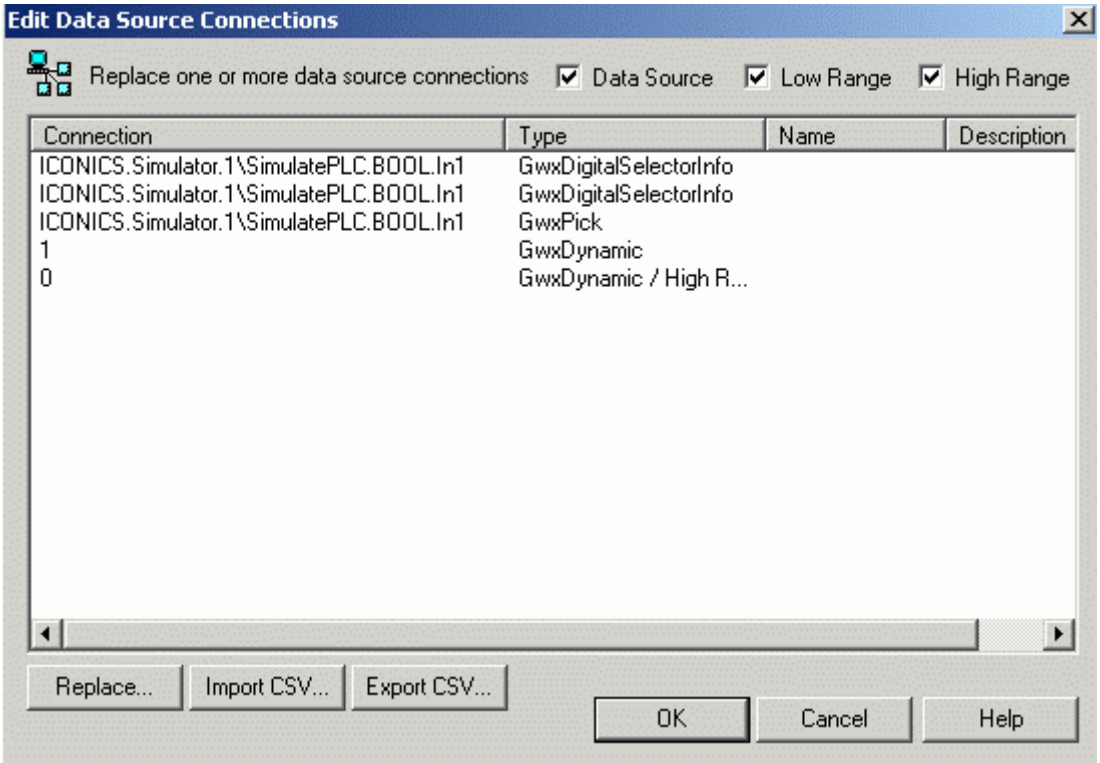
Simulator Tag Location

8. To ensure that your changes are consistent across the symbol, select the radio button **All** in the upper right corner of the **Replace Connections** dialog box. Click **OK.**

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

9. The **Edit Data Source Connections** dialog box should now appear as shown in the figure below. Click **OK.**

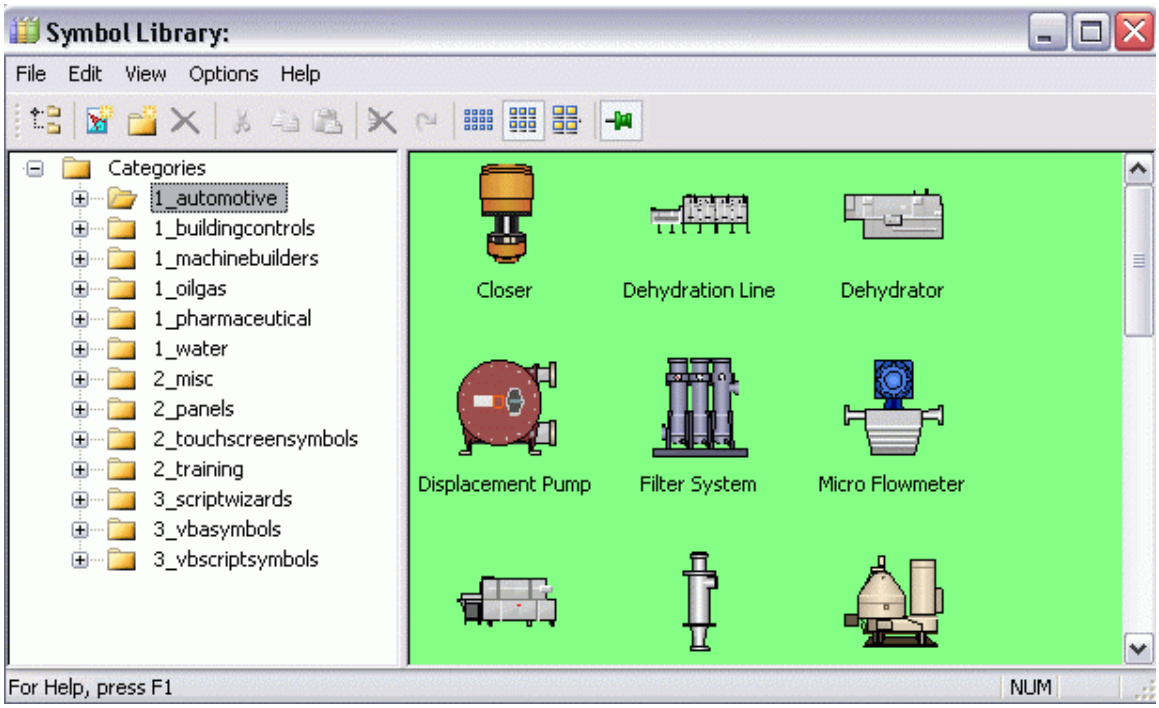
10. Go into runtime once more, and note the difference in the switch's behavior.



Edit Connections Box With Replaced Tags

Stand-Alone Symbol Library

The stand-alone Symbol Library is a GraphWorX32-independent program visually similar to Windows Explorer. As shown in the figure below, there is a **Tree View** showing directories and categories (.sdf files) on the left-hand side, and a **Control View** showing images of the current category on the right-hand side.



Stand-Alone Symbol Library Screen

The current design supports a **multilevel hierarchy** of categories. This hierarchy is characterized by a standard **directory** structure. This simplifies the reorganization of the library because standard Windows tools can be used for creating, renaming, moving, and deleting directories.

There can be an unlimited number of categories in any directory. A **category** is a file containing an unlimited number of GraphWorX32 symbols. It stores any type of symbol.

A useful feature that helps with a library organization is a **Root Directory**. You can mark any directory as a root for the Symbol Library. Only directories and categories below this root are shown in the tree view. This feature also allows the creation of several different libraries.

You can have multiple instances of the Symbol Library open at once. Two or more instances can simultaneously access the same category. Any changes made to a category are stored as soon as possible, and will appear in other instances.

Toolbar

To show or hide the Symbol Library toolbar, select **Toolbar** from the **View** menu. The toolbar, shown below, contains the following command buttons.

Set Root Directory: Specifies the target directory for the symbol category (.sdf) files.

New Category: Creates a new category file.

New Folder: Creates a new symbol directory.

Delete: Deletes current selection.

Cut: Removes the selected symbol from the list view.

Copy: Copies the selected symbol.

Paste: Pastes a symbol into the list view.

Delete Symbol: Deletes the selected symbol.

Undo Delete: Nullifies the last delete action.

Small Icons: Displays symbols as small icons.

Medium Icons: Displays symbols as medium icons.

Large Icons: Displays symbols as large icons.

Always on Top: Positions the Symbol Library on top of other open windows.

Menus

The menu bar of the stand-alone Symbol Library contains the following menus:

- File
- Edit
- View
- Options
- Help

You can create symbols using objects from the **Draw** menu. You can store symbols in the Symbol Library, and you can load symbols from the Symbol Library into the current display. You can also delete symbols you do not need. You can even change the size of the symbol button and choose a large or smaller size.

Note: You can also access many of the menu commands by right-clicking items in the tree control of the Configurator and selecting command functions from the pop-up menus.

File Menu

The **File** menu of the contains the following commands:

File Menu Commands

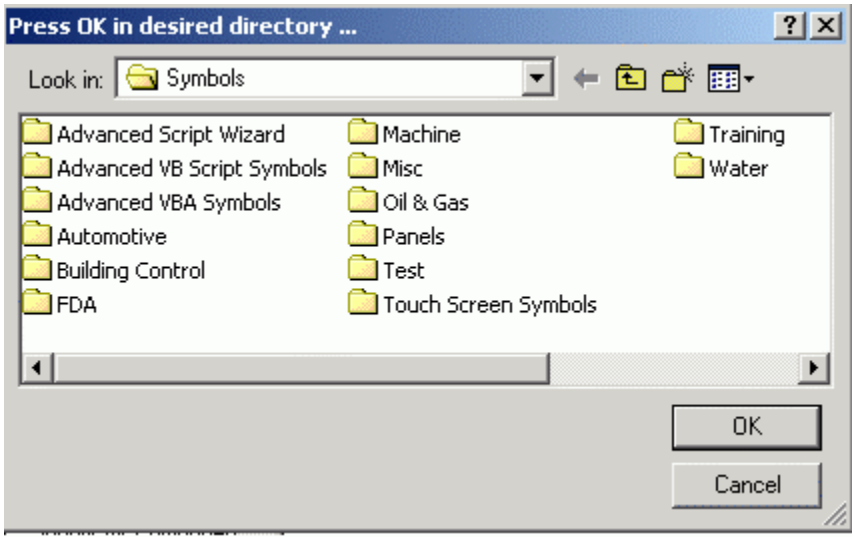
Command	Function
Root Directory	Specifies the target directory for the symbol category (.sdf) files.
Add > Category	Creates a new symbol category (.sdf) file.
Add > Folder	Creates a new symbol directory.
Rename	Renames the selected folder or category.
Delete	Deletes the current selection.
Passwords > Lock	Sets up the read-write password in the **Set Category Write Mode Password** dialog box. The password must be confirmed. Sets the read-write password to the category, which currently has no locks set, or the read-only password for the category, which is already opened in read-write mode. You cannot set the read-only password without setting the read-write password; this would result in a category file lock that no one would be able to remove (to modify the category file).
Passwords > Change Lock	Changes the category file password. You must be granted read-write permission to open the category with the read-write password.
Passwords > Unlock	Opens a locked symbol category. When the category file is initially locked, it is always locked with read-write access denied. (It could also have the read-only lock set). Simply enter the read-write password (or the read-only password if it is configured) to open the file. When the password is entered, it is recognized as either read-write or read-only. If the password is accepted, the category file transitions to the proper state to reflect the change.
Exit	Closes the application.

Specifying the Root Directory

To mark a directory as a root for the Symbol Library, do the following:

1. Select **Root Directory** from the **File** menu.
2. Select the root directory from the dialog box, as shown below.
3. Click **OK.**

Only directories and categories below this root are shown in the tree view of the Symbol Library.



Root Directory Dialog Box

Renaming Symbol Categories

To rename a directory or category, do the following:

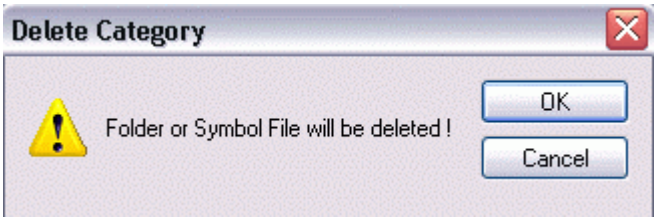
1. Select **Rename** from the **File** menu, or click on the **Rename** button on the toolbar.
2. Enter the desired name in the field in the dialog box, as shown below.
3. Enter the desired name in the field, and click **OK.**

Rename Directory/Category

Deleting Symbol Categories

To delete a directory or category from the tree, do the following:

1. Select **Delete** from the **File** menu, or click the **Delete** button on the toolbar. This opens the following message and allows you to delete a directory or a category.
2. Click **OK** or press **Enter** to delete the directory or category.



Delete Category Message

Note: The **Delete** key is not used to delete categories.

Password-Protecting Symbol Categories

A lock is activated when some password is entered or set for a category file. It is possible to set two different locks:

- Read-only lock
- Read-write lock

Read-Only Lock

When a **read-only lock** is applied to a category file, the file's content is visible but cannot be changed. A read-only lock prevents the user from doing the following:

- Changing the label of a symbol in the category file.
- Deleting a symbol from the category file.
- Pasting a symbol into the category file.
- Dragging and dropping a new symbol in the category file.

The Symbol Library also reflects when the category file is system read-only. Again, all symbols in the category file are visible, but the category itself cannot be modified. The user has to change the system attributes (remove the read-only access right) to allow for any changes to the category file.

Read-Write Lock

When a **read-write lock** is applied to a category file, the file's content is visible and can be modified. When a category file is system read-write and no lock is set, the user has unlimited access to the file. The unlimited access to the category file is gained when read-write access is granted.

The Symbol Library also tracks the status of every category file. The Symbol Library keeps track of each file that has been accessed by a user (who has entered the password to open the locked category file). Thus, when the user selects a category file that already has been accessed, the user does not have to re-enter the password for that particular category file.

Note: The Symbol Library maintains this information until the user closes the application. When the Symbol Library is restarted, the user will have to regain access to the locked category file.

Category Modes Representation

The Symbol Library category file can exist in the following modes. Each of these category file modes is represented in the Symbol Library tree view by a different icon, as shown below.

System read-only	
Read-only	
Read-write	
Locked	

Category File Transition

The simplest situation is when the file is system read-only. It cannot be changed. Therefore, it is read-only, and even the user is unable to change its mode unless the user modifies the file system attributes.

When the category is opened to any changes, the user can set the read-write password to it. The user can also set the read-only password to the same category file. The category, which is read-write, stays read-write even with the read-only password (lock) configured. The category must be read-write according to any changes (such as setting the read-only password).

The category file cannot have a read-only lock active. It always has to go with read-write so the administrator will be able to modify its content or passwords (locks) set.

Once the category file has the password set, it remains opened in read-write mode until the Symbol Library is closed.

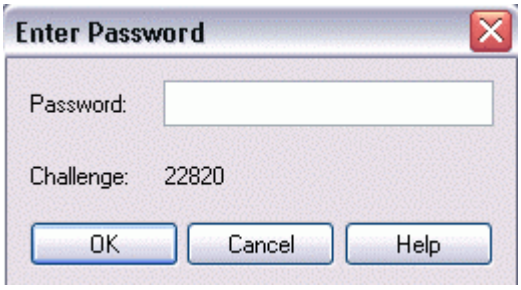
After launching the Symbol Library, those category files with locks set (active) are locked. Their content cannot be displayed, and they appear to be empty. They can be opened in either read-only mode or read-write mode.

Opening Locked Symbol Categories

When the category file is initially locked, it is always locked with read-write access denied. (It could also have the read-only lock set). To open such a file, select **Passwords > Unlock** from the **File** menu. This opens the **Enter**

Password dialog box, shown below. Simply enter the read-write password (or the read-only password if it is configured) to open the file. If you forget the password, you can use the **Challenge** number to get a substitute password from the ICONICS technical-support team (if applicable).

When the password is entered, it is recognized as either read-write or read-only. If the password is accepted, the category file transitions to the proper state to reflect the change.



Enter Password Dialog Box

Setting the Category Password

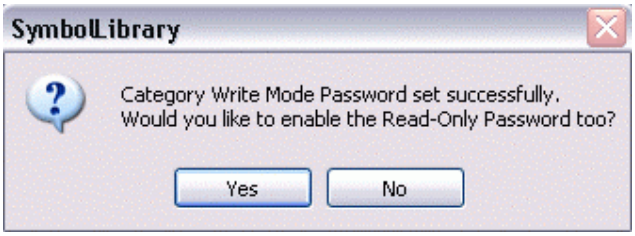
When the category file is system read-write, the **Passwords > Lock** command is enabled on the **File** menu.

It is possible to set the read-write password to the category, which currently has no locks set, or the read-only password for category, which is already opened in read-write mode. The user cannot set the read-only password without setting the read-write password; this would result in a category file lock that no one would be able to remove (to modify the category file).

You can set up the read-write password using the **Set Write Mode Password** dialog box, shown below. The password must be confirmed.

Set Write Mode Password Dialog Box

When the read-write password is successfully configured, you will be asked whether to set the read-only password to the category too, as shown below.



Enabling Read-Only Password Configuration

Clicking **Yes** opens the **Set Read-Only Password** dialog box, shown below.

Set Read-Only Password Dialog Box

When trying to set the password to the file that was previously opened in read-write mode, only the **Set Read-Only Password** dialog box will open to allow for the change of the read-only password. This time, the user is considered to be the administrator having full access to the category file. That is why the user is able to set the read-only password without entering the old read-only password.

Changing the Category Password

To change the category file passwords, select **Passwords > Change Lock** from the **File** menu. This opens the **Change Password for Current Category** dialog box, shown below. You must be granted read-write permission (to open the category with the read-write password).

Change Password for Current Category Dialog Box

The password entered the last time for the selected category is automatically entered into the **Old Password** field. Again, the Symbol Library will recognize the type of category password and change it if the entered values are correct (both the **New Password** and **Confirm New Password** strings match). When successful, a message box pops up with information about which lock was modified.

Edit Menu

The **Edit** menu commands are listed in the table below.

Edit Menu Commands

Command	Shortcut Keys	Function
Undo	CTRL+Z	Nullifies the last delete action.
Cut	CTRL+X	Removes the selected symbol from the list view.
Copy	CTRL+C	Copies the selected symbol.
Paste	CTRL+V	Pastes a symbol into the list view.
Delete		Deletes the selected symbol.
Edit Label	CTRL+T	Renames the selected symbol.

View Menu

The **View** menu commands are listed in the table below.

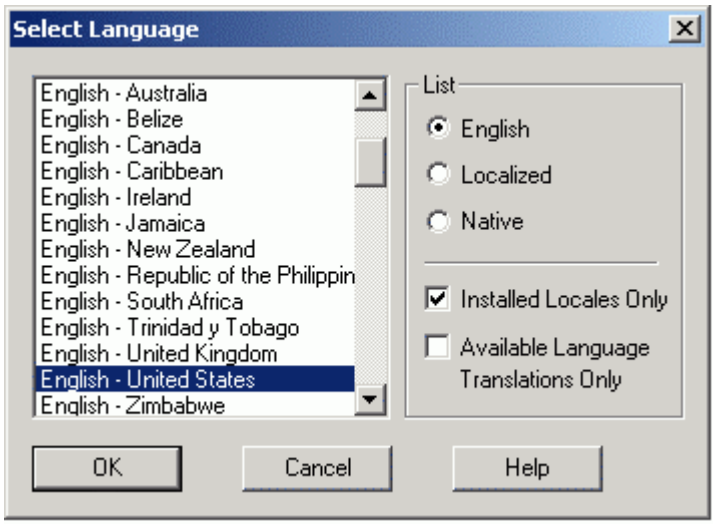
View Menu Commands

Command	Function
Toolbar	Shows/hides the toolbar.
Status Bar	Shows/hides the status bar at the bottom of the Symbol Library screen.
Split	Resizes the right and left panes of the Symbol Library window.
Select Language	Opens the **Select Language** dialog box (see below). Choose the language you wish to use for your system (Unicode version only) and click **OK.** For navigation purposes, use the buttons and check boxes in the **List** section.

Selecting Languages

The **Select Language** function on the **View** menu allows you to choose which language to use in your display. Choosing **Select Language** from the **View** menu opens the **Select Language** dialog box, shown in the figure below.

Note: A language resource .dll is required for language switching.



Select Language Dialog Box

Define the parameters listed in the table below. Then click **OK** to return to the work area.

Select Language Parameters

Parameter	Description
List	Lists available languages. Depending on which item you have selected, the view on the left will change. If **English** is checked, the languages will appear as their English name. If **Localized** is checked, the languages will appear with the native country in parentheses (for languages with several dialects only). When **Native** is checked, the languages are displayed the way they would be written in that language.
Installed Locales Only	If this is checked, local languages appear in the box.
Available Language Translations Only	Checking this box allows you to choose from available language translations only.

Options Menu

The **Options** menu commands are listed in the table below.

Options Menu Commands

Command	Function
Image Size > Small Images	Displays symbols as small icons.
Image Size > Medium Images	Displays symbols as medium icons.
Image Size > Large Images	Displays symbols as large icons.
3D Effect	Gives the symbols a three-dimensional look.
Always on Top	Positions the Symbol Library on top of other open windows.
Background Color	Changes the background color of the list view pane.

Help Menu

The **Help** menu contains the following commands:

Command	Shortcut Key	Function
Help Topics	F1	Launches the online help for the Configurator.
About Application		Launches the **ICONICS About Box,** which contains information about the product version number, copyright, and available disk space. It also contains information about how to contact ICONICS.

Creating and Storing Symbols

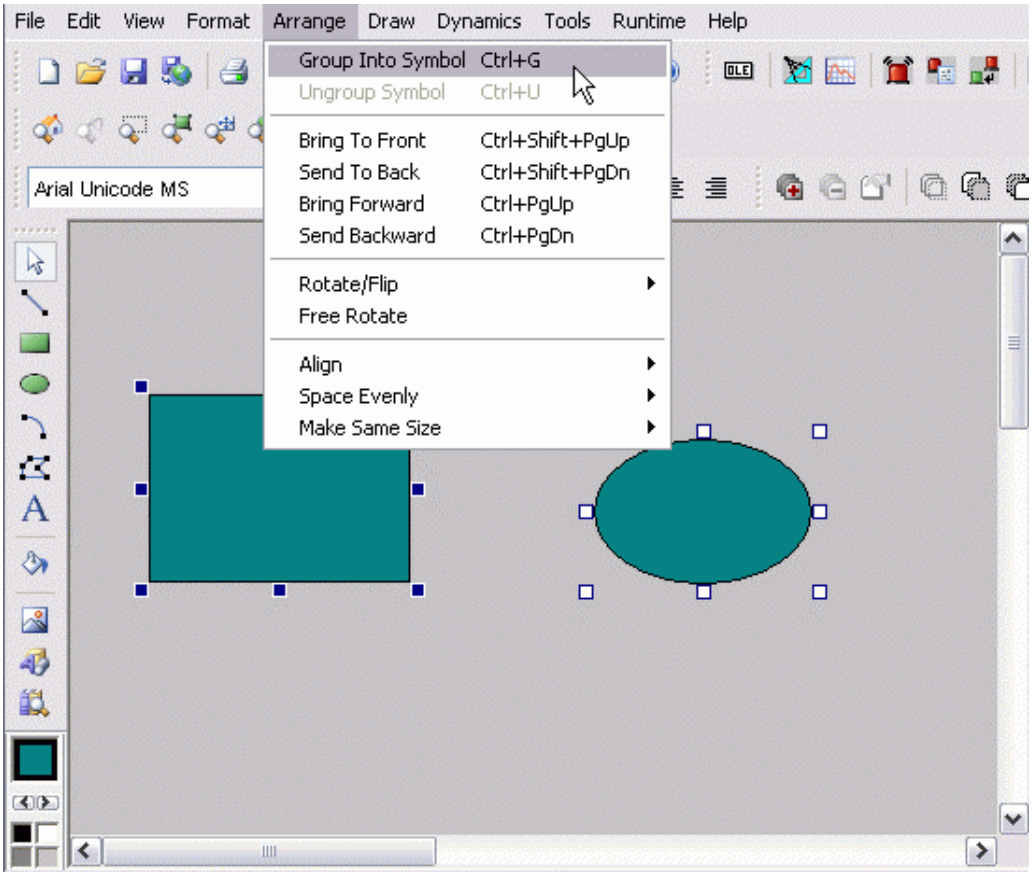
You can select symbols in your GraphWorX32 display to be saved and stored in the Symbol Library. To save (store) a symbol to the Symbol Library, do the following:

1. In GraphWorX32, create and group the object(s) to make into a symbol, as shown in the figure below.
2. Select the symbol.
3. Click on the **Symbols** button on the **Draw** toolbar. This opens the Symbol Library.

4. Select **Copy** from the **Edit** menu in GraphWorX32.

5. Drag-and-drop the new symbol into the desired category, or select **Paste** from the **Edit** menu of the Symbol Library.

6. The symbol from your display is now stored in this category in the Symbol Library.



Storing Symbols

Loading Symbols

The **Load** function allows you to copy a symbol from the Symbol Library into your GraphWorX32 display. Once a copy is placed in your display, the original remains in the library. To load a symbol, do the following:

1. Select the symbol in the Symbol Library that you want to insert into your display.

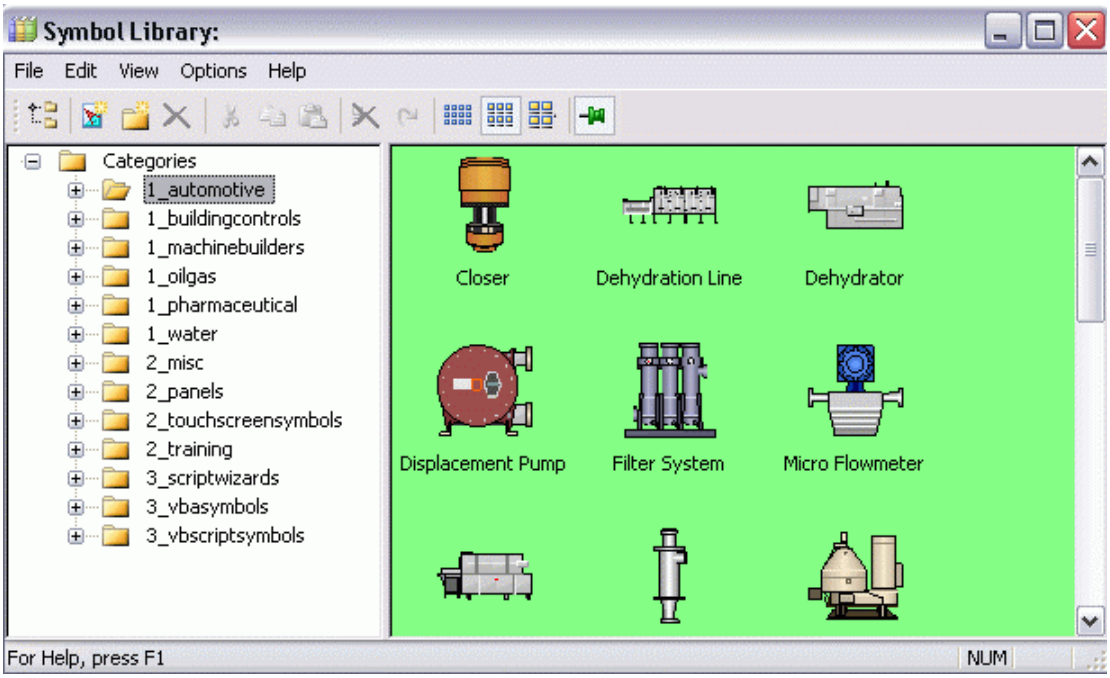
2. Select **Copy** from the **Edit** menu of the Symbol Library.

3. Go to your display and position the cursor where you want the symbol inserted. Then select **Paste** from the **Edit** menu in GraphWorX32. This inserts the symbol into your display. You can also drag-and-drop the symbol from the Symbol Library into your display.

Categories

While you are encouraged to import your own symbols into the Symbol Library, the Symbol Library comes loaded with several general categories of symbols commonly used in displays for industrial automation purposes, as shown in the figure below:

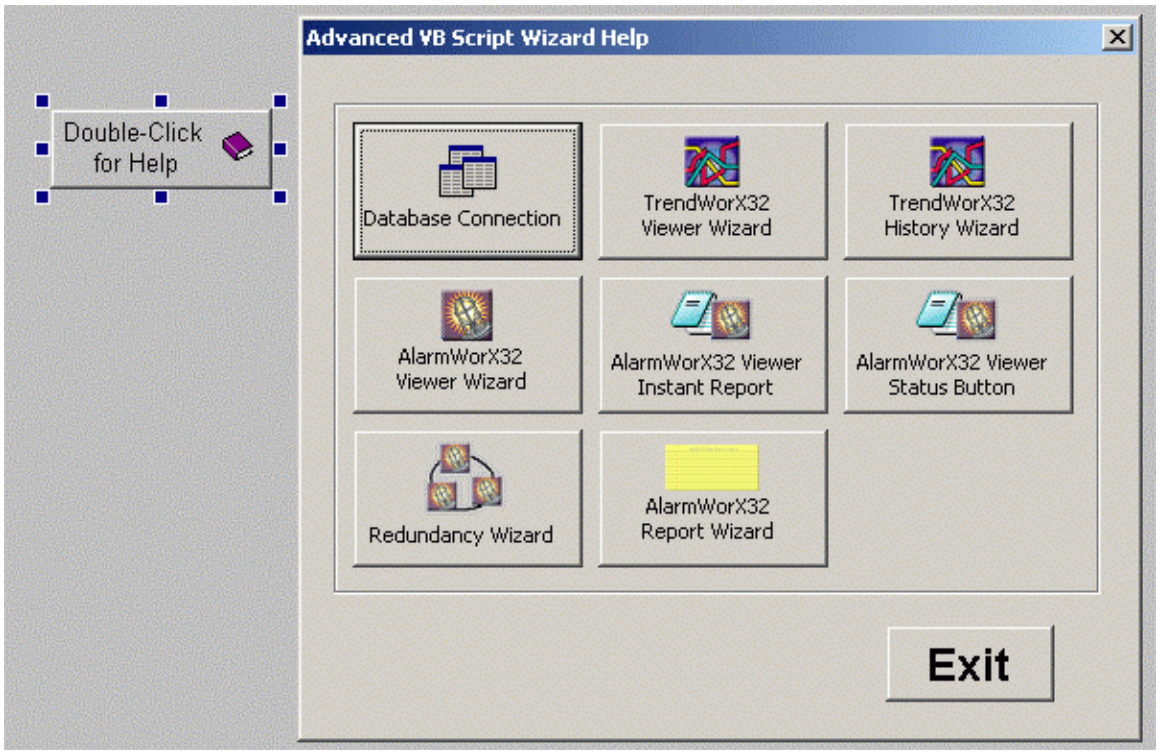
- Script Wizards
- VB Script Symbols
- VBA Symbols
- Automotive
- Building Controls
- Pharmaceutical
- Machine Builders
- Miscellaneous
- Oil and Gas
- Panels
- Touch Screen Symbols
- Training
- Water



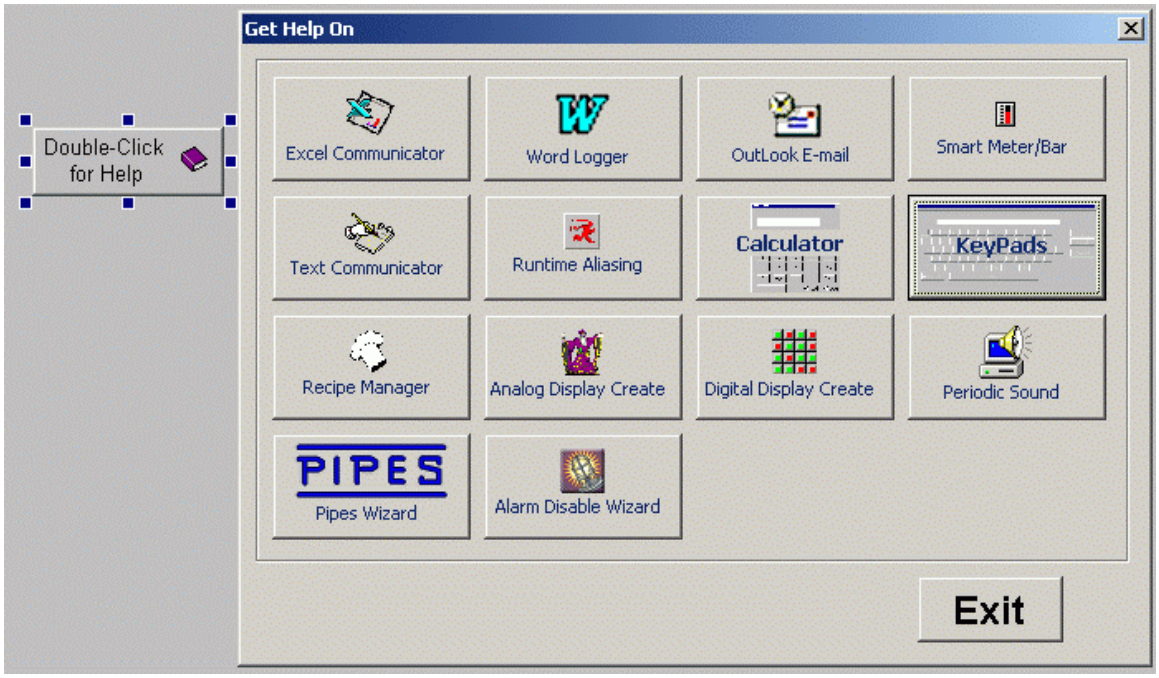
Symbol Categories

Script Wizards

The **Script Wizards** symbol category contains symbols that, when dragged into your GraphWorX32 display, generate various displays (e.g. TrendWorX32 Viewer, AlarmWorX32 Viewer, databases, etc.) using VBScript Wizards and Visual Basic for Applications (VBA) Wizards. Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties. This category also features built-in help symbols that provide information about how to use the wizard symbols. To get help on the wizards, simply drag the Help symbols into your GraphWorX32 display and click on the help button in configuration mode, as shown in the figures below. This opens the **Get Help On** dialog box, which contains a help button for each wizard symbol.



VBScript Wizards Symbols Help

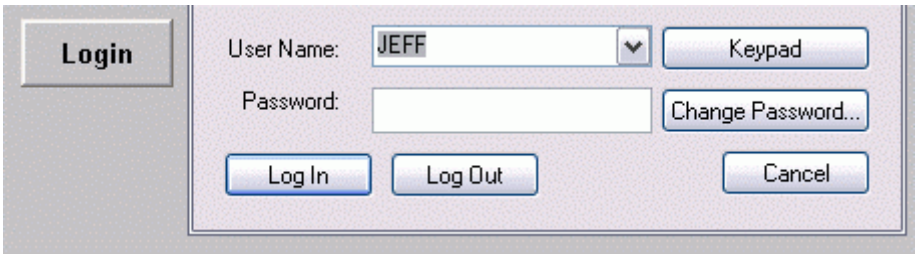


VBA Wizards Symbols Help

VBScript Symbols

The **VBScript Symbols** category contains symbols that, when dragged into your GraphWorX32 display, generate buttons that allow you to start and stop various GENESIS32 applications (e.g. Alarm Logger, GenBroker, etc.), as well as symbols that provide general information about your computer system. Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties. The buttons are activated when GraphWorX32 is in runtime mode.

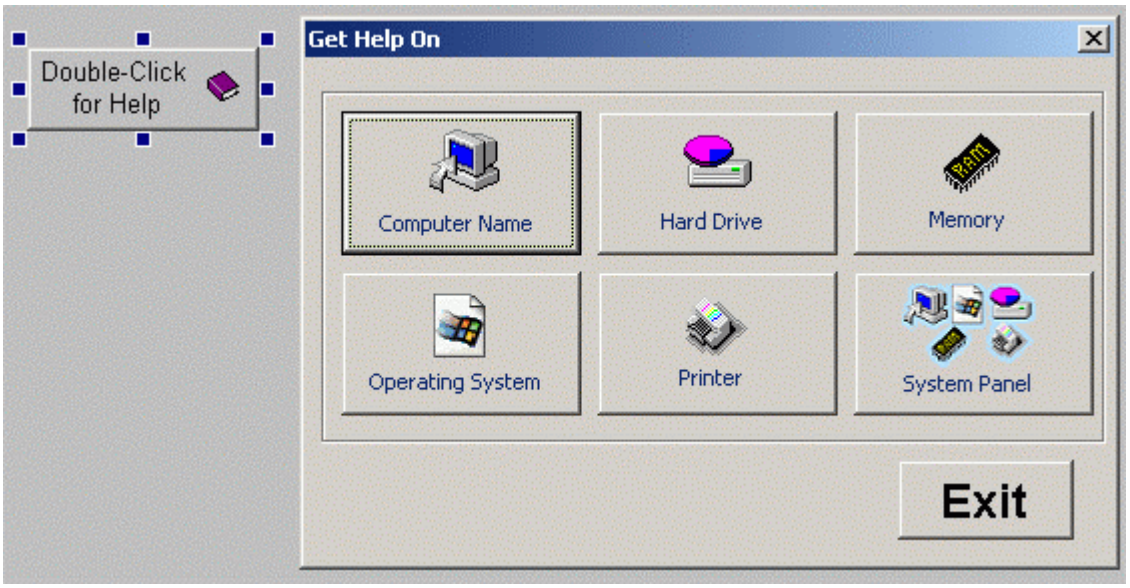
The **VBScript Symbols** category also contains a category file called "WebHMI Security Login.sdf," which contains several symbols that, when dragged into a GraphWorX32 display, allow users to gain access to the Security Server. The **WebHMI Security Login ActiveX** symbol enables WebHMI users on remote client machines to log in to the Security Server. For example, if the "Login" symbol button is placed in a GraphWorX32 display, the user can simply click on the symbol (button) in runtime mode to launch the Security Login dialog box, as shown in the figure below. Users can also log out and change their passwords. For more information, please see the Security Server or WebHMI help documentation.



Logging Into the Security Server

VBA Symbols

The **VBA Symbols** symbol category contains symbols that, when dragged into your GraphWorX32 display, generate buttons that provide general information about your computer system (e.g. available hard disk space an memory) as well as buttons that allow you to execute general Windows system commands (e.g. shut down your computer). The buttons are activated when GraphWorX32 is in runtime mode. The **VBA System Info Symbols** category also features built-in help symbols that provide information about how to use the symbols. To get help, simply drag the Help symbol into your GraphWorX32 display and click on the help button in configuration mode, as shown in the figure below. This opens the **Get Help On** dialog box, which contains a help button for each symbol in the category.



VBA System Information Symbols Help

Automotive Symbols

The **Automotive** symbols category contains standard graphical symbols for robotics and robot control panels. Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Building Control Symbols

The **Building Controls** symbols category contains standard graphical symbols for building control panels (e.g. coils, valves, HVAC, sensors, ducts, etc.). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Pharmaceutical Symbols

The **Pharmaceuticals** symbols category contains standard graphical symbols for food processing and manufacturing operations (e.g. ovens, stirrers, etc.). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Machine Builder Symbols

The **Machine Builders** symbols category contains standard graphical symbols for various machines (e.g. pulp and paper). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Miscellaneous Symbols

The **Miscellaneous** symbols category contains many additional graphical symbol categories for various industries, machines, and machine parts (e.g. tanks, pipes, valves, electrical, mining, motors, pumps, etc.). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Oil and Gas Symbols

The **Oil and Gas** symbols category contains several graphical symbols for equipment used in oil and gas operations (e.g. boilers, valves, tanks, heaters, etc.). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Panels Symbols

The **Panels** symbols category contains several graphical symbols for operator control panels. Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Touch Screen Symbols

The **Touch Screen** symbols category contains several graphical symbols ideal for operator control panels using touch-screen systems for their AlarmWorX32 and TrendWorX32 Viewer displays. The symbols include large buttons for navigating Viewer displays (e.g. acknowledge alarms, alarm ToolTips, scrolling, freezing, etc.). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Training Symbols

The **Training** symbols category contains several graphical symbols used in the GENESIS32 training course curriculum. For more information about training classes, please go to the ICONICS Web site at www.iconics.com.

Water Symbols

The **Water** symbols category contains several graphical symbols for equipment used in water and wastewater operations (e.g. static water treatment, smart water treatment, and water components.). Simply drag a symbol into your GraphWorX32 display and double-click the symbol to open the Property Inspector, where you can edit the symbol's properties.

Recipe Symbols

The **Recipe** symbols category contains graphical symbols (preconfigured recipe pick action buttons, recipe grid, and recipe settings) that allow you to easily visualize recipes. For complete information about recipe configuration and visualization, please see the Recipe Management System section in the GraphWorX32 help documentation.

Dockable Symbol Toolbar in GraphWorX32

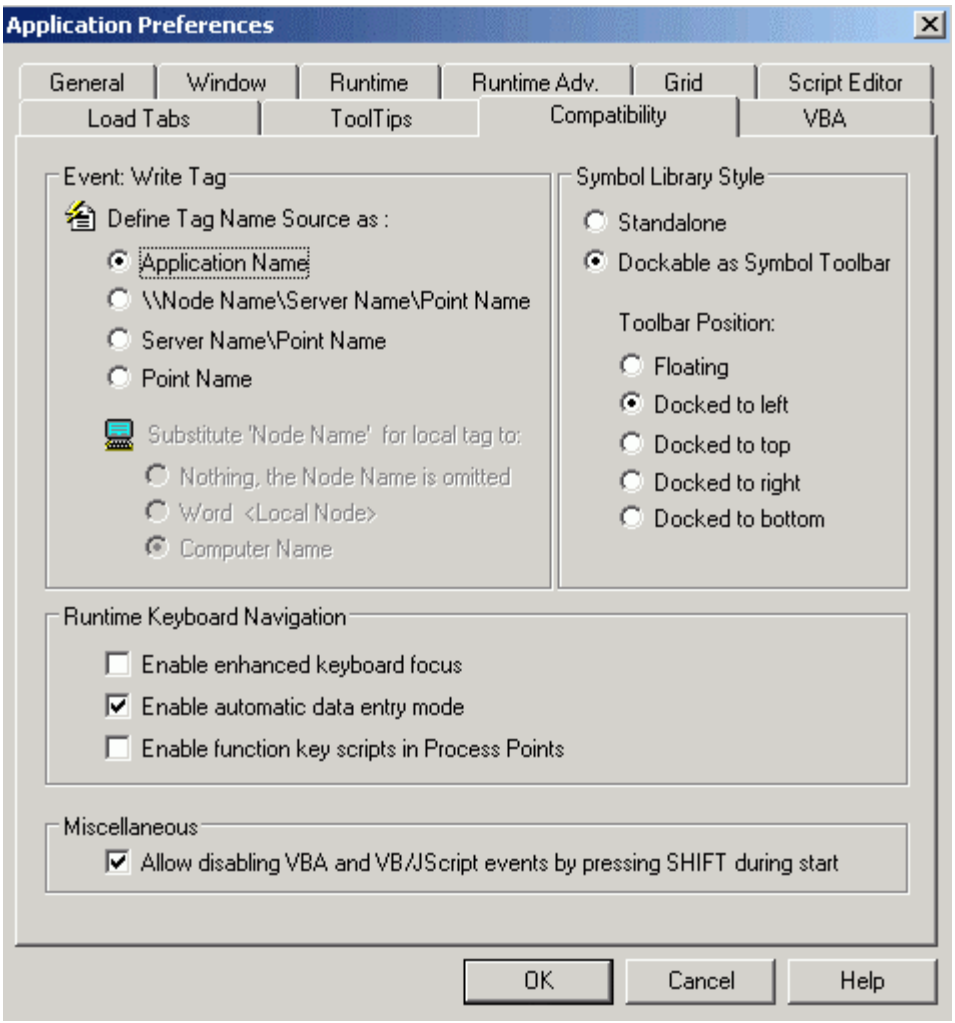
The Symbol Library is available in GraphWorX32 as a dockable toolbar or a floating window inside display files. It can be freely floating above a GraphWorX32 application or docked to any side of the GraphWorX32 display: left, top, right, or bottom, as shown in the figure below. Features of the Symbol Library are fully implemented in the Dockable Symbol Toolbar. You can create, rename, and delete both category (.sdf) files and symbols within the dockable toolbar, as well as unlock categories with passwords and change the look of the symbol icons in the view pane. The category (.sdf) file format of the Dockable Symbol Toolbar is fully compatible with previous versions of the Symbol Library, and vice versa.

GraphWorX32 uses two different modes for the Symbol Library:

- Standard stand-alone Symbol Library
- Dockable symbol toolbar

To switch between these two modes:

1. Select **Application Preferences** from the **Format** menu.
2. The **Application Preferences** dialog box appears, as shown in the figure below. Select the **Compatibility** tab. Under the **Symbol Library Style** field, you can select **Stand-alone** mode or **Dockable as Symbol Toolbar** mode.

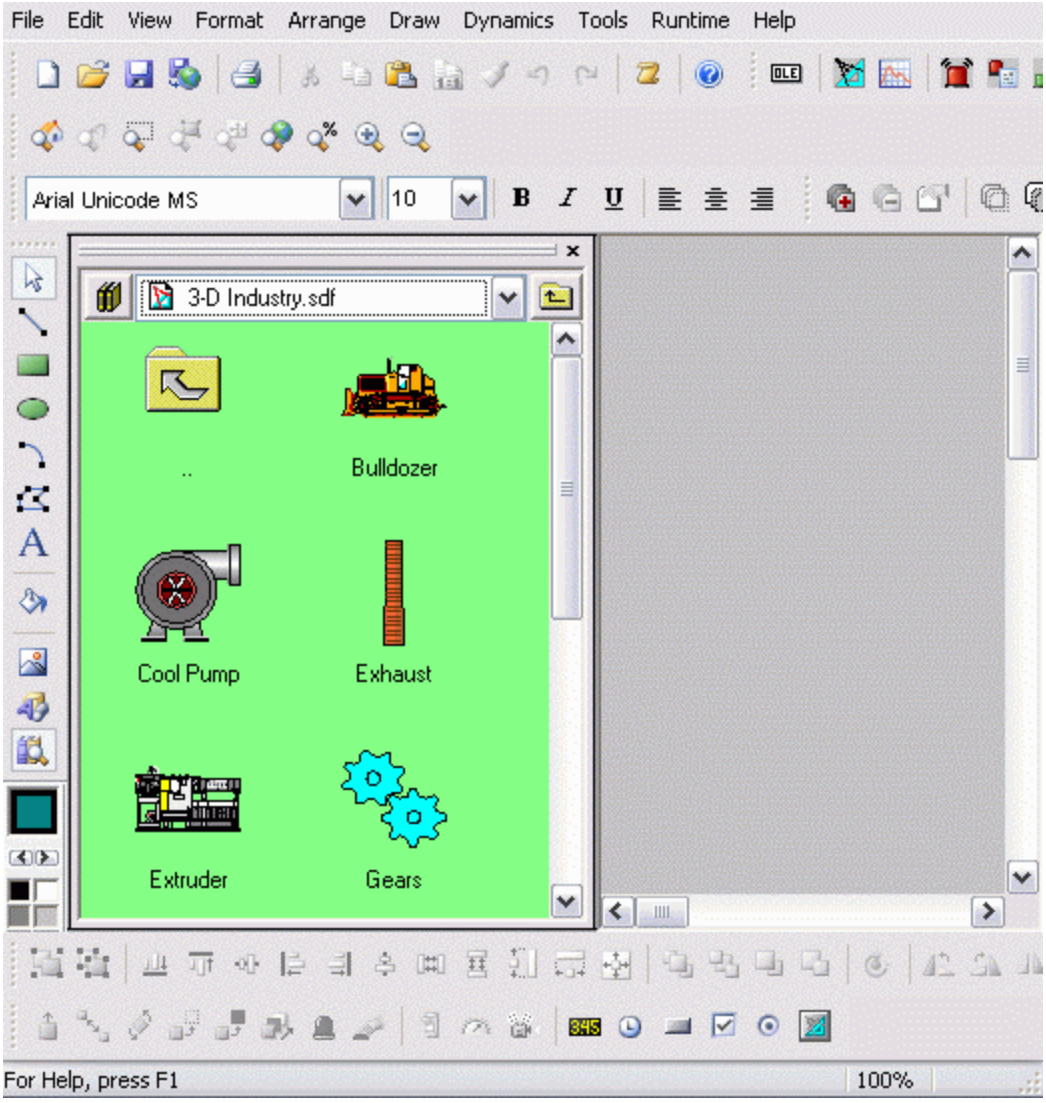


Switching Between Stand-alone and Dockable Symbol Modes

If **Stand-alone** mode is active, clicking the **Symbols** button on the **Draw** toolbar launches the **SymbolLibrary.exe.** You can also select **Import > Symbol** from the **Draw** menu, as shown in the figure below.

If the **Dockable as Symbol Toolbar** option is selected, then the Symbol Library is replaced with a Symbol toolbar inside the GraphWorX32 display, as shown in the figure below. You can also position the symbol toolbar as:

- A floating window
- Docked to left
- Docked to right
- Docked to bottom
- Docked to top



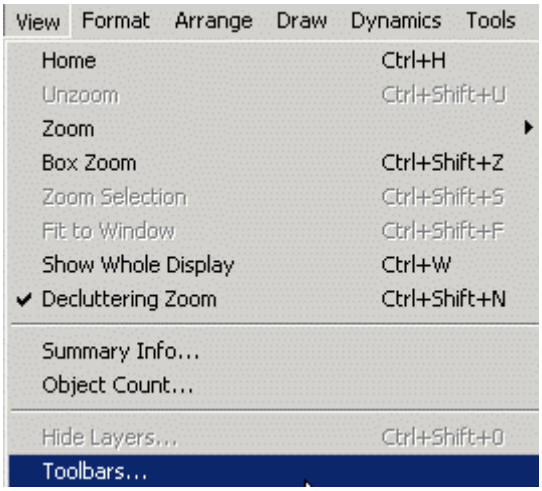
Symbol Library Docked Inside a GraphWorX32 Display

The **Symbols** button on the **Draw** toolbar toggles the visibility of the symbol toolbar (instead of launching the Symbol Library). The **Toggle Symbol Toolbar** command on the **View** menu is enabled, and the **Show Toolbars** dialog (which appears when you select **Toolbars** from the **View** menu) also includes a **Symbols** check box for showing/hiding the symbol toolbar. The **Import > Symbol** command is removed from the **Draw** menu.

Viewing the Dockable Symbol Toolbar in GraphWorX32

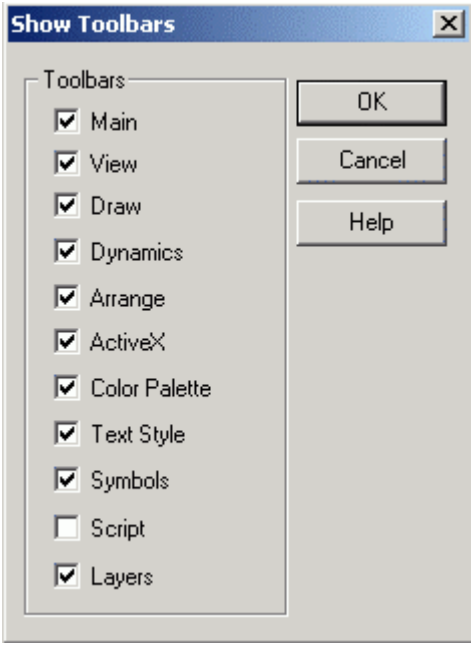
To view the Dockable Symbol Toolbar inside a GraphWorX32 display:

1. Select **Toolbars** from the GraphWorX32 **View** menu, as shown in the figure below.



Selecting Toolbars in GraphWorX32

2. In the **Show Toolbars** dialog box, check **Symbols,** as shown in the figure below. Click **OK.**

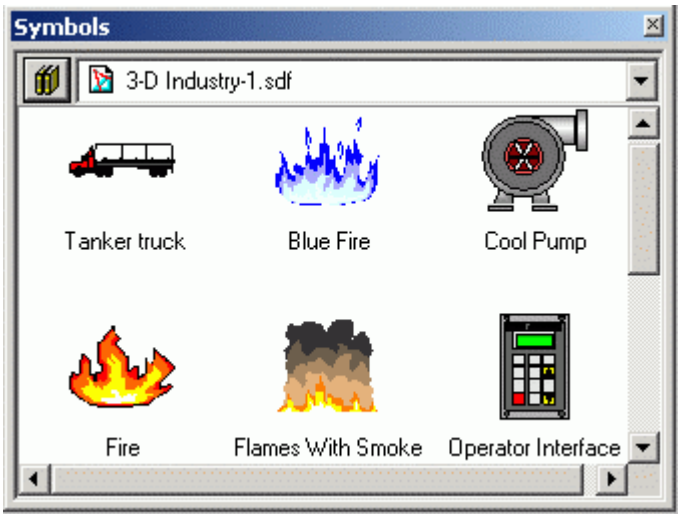


Selecting the Symbols Dockable Toolbar

3. This displays the Dockable Symbol Toolbar in the GraphWorX32 display.

Features of the Dockable Symbol Toolbar

The figure below shows the Dockable Symbol Library as a floating window. The drop-down list at the top of the window allows you to select from all available category (.sdf) files. GraphWorX32 symbols that are contained in the selected category file are displayed in the list control pane, as shown in the figure below. For maintenance and manipulation of symbols and categories, two pop-up menus are available.

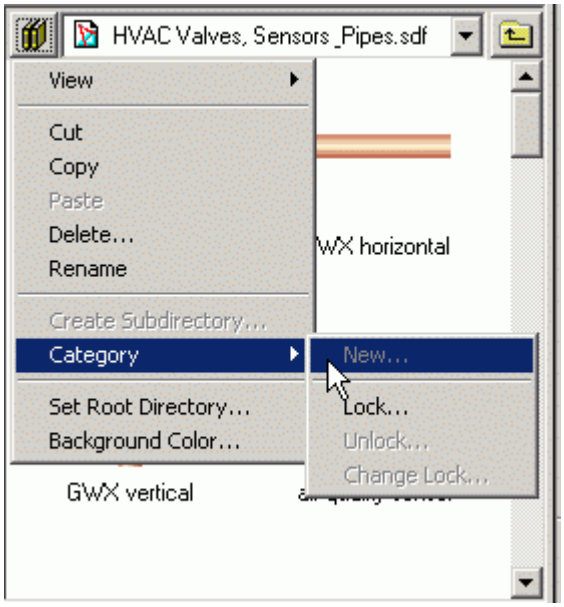


Dockable Symbol Library as a Floating Window

Symbol Toolbar Menu

The main pop-up menu for the Docking Symbol Toolbar is displayed when upper-left button is clicked, as shown in the figure below. You can also right-click anywhere inside the window to access this menu. This pop-up menu allows you to add, rename, delete, lock and unlock category files, as well as change the view of the symbol icons in the view pane. You can also cut, copy, paste, delete, and rename the symbols. The main pop-up menu also allows you to specify the root directory for the .sdf files. Previous versions of the Symbol Library maintained a tree control hierarchy of category files; in other words, category files could be placed under different subdirectories. The Dockable Symbol Library lists all category files as upper-level categories (even those included in subdirectories) and tracks where particular a category file came from so that no category files are lost.

Note: Please see the Symbol Library Help for complete information about menu functions.



Main Pop-up Menu in Dockable Symbol Toolbar

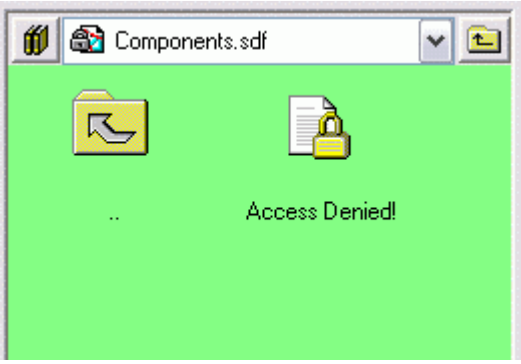
Main Pop-up Menu Commands

Command	Function
View	Specifies the size of the symbol icons in the selected category.
Cut	Removes the selected symbol from the list view.
Copy	Copies the selected symbol.
Paste	Pastes a symbol into the list view.
Delete	Deletes the selected symbol or symbol category.
Rename	Renames the selected symbol or symbol category.
Set Root Directory	Specifies the target directory for the symbol category (.sdf) files.
Create Subdirectory	Creates a new subdirectory (folder) in the root directory for the symbol category (.sdf) files.
Category	Adds, renames, deletes, locks, and unlocks symbol category files.
New	Creates a new category file.
Lock	Sets up the read-write password in the **Set Category Write Mode Password** dialog box. The password must be confirmed. Sets the read-write password to the category, which currently has no

	locks set, or the read-only password for the category, which is already opened in read-write mode. You cannot set the read-only password without setting the read-write password; this would result in a category file lock that no one would be able to remove (to modify the category file).
Unlock	Opens a locked symbol category. When the category file is initially locked, it is always locked with read-write access denied. (It could also have the read-only lock set). Simply enter the read-write password (or the read-only password if it is configured) to open the file. When the password is entered, it is recognized as either read-write or read-only. If the password is accepted, the category file transitions to the proper state to reflect the change.
Change Lock	Changes the category file password. You must be granted read-write permission to open the category with the read-write password.
Background Color	Changes the background color of the list view pane.

Category Locks

It is possible to lock any category file included in the drop-down list for writing to and even for reading from the symbol category. If particular category is locked, it is indicated by different category icon in the list box, as shown in the figure below. Also, if the category file is locked both for reading and writing, a message will appear on the Dockable Symbol Toolbar notifying the user about the lock.



Locked Symbol Category File Indicated

Symbol Library OLE Automation Reference

GraphWorX32 Display (configuration mode only)

- **Function InsertLibraryObjectByIndex(category As String, index As Long) As Boolean**

```
If Not ThisDisplay.InsertLibraryObjectByIndex("Misc", 0) Then
  MsgBox "Error with InsertLibraryObjectByIndex!"
Else
  Msgbox "Library object successfully inserted!"
End If
```

- **Function InsertLibraryObjectByName(category As String, symbol As String) As Boolean**

```
If Not ThisDisplay.InsertLibraryObjectByName("Misc", "Check")
Then
  MsgBox "Error with InsertLibraryObjectByName!"
Else
  Msgbox "Library object successfully inserted!"
End If
```

- **Function InsertLibraryObject(libraryObject As Object) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

Dim symbol As Object
Set symbol = sc.GetSymbolFromName("Check")
If symbol Is Nothing Then
  MsgBox "Failed to get symbol from name!"
  Exit Sub
```

```
End If

If Not ThisDisplay.InsertLibraryObject(symbol) Then
  MsgBox "Error with InsertLibraryObject!"
Else
  Msgbox "Library object successfully inserted!"
End If
```

- **Function ConvertGwxSymbolToLibraryObject(gwxSymbol As Object, libraryObject As Object) As Boolean**

```
Dim newSym As Object
Dim libObj As Object
If Not ThisDisplay.InsertLibraryObjectByName("Misc", "Check")
Then
  MsgBox "Error with InsertLibraryObjectByName!"
End If
Set newSym = ThisDisplay.GetHeadObject
MsgBox newSym.GetObjectTypeName

If Not ThisDisplay.ConvertGwxSymbolToLibraryObject(newSym,
libObj) Then
  MsgBox "Error converting symbol to library object!"
Else
  Msgbox "GwxSymbol successfully converted to library
object!"
End If
```

SymbolLibrary Interface

- **AlwaysOnTop As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

MsgBox sl.AlwaysOnTop
sl.AlwaysOnTop = Not sl.AlwaysOnTop
MsgBox sl.AlwaysOnTop
'Should display "False" and then "True"
```

- **Function AddNewCategory(category As String) As Object**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.AddNewCategory("Test")
If sc Is Nothing Then
  MsgBox "Symbol Category not added!"
  Exit Sub
Else
  Msgbox "Category successfully added!"
End If
```

- **Function RenameCategory(oldCategoryName As String, newCategoryName As String) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

If Not sl.RenameCategory("Test", "Test1") Then
  MsgBox "Error renaming category! "
  Exit Sub
Else
  Msgbox "Category successfully renamed!"
End If
```

- **Function DeleteCategory(category As String) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

If Not sl.DeleteCategory("Test1") Then
  MsgBox "Error deleting symbol category!"
```

```
Else
  MsgBox "Category successfully deleted!"
End If
```

- **Function GetNumberOfCategories() As Long**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

MsgBox "Number of categories: " & sl.GetNumberOfCategories
```

- **Function GetCategoryFromIndex(index As Long) As Object**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromIndex(0)
If sc Is Nothing Then
  MsgBox "Failed to get category from index!"
  Exit Sub
Else
  MsgBox "Category name: " & sc.GetCategoryName
End If
```

- **Function GetCategoryFromName(category As String) As Object**

```
(See InsertLibraryObject)
```

- **Function SetRootDirectory(directory As String) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

If Not sl.SetRootDirectory("C:\Program Files\ICONICS") Then
  MsgBox "Error setting root directory!"
Else
  MsgBox "Root directory has been set!"
  'Verify that it has actually been set correctly
  'by reopening the Symbol Library
```

```
End If
```

- **Function GetRootDirectory() As String**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

MsgBox sl.GetRootDirectory
```

- **Visible As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

MsgBox sl.Visible
sl.Visible = Not sl.Visible
MsgBox sl.Visible
'Should display "False" and then "True"
```

SymbolCategory Interface

- **EdgeEffect As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

MsgBox "Navigate to the Misc category and verify that the"
    & vbCr & "edge effect is opposite " & sc.EdgeEffect & "."
```

```
sl.Visible = True
sl.AlwaysOnTop = True
sc.EdgeEffect = Not sc.EdgeEffect
MsgBox sc.EdgeEffect
```

- **ImageSize As Integer**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

sc.ImageSize = BigSize
MsgBox sc.ImageSize        'Should be 80
sc.ImageSize = MediumSize
MsgBox sc.ImageSize        'Should be 60
sc.ImageSize = SmallSize
MsgBox sc.ImageSize        'Should be 40
```

- **Function AddSymbol(symbol As Object) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

Dim symbol As Object
Set symbol = sc.GetSymbolFromName("Check")
If symbol Is Nothing Then
  MsgBox "Failed to get symbol from name!"
```

```
  Exit Sub
End If

Set sc = sl.GetCategoryFromName("Test1")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

If Not sc.AddSymbol(symbol) Then
  MsgBox "Error with AddSymbol!"
Else
  MsgBox "Symbol successfully added!"
End If
```

- **Function RenameSymbol(oldName As String, newName As String) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If


Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Test1")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

If Not sc.RenameSymbol("Check", "NewCheck") Then
  MsgBox "Error with RenameSymbol!"
Else
  MsgBox "Symbol successfully renamed!"
End If
```

- **Function DeleteSymbol(symbolName As String) As Boolean**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If


Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Test1")
If sc Is Nothing Then
```

```
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

If Not sc.DeleteSymbol("NewCheck") Then
  MsgBox "Error with DeleteSymbol!"
Else
  MsgBox "Symbol successfully deleted!"
End If
```

- **Function GetNumberOfSymbols() As Long**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

MsgBox "Number of symbols in Misc category: " &
sc.GetNumberOfSymbols
```

- **Function GetSymbolFromIndex(index As Long) As Object**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

Dim symbol As Object
Set symbol = sc.GetSymbolFromIndex(0)
If symbol Is Nothing Then
  MsgBox "Failed to get symbol from name!"
```

```
  Exit Sub
Else
  MsgBox "Successfully retrieved symbol!"
End If
```

- **Function GetSymbolFromName(symbol As String) As Object**

```
(Tested in InsertLibraryObject)
```

- **Function GetSymbolName(index As Long) As String**

```
Dim sl As SymbolLibrary.SymbolLibrary
Set sl = New SymbolLibrary.SymbolLibrary
If sl Is Nothing Then
  MsgBox "Symbol Library not instantiated!"
  Exit Sub
End If

Dim sc As SymbolCategory
Set sc = sl.GetCategoryFromName("Misc")
If sc Is Nothing Then
  MsgBox "Failed to get category from name!"
  Exit Sub
End If

Dim symname As String
symname = sc.GetSymbolName(0)
If symname = "" Then
  MsgBox "Symbol doesn't exist or has no name!"
  Exit Sub
Else
  MsgBox "Symbol Name: " & symname
End If
```

- **Function GetCategoryName(index As Long) As String**

```
Dim sl As SymbolLibrary.SymbolLibrary
   Set sl = New SymbolLibrary.SymbolLibrary
   If sl Is Nothing Then
     MsgBox "Symbol Library not instantiated!"
     Exit Sub
   End If

   Dim sc As SymbolCategory
   Set sc = sl.GetCategoryFromIndex(1)
   If sc Is Nothing Then
     MsgBox "Failed to get category from index!"
     Exit Sub
```

```
End If
MsgBox "Category name: " & sc.GetCategoryName
```