AVEVA™
InTouch HMI
formerly Wonderware

Visualization Guide

Publication date: Wednesday, November 11, 2020

Contact Information

AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

Contents

CHAPTER 1

WindowMaker: Your Development Environment

About Your Development Environment

WindowMaker is the development environment you use to create InTouch HMI applications. You use WindowMaker to create the visual interface used by operators to view and manage your manufacturing processes. An InTouch HMI interface shows data from and writes data back to the production environment.

You can configure the following visual interface elements of your InTouch applications with WindowMaker:

- Windows contain interactive visual elements for plant operators to manage a production process.

- Basic objects are simple graphical elements, such as rectangles, circles, lines, and text.

- User-defined complex objects consist of one or more basic objects that represent elements in your production environment, such as valves and tanks.

- Pre-defined complex objects perform specific functions, such as alarming, historical trending, or Symbol Wizards.

- Animation links are properties of simple and complex objects to animate their appearance and transmit user input to tasks and production data changes.

- Wizards are pre-defined complex objects that perform specific functions or have specific appearance, such as sliders and meters.

- ActiveX controls are controls that you can place in InTouch windows to perform specific functions such as to show current alarms.

The main parts of the development environment are:

- Toolbars and status bar

- Classic view, which shows the scripts and tools.

- Project view, which shows the windows and scripts in a different format.

- Development area

- Fast switch buttons that open WindowViewer and the Web Client

- Boundary canvas to show the application within the specified target resolution if different than your screen resolution. Border style is not configurable.

The following graphic shows WindowMaker work areas:
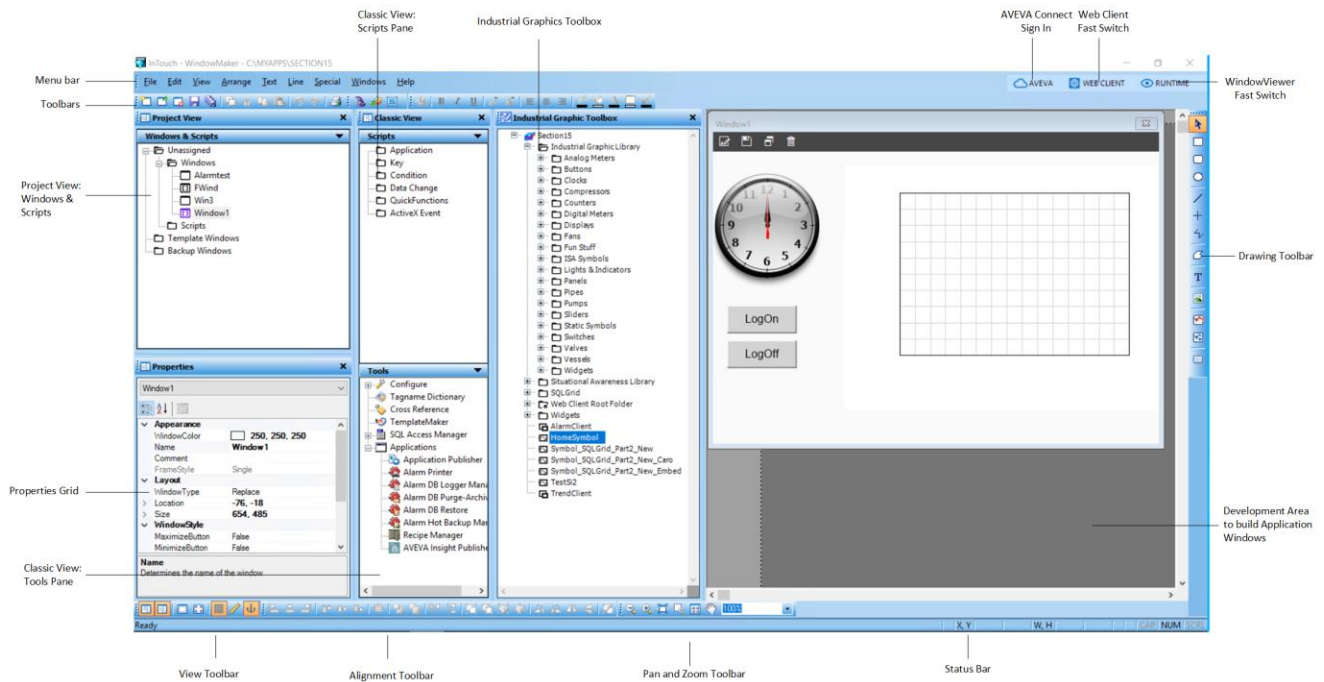
Setting Your WindowMaker Preferences

Using the WindowMaker Properties dialog box, you can configure preferences and options affecting the behavior of WindowMaker. You can:

- Change the title bar text.

- Display the grid or turn off the grid.

- Change the spacing between the pixels on the grid.

- Show the tag count.

- Change the default fonts for text and buttons.

- Set the precision for line selection.

- Set an option to close WindowMaker when switching to WindowViewer.

- Set an option to pick through hollow objects.

- Enable fast switch from WindowMaker to WindowViewer.

- Set the number of undo levels.

To set the properties for WindowMaker

1. On the **Special** menu, point to **Configure** and then click **WindowMaker**.The **WindowMaker Properties** dialog box appears.
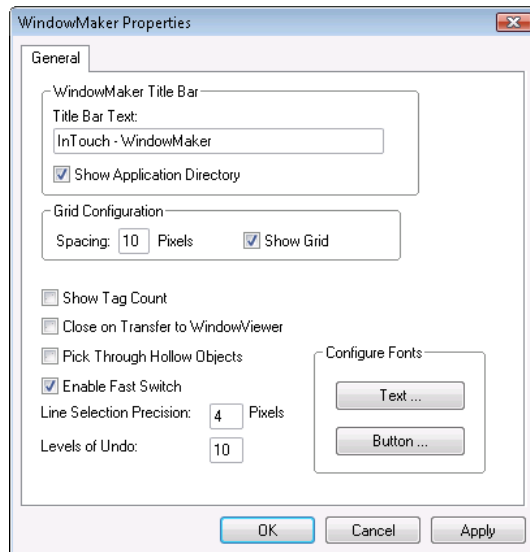
2. In the **WindowMaker Title Bar** area, configure the appearance of the title bar. Do any of the following:

   o In the **Title Bar Text** box, type the text to appear in the title bar during design time.

   o Select the **Show Application Directory** check box to include the path to the application folder in the title bar.

3. In the **Grid Configuration** area, configure the background grid. Do any of the following:

   o In the **Spacing** box, type the number of pixels between the grid coordinates.

   o Select the **Show Grid** check box to show the grid.

4. Configure miscellaneous window properties. Do any of the following:

   o Select the **Show Tag Count** check box to show the number of tagnames in your Tagname Dictionary in the menu bar. If you have a lot of tags, showing the tag count can impact the Tagname Dictionary performance.

   This is useful if you are creating an application with a limited Tagname Dictionary size. The tagname count does not include remote tagname references or system tags. Click **Update Use Counts** on the **Special** menu to find out your remote tagname reference usage.

   o Select the **Close on Transfer to WindowViewer** check box to close WindowMaker automatically when you start WindowViewer.

   The purpose of this option is to conserve limited memory. If memory is not an issue and you are moving often between WindowViewer and WindowMaker, do not select this option.

   When you select **Close on Transfer to WindowViewer**, the reciprocal command, **Close WindowViewer**, on the **General Properties** tab in the **WindowViewer Properties** dialog box is also selected.

   o Select the **Pick Through Hollow Objects** check box to select objects that are behind hollow objects.

   This enables you to select an object within a frame without having to send the frame to the background.

o Select the **Enable Fast Switch** check box to use the "fast switch" to toggle between WindowMaker and WindowViewer.

The fast switch is the word **Runtime** that appears in the upper right corner of WindowMaker. In WindowViewer, it is the word **Development**.

When you use the fast switch, WindowMaker automatically saves all changes made to all open windows before switching to WindowViewer.

o In the **Line Selection Precision** box, type the number of pixels your cursor can be away from a line and still be able to select it.

In most cases, the default setting of 4 works well.

o In the **Levels of Undo** box, type the number of **Undo** and **Redo** levels to maintain.

You can have up to 25 levels. If you type zero, the undo/redo function is turned off.

One level represents one action. The **Undo** and **Redo** stacks are empty when you create a new window or open an existing window. Both stacks are emptied when you close a window.

5. In the **Configure Fonts** area, click either **Text** or **Button** to set the default text font or button font. Select the font default, and then click **OK**.

You can override these defaults in any window by using the **Font** toolbar.

6. Click **OK**.

7. Restart WindowMaker to apply any changes you made.

Using the Screen Grid and Ruler

You can show a grid and ruler in the Development area to help you arrange and align objects.
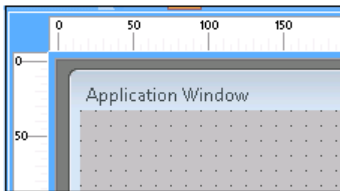
Snapping Objects to the Grid

You can make your objects snap to predefined grid points by selecting **Snap to Grid**.

By default, the grid is set to 10 pixels and to be visible when you start WindowMaker. You configure the pixel interval for the grid in the **WindowMaker Properties** dialog box.

To see the grid, you must select **Show Grid** on the **WindowMaker Properties** dialog box and select **Snap to Grid** on the **Arrange** menu.

To configure the grid

1. On the **Special** menu, point to **Configure** then click **WindowMaker**. The **WindowMaker Properties** dialog box appears.

2. In the **Spacing** box, type the number of pixels to space between coordinates.

3. Select the **Show Grid** check box if you want to see the grid when **Snap to Grid** is selected.

If you do not select **Show Grid**, no grid is visible in your windows when you select **Snap to Grid**.

Using The Ruler

Use the rulers for precision alignment of the objects in your windows. The rulers appear across the top and along one side of your development environment window.

To show or hide the rulers

1. On the **View** menu, click **Ruler**.

2. Repeat the step to hide the ruler.

Panning and Zooming

You can zoom in and out get a better look at the elements that you are editing to ensure objects line up exactly or are positioned correctly.

The Pan and Zoom toolbar appears by default at the bottom right of the screen. It can be floated or docked in other locations, like the other toolbars.

If you have specified a target resolution that is different than your screen resolution, the boundary canvas will adjust accordingly upon panning or zooming.

You can:

- Zoom in and out from 100% to 500%.

- Zoom to a specific area with the rubber band tool.

- Zoom the window to a specific percentage.

- Click and drag to pan the window.

- Return to the normal default view.

To show or hide the Pan and Zoom toolbar

- On the **View** menu, click **Pan and Zoom**.

See the *InTouch HMI Application Management and Extension Guide* for information about panning and zooming at run time.

Using the Thumbnail Window to Pan and Zoom

When you zoom in to a detail of your application window, the thumbnails window shows you the relationship of the detail to the whole application.

The **Thumbnails** window gives you both an overview and a detail of the development area.

Show or hide the **Thumbnails** window by clicking the **Thumbnails** button.

A red rectangle is the boundary of the zoomed area within a window.
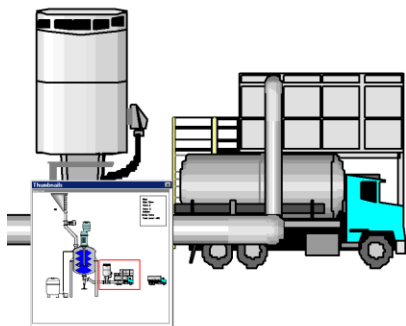
- Drag the red rectangle to show a different part of the window.

- Click on a different area of the window to move the rectangle to that area.

- Resize the rectangle to change the zoom level of the display area.

The thumbnail view shows a white rectangle for objects that you cannot zoom in on, such as an ActiveX control.

If you have specified an application target resolution that is different from your screen resolution, the boundary canvas outlining your specified target resolution dimensions will also appear in the thumbnail view.
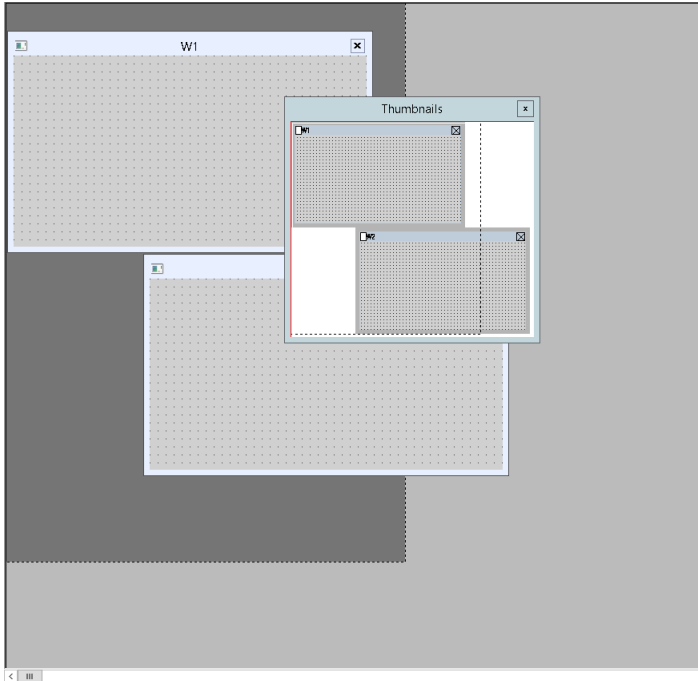
Note: Application windows exceeding the dimensions of the target resolution boundary will still appear in the thumbnail view as seen in the example above.

Using the Mouse Wheel to Zoom and Pan

If your mouse has a wheel, you can press the CTRL key and roll the wheel to change the zoom level of your image.

- As you roll the mouse wheel, each click changes the zoom level by 20%.

- You can also place the cursor in the InTouch window and press the mouse wheel to navigate within the window. When you press the mouse wheel, an icon with four arrows appears. Move the mouse to navigate in the window.

Pan and Zoom Limitations

Panning and zooming do not apply to the following controls:

- ActiveX controls

- The Distributed Alarm Object

- The 16 Pen Trend

- Text boxes

- Check boxes

- List boxes

- Combo boxes

- Radio Group objects

If one of these controls is in the visible area when the view is zoomed more than 100%, a rectangle with the name of the control appears in the area occupied by the control.

Managing Toolbars

You can show or hide any of the toolbars from the **View** menu.

You can move any toolbar from its default docked position to any other location within the development window. Floating toolbars have title bars and they allow you to change their size.

When you start WindowMaker, all toolbars are shown.

You can float or dock a toolbar by dragging it. When you show a docked toolbar that was hidden, it reappears in its last docked location in the window.

To show or hide a toolbar

- On the **View** menu, click the toolbar name.

To change the size of a floating toolbar

1. Move the cursor over any edge of the toolbar. The cursor changes to a double-ended arrow.
2. Drag the edge of the toolbar to move and resize the toolbar.

   As you move the cursor, a box appears to indicate the size the toolbar when you release the mouse button.

To hide all toolbars at once

- On the **View** menu, click **Hide All**.

Using the Application Explorer

The Application Explorer has two views: **Classic View** and **Project View**. These views show you both your InTouch application windows and the tools available to you. Like other toolbars, they can be opened or closed, docked or floating. The **Classic View** and **Project View** can be docked vertically, but not horizontally. The Wizards/ActiveX toolbar can be docked horizontally, but not vertically.
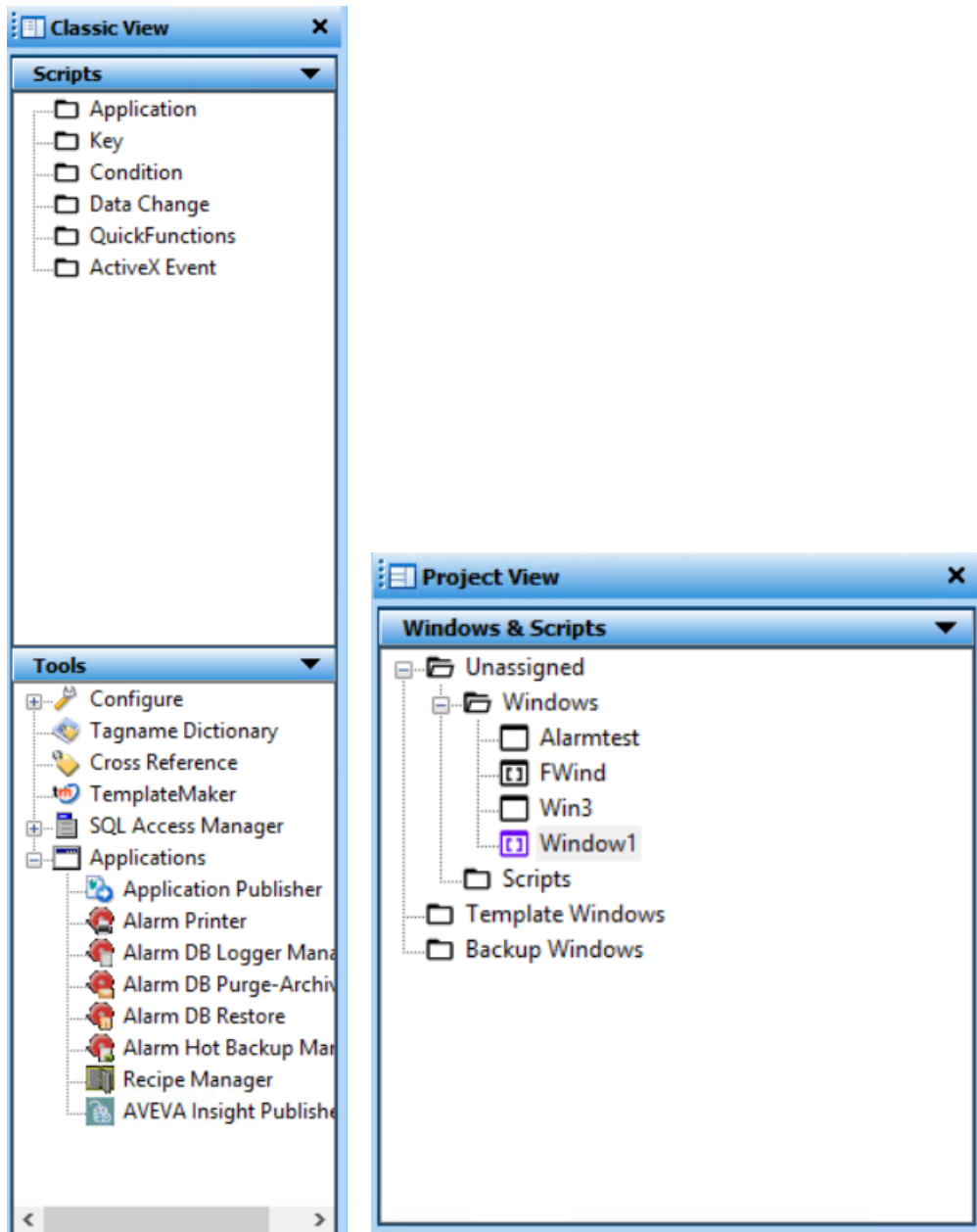
These views give you access to all application windows, scripts, configuration menus, the Tagname Dictionary, and wizards.

Do not add WindowViewer (view.exe) to the **Application Explorer**. The proper way to start WindowViewer is by clicking **WindowViewer** on the **File** menu, or by clicking the **Runtime** fast switch in the toolbar.

Navigating in the Application Explorer

You can expand or collapse the folders in either of the Application Explorer toolbars.

The **Applications** view shows other installed applications.

To expand or collapse the Application Explorer folders

1. Double-click a folder or icon to expand and show the group members.

2. Double-click on a member to open that member.

Adding Applications to the Application Explorer

The WindowMaker Application Explorer can start other applications from within WindowMaker. For example, you can run an I/O Server and configure it at the same time that you are developing your application. You can start third-party programs such as Windows Notepad, Wordpad, Microsoft Excel, Microsoft Word, Microsoft Paint and so on.

You can also configure the Application Explorer to open a specific file such as a document or spreadsheet.

To add a new application to Application Explorer

1. In the **Tools** pane of the **Classic** view, right-click **Applications**, and then click **New**. The **Application Properties** window appears.
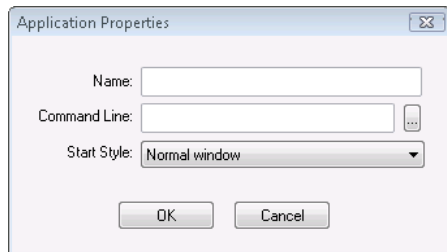
2. In the **Name** box, type the name of the application.

3. In the **Command Line** box, enter the full path for the application. Click the ellipsis button to browse for the application.

   You can add command line parameters for the application in the **Command Line** box.

4. In the **Start Style** list, click how you want the application to appear when it starts up.

5. Click **OK**. The application is added to the Application Explorer under Applications. You can now run the application at any time from WindowMaker.

Using Color Palettes

You can use color palettes to apply color to static and dynamic properties of lines, rectangles, round rectangles, ellipses, polylines, polygons and text. You can select the background color for your windows and the transparent color for bitmaps that enable you to view objects behind bitmaps.

The palette offers you a wide range of color selections, up to 16.7 million colors. The available colors may be limited by your video card capability.

You can also:

- Define and add custom colors.

- Import custom color palettes created in other Windows applications.

- Export your palettes to other Windows applications.

Opening the Color Palette

The color palette appears whenever you click a colored square in a dialog box or you click one of the color tools to apply line, fill, or text color to a selected object.

To open the color palette

1. Click a colored square in a dialog box. The **Standard Palette** appears.
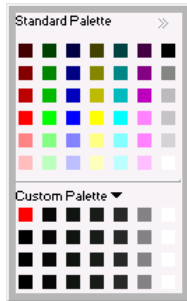
2. Click the right arrows to access the classic colors.

3. Click the color to use. The color palette closes and the color you selected is applied.

Creating Custom Colors

You can create a palette of custom colors.

To create a custom color

1. Open the color palette.

2. In the **Custom Palette** area, right-click one of the blank squares and then click **Edit Custom Color**. The **Add a Color** dialog box appears.
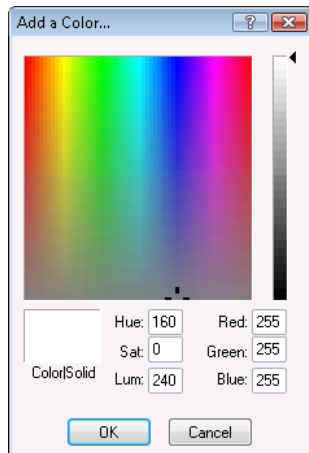
3. Do any of the following:

   o Click anywhere in the color display and use the slider to adjust the color. To specify that you want 100% of the color with no white or black, press ALT + O.

   o Type values in the **Red, Green** and **Blue** boxes to define a color. You can experiment with these values by observing the color matrix. Notice the values for hue, saturation and luminosity also change.

   o Type values in the **Hue**, **Sat**,and **Lum** boxes to define a color. As you change any of these values, the red, green, and blue scales change to match.

   Hue is a discrete color value, where 0 is red, 60 is yellow, 120 is green, 180 is cyan, 200 is magenta, and 240 is blue.

   Saturation is the amount of color in a specified hue, up to a maximum of 240.

   Luminosity is the brightness of a color.

4. View the resulting color in the **Color|Solid** box.

   If your monitor is set to display 256 colors, the **Color|Solid** box might show two colors. The right side shows how the selected color appears as a solid color. The left side shows the dithered color, or the approximation of the specific color using two of the 256 colors.

5. Click **OK**.

You can also create custom colors using the eye dropper tool.

Note: Use this feature for creating transparent bitmaps.

To select a custom color with the eye dropper tool

1. Open the **Color Palette**.

2. Right-click one of the blank squares in the **Custom Palette** section at the bottom of the color palette.

3. Click the eye dropper tool and then click the color that you want to add.

Importing and Exporting Custom Colors

If you defined a custom color palette, you can export it from an InTouch application and then import it into another InTouch application.

To import a custom palette

1. Open the color palette.

2. Click the **Custom Palette** down arrow.

3. Click **Load Palette**. The standard Windows **Open** dialog box appears.

4. Locate and select the .pal palette file with the color definitions you want.

5. Click **Open**. The colors contained in the palette file are loaded into the **Custom Palette**.

To export a custom palette

1. Open the color palette.

2. Click the **Custom Palette** down arrow.

3. Click **Export Palette**. The standard Windows **Save As** dialog box appears.

4. Specify the name for the palette file and click **Save**.

Setting Font Defaults

You can set font defaults for text objects and button objects that have text.

Tip: You can override these defaults in any window by using the toolbar to customize window or button text.

To set the font defaults

1. On the **Special** menu, point to **Configure** and then click **WindowMaker**. The **WindowMaker Properties** dialog box appears.

2. In the **Configure Fonts** area, click either **Text** or **Button** to set the default text font or button font. Select the font default, and then click **OK**.

   You can override these defaults in any window by using the **Fonts** tool in the **Format** toolbar.

3. Click **OK**.

Using Full Screen Mode

Full screen mode hides all program elements except open windows and floating toolbars.

To toggle full screen mode on or off

- On the **View** toolbar, click the **Full Screen** button to switch from normal to full screen mode.

  The **View** toolbar changes to the **Restore** toolbar automatically, and floats on top.

Windows Dialog Box Features

When you are opening, saving, closing, deleting or duplicating windows using the **File** menu, the names of all the windows that are valid candidates for the selected command appear in a list.

Tip: Click **Details** to change from the list view to the details view. The details view includes any **Comments** you made for the window in the **Window Properties** dialog box.

To view the windows and details in a multi-column format

1. Click the **Details** button. A dialog box appears, showing the list of windows.
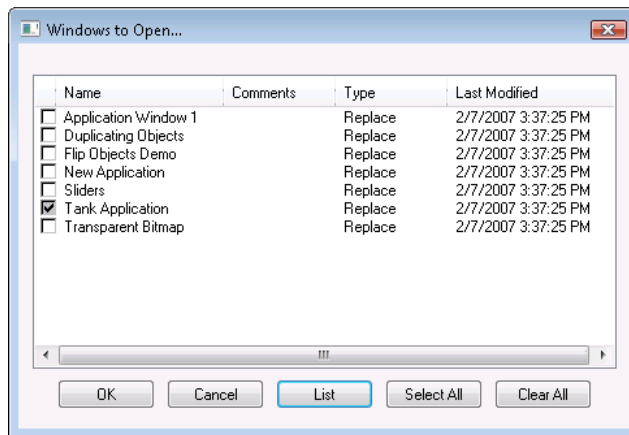
2. Select a window, click on the window a second time to clear it.

   o To select all windows, click **Select All**.

   o To clear all windows, click **Clear All**.

   o To open selected window(s), click **OK**.

Mouse Short Cuts

Use the following short cuts to open dialog boxes and do other common tasks.

To access menu commands for items in WindowMaker

- Right-click on the item. Items include graphic objects, folder names in the views, and so on.

To open the Animation Links dialog box

- Double-click an object or symbol.

To open the Select Tag dialog box (Tag Browser)

- Double-click a blank expression input field within a link definition dialog box. The **Select Tag** dialog box appears.
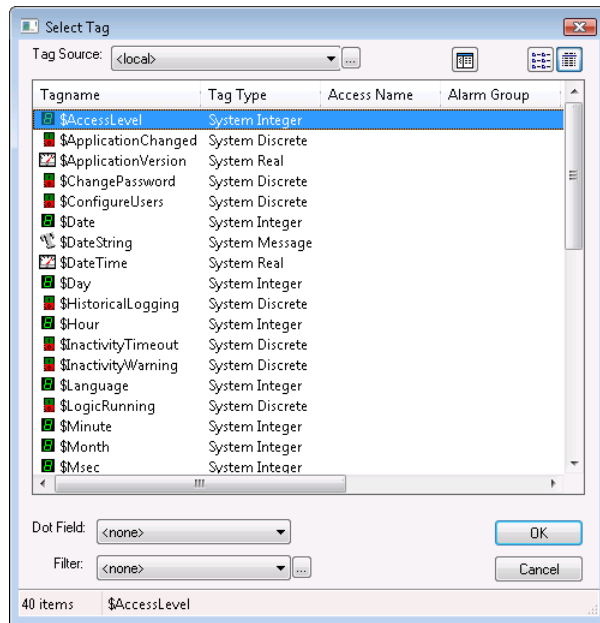
To access the tag dotfields

- In any Tagname or Expression input box, type a tagname plus a period, then double-click to the right of the period. You can also type just a period and double-click to the right of it. The **Choose field name** dialog box appears showing all tagname dotfields.

To open a tagname definition in the Tagname Dictionary

- Double-click the tagname.

Moving Objects with the Arrow Keys

In WindowMaker, you can use the **arrow keys** to move a selected object or group of objects.

When moving objects with the arrow keys, how far an object moves depends upon whether or not the grid is showing.

When the grid is showing, how many pixels an object moves depends upon the grid spacing, which is set on the WindowMaker properties dialog. The default setting is ten pixels between grid points.

When the grid is showing:

- Pressing an arrow key moves the object one grid point.

- Pressing SHIFT + an arrow key moves the object two grid points.

- Pressing CTRL + an arrow key moves the object four grid points.

When the grid is not showing:

- Pressing an arrow key moves the object one pixel.

- Pressing SHIFT + an arrow key moves the object ten pixels.

- Pressing CTRL + an arrow key moves the object 50 pixels.

CHAPTER 2

Application Windows

About Application Windows

An application window is a container for one or more graphics that model your production processes. For example, you can have a window that shows equipment in a unit. Another window can show a grid of alarm information related to that unit.
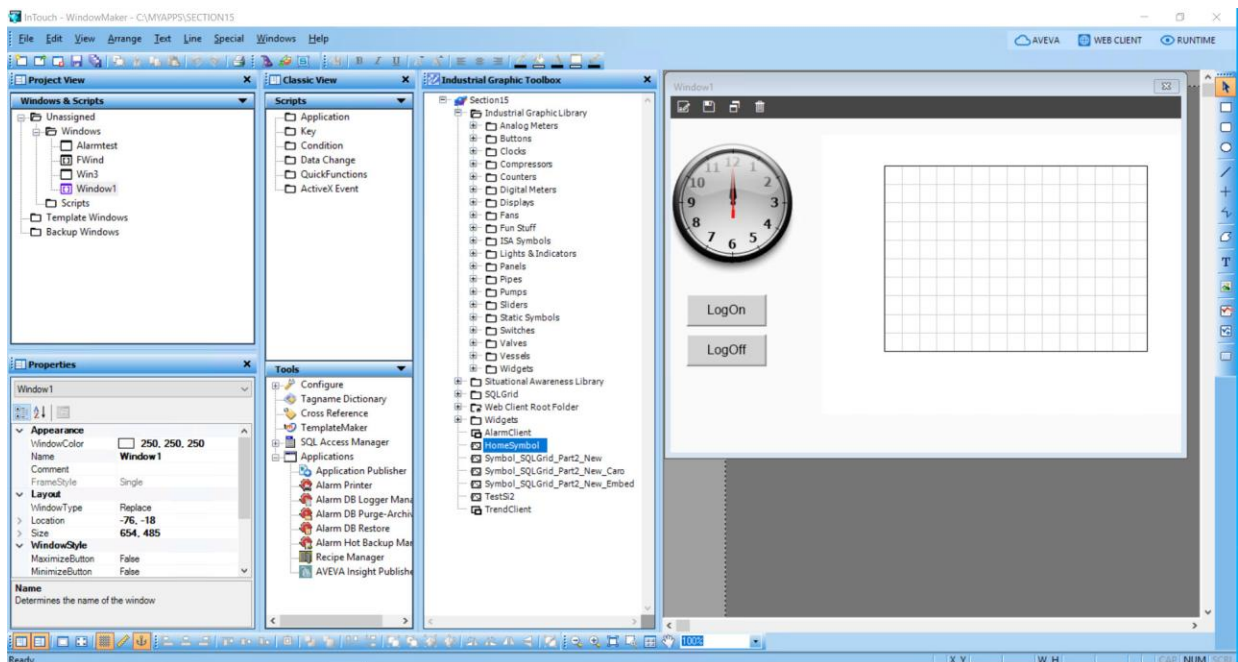
You can create any number of windows and you can define window properties such as background color, screen position, window title, and so on.

Creating Application Windows

When you create a new application window, the default settings reflect those of the previously created or of the currently active InTouch window. The following settings will reflect those of the previously created window:

- Window names: can have up to 32 characters and cannot include any special characters except dollar ($), pound (#) and underscore(_). When you create a window, you are only required to provide a window name. All other items are optional.

Note: If the window name contains unsupported characters, an underscore (_) replaces each unsupported character including the space character. If an application is migrated, editing the window or modifying any of the window properties will replace any special characters in the window name to underscore (_). This is applicable for all types of windows including application, frame and template.
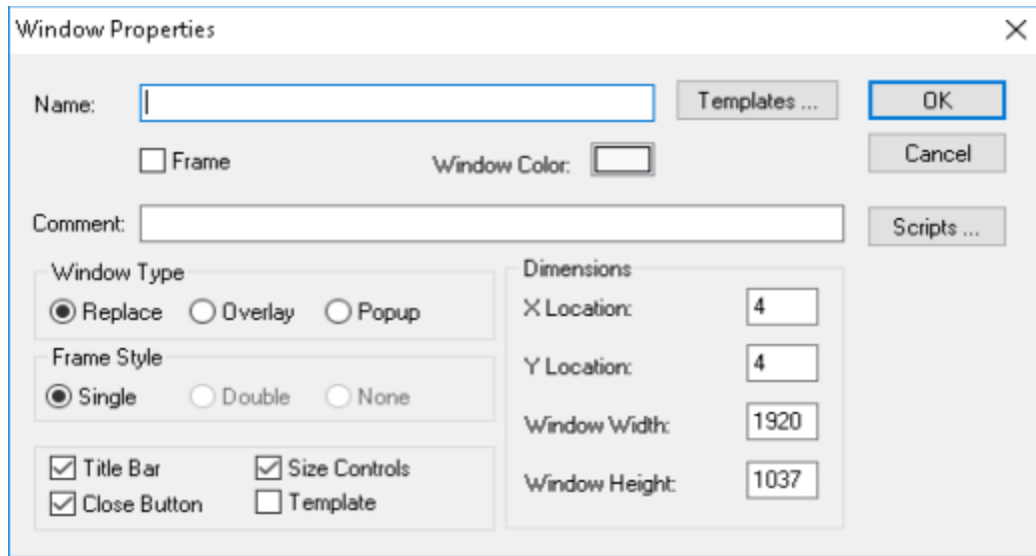
- Comments: can be included for a window, but is for information purposes only. The comment appears in file listings but is not used by the application.

- Window dimension values: by default, the window dimension values are set to the dimensions of the previously created window. These values are also automatically modified if you manually change the window size by dragging the window border.

Note: If you have specified an application target resolution that is different than your screen resolution, your application windows can exceed the canvas boundary. However, only windows within the target resolution canvas boundary will display at Run Time.

To create a new window

1. On the **File** menu, click **New Window**. The **Window Properties** dialog box appears.

2. Configure the basic window properties. Do the following:

   o In the **Name** textbox, type a unique name that identifies the window.

   o In the **Comment** textbox, type any comments you want to associate with the window. The comment must be 50 characters or fewer.

   o Select the **Frame** checkbox to create windows that can be viewed on the web browser.

   o Click the **Window Color** button to select the background color for the window.

3. Configure the **Window Type** to see how the window opens at run time.

   o Select the Window Type as **Replace** for the window to automatically close any windows it intersects with when it appears on the screen.

   o Select the Window Type as **Overlay** for the window to appear on top of currently open windows. It can be larger than the window(s) it is overlaying. When an overlay window closes, any windows behind it reappear. Clicking on any visible portion of a window behind an overlay window brings that window to the foreground as the active window.

   o Select the Window Type as **Popup** for the window to always stay on top of all other windows. Popup windows usually require a response from the user to be removed.

4. Configure **Frame Style** for the window to display a border around the window.

   o Click **Single** for a single, one-dimensional border.

   o Click **Double** for a three-dimensional bordered window.

   o Click **None** for a window with no border.

5. Select the **Size Controls** check box to enable users to resize a window in WindowMaker.

6. Select the **Title Bar** check box to include a title bar.

7. Select the **Close Button** check box to include a button on the title bar to close a window at design time or run time. **Windows retain the close window button even after they are exported to other applications.**

8. In the **Dimensions** area, specify the window location and dimensions. Do the following:

   o In the **X Location** box, type the number of pixels between the left edge of the design area and the left edge of the window being defined.

   o In the **Y Location** box, type the number of pixels between the top edge of the design area and the top edge of the window being defined.

   o In the **Window Width** and **Window Height** boxes, type the window width and height in pixels.

9. Click **OK**.

Setting an Application Window as a Template Window

You can set an application window to be a template window using the Window Properties. This will allow you to reuse windows and reduce configuration time.

To set an application window as a template window:

1. Open the **Window Properties** dialog box by doing one of the following:

   a. Click **File**, **New Window**

   b. Right click an existing application window

2. Select the **Template** checkbox in the **Window Properties** dialog box.
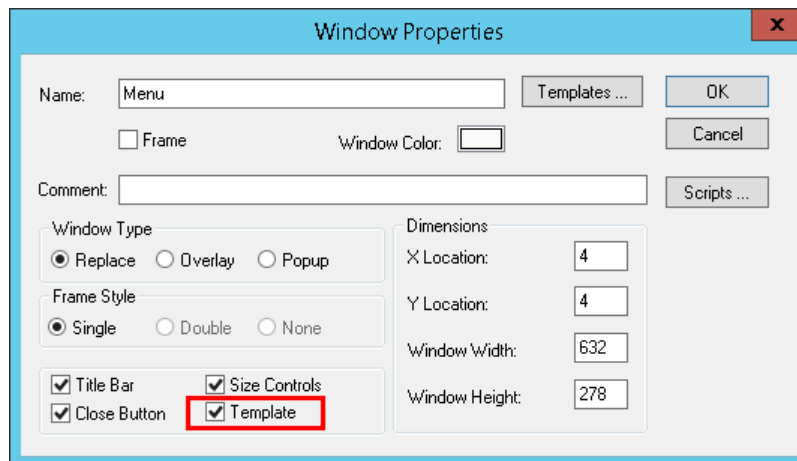
3. Click **OK**.

   The window now appears in the Template Windows folder under the Windows & Scripts pane. It is also available in the Template Windows Browser.

Note: You can change a template window back to an application window by unchecking the Template checkbox in the Window Properties dialog box. You can also drag and drop windows between the Template Windows folders to change this property.

Creating an Application Window from a Template Window

You can create an InTouch application window from any of the template windows available in the Template Windows Browser.

Note: Only windows with the Template property selected will be available in the Template Windows Browser. See *Setting an Application Window as a Template Window* on page 25 for the detailed procedure.

To create an application window from a template window:

1. Launch the Template Window Browser by doing one of the following:

    a. Click **File**, **New Window from Template**.

        The Template Window Browser appears.

    b. Click **File**, **New Window**.

    c. In the **Window Properties** dialog box, click the **Templates** button next to the **Name** box.
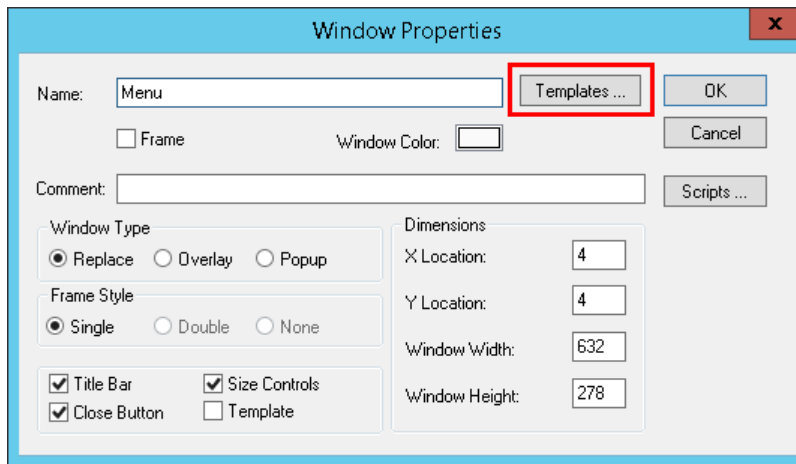
The Template Window Browser will appear.
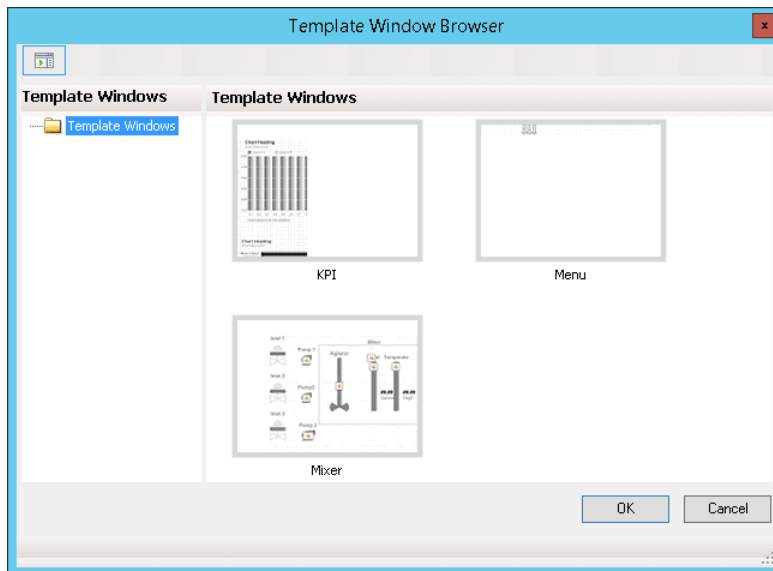
2. Browse the template window thumbnails and select your window.

Note: Each template window is shown in thumbnail view by default. The window thumbnails are shown proportional to the actual window location in design time with respect to the whole application.

3. Click **OK**.

The **Window Properties** dialog box will appear again. All of the template window properties will be populated except the window name.

Note: If the window name of a template window from a migrated application contains unsupported special characters, an underscore (_) replaces each unsupported character including the space character in the new application window.

4. Click **OK**.

Working with Frame Windows

If you are working in a Standalone or Managed InTouch application, you can create and develop frame windows in addition to or in place of application windows. Frame windows enable you to host Industrial Graphics that can support pan, zoom and touch capabilities.

Frame windows support most of the same properties as application windows. However, several limitations apply.

Frame windows do not:

*   support multiple Industrial Graphics

*   host native InTouch controls

*   host ActiveX controls

*   host SmartSymbols

*   support undo/redo actions for window components

*   flip or rotate actions for an embedded symbol

*   support cut, copy, paste or duplicate operations

*   provide cross reference support for Industrial Graphics embedded in frame windows

*   support window conversion to Industrial graphics

*   support creating new instances, editing instances or selecting alternate instances of embedded automation object graphics

To create a frame:

1. From WindowMaker, click **File**, **New Window**.

    The Window Properties dialog box appears with an checkbox for a Frame window.
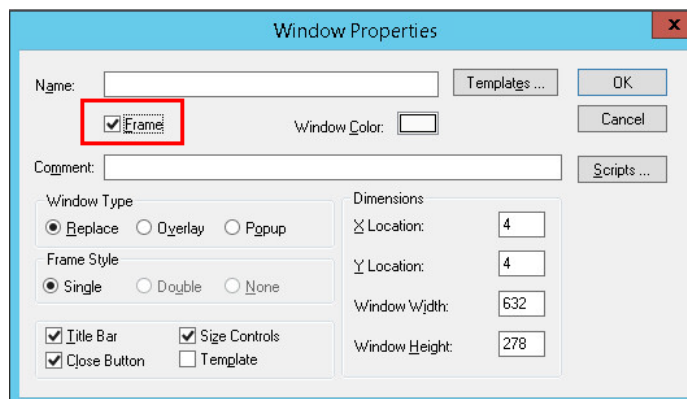
2. Select **Frame**.
3. Configure other frame properties and click **OK**.

Using the Properties Grid

You can edit frame properties directly from the Properties Grid on the WindowMaker canvas. The frame properties will populate in the Properties Grid upon creating the window:
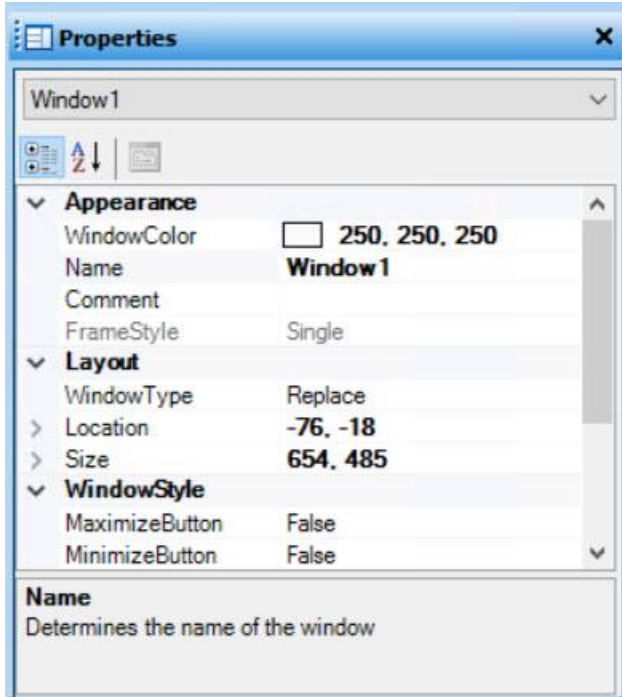
The Industrial Graphic Toolbox and the Properties Grid appear to the right of the canvas by default. You can dock the Graphic Toolbox or the Properties Grid to the left side of the canvas. To restore WindowMaker to the default, click **View**, **Restore Layout**.

Note: Application window properties will not populate in the **Properties Grid**.

Frame windows support the same properties as application windows, with three additional properties. They are described below.

See *Creating Application Windows* on page 23 for a full description of each window property.

Frame Property	Functionality
MaximizeButton	Enables the Maximize button in the upper right of the frame.
	Default is **False**.
MinimizeButton	Enables the Minimize button in the upper right of the frame.
	Default is **False**.
WindowState	Displays the initial state of the window: **Normal**, **Minimized** or **Maximized**.
	Default is **Normal**.

Changes to the frame properties will immediately reflect in the frame, with the exception of the **FrameStyle** property. To configure this property, the **SizeControl** and **TitleBar** properties must first be set to **False**.

The **FrameStyle** dropdown menu will become enabled.

Working with Graphics Embedded in Frames

Industrial Graphics can be embedded in frame windows the same way as application windows. The Industrial Graphic Toolbox is integrated into both the ArchestrA IDE and InTouch WindowMaker. The Graphic Toolbox includes separate folders containing the Industrial Graphic Library and Situational Awareness Library of predefined symbols. The Industrial Graphic Library contains realistic symbols of standard industrial objects.

Situational Awareness Library symbols have a simplified look and provide minimum visual detail to efficiently convey their functional purpose and status without showing irrelevant information to operators. Most Situational Awareness Library symbols are designed as Symbol Wizards that incorporate multiple visual and functional configurations in each symbol. Selecting a functional configuration of a symbol is a simple matter of selecting options from the symbol's Wizard Options.

Do either of the following to embed a symbol in a frame:

- drag an Industrial Graphic from the Graphic Toolbox and drop it onto the frame

- select Embed Industrial Graphic toolbar button and browse for a graphic

- embed graphics using the context menu from the frame window

You can automatically fit a frame to an embedded Industrial Graphic. To fit the frame to the embedded symbol, right click the window select **Fit to Symbol**.

The frame will resize to fit the symbol:
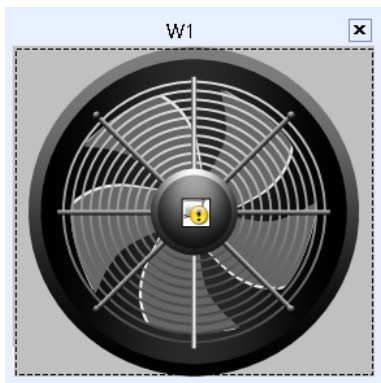
Note: Undo is not supported for **Fit to Symbol**. If you manually change the dimensions of the frame window or changes the Size property, you need to perform the **Fit to Symbol** action again.

When you embed an Industrial Graphic in the frame, the symbol will populate in the **Properties Grid** dropdown menu as shown below:
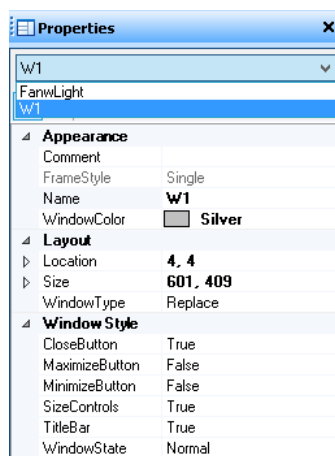
You can toggle between the frame window and symbol properties. Do any of the following:

- Select the symbol or frame window name from the dropdown menu

- Right-click on the window frame and select **Window Properties**.

- Click the title bar or frame border to display frame window properties in the grid

  Click on the embedded symbol to display the symbol properties.

You must configure the symbol properties to enable pan and zoom functionality using keyboard, mouse and touch gestures. The configurable symbol properties are listed below.

Symbol Property	Functionality
MaintainAspectRatio	Maintains the symbol's aspect ratio when the modern frame is resized. Default is **True**.
SymbolName	Sets the symbol name being hosted by the modern frame
InteractionMode	Sets the interaction mode. Options are: o **None**: disables Pan and Zoom o **PanZoom**: default, enables Pan and Zoom
ShowZoomControl	Shows the symbol and Zoom control. Options are: o **Auto**: control is shown as required o **Visible**: control is always shown

For details on using pan and zoom capabilities at run time, see the *InTouch HMI Application Management and Extension Guide*.

Using the Ribbon on a Frame Window

On creating a frame window, the following options are made available in the ribbon of the window.

- Edit Symbol - Launch the Graphic Editor where you can edit the symbol. For more information, see the Industrial Graphic Editor User Guide.

- Save Window - Save the window, see *Opening, Saving, and Closing Windows* on page 31.

- Save Window As - Create a duplicate window, see *Duplicating Windows* on page 33.

- Delete Window - Delete the window, see *Deleting Windows* on page 33.
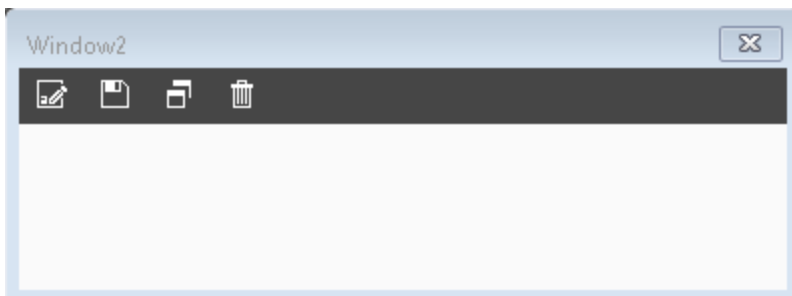
The ribbon highlights some of the options available when you right-click the canvas of the frame window.

Importing and Exporting Windows

You can import and export windows and their contents from one InTouch application to another. For more information, see Exporting and Importing Tag Definitions, Windows, and Scripts in the InTouch HMI Application Management and Extension Guide.

Importing Windows Using XML Utility

You can add the close button script to the XML script while importing a window using the XML utility.

Note: The windows that you want to import are specified in the batch file (wm.bat).

```
<CloseButton>True</CloseButton>
```

If your XML file contains this script, the window you import will have the close window button on the title bar.

Setting Windows to Appear at Run Time

"Home" windows are windows that appear in WindowViewer when the user starts WindowViewer directly, either from an icon or a menu command.

Home windows do not appear if you use the **Runtime** fast switch to start WindowViewer.

You can show home windows at any time during run time by using the ShowHome() function in a script.

To set home windows

1. On the **Special** menu, point to **Configure** and click WindowViewer. The WindowViewer **Properties** dialog box appears.

2. Click the **Home Windows** tab.

3. Select the window or windows to open when WindowViewer starts.

4. Click **OK**.

Modifying Application Windows

When developing your application, you can modify the properties of windows at any time.

To modify properties of an application window

1. Right click on the window name in the **Project** view, and click **Properties**. The **Window Properties** dialog box appears.

2. Make your changes. For more information about the window options, see *Creating Application Windows* on page 23.

3. Click **OK**.

Opening, Saving, and Closing Windows

While developing your application, you can open as many windows as your computer memory supports.

Opening a Window

To open a window

1. On the File menu, click Open Window. The Windows to Open dialog box appears listing the names of all windows in your application.

2. Do either of the following:

- To open a single window, double-click the window name.

- To open multiple windows, select the check boxes for the windows to open and then click OK.

When you save a window, all graphics, QuickScripts, properties, and so on associated with the window are also saved.

Saving a Window

To save a window

1. On the File menu, click Save Window. The Windows to Save dialog box appears, listing the names of all windows.

2. Select the windows that need to be saved.

3. Click OK.

When you close a window that has been modified, you are prompted to save your changes.

Closing a Window

To close a window

1. On the File menu, click Close Window. The Windows to Close dialog box appears listing the names of all currently open windows.

2. Select the check box next to the window name.

3. Click OK.

Viewing a Thumbnail Preview of a Window

You can view the thumbnail preview of a window in WindowMaker. This is especially useful if you have an application with many windows. You can then view the thumbnail preview of a window to check if it is the right one before opening it.

You can view the thumbnail preview of a window in the Project view only. The thumbnail preview is available and updated only after you have saved a window. If you migrate an application or import a window from an XML file, you can view the thumbnail preview without having to explicitly save the window. If you import a window from another application, you can view its thumbnail preview without having to explicitly save the window. However, if you have modified the window after importing it, you must save it to view the changes in the thumbnail.

You can also update the thumbnail of all windows in an InTouch application. You must close all windows before you can update the thumbnail.

To update the thumbnail of all windows

1. On the **Special** menu, click **Update All Window Thumbnails**. A confirmation message appears.

Note: You can also update the thumbnail of a particular window by right-clicking the window, and then clicking **Update Thumbnail**.

2. Click **Continue**. The thumbnail of all windows in the InTouch application is updated.

Note: The system will open and close all windows to update the thumbnail.

To view a thumbnail preview of a window

1. Open WindowMaker.

2. Open an application and save a window.

3. Move the pointer over the rectangle area that covers the window name and the associated icon. The thumbnail preview appears.
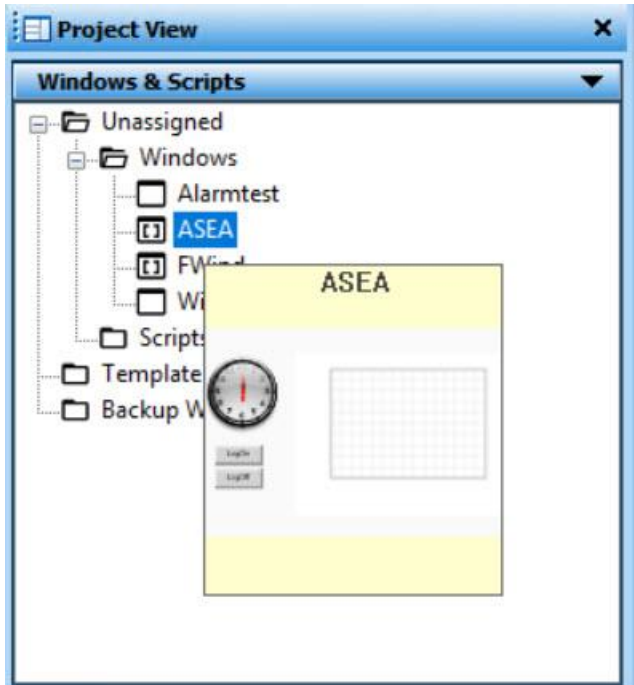
Duplicating Windows

When you have very similar processes to simulate and control, you may want to duplicate a window and then customize it for a secondary process or unit.

You can duplicate windows with all graphics, QuickScripts, properties, and so on, associated with the window.

Before you start, the window that you want to duplicate must be open and saved at least one time. You can only duplicate one window at a time.

To duplicate a window

1. On the **File** menu, click **Save Window As**. The **Window to save under new name** dialog box appears, listing the names of all currently open windows.

2. Select the check box next to the window name.

3. In the **New Name** box, type a name for the new window.

4. Click **OK**.

Note: If the original window name contained unsupported characters, an underscore (_) character will replace each unsupported character in the duplicate window name.

Deleting Windows

To conserve computer storage space, or if the list of windows in the Application Explorer becomes too long to manage, you can delete unused windows.

Caution: Make sure you delete the correct window. You cannot restore a deleted window with **Undo**.

To delete a window

1. On the **File** menu, click **Delete Window**. A dialog box appears listing the names of all windows.

2. Select the window name you want to delete and click **OK**. When the message appears, click **Yes**.

3. Click **OK**.

Printing Information about InTouch Windows

You can print the following information for InTouch windows:

- Details for the graphical objects placed in a window. For example, the window location for all types of objects, the font used for a text object, the custom properties defined for an Industrial graphic, and so on.

- Details about the animation links used in a window.

- Scripts associated with a window.

- Tags used in a window.

You can print details for graphical objects to either a printer or an .html file. An .html file is created for each window. The .html file contains:

- A picture of the window as a referenced .png file.

- A listing of details about each graphical object in the window.

If you print graphical object details for multiple windows, a summary .html file is created that has links to the window-specific .html files.

For animation link, script, or tag details, you can print to either a printer or a .txt file.

To print information about InTouch windows

1. Open an InTouch application in WindowMaker.

2. On the **File** menu, click **Print**. The **WindowMaker Printout** dialog box appears.
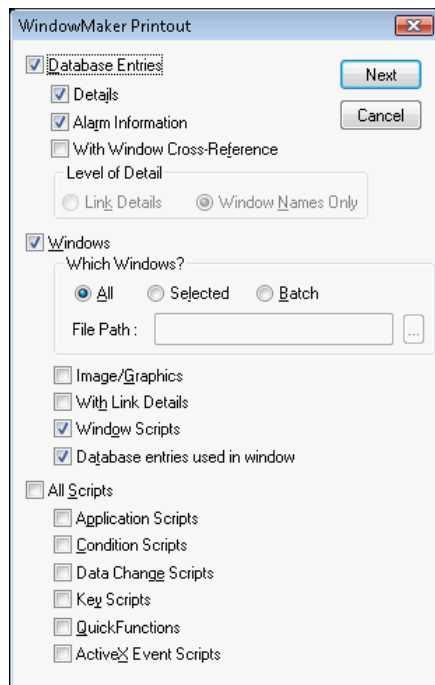
3. Select **Windows**.

4. Select the windows to print:

- o **All** prints the information for all windows in the application, including names of any embedded graphics.

- o **Selected** prints only the information for specific windows, including names of any embedded graphics on the selected windows. The **Windows to Print** dialog box appears. Select the windows in your application you want to print and click **OK**.

- o **Batch** prints only the information for windows specified in a .csv file, including names of any embedded graphics.

    For details on the .csv format, see.*CSV Format for Printing Windows* on page 35.

5. Select what you want printed for the selected windows:

    - o **Image/Graphics** prints the information about all of the graphic objects placed in the windows.

    - o **With Link Details** prints the link details for the windows.

    - o **Window Scripts** prints the scripts associated with the windows.

    - o **Database entries used in window** prints the tags used in the windows.

6. Click **Next**. The **Select Output Destination** dialog box appears.

7. Do one of the following

    - o Click **Send output to Printer** to print the information.

    - o Click **Send output to Text File** to create a single .txt file.

    - o Click **Send output to HTML File** to create a summary .html file and a single .html file and .png file for each window you specified. If .html and .png files exist, they are automatically overwritten.

8. Click **Print**.

Printing Windows Information from a Command Prompt

You can print window information from a command prompt. To do this, you create a .csv file that contains the window names and then reference the .csv file in the print command.

When the print command runs, WindowMaker automatically opens the default InTouch application, performs the print operation, and then closes. Printing from a command prompt only works for stand-alone InTouch applications.

To print window information from a command prompt

1. Create the .csv file that contains the names of the windows to print.

    For details on the .csv format, see *.CSV Format for Printing Windows* on page 35.

2. Exit WindowMaker and close the .csv file.

3. Open a command prompt.

4. Type the command to print the information.

    For details on the command syntax, see *Syntax for Printing from a Command Prompt* on page 36.

5. Press the **Enter** key. WindowMaker starts and prints the information.

.CSV Format for Printing Windows

You ca use a .csv file to use your window information to multiple formats, such as Microsoft Excel. Create a single row in a .csv file and include each window name separated by a comma:

Window1,Window2,Window3,WindowN

For example:
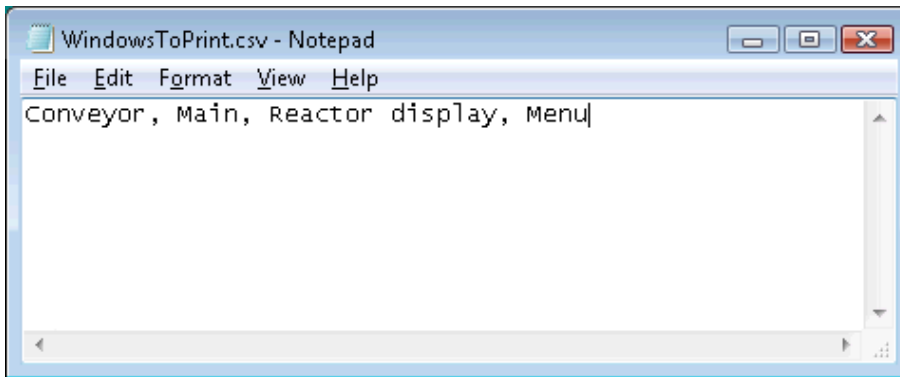
If you use Microsoft Excel to create the file, there should be one row, with the name of each window in each column cell of the row.

You cannot use the backslash character (\) in a window name.

You must use a comma and not any other separator.

Syntax for Printing from a Command Prompt

The command syntax is:

```
"<WindowMaker path name>" COMMANDFILE="<CSV file>" ALL OUTPUTTARGET=<target
name>
```

where:

`<WindowMaker path name>` is the path to the WindowMaker application (WM.exe).

- o `<CSV file>` is the name of the .csv file that specifies the windows to print.
- o `<target name>` is output location, either to a printer or .html file.
- o `ALL` is the command to print all link, database entry, and script information. If you do not include the ALL command, only graphical object information prints.

In this example, all link, database entry, and script information is sent to the default printer:

```
"C:\Program Files\Wonderware\InTouch\wm.exe" COMMANDFILE="D:\print.csv"
ALL OUTPUTTARGET = PRINTER
```

In this example, graphical object information is sent to an .html file:

```
"C:\Program Files\Wonderware\InTouch\wm.exe" COMMANDFILE="D:\print.csv"
OUTPUTTARGET = HTML <DEMOAPP.html>
```

CHAPTER 3

WindowMaker Objects

About WindowMaker Objects

Graphical objects are the key parts of the human machine interface (HMI) applications you build.

As you build your applications, you create simple objects, combine simple objects to make more complex objects, and use some pre-defined complex objects.
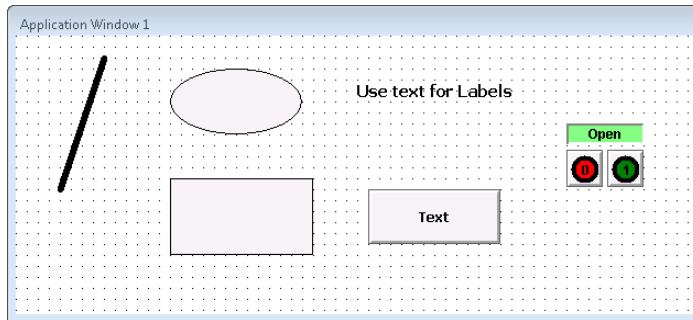
The individual elements of a complex object are usually grouped together to:

- Prevent a complex object from breaking apart while editing.

- Duplicate the entire object.

- Assign a common set of properties to its individual elements.

You can use Industrial graphics created with the Industrial Graphic Editor in your managed or Advanced InTouch applications. You can also add Industrial Graphics directly from WindowMaker's Industrial Graphic Toolbox. For more information about working with Industrial and Situational Awareness Library symbols, see the *Industrial Graphic Editor User Guide* or WindowMaker help.

Simple Objects

You can create the following types of simple objects:

- Lines

- Shapes

- Text

- Buttons

Each simple object has attributes that control how it appears:

- Line color and weight

- Fill color

- Height

- Width

- Orientation

Creating Lines and Shapes

The following table describes how to do basic drawing tasks. The drawing buttons are located on the Drawing toolbar.

To draw this	Click this	Button
Line	Line button	╱
Horizontal or vertical line	H/V Line button	+
Rectangle	Rectangle button	□
Rectangle with rounded corners	Rounded Rectangle button	▢
	Note: To adjust the radius of the rounded rectangle corners, see *Changing the Radius of a Rounded Rectangle* on page 51.	
Circle or an ellipse	Ellipse button. Press and hold the SHIFT key to draw a circle.	○

Creating Buttons

You can use buttons to create points of interaction with your application. The process is very similar to creating simple drawing objects.

For information about creating polygons, see *Creating Polylines and Polygons* on page 38.

To create a button

▢

1. On the **Drawing** toolbar, click the **Button** button.
2. Click and drag to place and size the button.
3. Edit the default button text. Do the following:
    a. Right-click the button and click **Substitute Strings**.
    b. In the **New String** box, type the text for the button.
    c. Click **OK**.

Creating Polylines and Polygons

Drawing polylines is slightly different than drawing lines.

To create a polyline or polygon

1. On the **Drawing** toolbar, click the **Polyline** or **Polygon** button.
2. Click on the application window to set the first point.
3. Click on the application window again to set more points to define your polyline or polygon.
4. Double click the last point.

Creating Text

You can use text to label visual items in your application.

When you create text, the text formatting settings match those set in the WindowMaker **Properties** dialog box. You can change the appearance of selected text. For more information, see *Changing Text Appearance* on page 45.

When you type multiple lines of text, they become objects which can be moved and edited independently. You can also combine text objects into a symbol and edit them as a group.

To create text

1. On the **Drawing** toolbar, click the **Text** button.

2. Click the place for the text to start.

3. Type your text and press ENTER. A new line of text appears.

Complex Objects

Complex objects provide more functionality than simple objects. The following table describes the different types of complex objects.

Complex **Object**	Description
Cell	A group of two or more objects, including symbols or other cells, that are joined together to form a single unit. You can use cells to create virtual devices such as slide controllers. Cells are useful for creating multiple devices to be associated with different tags.
Symbol	A group of simple objects, such as lines, shapes, and text, that joined together and treated as a single object. Any attribute change applied to a symbol affects all the component objects of a symbol. Symbols cannot contain bitmaps, buttons, cells, wizards, or trends.
SmartSymbol	An InTouch cell that has been converted into a reusable graphic template. You can place one or more instances of a SmartSymbol template in your application windows. Any change to a template propagates to the instances. For more information, see About SmartSymbols in the InTouch® HMI SmartSymbols Guide.
Industrial Graphic	A highly versatile graphic created using the Industrial Graphic Editor in the ArchestrA Integrated Development Environment (IDE). For more information, see About InTouch HMI and ArchestrA Integration in the InTouch HMI and ArchestrA Integration Guide.
Bitmap Container	Objects that enable you to import images such as photographs, drawings, and screen shots. You can rotate a bitmap and you can give it a transparent background. For more information, see *Working with Bitmap Containers* on page 49.
Trend Objects	Charts of real-time or historical data value changes of multiple tags over time. For more information, see *Trend Objects* on page 92.

Complex **Object**	Description
Wizard	Pre-built object that you only need to select, place, and configure for your application.
	For more information, see *Wizards* on page 89.
ActiveX Control	Software component that runs within your application. WindowMaker supports both AVEVA and third-party ActiveX controls. For more information, see *Using ActiveX Controls* on page 117.

Cells and Symbols

You can combine multiple objects into two different types of single units: cells and symbols. A cell can contain *any* object. A symbol can only contain simple objects. Symbols cannot contain cells.

To find out if a specific object is a cell or a symbol, double-click the object.

- A cell opens, either the **Substitute Tagnames** dialog box, or if the cell doesn't contain tagnames, the Substitute Names warning message appears.
- A symbol or simple graphic object opens the **Animation Links Selection** dialog box.

About Cells

Use a cell to combine and maintain a fixed spatial relationship among multiple elements. You can also use a cell to move multiple elements around and align with other graphical elements.

To change the elements of a cell, you must break the cell, change the elements, and then combine the elements into a cell again.

You can animate elements of a cell but you cannot animate a cell. You also cannot resize a cell.

About Symbols

You can animate a symbol and simple objects. You can also use a symbol to animate parts of a complex graphic.

You cannot make a symbol if more than one of the selected objects has links.

If you combine two symbols into a new symbol, the original symbol structure is lost. If you break the new symbol, it is broken into the individual components of each original symbol. The two original symbols are lost.

Grouping Objects to Cells

You can combine symbols, bitmaps, trends, buttons, wizards and other cells into a cell. If you include a symbol in a cell, all animation links associated with that symbol remain intact.

After you use a cell in a SmartSymbol, if you break the SmartSymbol, you cannot resize the cell.

To create a cell

1. Select the objects that you want to include.
2. On the **Arrange** menu, click **Make Cell**.

To break a cell

1. Select the cell.
2. On the **Arrange** menu, click **Break Cell**.

Grouping Objects to Symbols

Symbols cannot contain bitmaps, buttons, cells, wizards or trends. If one of the selected objects has animation links attached to it, the links are attached to the new symbol.

To create a symbol

1. Select the objects that you want to include.
2. On the **Arrange** menu, click **Make Symbol**.

To break a symbol

1. Select the symbol.
2. On the **Arrange** menu, click **Break Symbol**.

Common Manipulations

Right-click an object to see a menu showing the valid commands or actions you can apply to that object. You can:

- Select objects
- Align objects
- Layer objects
- Flip objects
- Resize objects
- Change font
- Change fill
- Control horizontal and vertical spacing

- Move objects
- Arrange objects
- Undo changes
- Flip symbols
- Rotate objects
- Change line or outline
- Delete objects

Selecting Objects

You must select an object before you can modify it. When you select an object, handles appear on the perimeter of the object. You can use these handles to resize and/or reshape the object.
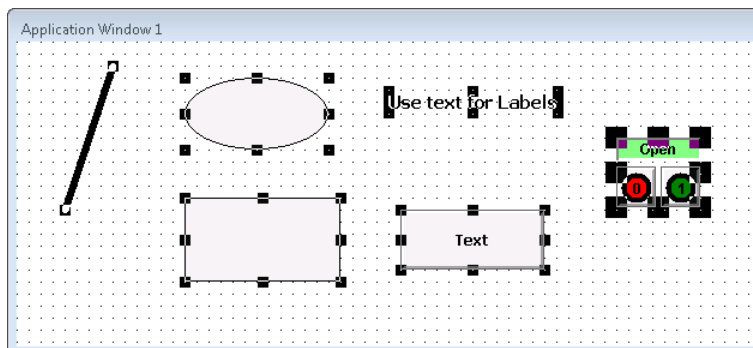
To select all objects in the active window

- On the **Edit** menu, click **Select All**, or press **F2**.

To select an object

- Click the **Select Mode** button and click the object you want to select.

To un-select an object

- Click a blank area of the window.

To select multiple objects

- Click the **Select Mode** button, select the first object, then SHIFT+click the other objects.

To select a group of objects

- Click the **Select Mode** button and drag a box around the objects. All the objects that are completely within the rectangle are selected.

To un-select a specific object or objects from a group of objects

- SHIFT+click the object.

Moving Objects

You can move objects by:

- Dragging them.

- Using the arrow keys on your keyboard.

- Typing window coordinates in boxes in the status bar.

When you are moving an object, notice how the coordinates in the status bar change.

To move an object by dragging

- Select the object and drag it.

When moving objects with the arrow keys, how far an object moves depends upon whether or not the grid is showing.

When the grid is showing, how many pixels an object moves depends upon the grid spacing, which is set on the WindowMaker properties dialog. The default setting is ten pixels between grid points.

When the grid is showing,

- Pressing an arrow key moves the object one grid point.

- Pressing SHIFT + an arrow key moves the object two grid points.

- Pressing CTRL + an arrow key moves the object four grid points.

When the grid is not showing,

- Pressing an arrow key moves the object one pixel.

- Pressing SHIFT + an arrow key moves the object ten pixels.

- Pressing CTRL + an arrow key moves the object 50 pixels.

To move an object with the Arrow keys

- Select an object and
  - ○ press an arrow key.
  - ○ SHIFT + press an arrow key.
  - ○ CTRL + press an arrow key.

To move an object with the status bar

1. Select the object.

   X, Y | 188 | 121

2. Type the X and Y coordinates in the status bar.

3. Press Enter.

Aligning Objects

You can align objects by their left or right edges, centers, center points, tops, middles, or bottoms.

Using the menu commands or the buttons, you can align in several ways.

Select	Or click	To do this
Align Left	吕	Align left edges of the objects with the left edge of the object farthest to the left.
Align Center	盂	Align objects with the vertical center line of the group.
Align Right	凸	Align right edges of objects with the right edge of the object that is farthest to the right.
Align Tops	帀	Align top edges with the top edge of the highest object.
Align Middle	┼o	Align horizontal centers with the middle of the group.
Align Bottom	凸	Align bottom edges with the bottom edge of the lowest object.
Align Center points	回	Align center points with the center point of the group.

To align objects

1. Select multiple objects.

2. On the **Arrange** menu, point to **Align** and click the appropriate align command.

Layering Objects

You can position objects in front of or behind others.

To position an object behind another object

1. Select the objects(s).

2. On the **Arrange** menu, click **Send to Back**.

To position an object in front of another object

1. Select the objects(s).

2. Do one of the following:

   o On the **Arrange** toolbar, click **Bring to Front**.

   o On the **Arrange** menu, click **Bring to Front**.

   o Press SHIFT+F9.

Controlling Horizontal and Vertical Spacing

You can space objects horizontally between the left most selected object and the right most selected object.

You can also control the vertical spacing between the top most selected object and the bottom most selected object.

To space objects horizontally or vertically

1. Select the objects.

2. On the **Arrange Toolbar**, click **Space Horizontally** or click **Space Vertically**.

Flipping Objects and Cells

You can flip most objects horizontally or vertically. You can flip objects singly or in groups.

When you flip an object, you transform it to its mirror image. You cannot flip text.
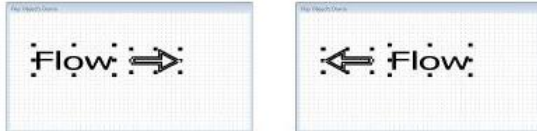
When you flip cells, they are not mirrored. Only the position of the cell in the group of objects is mirrored.
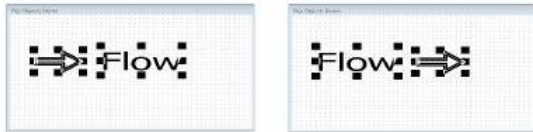
Compare the location of the cells before and after they are flipped. The position is flipped, not the contents.

To flip an object or a cell

1. Select the object(s).

2. On the **Arrange** toolbar, click **Flip Horizontal** or click **FlipVertical**.

Resizing Objects

You can resize an object using two methods. You can drag it or specify an exact width and height.

If snap to grid is turned on, the object snaps to the grid during the proportional resizing operation. The result is a slight deviation in the ratio between vertical to horizontal size. To avoid this deviation, turn off the snap to grid.

To resize an object by dragging

1. Select the object and then position the point of the arrow cursor in the center of a handle.

2. Drag the handle to resize the object.

To resize an object by proportional resizing

- Select the object and SHIFT+drag.

To resize an object by dimensions

1. Select the object.

    W, H 30   47
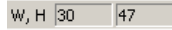
2. Type the width and height dimensions in the **W, H** boxes on the status bar.

Rotating Objects

You can rotate most objects including symbols, text, and bitmaps. You cannot rotate cells.

You can rotate objects clockwise or counter clockwise 360 degrees in 90 degree increments.

Rotating objects in WindowMaker has nothing to do with dynamically rotating objects at run time or in WindowViewer. You rotate objects in WindowViewer by linking them to an orientation animation.

To rotate an object

1. Select the object(s).

2. On the **Arrange** toolbar, click **Rotate Clockwise** or click **Rotate Counter Clockwise**.

Changing Text Appearance

You can set the appearance of fonts before you create text by setting defaults in advance or you can change the appearance of fonts after you create them. For information about setting font defaults, see *Setting Font Defaults* on page 19.

The text justification attribute settings are important for text objects showing dynamic values. The justification determines how fields of varying length appear at run time.

For example, if you are showing a numeric value at the end of a text string that is centered or is right justified, the entire text string, including the value, is centered or justified each time there is a change in the number of digits shown.

Using the menu commands or the buttons, you can configure text in several ways.

To	Click	Button
Change the font, style, color, or text size	**Font**	
Make the text bold	**Bold**	B
Make the text italic	**Italic**	I
Make the text underlined	**Underline**	U
Reduce or enlarge font size	**Reduce Font** or **Enlarge Font**	A˅ A˄
Change the justification	**Left Justified**, **Centered**, or **Right Justified**	

To configure the text appearance

1. Select the text object.

2. On the **Text** menu, click the appropriate text command.

Changing Lines and Outlines

You can change the color and either the pattern or width of lines or outlines around outlined objects. Outlined objects include filled shapes such as ellipses, rectangles, and polygons, as well as bitmaps and other imported images.

Wider lines take longer to draw in run time. Dashed and dotted lines can only be one pixel wide.

To set the line appearance default settings

1. Click a blank area of the window.

2. On the **Line** menu, select the line width or pattern.

3. On the **Format** toolbar, click the **Line Color** tool.

4. Select a color.

To change the color of a line

1. Select a line, a group of lines, or an object with an outline.

2. On the **Format** toolbar, click the **Line Color** tool.

3. Select a color.

To change the style or width of a line or outline

1. Select the object.

2. On the **Line** menu, click the line style or width.

To remove an outline

1. Select the object.

2. On the **Line** menu, click **No Line**.

Changing Fill

Filled shapes include shape surrounded by a line. Examples of filled shapes are rectangles, rounded rectangles, circles, ellipses, and polygons.

To change the fill color of an object

1. Select the object.

2. On the **Format** toolbar, click the **Fill Color** tool.

3. Select a color.

To set the default color for filled shapes

1. Click a blank space on the window.

2. On the **Format** toolbar, click the **Fill Color** tool.

3. Select a color.

Deleting Objects

You can delete one or more objects.

To delete an object

- Do either of the following:
  - Right-click the object and click **Erase**.
  - Select the object and then press Delete.

Undoing Changes

WindowMaker records your editing and formatting changes for each window. By default, WindowMaker supports 10 levels of undo/redo, where one level represents one action. You can set WindowMaker to retain up to 25 levels of actions. You can also turn off undo/redo by setting the undo/redo level to zero.

If you close the window, all recorded actions are cleared.

To undo a command

- On the **Edit** menu, click **Undo**.

To redo a command

- On the **Edit** menu, click **Redo**.

To set the number of undo/redo levels

1. On the **Special** menu, point to **Configure**, and then click WindowMaker. The **WindowMaker Properties** dialog box appears.

2. In the **Levels of Undo** box, type the number of levels.

Special Manipulations for All Objects

You can modify and manipulate simple objects. You can:

- Cut, copy, and paste objects.
- Cut, copy, and paste object links.
- Duplicate objects.

Cutting, Copying, and Pasting Objects

Cut, copy, and paste operations in WindowMaker are much the same as in other Windows based applications but there are some significant differences you need to be aware of.

When you cut, copy, or paste an object, attributes and animation links are also cut, copied, or pasted with it.

All pasted objects remain selected after being pasted and you can move them to adjust their location.

To cut an object

- Right-click the object and then click **Cut**.

To copy an object

- Right-click the object and then click **Copy**.

To paste an object

1. Right-click a blank space in the window and click **Paste**. The cursor changes to a corner symbol.

2. Hold down the left mouse button. The cursor changes to a dotted rectangle the size of the copied object.

3. Drag the rectangle to locate the object.

4. Release the mouse button.

Cutting, Copying, and Pasting Object Links

The Clipboard is a temporary storage area for links you cut or copy.

* The Clipboard only stores the links for your most recent cut or copy action.

* You can paste the links to any object or symbol that supports the links in the Clipboard.

* If a pasted link is not supported by the object, for example, a line color link on a text object, the link is not pasted.

* If you select multiple objects for pasting, the links are pasted to all objects.

To cut, copy, paste and clear links

* Right-click the object, point to **Links** and click the appropriate command.

Duplicating Objects

Duplicating objects is similar to copying objects and their animation links, but has the advantage of also duplicating the offset distance and direction when the objects are duplicated more than once.

When you move a duplicated object without un-selecting it, then duplicate it again, the third iteration is offset the same distance and direction between the first two iterations.

You can repeat this procedure as many times as necessary.

To duplicate an object

1. Right-click the object and click **Duplicate**. The object is copied and pasted to an offset position from the original object.
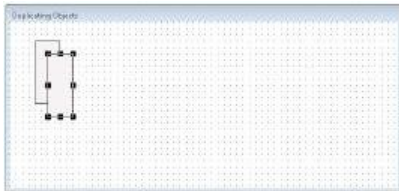
2. Keeping the duplicated object selected, drag it to a different position.
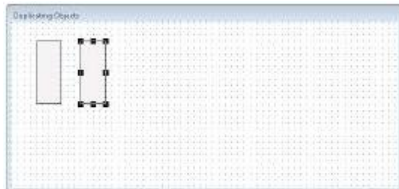
3. Again, without un-selecting the duplicated object, right click the object and click **Duplicate** again. The third iteration of the object appears in the same relative position as the first two.
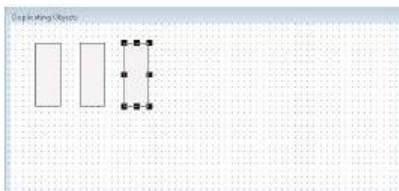
Special Manipulations for Special Objects

The following object types have unique attributes that you can edit:

- Polylines and polygons

- Bitmap containers

- Bitmap transparencies

- Rounded rectangles

- Object text

Reshaping Polyline and Polygon Objects

You can adjust the shapes of polylines and polygons.

To reshape a polyline or polygon

1. Select the object.

2. Do one of the following.

   o On the **Edit** menu, click **Reshape Object**.

   o Right-click the object and click **Reshape Object**.

   o On the **Arrange** toolbar, click the **Reshape Object** button.

   Each shape definition point becomes a handle.

3. Drag a handle to reshape.

To add a point to a polygon

1. Select the object.

2. Do one of the following.

   o On the **Edit** menu, click **Add Point**.

   o Right-click the object and click **Add Point**.

3. Click on an edge of the polygon, then drag the point to change the shape of the polygon.

To delete a point from a polygon

1. Select the object.

2. Do one of the following.

   o On the **Edit** menu, click **Del Point**.

   o Right-click the object and click **Del Point**.

3. Click on a point of the polygon, the point is deleted and the shape of the polygon changes.

Working with Bitmap Containers

Bitmap containers are objects that enable you to import graphic objects such as pictures, screen captures, and drawings into your application.

Supported bitmap container file types are .bmp, .jpeg, .jpg, .pcx and .tga.

When you import a bitmap, it automatically fills the bitmap container, but you can resize it to its original size and proportions.

You can rotate bitmaps in 90 degree increments.

You can include bitmaps in a cell but not in a symbol.

Using WindowMaker, you are able to place more bitmaps into a window than can load in WindowViewer. If you need to place a large number of bitmaps in a window, be sure to test the window in WindowViewer before releasing the application.

To   import a bitmap image

1. On the **Drawing** toolbar, click the **Bitmap** tool. The cursor turns into a cross-hair.

2. Drag the cursor to draw a bitmap container.

3. On the **Edit** menu, click **Import Image**. The **Select Image File** dialog box appears.

4. Select the image filename and click **OK**.

To make the bitmap its original size

1. Select the image.

2. On the **Edit** menu, click **Bitmap - Original Size**.

To paste a bitmap image

1. Copy the graphic to the Windows Clipboard.

2. Click the **Bitmap** tool and draw a bitmap container in your window.

3. Right-click the bitmap container and click **Paste Bitmap**.

To edit a bitmap

1. Select the bitmap.

2. On the **Edit** menu click **Edit Bitmap**. Microsoft Paint opens showing the bitmap.

3. Edit the bitmap in MS Paint.

4. Save and close MS Paint.

Defining Bitmap Transparency

When you define a transparent color in a bitmap, the window background or any objects behind the bitmap shows through it everywhere the transparent color is used. You can define only one transparent color per image.
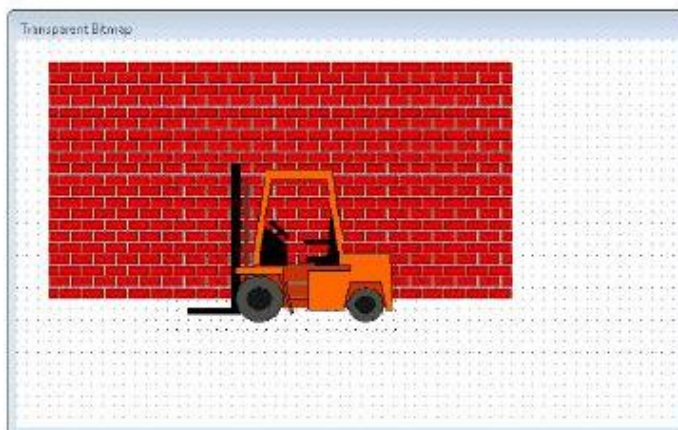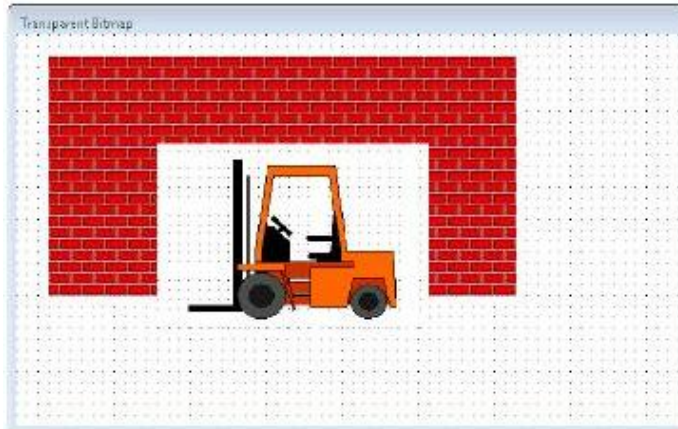
To create a transparent bitmap

1. With the bitmap selected, click the **Transparent Color** button on the **Format** toolbar to open the transparent color palette.

2. Right-click a color square in the **Custom Palette**. The **Edit Custom Color** dialog box appears.

3. Click the eye dropper tool.

4. Click the color in the bitmap that you want to make transparent. The color is copied to the color square that you selected in the color palette.

5. Click the color square to apply the transparent color to the bitmap. All the pixels in the image that are that color become transparent.

Changing the Radius of a Rounded Rectangle

You can increase and/or decrease the corner radius of a Rounded Rectangle.

To increase or decrease a rounded object's radius

1. Select the object.

2. On the **Edit** menu, click **Enlarge Radius** or **Reduce Radius**.

Substituting Object Text

You can edit the text of objects that have text, such as symbols, cells, or buttons.

When you change a text string, it retains all of its original attributes, font, style, color, and so on. Text formatting also applies to numeric values.

To change the text in an object

1. Select the object or button with the text. Do either of the following:

    o   On the **Special** menu, click **Substitute Strings**.

    o   Right-click the text object, point to **Substitute** and then click **Substitute Strings**.

2. In the **New String** box, type the new string and click **OK**.

To change a portion of text in a series of text objects

1. Select all the text objects.

2. On the **Special** menu, click **Substitute Strings**.

3. Click **Replace**. The **Replace Text** box appears.

4. In the **Old Text** box, type the portion of text to replace.

5. In the **New** box, type the new text string.

6. Click **OK**. The new text string replaces the old text string in all the selected objects.

Printing Window Information about WindowMaker Objects

You can print information about the InTouch graphical objects that you have placed in an application window. InTouch graphical objects include simple objects, such as lines, buttons, and text, as well as complex objects, such as cells, ActiveX controls, SmartSymbols, and Industrial graphics.

You can print details for graphical objects to either a printer or an .html file. An .html file is created for each window. The .html file contains:

*   A picture of the window as a referenced .png file.

*   A listing of details about each graphical object in the window.

If you print graphical object details for multiple windows, a summary .html file is created that has links to the window-specific .html files.

To print information about WindowMaker objects

1. Open an InTouch application in WindowMaker. On the WindowMaker **File** menu, click **Print**. The **WindowMaker Printout** dialog box appears.
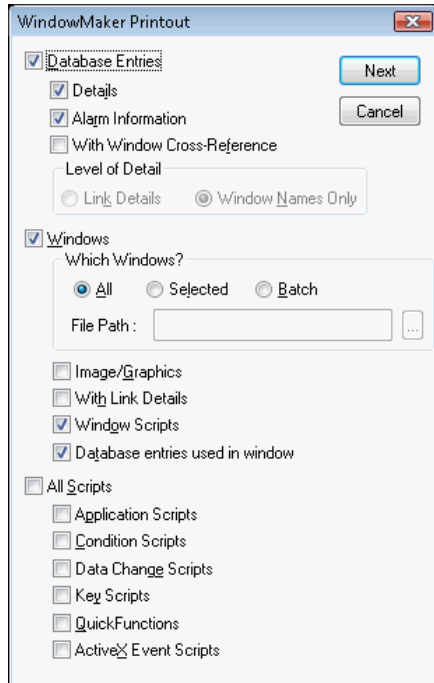
2. Select the **Windows** check box.

3. Specify the windows to print:

   o   **All** prints the images/graphics for all windows in the application.

Note: Printing all windows of an application to an XPS file can cause WindowMaker to stop responding if the windows contain many graphics or large graphics that exceed the available memory of the computer hosting WindowMaker. You should print these windows one at a time to avoid problems caused by memory shortages.

---

   o   **Selected** prints only the images/graphics for specific windows. The **Windows to Print** dialog box appears. Select the windows in your application you want to print and click **OK**.

   o   **Batch** prints only the information for windows specified in a .csv file.

   For details on the .csv format, see *.CSV Format for Printing Windows* on page 35.

4. Select the **Image/Graphics** check box.

5. Click **Next**. The **Select Output Destination** dialog box appears.

6. Do one of the following

   o   Click **Send output to Printer** to print the information.

   o   Click **Send output to HTML File** to create a summary .html file, a single .html file, and .png file for each window you specified. If .html and .png files exist, they are automatically overwritten.

7. Click **Print**.

CHAPTER 4

Animating Objects

About Animating Objects

You can animate objects or symbols by means of animation links. Animation links connect the values of tags or expressions to your objects or symbols.

For example, you can:

• Create a slider or tank symbol that shows the liquid level in the tank.

• Create a meter that shows a range of values.

• Create touch screen symbols for operator control.
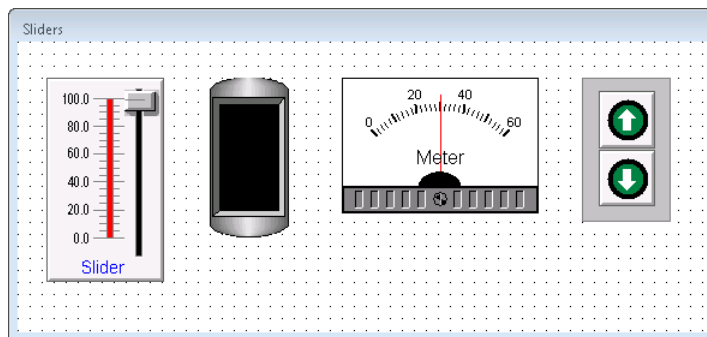
Two Types of Animation Links

There are two basic types of animation links: display links and touch links.

• Display links show information to operators. Examples of display links are changing colors, changing fill levels, horizontal or vertical movements, and blinking objects.

• Touch links enable operator input into the system. Examples of touch links are sliders or push buttons that respond to operator input.

You can define multiple links for objects or symbols. By combining various links, you can create almost any screen animation effect.

Data Display Animations

Data display animations only show information to operators. These animations do not support operator input.

Creating Value Displays

Use a value display text object to show the value of a tag. This lets you show things like fill levels, on/off status, or alarm messages.

You can use any one of three types of *value display links* to show messages at run time.

Value Display Type	Shows
Discrete	Discrete values such as on or off

Value Display Type	Shows
Analog	The value of an analog expression such as fill level or speed.
String	The value of a string expression such as "Fill Level = 100".

You can use up to 1023 characters in an expression. If you need a larger expression, create a QuickFunction and call it in your expression.

Messages appear in the location of the original text object using the font, size, color, alignment, and linked attributes set for that object. The original contents of the field have no effect on the message at run time.

To create a discrete value display link

1. Right-click the text object and click **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Value Display** area, click **Discrete**. The **Output** -> **Discrete Expression** dialog box appears.

3. In the **Expression** box, type the name of a discrete tag or an expression that equates to a discrete value. For example:
```
Cooling_Pump
```

4. In the **On Message** box, type the message to appear when the value of the expression equals 1, true, on, or yes. For example:
```
Pump is ON
```

5. In the **Off Message** box, type the message to appear when the value of the expression equals 0, false, off, or no. For example:
```
Pump is OFF
```

6. Click **OK**.

To create an analog value display link

1. Right-click the text object and click **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Value Display** area, click **Analog**. The **Output -> Analog Expression** dialog box appears.
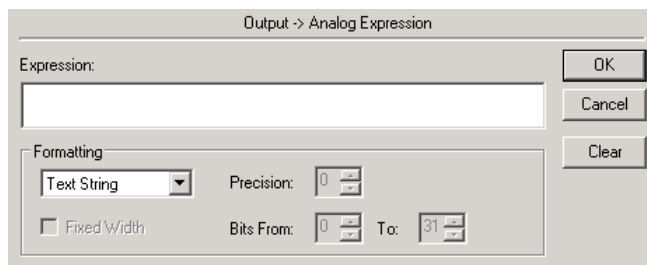
3. In the **Expression** box, type an analog (integer or real) tagname or an expression that equates to an analog value. For example:
```
Tank_CV*0.06
```

4. In the **Formatting** area, in the list, click each data type for which you want to configure run time advanced formatting. The **Fixed Width** check box, **Precision** box, and **Bits From** and **To** boxes become available based on the data type you select. For information on configuring these options, see *Advanced Formatting for Text* on page 84.

Note: During run time, the analog value input link field can be resized by clicking and dragging with the pointer and mouse.

5. Click **OK**.

To create a string value display link

1. Right-click the text object and click **Animation Links**. The Animation Links dialog box appears.

2. In the **Value Display** area, click **String**. The **Output -> String Expression** dialog box appears.
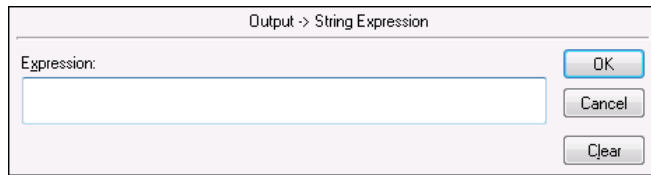
3. In the **Expression** box, type the name of a message tag or an expression that uses a message tag. For example:
```
"The Tank Level is:" + Text(TankLevel,"#")
```

4. Click **OK**.

Note: During run time, the string value input link field can be resized by clicking and dragging with the pointer and mouse.

Creating Movement

You make objects move at run time by using location links. You can make an object move horizontally, vertically, or both, as the value of an analog tag or expression changes. For example, as a tank level increases and decreases, an indicator moves up and down.

To create horizontal movement

1. Place the object on the screen in the starting location.

2. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

3. In the **Location** area, click **Horizontal**. The **Horizontal Location** dialog box appears.
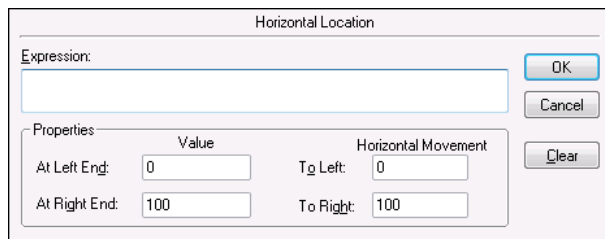
4. In the **Expression** box, type an analog tag name or an expression that equates to an analog value.

5. In the **Properties** area, configure how far the object moves. Do the following:

   a. In the **At Left End** box, type the value of the analog tag when the object should be at its farthest left position.

   b. In the **At Right End** box, type the value of the analog tag when the object should be at its farthest right position.

   c. In the **To Left** box, type the number of pixels the object should move to the left of its starting position.

   d. In the **To Right** box, type the number of pixels the object should move to the right of its starting position.

6. Click **OK**.

To create vertical movement

1. Place the object on the screen in the starting location.

2. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

3. In the **Location** area, click **Vertical**. The **Vertical Location** dialog box appears.
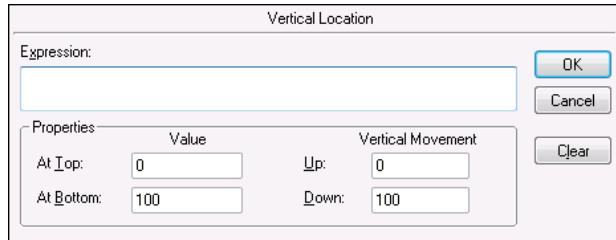
4. In the **Expression** box, type an analog tag name or an expression that equates to an analog value.

5. In the **Properties** area, do the following:

   a. In the **At Top** box, type the value of the analog tag when the object should be at the top position.

   b. In the **At Bottom** box, type the value of the analog tag when the object should be at the bottom position.

   c. In the **Up** box, type the number of pixels the object should move up from the starting position.

   d. In the **Down** box, type the number of pixels the object should move down from the starting position.

6. Click **OK**.

Creating Rotation

You can make an object move around a center point as the value of an analog tag changes by using orientation links. For example, as pressure increases or decreases, a pointer can move around a dial.

The orientation link uses the center of the object or symbol as the default center of rotation. You can offset the center of rotation.

Orientation links are not supported for Industrial graphics.

Tip: Draw a temporary rectangle from the center of the object to the rotation center point. Now you can read the X and Y offset dimensions in pixels from the W, H boxes in the status bar.

To create an orientation link

1. Right-click the object and click **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Miscellaneous** area, click **Orientation**. The **Orientation -> Analog Value** dialog box appears.
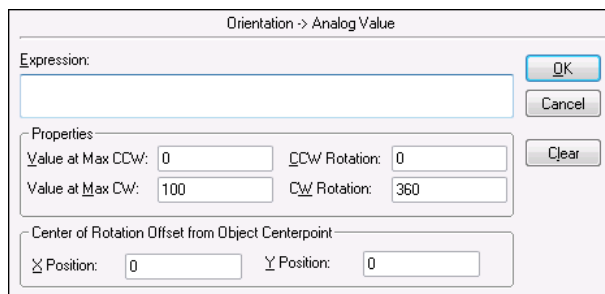
3. In the **Expression** box, type the name of an analog tag or an expression that equates to an analog value.

4. In the **Properties** area, do the following:

   a. In the **Value at Max CCW** box, type the value the expression must be for the object to rotate to the maximum counter-clockwise position.

   b. In the **Value at Max CW** box, type the value the expression must be for the object to rotate to its maximum clockwise position.

   c. In the **CCW Rotation** box, type the degrees the object rotates counter-clockwise when the **Value at Max CCW** is reached.

   d. In the **CW Rotation** box, type the degrees the object rotates clockwise when the **Value at Max CW** is reached.

5. In the **Center of Rotation Offset from Object Centerpoint** area, do the following:

   a. In the **X Position** box, type the horizontal offset of the rotation centerpoint. Enter the offset in pixels from the centerpoint of the object.

   b. In the **Y Position** box, type the vertical offset of the rotation centerpoint. Enter the offset in pixels from the centerpoint of the object.

6. Click **OK**.

Animating Sizes

You can vary the height and/or width of an object according to the value of an analog tag or expression by using object size links.

For example, a pressure indicator can become larger as pressure increases, or an object on a conveyor can appear to move toward the viewer by becoming larger.

Object size links not only control the size of an object, but the direction in which the object changes size through the use of an anchor for the animation.

To create an object size height link

1. Right-click the object and click **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Object Size** area, click **Height**. The **Object Height -> Analog Value** dialog box appears.

3. In the **Expression** box, type the name of an analog tag or an expression that equates to an analog value.

4. In the **Properties** area, do the following:

   a. In the **Value at Max Height** box, type the value of the tag or expression when the object reaches maximum height.

   b. In the **Value at Min Height** box, type the value of the tag or expression when the object reaches minimum height.

  c. In the **Max % Height** box, type the percentage of its original height that the object will be when the tagname or expression reaches the value set in the **Value at Max Height** box. The percent figures are expressed as a percentage of the drawn size of the object. The drawn size is always the 100% size.

  d. In the **Min % Height** box, type the percentage of its original height that the object will be when the tagname or expression reaches the value set in the **Value at Min Height** box. The percent figures are expressed as a percentage of the drawn size of the object. The drawn size is always the 100% size.

5. Select the **Anchor** point from which the object enlarges.

  o Select **Top** for the object to enlarge from its top downward.

  o Select **Middle** for the object to enlarge from its centerpoint outwards in both directions.

  o Select **Bottom** for the object to enlarge from its bottom upwards.

6. Click **OK**.

To create a object size width link

1. Right-click the object and click **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Object Size** area, click **Width**. The **Object Width -> Analog Value** dialog box appears.
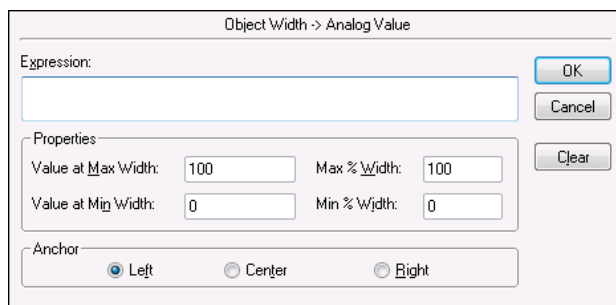
3. In the **Expression** box, type the name of an analog tag or an expression that equates to an analog value.

4. In the **Properties** area, do the following:

  a. In the **Value at Max Width** box, type the value of the tag or expression when the object reaches maximum width.

  b. In the **Value at Min Width** box, type the value of the tag or expression when the object reaches minimum width.

  c. In the **Max % Width** box, type the percentage of its original width that the object will be when the tagname or expression reaches the value set in the **Value at Max Width** box. The percent figures are expressed as a percentage of the drawn size of the object. The drawn size is always the 100% size.

  d. In the **Min % Width** box, type the percentage of its original width that the object will be when the tagname or expression reaches the value set in the **Value at Min Width** box. The percent figures are expressed as a percentage of the drawn size of the object. The drawn size is always the 100% size.

5. Select the **Anchor** point from which the object enlarges in width.

  o Select **Left** for the object to enlarge from its left side.

  o Select **Center** for the object to enlarge from its centerpoint outwards in both directions.

  o Select **Right** for the object to enlarge from its right side.

6. Click **OK**.

Animating Colors

You can animate color changes to any object by using color links. Changes can be based on the value of an analog or discrete tag, the value of an analog or discrete expression, or a discrete or analog alarm status.

You can use three kinds of color links to animate objects:

* Line Color

* Fill Color

* Text Color

For each of these three kinds of color links, four types of expressions can control color changes.

Expression Type	Changes the color based on
Discrete	The value of a discrete tag or expression.
Analog	The value of an analog tag or expression. You can define ten colors to represent differing values.
Discrete Alarm	The alarm state of a tag, Alarm Group, or Group Variable.
Analog Alarm	The alarm state of an analog tag, Alarm Group, or Group Variable. You can define five colors to represent five alarm conditions.

WARNING! Objects do not go into an alarm state when using an analog alarm link if the link is to a remote tag from an unconverted application created before InTouch version 7.11.

All discrete color links are created in the same way. The following procedure describes creating a fill color link.

To create a discrete fill color link

1. Right-click the object and click **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Fill Color** area, click **Discrete**. The **Fill Color -> Discrete Expression** dialog box appears.
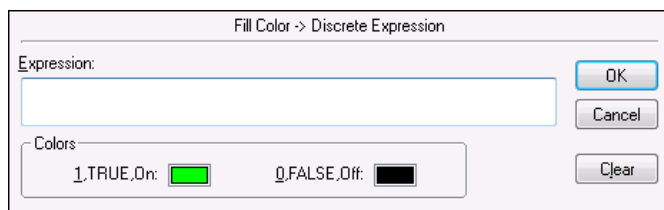
3. In the **Expression** box, type the name of a discrete tag or a discrete expression that equates to true or false.

   Discrete expressions can contain analog tags. For example, TankLevel >= 75. In this example, when the value of the variable "TankLevel" is greater than or equal to 75, the fill color of the object changes.

4. In the **Colors** area, click each color box to open the color palette. Select the color to use for each state.

5. Click **OK**.

To create an analog expression color link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Fill Color** area, click **Analog**. The **Fill Color -> Analog Expression** dialog box appears.
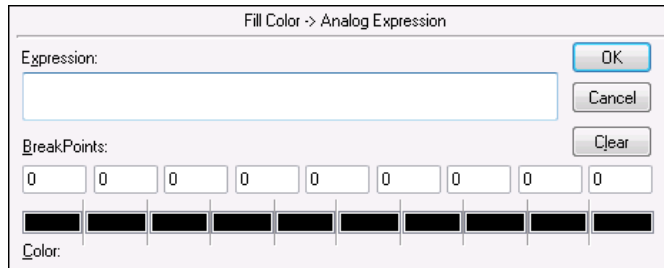
3. In the **Expression** box, type the name of an analog tag or an expression that equates to an analog value.

4. In the **Break Points** area, do the following.

    o   Specify the breakpoint values where the object changes color.

Tip: You do not have to use ten different colors. For example, if you only want the object to change color three times, type three values and use the same color for the remaining values. If you need a more versatile range, review the analog fill capabilities of Industrial graphics. For more information see, About InTouch HMI and ArchestrA Integration in the InTouch® HMI and ArchestrA® Integration Guide.

---

    o   In the **Colors** area, select a color for each breakpoint.

5. Click **OK**.

To create a discrete alarm status color link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Fill Color** area, click **Discrete Alarm**. The **Fill Color -> Discrete Tagname Alarm Status** dialog box appears.
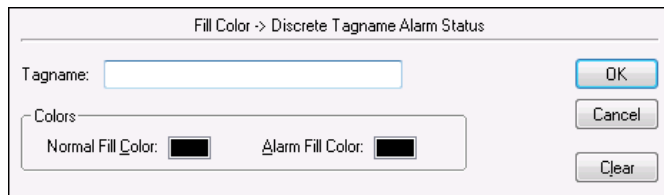
3. In the **Tagname** box, type the name of the discrete tag to associate with the object.

4. In the **Colors** area, select a color for each alarm state.

5. Click **OK**.

To create an analog alarm status color link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Fill Color** area, click **Analog Alarm**. The **Fill Color -> Analog Tagname Alarm Status** dialog box appears.
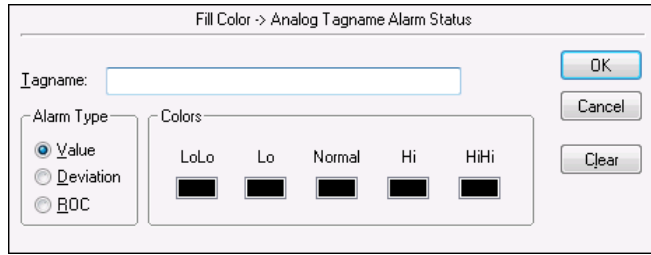
3. In the **Tagname** box, type the name of an analog tag to associated with the object.

In the **Alarm Type** area, select from one of the three types of alarms to associate with the object.

Alarm Type	Use up to
Value	Five colors to show the status of the value alarms.
Deviation	Three colors to show the status of the deviation alarms.
ROC (Rate of Change)	Two colors to show the status of a rate-of change alarm.

4. In the **Colors** area, select a color for each alarm state.

5. Click **OK**.

Animating Fill Levels

You can vary the fill level of an object using a percent fill link. The percent fill is based on the value of an analog tag or expression. You can create horizontal fills, vertical fills, or both.

For example, you can use a vertical fill link to show the level of liquid in a tank or a horizontal fill link to show the progress of a process.

You create the horizontal and vertical percent fill links the same way.

To create a percent fill link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Percent Fill** area, do one of the following:

    o   Click **Vertical**, the **Vertical Fill -> Analog Value** dialog box appears.
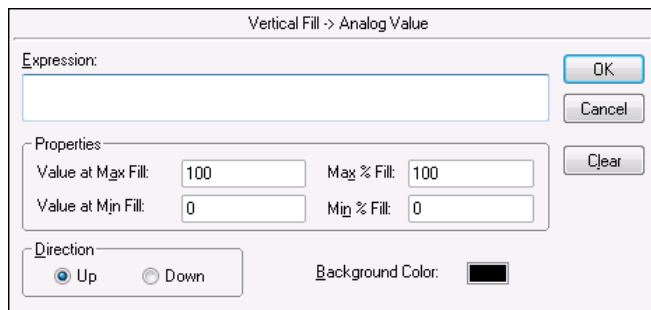
o   Click **Horizontal**, the **Horizontal Fill -> Analog Value** dialog box appears.
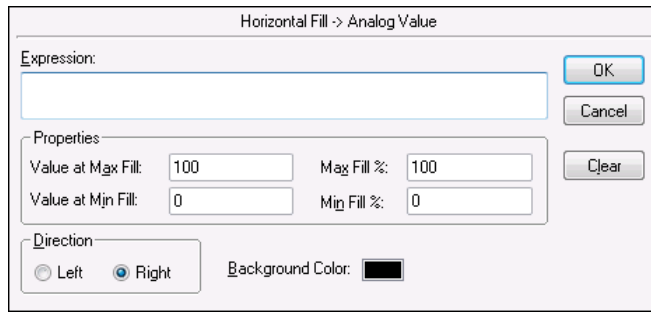
Horizontal Fill -> Analog Value	
Expression:	OK
	Cancel
Properties	
Value at Max Fill: 100 Max Fill %: 100	Clear
Value at Min Fill: 0 Min Fill %: 0	
Direction	
○ Left ● Right Background Color: ■	

3. In the **Expression** box, type the name of an analog tag or an expression that equates to an analog value.

4. In the **Properties** area, do the following:

    a. In the **Value at Max Fill** box, type the value of the expression that will result in the object being filled to its maximum level.

    b. In the **Value at Min Fill** box, type the value of the expression that will result in the object being filled to its minimum level.

    c. In the **Max % Fill** box, type the percentage (0-100) that the object will be filled when the expression reaches the level set in the **Value at Max Fill** box.

    d. In the **Min % Fill** box, type the percentage (0-100) that the object will be filled when the expression reaches the level set in the **Value at Min Fill** box.

5. In the **Direction** area, click the direction to fill from.

6. In the **Background Color** box, select the color of the unfilled portion of the object.

    o   The actual fill color is the color that you select for the object when you draw it.

    o   If you link both vertical percent fill and horizontal percent fill links to the same object, the last color you select is the background color.

7. Click **OK**.

Making Objects Blink

You can create animated objects that blink based on the values of tags by using blink links. For example, you can create an object that blinks red when a certain piece of equipment is on, or when an alarm set point is reached.

The blink animation synchronization enables all active blink animations on all open windows to blink in unison based on their common blink speed (slow, medium, or fast).

Tip: Discrete expressions can contain analog tagnames. For example, TankLevel > 75. In this example, when the value of the TankLevel tag is greater than 75, the object begins blinking.

To create a blink link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Miscellaneous** area, click **Blink**. The **Object Blinking -> Discrete Value** dialog box appears.
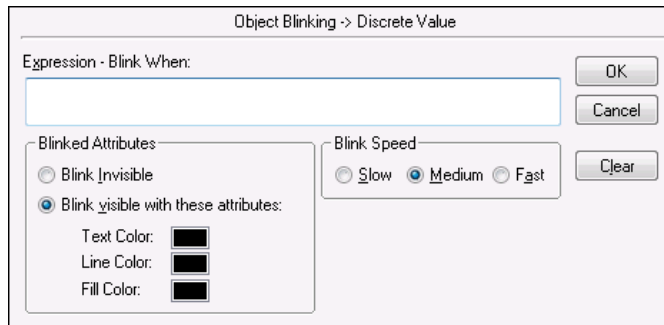
3. In the **Expression - Blink When** box, type the name of a discrete tag or an expression that equates to a discrete value.

4. In the **Blinked Attributes** area, do the following:

   o Click **Blink Invisible** to set the object to blink by disappearing and reappearing in the window.

   o Click **Blink visible with these attributes** to set the object to remain visible, but change color when activated.

   o Click the **TextColor**, **Line Color** or **Fill Color** boxes to select colors for those parts of the object. The color palette appears.

Note: If you select a fill blink color that is the same as the object's fill color, the object does not appear to blink.

5. In the **Blink Speed** area, set the blinking speed of the object. Click either **Slow**, **Medium**, or **Fast**.

6. Click **OK**.

To set the blink frequency for WindowMaker

1. On the **Special** menu, point to **Configure** and then click WindowViewer. The WindowViewer **Properties** dialog box appears.

2. In the **Blink Frequency** area, type the number of milliseconds to use for the three speeds.

Note: Changes you make to these settings are global and affect all blink speeds in your application.

3. Click **OK**.

Enabling Visibility

You can create links to hide objects based on the values of various tags by using visibility links. Using visibility links, you can:

- Create the impression that moving objects only move in one direction, by hiding them when they move in the wrong direction.

- Create the impression that a moving object has stopped.

- Cause an object such as an alarm or error message to become visible only when it is activated.

Tip: Discrete expressions can contain analog tags, for example, TankLevel >= 75. In this example, when the value of the tag, TankLevel is greater than or equal to 75, the object becomes visible in the window.

To create a visibility link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Miscellaneous** area, click **Visibility**. The **Object Visibility -> Discrete Value** dialog box appears.
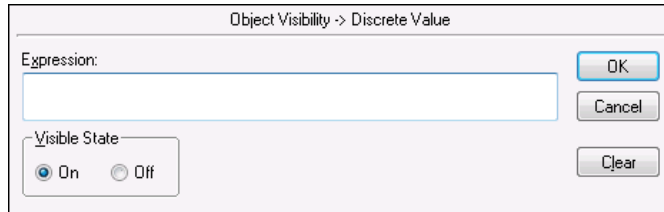
3. In the **Expression** box, type the name of a discrete tag or an expression that equates to a discrete value.

4. Select the **Visible State** for the object. If you select **Off**, the object is invisible when the value of the expression is true. If you select **On**, the object is visible when the value of the expression is true.

5. Click **OK**.

Disabling Objects

You can impose a level of security on your application with disable links. For example, you can disable touch sensitive objects based on operator access level or name. Or you can secure a button from tampering if no one is logged on.

A disabled state of ON means the touch functionality of the object or button is turned off and is not active as long as the expression is true.

Tip: Discrete expressions can contain analog tagnames. For example, TankLevel >= 75. In this example, when the value of the variable "TankLevel" is greater than or equal to 75, the object is disabled.

To create a disable link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Miscellaneous** area, click **Disable**. The **Object Disabled -> Discrete Value** dialog box appears.
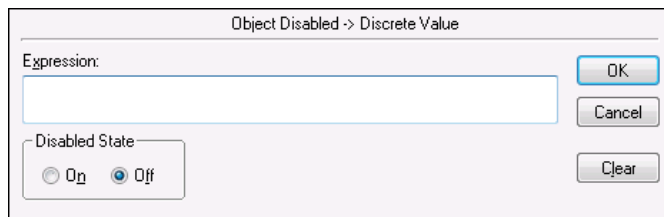
3. In the **Expression** box, type the name of a discrete tag or expression that equates to a discrete value.

4. In the **Disabled State** area, do one of the following:

   o Select **On** to set the disabled state so that the object does not activate while the discrete tag or expression is true.

   o Select **Off** to remove the disabled state and enable the object to function while the discrete tag or expression is true.

5. Click **OK**.

Configuring ToolTips

You can create ToolTips to give users information about an object on the screen by using tooltip links. ToolTips appear when the pointer moves over the object, and disappears when the pointer is moved away. The duration of time the tooltip appears and the positioning of the tooltip are determined by the operating system.

You can set either an expression or static text for your tooltips.

- Create a static tooltip to show the same message every time the tooltip appears.

- Create an expression tooltip so that every time the tooltip appears, the expression is evaluated and shown as the tooltip text.

For the following example expression, the text appears as the current value of the msgTooltipTag01 message tag.

```
msgTooltipTag01
```

For this example, the literal string appears followed by the current value of the iTemp tag and the result appears as the tooltip text:

```
"Current temp. is " + StringFromIntg (iTemp,10)
```

The width of the Tooltip window can be changed in Windows 7 and later versions of Windows. You can edit the InTouch.INI file to add an entry for Tooltip width.

Examples:

- Tooltipwidth=200

- A value of -1 (Tooltipwidth=-1) shows tooltip text on a single line without a line break

- If no tooltip width is specified, the default width of the tooltip window is 88.

To create a tooltip link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box opens.

2. In the **Miscellaneous** area, click **Tooltip**. The **Object Tooltip -> String Tagname** dialog box appears.
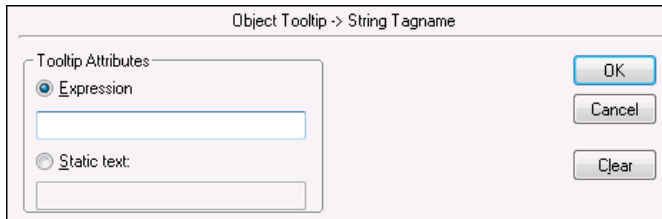
3. In the **Tooltip Attributes** area, select either **Expression** or **Static** text.

   o If you select **Expression**, enter an expression that evaluates to a message value. This can be a simple message tagname or a more complex expression.

   o If you select **Statictext**, enter a static message, up to 131 characters, as the tooltip text.

4. Click **OK**.

Positioning a Touch-Sensitive Window

You can make a window appear during run time in a precise location relative to a touch sensitive object. For example, an operator can select an object to see the status, name, or other data linked to that object. When the operator selects the object, either by a click or mouse-over, the window appears in the location you specify.

Use the script functions ShowAt() or ShowTopLeftAt() with the system read only tags $ObjHor and $ObjVer to locate the window relative to the object. You can also use fixed positions in these functions.

If the Windows **Display Properties** is set to the **Windows XP** theme, this functionality is erratic under certain circumstances.

The syntax looks like this:
```
ShowTopLeftAt (windowname, $ObjHor, $ObjVer);
```

where

windowname: the name of the window to be opened.

$ObjHor: the horizontal position of the center of the object selected.

$ObjVer: the vertical position of the center of the object selected.

The new window appears with its top left corner at the center of the selected object.

A similar script function opens the window with its center at the center of the selected object. The syntax looks like this:
```
ShowAt (windowname, $ObjHor, $ObjVer);
```

To open a window at the selected object

1. Design, name, and create the window to appear.

2. Right-click the object to trigger the window, and click **Animation Links**. The **Animation Links** dialog box appears.

3. In the **Touch Pushbuttons** area, select **Action**. The **Touch -> Action Script** dialog box appears.
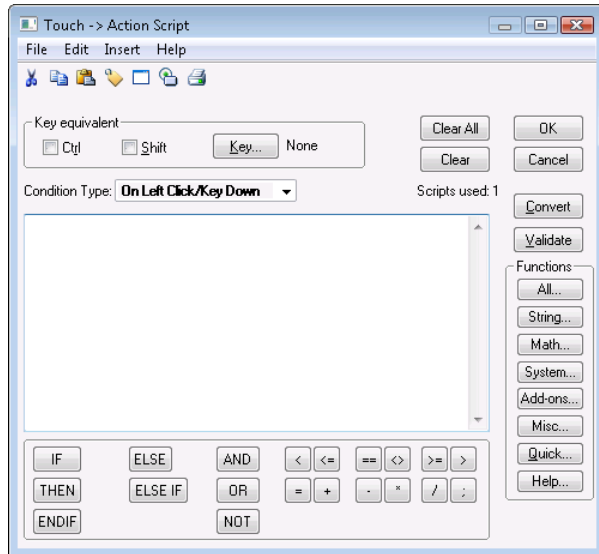
4. Type one of following scripts according to the syntax defined previously:
```
ShowTopLeftAt (windowname, $ObjHor, $ObjVer);
```

or
```
ShowAt (windowname, $ObjHor, $ObjVer);
```

5. In the **Condition Type** box, click the mouse action to open the window.

6. Click **OK**.

$ObjHor System Tag

Contains the horizontal pixel location of the center of the object in focus.

Category

system

Usage

`$ObjHor`

Data Type

Integer (read only)

See Also

$ObjVer

$ObjVer System Tag

Contains the vertical pixel location of the center the object in focus.

Category

system

Usage

`$ObjVer`

Data Type

Integer (read only)

See Also

$ObjHor

Data Entry Animations

To create objects that enable operator interaction, use touch links. Touch links enable operators to input data into a running application. For example, an operator can log on with a keyboard, turn a valve on or off, enter a new alarm setpoint, or start or stop a process.

A frame appears around a touch-sensitive object when it has the focus. An object gets the focus when the user moves the cursor over it or presses the TAB or arrow keys to move the focus to the object.

If you expect your users to use the TAB key to select touch sensitive objects, try to arrange them in horizontal patterns. Pressing the Tab key moves the focus from one object to another from left to right beginning at the top of the window then moving down.

If a touch link object contains text objects that are placed on top of each other, the top text object is the only one that appears.

An operator can activate a touch-sensitive object by clicking it, pressing an assigned key equivalent, pressing **Enter** when the object frame appears, or by actually touching it if using a touch screen display device.

You can create nine types of user input touch links:

Touch Link	Action
User Inputs	• Discrete
	• Analog
	• String

Touch Link	Action
Sliders	• Vertical
	• Horizontal
Pushbuttons	• Discrete Value
	• Action
	• Show Window
	• Hide Window

When a text field is used for input, text appears on the screen as the keys are pressed.

If you don't want text to appear as you type it, select the **Input Only** option in the configuration panel for the link.

Enabling Discrete Input

You can design operator input touch links to change the value of discrete tags from one state to another. For example to turn a pump on or off, use discrete links.

To create a discrete input link

1. Right-click an object and select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Touch Links** area, under **User Inputs**, select **Discrete**. The **Input -> Discrete Tagname** dialog box appears.
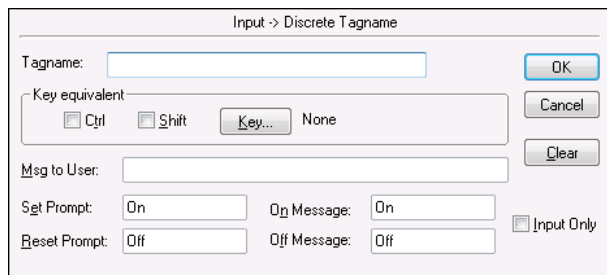
3. In the **Tagname** box, type the name of a discrete tag.

4. Optionally, in the **Key Equivalent** area, assign a key equivalent to the link. For more information, see *Creating Keyboard Shortcuts* on page 83.

5. Configure the discrete input details. Do the following:

   o In the **Msg to User** box, type the message to appear in the **Input** dialog box.

   o In the **Set Prompt** and **Reset Prompt** boxes, type the messages to appear on the buttons the operator will click to turn the discrete value on and off.

   o In the **On Message** and **Off Message** boxes, type the messages to appear in the text field associated with the object when the value is on or off.

6. Select the **InputOnly** check box to prevent the input from appearing in a text field associated with the object.

7. Click **OK**.

Enabling Analog Input

You can create an object for operators to input real values such as alarm set point or conveyor speed by using an analog input link.

To create an analog input link

1. Right-click an object and select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Touch Links** area, under **User Inputs**, select **Analog**. The **Input -> Analog Tagname** dialog box appears.

```
┌─────────────────────────────────────────────────────────────┐
│                     Input -> Analog Tagname                   │
│  Tagname: [_____]      [   OK   ]   │
│  ┌Key equivalent─────────────────────────┐                    │
│  │ □ Ctrl    □ Shift    [ Key... ] None   │     [ Cancel ]    │
│  └────────────────────────────────────────┘                   │
│  Msg to User: [_____]    [  Clear  ]  │
│  ┌Keypad?──┐  Minimum: [0      ]    □ Input Only               │
│  │ ○ Yes ● No│ Maximum: [100    ]                              │
│  └──────────┘                                                  │
│  ┌Formatting────────────────────────────────────────┐         │
│  │ [Text String    ▼]  Precision: [0 ▲▼]             │         │
│  │ □ Fixed Width    Bits From: [0 ▲▼] To: [31 ▲▼]    │         │
│  └────────────────────────────────────────────────────┘        │
└─────────────────────────────────────────────────────────────┘
```

3. In the **Tagname** box, type the name of an analog tag.

4. In the **Key Equivalent** area, assign a key equivalent to the link. For more information, see *Creating Keyboard Shortcuts* on page 83.

5. Configure the analog input details. Do the following:

    o In the **Keypad?** area, click **Yes** if you want to show an on-screen numeric keypad to enter a value. In the **Msg to User** box, type the prompt message that you want to appear in keypad.

    o In the **Minimum** and **Maximum** boxes, type the minimum and maximum input values for the tag. These settings limit the user input for the animation at run time. Constant analog values (integer or real) are supported, as well as references. The default values are 1 and 100, respectively. The maximum value must be greater than the minimum value.

       If you are using a reference, see the Using References for the Minimum and Maximum Limits of an Analog Input Animation section for more information.

6. Select the **InputOnly** check box to prevent the input from appearing in a text field associated with the object.

7. In the **Formatting** area, in the list, click each data type for which you want to configure run time advanced formatting. The **Fixed Width** check box, **Precision** box, and **Bits From** and **To** boxes become available based on the data type you select. For information on configuring these options, see the Advanced Formatting for Text section.

8. Click **OK**.

Using References for the Minimum and Maximum Limits of an Analog Input Animation

Constant analog values and references are supported for the minimum and maximum limits of an analog user input animation. You configure these limits using the **Input -> Analog Tagname** dialog box.

A reference can either be a remote analog tag or a local analog tag like a memory tag or an I/O tag.

- When InTouch I/O tags are used for the minimum and maximum values, only the InTouch I/O tags are evaluated. If the I/O tag refers to a remote tag reference that is assigned a string value, then the I/O tag value is evaluated as 0.

- When remote tag references are used for the minimum and maximum values, the references are internally evaluated as strings. If the remote references are assigned with string values that fail to get converted to analog values, a warning message appears in the Logger. The **Minimum** and **Maximum** boxes show values based on the data type of the reference to which user input is configured.

- When the user input in the **Minimum** or **Maximum** field contains the letter 'e', the input is either a string or an exponential value. When the user input contains numbers followed by the letter 'e', followed by numbers (±N e ±N format, where N is a number), the user input is considered as an exponential value. When the user input containing 'e' contains any other alphabet (even another 'e'), the user input is considered as a reference name.

If a reference goes into bad quality at run time, the following occurs.

For the reference in the **Minimum** box:

- If the primary reference of the animation is an InTouch tag, the tag's configured minimum value is retrieved from the tag database and used for the minimum value.

- If the primary reference of the animation is an external reference, the minimum value of the data type is used. If the configured data point is a string, the last known value is used to determine the data type.
  - If the last known value contains a decimal point, the data type is real.
  - If the last known value does not contain a decimal point, the data type is an integer.
  - If there is no last known value, then integer is used.

For the reference in the **Maximum** box:

- If the primary reference of the animation is an InTouch tag, the tag's configured maximum value is retrieved from the tag database and used for the maximum value.

- If the primary reference of the animation is an external reference, the maximum value of the data type is used. If the configured data point is a string, the last known value is used to determine the data type.
  - If the last known value contains a decimal point, the data type is real.
  - If the last known value does not contain a decimal point, the data type is an integer.
  - If there is no last known value, then integer is used.

Enabling String Input

You can create an object to enable users to input text strings such as batch names, operator ID, or passwords by using string input links.

To create a string input link

1. Right-click an object and select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Touch Links** area, under **User Inputs**, select **String**. The **Input -> String Tagname** dialog box appears.
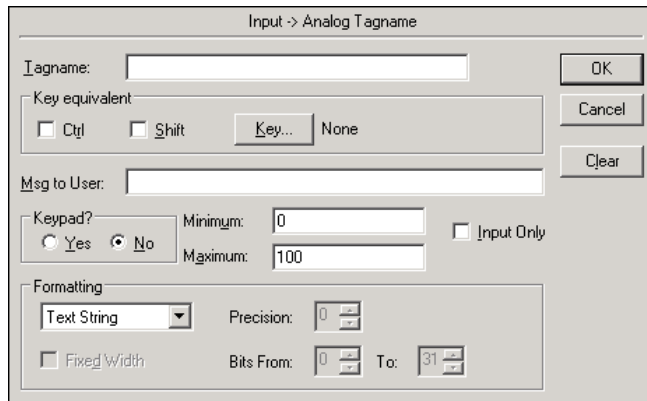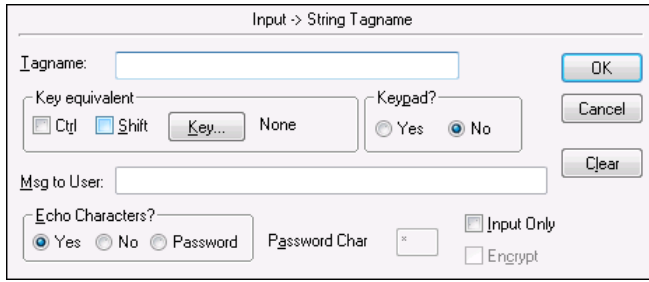

3. In the **Tagname** box, type the name of a message tag.

4. Optionally configure a key equivalent and/or keyboard.

   o In the **Key Equivalent** area, assign a key equivalent to the link. For more information, see *Creating Keyboard Shortcuts* on page 83.

   o In the **Keypad?** area, click **Yes** if you want to show a keyboard for inputting the new value. In the **Msg to User** box, type the message to appear in keyboard.

5. In the **Echo Characters?** area, select if you want characters to appear in the input box as the user types the string.

   o Click **Yes** to show the typed text in the input box.

   o Click **No** to not show the typed text.

   o Click **Password** to use a "masking" character instead of the typed text. In the **Password Char** box, type the masking character. Select the **Encrypt** check box to encrypt the password.

Important: Password encryption only works within the context of the InTouch HMI. Do not encrypt the string if you want to pass it to an external security system, such as the operating system or a SQL Server database. The external security system cannot read the encrypted password string and access will fail.

6. Select the **InputOnly** check box to prevent the input from appearing in a text field associated with the object.

7. Click **OK**.

Enabling Sliders

You can create objects that users can drag back and forth through the use of slider touch links. As the user moves the object, it alters the value of the tag linked to it. You can link an object to a horizontal or a vertical slider.

When you make an object into a slider, you set the reference location, which is the point on the object the cursor uses to lock onto it.

You can use both horizontal and vertical links on a single object, so that the value of two analog tags are altered simultaneously.

To create a horizontal slider link

1. Right-click the object and click **Animation Links**. The **Animation Link** selection dialog box appears.

2.   In the **Slider** area, click **Horizontal**. The **HorizontalSlider** dialog box appears.
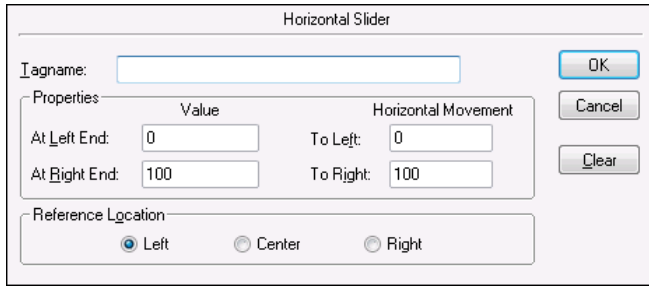
3.   In the **Tagname** box, type the name of an analog tag.

4.   In the **Properties** area, do the following:

   a.   In the **At Left End** box, type the value for the tag when the slider is in its farthest left position.

   b.   In the **At Right End** box, type the value for the tag when the slider is in its farthest right position.

   c.   In the **To Left** box, type the number of pixels the slider can move to the left.

   d.   In the **To Right** box, type the number of pixels the slider can move to the right.

5.   In the **Reference Location** area, click the location on the object that the cursor will use to lock onto the object.

6.   Click **OK**.

To create a vertical slider link

1.   Right-click the object and click **Animation Links**. The **Animation Links** dialog box appears.

2.   In the **Slider** area, click **Vertical**. The **VerticalSlider** dialog box appears.
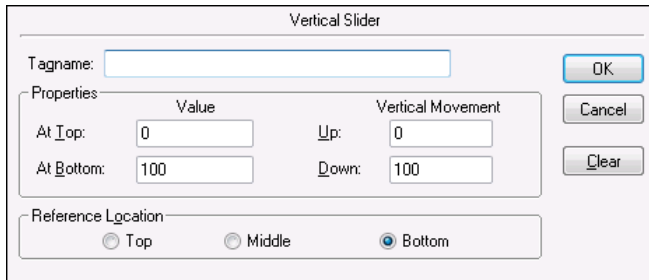
3.   In the **Tagname** box, type the name of an analog tag.

4.   In the **Properties** area, do the following:

   a.   In the **At Top** box, type the value for the tag when the slider is in its farthest up position.

   b.   In the **At Bottom** box, type the value for the tag when the slider is in its farthest down position.

   c.   In the **Up** box, type the number of pixels the slider can move up.

   d.   In the **Down** box, type the number of pixels the slider can move down.

5.   In the **Reference Location** area, click the location on the object that the cursor will use to lock onto the object.

6.   Click **OK**.

Enabling Push Buttons

You can create touch sensitive objects that start action scripts by using touch push button links.

Action scripts can set tags to specific values, start and control other applications, execute functions and so on.

To create a discrete value touch link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Touch Pushbutton** area, click **Discrete Value**. The **Pushbutton -> Discrete Value** dialog box appears.
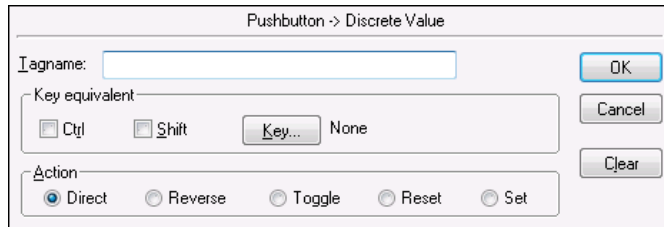
3. In the **Tagname** box, type a discrete type tagname.

4. Click **Key** to assign a key equivalent to the link.

5. In the **Action** area, click one of the following types:

    o   Click **Direct** to set the value equal to 1 as long as the push button is pressed and held down. The value automatically resets to 0 when the button is released.

    o   Click **Reverse** to set the value equal to 0 when the push button is pressed and held down. The value automatically resets to 1 when the button is released.

    o   Click **Toggle** to reverse the state of the discrete tag.

    o   Click **Reset** to set the value equal to 0.

    o   Click **Set** to set the value equal to 1.

6. Click **OK**.

To create an action script link

1. Right-click the object and then select **Animation Links**. The Animation Links dialog box appears.

2. In the **Touch Pushbutton** area, click **Action**. The **Touch-> Action Script** editor appears.
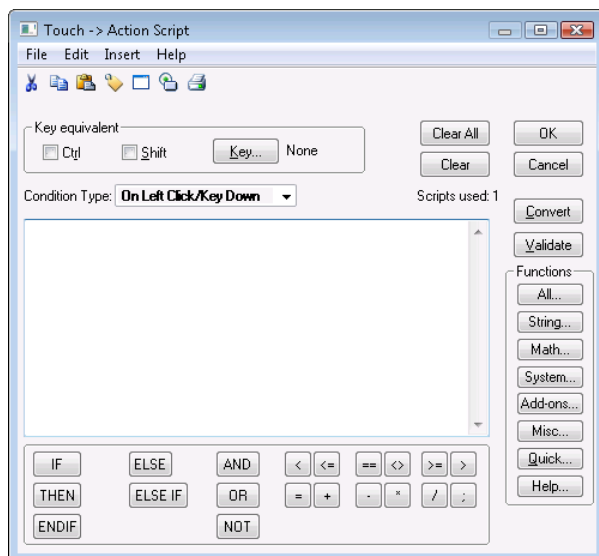
3. In the **Condition Type** list, select a script type. For more information on the types of action scripts, see Script Triggers in the InTouch® HMI Scripting and Logic Guide

Note: If you assign a key equivalent link to an action push button and to the same key is used for a key script, the key equivalent link you assign takes precedence over the key script.

4. In the **Script Editor** window, type the script to execute when the object is activated.

5. Click **OK**.

Opening and Closing Windows

You can create touch links to open and close other InTouch windows by using show window and hide window links.

You can program objects to open more than one window at a time. However, if you program your link to open more than one window, be aware of any intersections and window types.

If one of the opening windows is a replace type window and intersects another opening window, the other window closes before opening.

To create a show (or hide) window link

1. Right-click the object and then select **Animation Links**. The **Animation Links** dialog box appears.

2. In the **Touch Pushbutton** area, click **Show Window** or **Hide Window**. The **Windows to Show when touched** or **Windows to Hide when touched** dialog box appears.
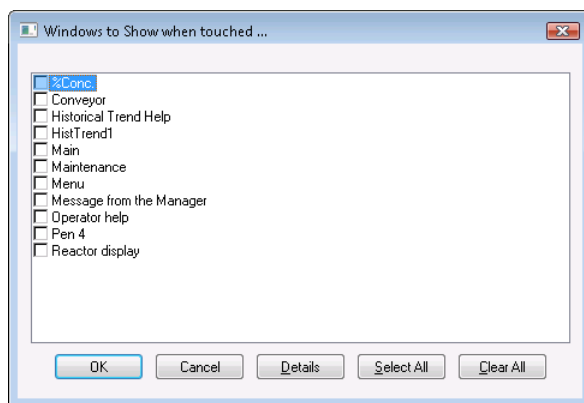
3. Select the window(s) to open or hide.

4. Click **OK**.

Configuring On-Screen Keyboards

On-screen keyboards enable operator input in situations where a keyboard is not connected to the computer.

You can designate one of three on-screen keyboard types.

- The standard InTouch keyboard or keypad. This is the default keyboard.

- The Windows system keyboard. The Windows keyboard is a fully functional QWERTY-type keyboard with function keys, print screen key, number lock key, directional arrows, and so on.

- The re-sizable keyboard or keypad. This keyboard can be resized during run time.

You can also open the keyboard by using the DialogStringEntry() and DialogValueEntry() function in a script. For more information, see *DialogStringEntry() Function* on page 77 and *DialogValueEntry() Function* on page 78.

To configure the on screen keyboard type

1. On the **Special** menu, point to **Configure** and then click WindowViewer. The WindowViewer **Properties** dialog box appears.

2. In the **Keyboard** area, select type of keyboard you want.
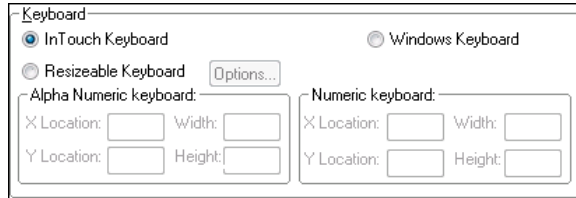
3. If you select the re-sizeable keyboard, click **Options** to select font, location, and dimension properties of the keyboard.

4. Click **OK**.

To make an on screen keyboard appear

1. Configure the on-screen keyboard type.

2. Right-click on a text object and select **Animation Links**. The **Animation Links** dialog box appears.

3. In the **Touch Links** area, under **User Inputs**, select **String**. The **Input->String Tagname** dialog box appears.
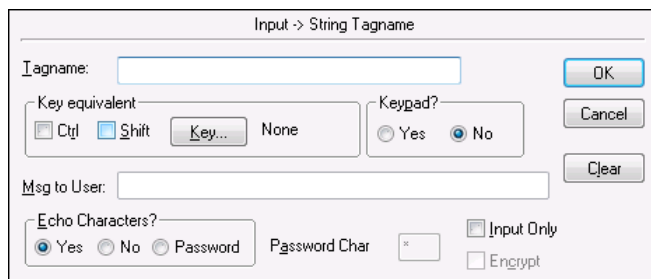
4. In the **Keypad?** area, select **Yes**.

5. Click **OK**.

DialogStringEntry() Function

Shows an alphanumeric keyboard on the screen, enabling the operator to change the current string value of a message tag in the Tagname Dictionary.

Category

misc

Syntax

```
[Result=]DialogStringEntry(MessageTag_Text, UserPrompt_Text);
```

Parameters

MessageTag_Text
The name of the message tag to be modified. This value is a string value. Specify the tagname within quotes or use the .Name dotfield without quotes. You can also use a message tag as a pointer.

UserPrompt_Text
The user message to show at the top of the keyboard.

Return Value

Returns one of the following integer values:

      0 = Cancel was pressed.

      1 = OK was pressed.

     -1 = Internal error.

     -2 = Could not initiate.

     -3 = Tagname not defined.

     -4 = Tagname is not a Message type.

     -5 = Unable to write.

Remarks

This function is used primarily in applications containing touch screens.

Example
```
Errmsg=DialogStringEntry(MyMessageTag.Name,  "Enter a new string...");
Errmsg=DialogStringEntry("MyMessageTag","Enter a new string...");
```

For example, the following script opens an alphanumeric keyboard, allowing modification of
MyMessageTag while showing the message "Enter a new string..." at the top of the keyboard:
```
MessageTagX="MyMessageTag"; {assign the string MyMessageTag (which is actually
the tagname to be modified) to the Memory Message tagname MessageTagX}
MessageDisplay="Enter a new string..."; {assign the new message string to the
Memory Message tagname MessageDisplay}
Errmsg=DialogStringEntry(MessageTagX, MessageDisplay); {quotes are not
required because MessageTagX was defined as a Message tagname}
```

See Also

DialogValueEntry()

DialogValueEntry() Function

Shows the numeric keypad on the screen, enabling the user to change the current value of a discrete,
integer or real tag.

Category

misc

Syntax
```
[Result=] DialogValueEntry(ValueTag_Text, LowLimit, HighLimit,
UserPrompt_Text);
```

Parameters

ValueTag_Text
The name of the discrete, integer, or real tag to modify. This value is a string value. Specify the
tagname within quotes or use the .Name field without quotes. You can also use a message tag as a
pointer.

LowLimit
The minimum value of the tag. (This should be >= the tagname's definition for minimum value,
minimum raw or minimum engineering unit, as applicable).

HighLimit
The maximum value of the tag. (This should be <= the tagname's definition for maximum value,
maximum raw or maximum engineering unit, as applicable).

UserPrompt_Text
The user message to show at the top of the keypad.

Return Value

Returns one of the following integer values:

0 = Cancel was pressed.

1 = OK was pressed.

-1 = Highlimit<=Lowlimit.

-2 = Could not initiate.

-3 = Tagname not defined.

-4 = Tagname is not a Discrete, Integer or Real type.

-5 = Write failed.

Remarks

This function is used primarily in applications containing touch screens.

Example(s)

```
Errmsg=DialogValueEntry(MyIntegerTag.Name, MyIntegerTag.MinEU,
MyIntegerTag.MaxEU, "Enter a new value...");
Errmsg=DialogValueEntry("MyIntegerTag", -100, 100, "Enter a new value...");
```

For example, the following script brings up the numeric keypad, allowing modification of MyIntegerTag using a minimum and maximum limit of -100 and 100 (respectively), while showing the message "Enter a new value..." at the top of the keypad:

```
TagnameX="MyIntegerTag"; {assign the string MyIntegerTag (which is actually the
tagname to be modified) to the Memory Message tagname TagnameX}
Min=-100; {assign the minimum value allowed for the tagname to the Memory
Real/Integer tagname Min}
Max=100; {assign the minimum value allowed for the tagname to the Memory
Real/Integer tagname Max}
MessageDisplay="Enter a new value..."; {assign the new message string to the
Memory Message tagname MessageDisplay}
Errmsg=DialogValueEntry(TagnameX, Min, Max, MessageDisplay); {quotes are not
required because TagnameX was defined as a Message tagname. By assigning a
Discrete, Integer or Real tagname to TagnameX, the function will modify that
assigned tagname}
```

See Also

DialogStringEntry()

Common Animation Tasks

Common animation tasks include selecting tags, creating keyboard shortcuts, changing tagname references and replacing placeholder tags.

Selecting Tags or Attributes

Use the **Select Tag** dialog box, also called the tag browser, to select:

• Tags defined in a local or remote InTouch application.

• ArchestrA object attributes with the Galaxy Browser.

You open the **Select Tag** dialog box by double-clicking in any text box that requires a tag name for input.

Selecting an InTouch Tag

You can select tags defined in a local or remote InTouch application. You can create references to tags in any tag source that supports the Tagname Dictionary interface.

For example, remote tag references enable your application to access data from I/O Servers without creating tags in the local Tagname Dictionary.

You can set a dotfield for each InTouch tag you select. Dotfields can access, monitor, and modify tag properties. If you do not select a dotfield, the .Value dotfield is used. For more information about dotfields, see Using Tag Dotfields to View or Change Tag Properties in the InTouch® HMI Data Management Guide

To select an InTouch tag

1. Open the **Select Tag** dialog box by double-clicking in any text box that requires a tag name for input.
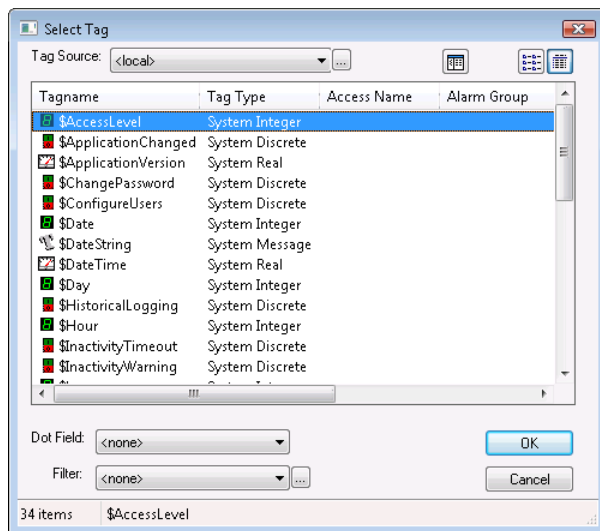
2. In the **Tag Source** list, click the name for the tag source or click the browse button to define a new tag source to use.

    For more information on defining tag sources, see Data Access with I/Oin the InTouch® HMI Data Management Guide

3. In the **Filter** list, click a filter to reduce the number of tagnames shown in the window. To define a filter, click the ellipsis button. For more information, see *Creating a Tag Filter* on page 81.

4. Select a tagname in the window.

    You can change the way the tagnames are shown in **Select Tag** dialog box. For more information, see *Changing the View in the Select Tag Dialog Box* on page 82.

5. In the **Dot Field** list, click a dotfield to append to the selected tagname.

    Dotfields can access, monitor and modify tag properties. If you do not select a dotfield, the .Value dotfield is used. Click **OK**.

Selecting an ArchestrA Object Attribute

You can select attributes that are associated with an ArchestrA object. To do this, you must first add the Galaxy that contains the object as a data source for your InTouch application. For more information about data sources, see Data Access with I/O in the InTouch® HMI Data Management Guide

To select an object attribute

1. Open the **Select Tag** dialog box by double-clicking in any text box that requires a tag name for input.
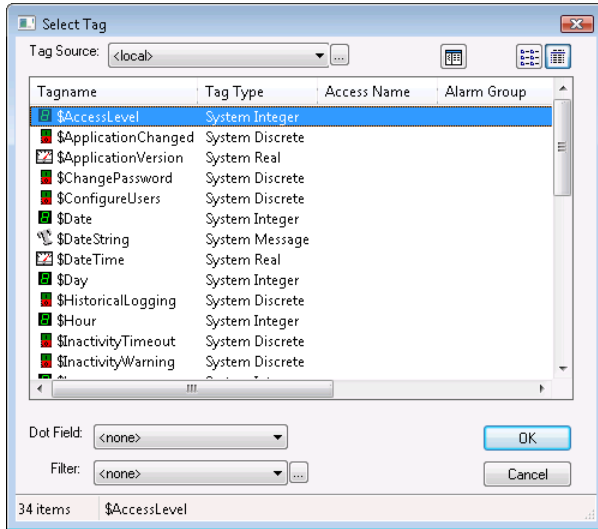
2. In the **Tag Source** list, click the name of the tag source that uses the Galaxy or click the browse button to define a new tag source to use. The **Galaxy Browser** dialog box appears.

3. Use the **Galaxy Browser** to find and select an ArchestrA object attribute. For more information, see the Application Server documentation.

4. Click **OK** to close the **Galaxy Browser**. The attribute reference appears in the text box that requires a tag name.

Note: To return to the **Select Tag** dialog box from the **Galaxy Browser**, click the **Back** button in the bottom right corner of the **Galaxy Browser**.

Creating a Tag Filter

If you have a large number of tags in your Tagname Dictionary, finding the tag for an animation can be cumbersome. For example, you may only want to see the tags assigned to a particular Access Name or Alarm Group. You can configure a filter so that only a subset of tags are shown in the **Select Tag** dialog box.

You can use the following wildcard expressions in your filter.

- The multiple character wildcard is the asterisk (*). For example, "Asyn*" searches for all tagnames beginning with the characters "Asyn".

- The single character wildcard is the question mark (?). For example, the "Tag?" filter searches for all four-character tagnames that begin with "Tag". The "Tag*" filter searches for all tagnames that begin with "Tag".

- Any sequence of valid tagname characters, together with the two wildcard characters, is acceptable in a filter. Valid tagname characters are: A-Z, a-z, 0-9, !, @, -, #, $, %, _ and &.

To define a search filter

1. In the **Select Tag** dialog box, click the ellipsis button next to the **Filter** list. The **Define Tag Filter** dialog box appears.
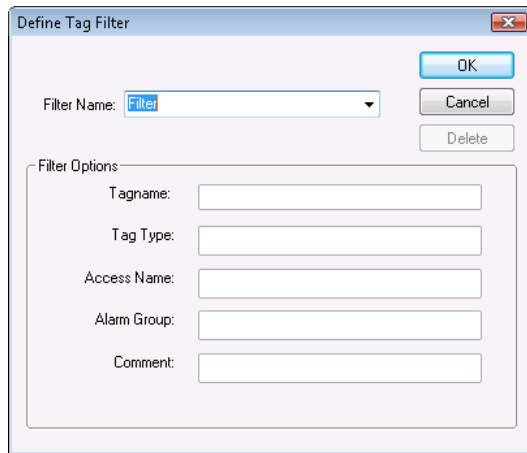
2. In the **Filter Name** box, type a filter name.

3. In the **Filter Options** area, configure the filter criteria. Do any of the following:

    o   In the **Tagname** box, type the tag name.

    o   In the **Tag Type** box, type the tag type.

    o   In the **Access Name** box, type the local Access Name.

    o   In the **Alarm Group** box, type the Alarm Group name.

    o   In the **Comment** box, type the comment expression.

4. Click **OK**. The **Filter Name** appears in the **Filter** list in the **Select Tag** dialog box. You can select the filter to show the tags meeting the filter criteria.

To delete a search filter

1. In the **FilterName** box, click the filter.

2. Click **Delete**.

Changing the View in the Select Tag Dialog Box

The **Select Tag** dialog box has three different views: List, Details, and Tree View.

To see the	Click the	Description
List view	List view button	Small icons appear next to the tagnames according to the type of each tagname.
Details view	Details view button	You see the same small icons and tagnames, plus Tag Types, Access Name, Alarm Group, and Comments. Sort the list by clicking the column headers.
Tree View	Tree view icon	The Tree View displays the tagnames in two views.
		You can access the member tagnames in any SuperTag template.

Creating Keyboard Shortcuts

You can assign a specific key on the keyboard to activate certain animation links by using key equivalent links. The key equivalent is only operational when the object with the link is visible or selected. If the object has a visibility or disable link, the key equivalent is not active when the object is disabled.

You can define the same key in multiple windows. However, the definition in the most recently opened window is the active one. In the case of overlay windows, the key is active in the window on top.

Note: If any object or action push button in the active window is assigned to the same key used for a key action script, the key equivalent link on the key in the active window takes precedence over the execution of the key action script.

The animation links that support key equivalents appears the **Key Equivalent** area in their link dialog boxes.

Key links are only available for function keys 1-16. If you are using a custom keyboard that has more than 16 function keys, get a device driver from your manufacturer to access the extended function keys on your system.

To assign a key to a link

1. Open the **Animation Links** dialog box for the type of link you are configuring.

2. Select **Ctrl** and/or **Shift** if you want the operator to hold down either or both of these keys when pressing the key equivalent.

3. Click **Key**. The **Choose key** dialog box appears.

4. Click the key to assign to the link. The **Animation Link** dialog box reappears with the name of the selected key next to the **Key** button.

5. Click **OK**.

Changing Tagname References

When you duplicate an object, you get an exact copy of the original including links, animation, scripts and so on. However, if you need to use a different tag on a duplicated object, you must substitute a tag.

You can select and substitute a tag for any object and you can select multiple objects and substitute all their tags at the same time.

If your system license supports a limited number of tagnames, convert your local tags to remote tag references to reduce the number of tags defined in your local Tagname Dictionary.

Substituting a tag works for all tags and references.

To substitute a tag with another tag

1. Select the object(s) associated with the tag to change to another tag.

2. On the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box appears.

3. In the **New Name** box, type the new tagname.

   o If you double-click a tag in the **New Name** box, its definition in the Tagname Dictionary appears.

   o If you erase the tagname then double-click the blank box, the **Select Tag** dialog box appears.

4. Click **OK**. The tag associated with the object is automatically changed.

Converting Placeholder Tags

If you import or export a window or QuickScript to or from your current application, all the tags associated with that window or QuickScript are transferred with the window.

The local tags are not automatically added to your application database. Instead, they are automatically marked as placeholder tags.

Remote tag references are not affected, and are not marked as placeholder tags.

You must convert the placeholder tags to existing local tags, define them in the new application's Tagname Dictionary, or make them remote tag references.

Notice the placeholders ?d:, ?i:, ?m: and ?r: preceding the tags. They indicate the type that the tag was originally defined as:

Placeholder Symbol	Tagname Type
d	Discrete
i	Integer
m	Message
r	Real

Note: Remote tag references are not shown as placeholders but as remote tag references such as: **PLC2:Temperature**.

You can use several methods in the **Substitute Tagnames** dialog box to convert placeholder tags to local tags. For more information, see Exporting and Importing Tag Definitions, Windows, and Scriptsin the InTouch® HMI Application Management and Extension Guide.

Tip: If you manually convert tagnames and you no longer need the original tag defined in the original Tagname Dictionary, you can update the tagname use counts and then delete the unused tag.

By importing a window or QuickScript from another application, and converting all of the tagnames associated with the animation links or QuickScript(s) to remote tagname references, you can create an application that instantly receives data from hundreds of remote tagnames without defining a single tag in your local Tagname Dictionary.

Advanced Formatting for Text

Using InTouch WindowMaker, you can configure the intelligent or advanced formatting options, which will be used at run time by InTouch WindowViewer. These options only apply if you are using the advanced formatting features on an analog value display or analog user input animation. The present value of InTouch WindowViewer will be retained if you do not configure advanced formatting options.

The advanced formatting options appear in the **Input -> Analog Tagname** dialog box and the **Output -> Analog Expression** dialog box.

Advanced formatting of numeric value display provides you with a native way of formatting numerical data based on data type and data quality. Advanced formatting enables you to format analog values at run time in InTouch WindowViewer.

You can either use text string or enable the advanced formatting options on an individual animation basis. When you migrate an animation from an earlier InTouch version, or create an animation for the first time, text string is set as the default setting.

The advanced formatting configuration options are:

- Text string, which uses the existing InTouch formatting.

- Real, which is formatted as the real value based on the size of the value at run time. The global decimal precision configuration is used to determine the number of decimal places.

- Fixed decimal point, of which the formatting depends on the type of values:

  o Real values: The value is formatted with the user-specified number of decimal places configured using the **Precision** control.

  o Integer values: The value is formatted as an integer without a decimal point and is indented on the right to accommodate for where the decimal places would have been.

  o Discrete values: The value is formatted as a 0 or 1 without a decimal point and is indented on the right to accommodate for where the decimal places would have been.

- Integer, which is always formatted as an integer without a decimal point.

- Exponential, which is always formatted as an exponent taking into consideration the fixed decimal point.

- Hex, which is displayed in hexadecimal format based on the configured Boolean range.

- Binary, which is always formatted as a binary string taking into consideration the configured Boolean range.

In the **Fixed Width** box, you can increase the text size specified in design time. This option is not available if **Text String** formatting is selected for the formatting style.

At design time, if you have selected the **Fixed Width** check box in the animation configuration, then the value rendered at run time does not exceed the length of the text field. If the value exceeds the length of the text field, the globally configured "Fixed Field Width Too Large Character" takes the length of the text field. Formatting with fixed width stops the text element from expanding horizontally at run time to accommodate a value size.

If you do not specify any text for the text field at design time, the fixed width setting is ignored and the full value is shown. For instance, if you enter "" (no text) at design time, it is equivalent to 0 for the text field width. This does not allow any text to be rendered if the **Fixed Width** option is enabled. In this situation, the **Fixed Field Width** setting is set to "None" as default at run time.

If you click **Clear**, the advanced formatting options are reset to the following:

- **Text String** is selected for the **Formatting** option.

- The **Fixed Field Width** option is cleared and subsequently disabled since this option is not available when **Text String** formatting is selected.

The following values are set by default:

- 0 is set for the Fixed Decimal precision control.

- 0 is set for the Exponential precision control.

- 0 and 31 are set for the Hex bit range controls.

- 0 and 31 are set for the Binary bit range controls.

Format Controls:

- The Precision control accepts an integer value and is only enabled if you have selected the **Fixed Dec. Point** or **Exponential** options. The valid range of integer values for this control is from 0 through 8, with 0 being the default value. The bit range consists of two spin controls, each of which takes integers as inputs. These controls are only available if you have selected either the **Hex** or the **Binary** option. For these two controls, you need to specify the bit value, ranging from 0 through 31. You can specify the bit range in the forward or reverse order. You can also specify a single bit, using the same bit specification for the range. The default numeric range for the bit specification is 0 through 31. The range is zero based to stay consistent with other places in InTouch where bits are specified.

For versions of InTouch earlier than 3.5 that contain configured analog user input animations or analog value display animations, the migrated animation is set to **Text String** formatting and advanced formatting is off.

For later versions, the advanced formatting options are migrated forward.

Formatting Text Objects as Numbers

Text objects can be used to display static or dynamic numeric values. You can show the integer or real value of an analog tag by associating a Value Display - Analog animation link to a text object.

To specify the format of an analog tag value shown by animating a text object, the following characters are used:

Character	Description
#	Forces the display of an analog value to an integer.
0	Represents a digit at each specified position of a real number. Forces leading zeros to the integer part of a number.
,	Inserts a comma at the specified position of a real number.
.	Places a decimal point at the specified position of a real number.

Other text format properties like font, size, and color apply to numeric values shown in the string.

Examples of Formatting Text Objects as Numbers

The following table shows examples of displayed analog values when animated format characters are used in a text object. In these examples, the current value of a temperature I/O tag is 123.45 C°. This tag is assigned as the reference value of a text object using an Analog animation link.

Formatted Text Object	Displayed Value of the Text Object
Temp = # C	124 C Displays the temperature as an integer. Only one # needs to be entered in the text object.
Temp = #.# C	123.5 C The fractional part of the temperature is rounded to a single number using the # character. The integer part of the temperature is unchanged.
Temp = 00000 C	00124 C Adds leading zeros to the integer part of the temperature.
Temp = 000.0 C	123.5 C The fractional part of the temperature is rounded to a single digit because of the single zero in the format.

Formatted Text Object	Displayed Value of the Text Object
Temp = 00.00 C	123.45 C The original temperature value remains unchanged.
Temp = 0,000.#	0,123.5 C A leading zero and comma are forced at the integer part of the temperature reference value. The fractional part is rounded to a single number using the # character.

To specify the format of text objects as numbers

1. Create a text object in a window and insert one or more characters to format a reference value associated with the text object.

2. Double-click the text object to show the **Animation Links** selection dialog box.

3. In the **Value Display** section, click **Analog**. The **Output -> Analog Expression** dialog box appears.
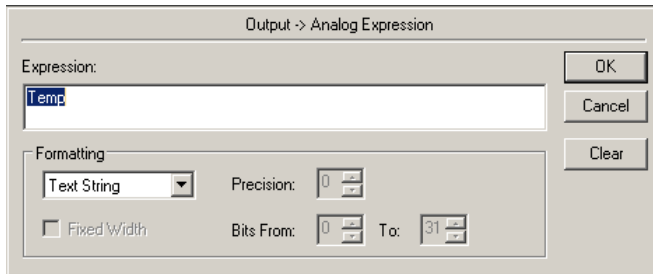
4. In the **Expression** field, type an analog tagname or expression.

5. Click **OK**.

6. Save your changes to the window.

7. Run the application to verify the number appears correctly in the text object.

CHAPTER 5

Wizards

About Wizards

Wizards are pre-designed, pre-built, and pre-programmed objects you only need to select, place and configure for your application.
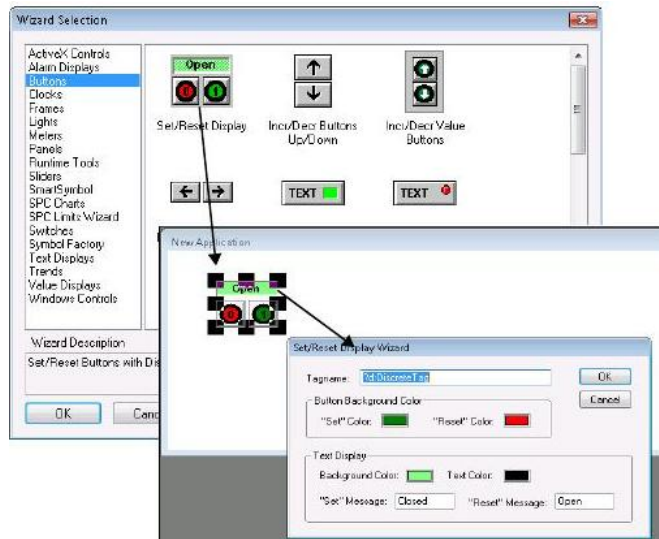
Working with Wizards

Using wizards, you do not spend time drawing the individual components for the object, or entering the value ranges for the object, or animating the object.

- You select wizards from the Wizards/ActiveX toolbar.

- You configure wizards by typing tags and QuickScripts in configuration dialog boxes.

- When you paste a selected wizard into a window and then double-click it, the **Configuration** dialog box appears.

For example, in the case of a slider wizard, the configuration includes items such as the tagname to effect, the minimum and maximum range labels for the slider, the fill color, and so on. After the required configuration information is provided, the wizard is ready to use.

You can also develop your own complex wizards to provide behind the scenes types of operations. These operations can include creating complete display windows, creating or converting a database, importing an AutoCAD drawing, and configuring other applications.

Before creating your own wizards, you should investigate Industrial graphics, which offer wizard functionality, but do not require programming.

Types of Wizards

Categories of wizards are shown in the **Wizard Selection** dialog box.
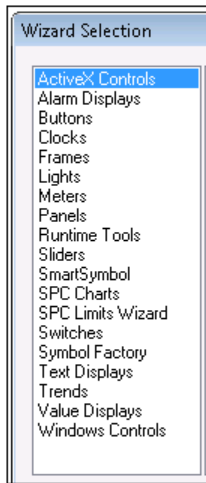
Trend Objects and Windows Controls Wizards are special wizards with unique parameters. For more information, see *Trend Objects* on page 92 and *Windows Controls Wizards* on page 93.

Adding Wizards to the Toolbar

You can add frequently used wizards to the Wizard/ActiveX Toolbar for quick availability.

To add a wizard to the Wizards/ActiveX Toolbar

1. Click the **Wizard** button in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears.
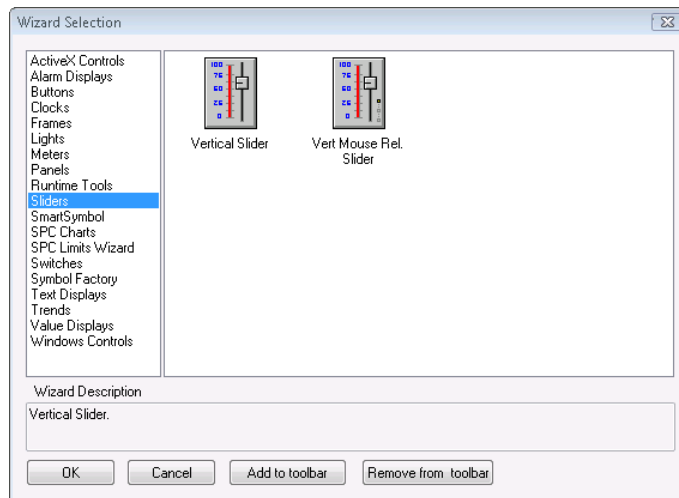
2. In the left pane, select a wizard category, such as **Sliders**.

3. In the right pane, select a wizard, and then click **Add to toolbar**. The wizard button appears in the toolbar.

To remove wizards from the toolbar

1. Click the **Wizard** button in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears.

2. Click **Remove from toolbar**. The **Remove Wizard from Toolbar** dialog box appears.

3. Select the wizard to remove and click **OK**.

Pasting Wizard Instances

You can place instances of a wizard in a window.

To place a wizard in a window

1. Click the Wizard button in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears. In the left pane, select a wizard category.

2. In the right pane, select a wizard.

3. Click **OK**. The dialog box closes, and the cursor becomes a corner symbol.

4. Click the location to put the wizard.

Configuring Wizards

After you place a wizard in your application, double-click it to configure its properties. A properties dialog box appears that is custom to the selected wizard.

For more information on each particular type of wizard, see the wizard Help, if available.

Performing Standard Operations on Wizards

You can cut, copy, paste, delete, and duplicate wizards, in the same way and with the same results as with other objects.

Installing and Removing Wizards

You must install a wizard to WindowMaker so that you can use it in your application. When you remove a wizard from WindowMaker, it is not deleted from your computer.

To install a wizard

1. On the **Special** menu, point to **Configure** and then click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.
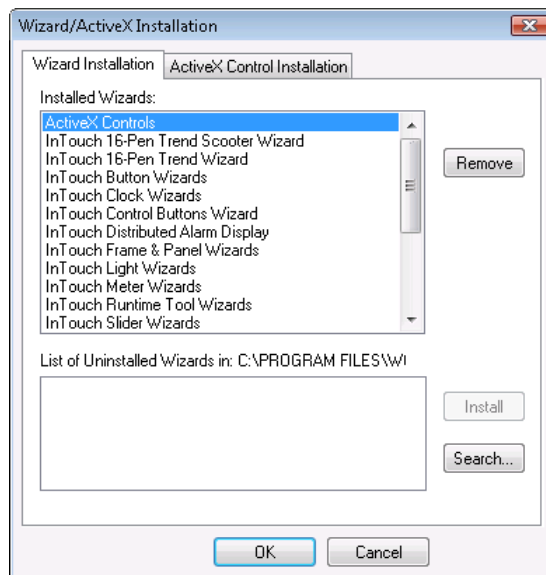
2. Select from the **List of Uninstalled Wizards** and then click **Install**.

To remove a wizard

1. On the **Special** menu, point to **Configure** and then click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.
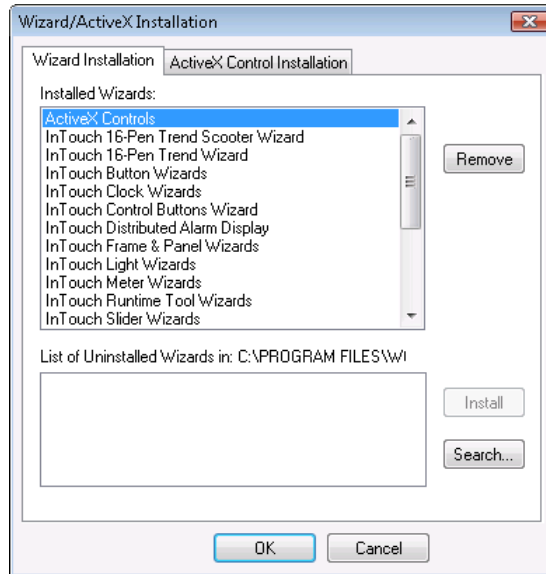
2. In the **Installed Wizards** list, select the wizard to remove and then click **Remove**. A message box appears asking you to confirm the deletion.

Tip: SHIFT+click or CTRL+click to make multiple selections.

3. Click **Yes** to remove the wizard. The wizard appears in the **List of Uninstalled Wizards** list.

To import wizards from another directory

1. On the **Special** menu, point to **Configure** and then click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.

2. Click **Search**. The **Browse for Folders** dialog box appears.

3. Browse to the directory containing the wizards to install and click **OK**. The **Wizard Installation** dialog box reappears with the imported wizards in the **List of Uninstalled Wizards**.

Trend Objects

Trend objects are wizards that chart the values of tags over time.

There are three main types of trend objects:

- Real-time trends chart up to four tags in real time.

- Historical trends chart up to eight tags over a past time period.

- 16 Pen trends chart real-time and historical data for up to sixteen tags.

There is no limit to the number of trend objects, real-time or historical, you can place in a window.

Configure trend objects with the following items:

- Time span

- Value range

- Grid resolution

- Location of time and value stamps

- Pens and colors

Before you can use a trend wizard, you must enable logging for each tag to track, and also enable logging within the InTouch application.

To enable logging for tags

1. From the **Tagname Dictionary**, select a tag and then select **Log Data**.

2. If you have not done so previously, enable logging in the InTouch application.

   a. On the **Special** menu, point to **Configure**, and then click **Historical Logging**. The **Historical Logging Properties** dialog box appears.

   b. Select the **Enable Historical Logging** check box or **Enable Storage to Historian** checkbox and click **OK**.

For more information on configuring and using trend objects, see Trending Tag Data in the InTouch HMI Data Management Guide.

Windows Controls Wizards

The Windows Controls Wizards are user interface objects such as drop-down lists, combo boxes, option selections or check boxes.

Use Windows Control Wizards to present predefined sets of choices to your users. For example, you might create a drop down list of processes, recipes, or operator IDs. You can also enable and disable specified controls. You can even load and modify the contents of drop-down lists.

Run-time properties for Windows Controls Wizards are accessed through QuickScript functions rather than animation link expressions.

Windows Controls Wizards have properties like InTouch tag dotfields. They can be read-write or read-only. Some properties are accessible at development and some at run time. They are identified as ControlName.x, where x is the property.

For example, if the .Visible property of a windows control is equal to 0, the control is not visible in the window. As with InTouch tags, .Value is the default property for Windows Control Wizards.

Note: A more robust and flexible set of .NET based Windows controls is available if you use Industrial graphics.

Creating and Configuring Windows Controls

When you create and configure Windows Control Wizards, keep in mind that:

- Windows Control Wizards cannot overlap one another.

- During run time, Windows Control Wizards will always be on top if they overlap with other objects in the window.

- Each Windows Control Wizard has a unique control name—which does not add to the tag count.

- The initial value of tags assigned to either a list box or combo box does not initialize the value of the list box or combo box. You must use the SetPropertyX QuickScript functions in a script to assign initial values that need to be different than the default values.

Tip: Paste Windows Controls Wizards into your windows just like other wizards. To achieve the best readability, select a gray background for your windows controls. If your background color cannot be gray, place a gray Panel Wizard behind the Windows Control Wizards.

For each Windows Controls Wizard, you must specify an alphanumeric control name, where the first character is a letter. Underscores are allowed, but other special characters are not. For example, "Checkbox_1" is allowed, but "Checkbox#1" is not.

You can configure Windows Control Wizards using InTouch QuickScripts.

Creating a Text Box

You use text boxes in your application to enable operators to enter text strings.

To create and configure a text box

1. Create and place a text box. Do the following:

   a. From the **Wizard Selection** dialog box, select **Windows Control Wizards**. The control wizard icons appear.

   b. Double-click the text box icon. Your application window reappears and the cursor changes to a left corner symbol.

   c. Click a location in the application window to place the wizard. The text box wizard appears with heavy black handles at the corners.
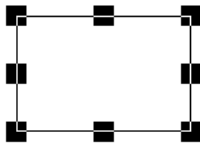
   d. Drag and resize the wizard to suit your application.

2. Double-click the wizard. The **Textbox Control** dialog box appears.
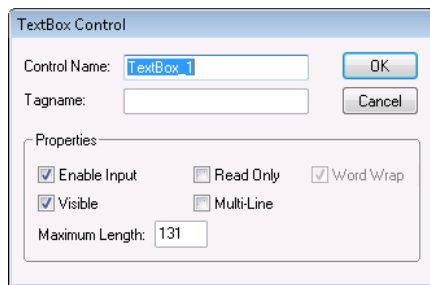
3. Configure the dialog box. Do the following:

   a. Type a control name, such as *TextBox_1*, in the **ControlName** box.

   b. Type a memory message tag name, such as *New_Value*, in the **Tagname** box.

   c. In the **Properties** area, select **Enable Input** and **Visible**.

4. Click **OK**. The **Textbox Control** dialog box closes.

Creating a List Box

You can create applications that enable your users to select items from a list of options.

When adding a list box to your application, you place the control on the screen, set the properties to configure the list, and then write any scripts you may need.

List boxes can be loaded with items from a file or from keyboard input during run time. For more information, see *Scripting Windows Controls* on page 98.

To create a list box

1. Create and place the list box control. Do the following:

   a. In the **Wizard Selection** dialog box, select **Windows Controls**.

    b.   Double-click the list box icon. Your application window reappears and the cursor changes to a left corner symbol.

    c.   Click a location in the application window to place the control. The list box control appears.
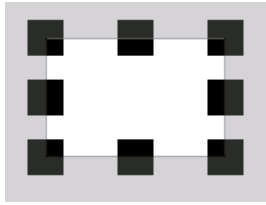
2.   Double-click the control. The **ListBox Control** dialog box appears.
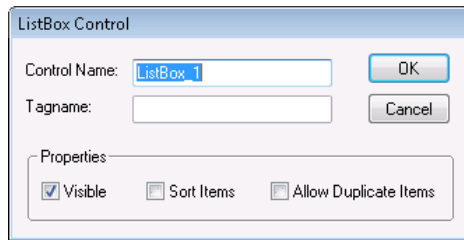
3.   Configure the control. Do the following:

    a.   Type a control name, such as *ListBox_1,* in the **ControlName** box.

    b.   Type a memory message tag name, such as *LB1_Value*, in the **Tagname** box.

    c.   In the **Properties** area, configure how the control appears and functions.

4.   Click **OK**.

Creating a Combo Box

You can create applications that enable your users to select items from lists of options with a combo box. A combo box is a Windows control combining a text box and a list box.

When adding a combo box to your application, place the control on the screen, set the properties to configure the combo box, and then write any scripts you may need.

Combo boxes can be loaded with items from a file or from keyboard input during run time. For more information, see *Scripting Windows Controls* on page 98.

To create a combo box

1.   Create and place the combo box control. Do the following:

    a.   In the **Wizard Selection** dialog box, select **Windows Controls**.

    b.   Double-click the combo box icon. Your application window reappears and the cursor changes to a left corner symbol.

    c.   Click a location in the application window to place the control. The combo box control appears.
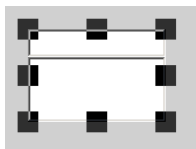
2. Double-click the control. The **ComboBox Control** dialog box appears.
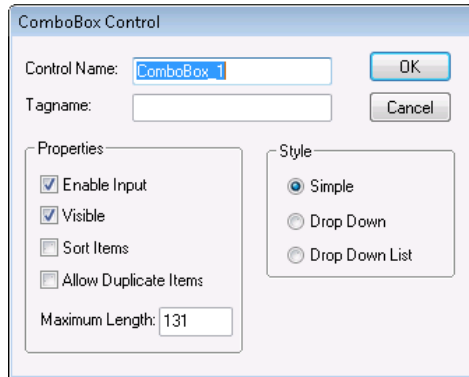
3. Configure the control. Do the following:

    a. Type a control name, such as *ComboBox_1,* in the **ControlName** box.

    b. Type a memory message tag name, such as *CB1_Value*, in the **Tagname** box.

    c. In the **Properties** area, configure how the control appears and functions.

    d. In the **Style** area, select the type of combo box.

4. Click **OK**.

Creating a Check Box

A check box control enables an operator to select an option.

To configure a check box control

1. Create the check box. Do the following:

    a. From the **Wizard Selection** dialog box, select **Windows Control Wizards**. The control wizard icons appear.

    b. Double-click the check box icon. Your application window reappears and the cursor changes to a Left corner symbol.

    c. Click a location in the application window to place the wizard. The check box wizard appears.

    d. Drag and resize the wizard.

2. Double-click the wizard. The **Checkbox Control** dialog box appears.
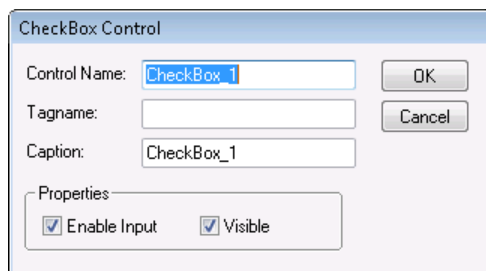
3. Configure the wizard. Do the following:

    a. Type the control name.

   b.   Type the discrete tag name, or double-click the empty tagname box to show the **Select Tagname** dialog, and select a tag.

   c.   Type the caption to appear on the face of the button.

4. Click **OK**.

Creating a Radio Button Group

You use radio buttons when the user must select from one of several choices. When a user selects an option, the previously selected option is un-selected.

You create the options for a user as a group of radio buttons. Each radio button has a caption, and provides a unique value to your script.

You can only assign integer tags to radio button controls.

To create a Radio Button group

1. Create the Radio Button control wizard. Do the following:

   a.   From the **Wizard Selection** dialog box, select **Windows Control Wizards**. The control wizard icons appear.

   b.   Double-click the Radio button icon. Your application window reappears and the cursor changes to a left corner symbol.

   c.   Click a location in the application window to place the wizard. The Radio Group control wizard appears with three radio buttons.
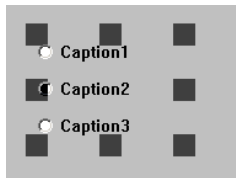
   d.   Drag and resize the wizard to suit your application.

2. Double-click the wizard. The **Radio Button Group Control** dialog box appears.
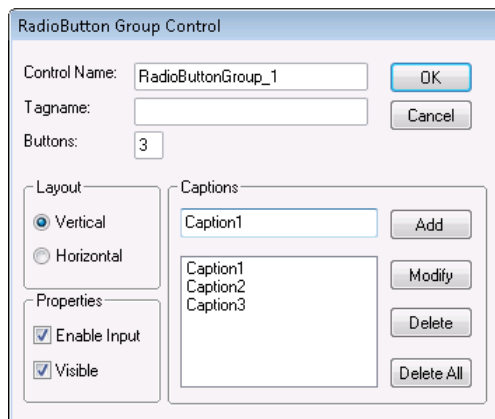
3. Configure the wizard. Do the following:

   a.   Type the control name.

   b.   Type an integer type tag name to link to this control.

   c.   Type the number of buttons to show.

   d.   Type the captions for each button.

e.   Set the layout and properties.

4.   Click **OK**.

Scripting Windows Controls

You can use QuickScript functions in scripts to:

- Get or set the value of a control.

- Enable, disable, or hide a control.

- Work with items in combo boxes, list boxes, text boxes, and check boxes.

The run-time properties can be either read-write or read-only, depending on the property.

Use the GetPropertyX() and SetPropertyX() functions to control or retrieve these properties.

Getting or Setting the Value of a Control

The .Value property is the default property for all InTouch Windows Controls Wizards.

Changes made to this property are synchronized in the InTouch tagname and the Windows Controls Wizards.

.Value Dotfield

Default property for all InTouch windows control wizards. Changes made to this property are synchronized in the InTouch tagname and the windows control wizards.

Category

windows control

Usage

The M, I and D are for the Memory, Integer and Discrete versions of the GetProperty and SetProperty functions.

```
[ErrorNumber=]GetPropertyM("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyM("ControlName[.Value]", Value);
[ErrorNumber=]GetPropertyI("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyI("ControlName[.Value]", Value);
[ErrorNumber=]GetPropertyD("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyD("ControlName[.Value]", Value);
```

Parameters

ControlName
Name of the windows control, e.g., ChkBox_4.

Tagname
The tagname to which the value of the property is written.

[.Value]
This property is optional. If not specified, the function always assumes the .Value property is being used.

Value
The actual value to be written or a valid InTouch tagname (of the same type as the property to be written to) that holds the property value to be written when the function is processed.

Remarks

The initial value of tagnames assigned to either a list box or combo box cannot be used to initialize the value of the list box or combo box.

This dotfield is read/write during development and run time. If the .Value dotfield is accessed by associating a tagname to either a list box or a combo box, it is read-only. If the .Value dotfield is assigned to a check box, radio button, or text box, it is read/write. The value you specify at development serves as the default for run time.

Data Type

Message (read/write) for text boxes, list boxes and combo boxes.

Integer (read/write) for radio buttons.

Discrete (read/write) for check boxes.

Applies To

Text boxes, list boxes, combo boxes, check boxes and radio buttons.

Example(s)

The following statement sets the .Value dotfield of the radio button object "RadioButton_1" to a value of 4:

```
SetPropertyI( "RadioButton_1.Value", 4 );
```

See Also

GetPropertyM(), SetPropertyM(), GetPropertyI(), SetPropertyI(), GetPropertyD(), SetPropertyD()

Enabling or Disabling a Control for User Input

Use the .Enabled property to determine whether the control object can respond to operator-generated events.

.Enabled Dotfield

Determines whether the control object can respond to user-generated events.

Category

windows control

Usage

```
[ErrorNumber=] GetPropertyD("ControlName.Enabled",
Tagname);
[ErrorNumber=] SetPropertyD("ControlName.Enabled",
Discrete);
```

Parameters

ControlName
Name of the windows control. For example, ChkBox_4.

Tagname
A discrete tagname that holds the property requested.

Discrete
A discrete value or a discrete tagname that holds the value to be written when the function is processed. For a discrete value:

0 = Control is disabled.

1 = Control is enabled.

Remarks

This property is read/write during both development and run time.

Data Type

Discrete (read/write)

Applies To

Text boxes, list boxes, combo boxes, check boxes and radio buttons.

Example(s)

The following statement disables the list box object named "ListBox_1".
```
SetPropertyD("ListBox_1.Enabled", 0);
```

See Also

GetPropertyD(), SetPropertyD()

Hiding Windows Controls Dynamically

Use the .Visible property to determine whether the windows control is visible in the window.

.Visible Dotfield

Determines whether the windows control is visible in the window.

Category

windows control

Usage
```
[ErrorNumber=]GetPropertyD("ControlName.Visible", Tagname);
[ErrorNumber=]SetPropertyD("ControlName.Visible", Number);
```

Parameters

ControlName
Name of the windows control. For example, ListBox_1.

Tagname
A tagname (of the same type to be returned) that holds the property value when the function is processed.

Number
A discrete value or a discrete tagname that holds the value to be written when the function is processed. For a discrete value:

0 = Control is invisible.

1 = Control is visible.

Remarks

This property is read/write in both development and run time.

Data Type

Discrete (read/write)

Valid Values

Applies To

Text boxes, list boxes, combo boxes, check boxes and radio buttons.

Example(s)

The following statement hides the text box named "TextBox_1".
```
SetPropertyD("TextBox_1.Visible",0);
```

See Also

GetPropertyD(), SetPropertyD()

Adding and Deleting Items in Combo Boxes

Use the following script functions to add and delete items from combo boxes and lists

Script function	Effect
wcAddItem()	Adds an item to the end of the list of a list box or combo box. If sorting is enabled, the list is sorted after the item is added.
wcInsertItem()	Adds an item at a specified position in the list of a list box or combo box.
wcDeleteItem()	Deletes an item from a specified position in the list of a list box or combo box.
wcDeleteSelection()	Deletes the currently selected item from the list or combo box.
wcClear()	Removes all items from the list or combo box.

wcAddItem() Function

Adds an item to the end of the list of a list box or combo box. If sorting is enabled, the list is sorted after the item is added.

Category

windows control

Syntax

```
[ErrorNumber=]wcAddItem("ControlName", "MessageTag");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

MessageTag
The message string to be shown. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement adds the contents of the message string to the list box when the window (using On Show Window QuickScript) containing the list box is opened:

```
wcAddItem("ListBox_1", "Chocolate");
wcAddItem("ListBox_1", "Vanilla");
wcAddItem("ListBox_1", "Strawberry");
```

See Also

wcInsertItem()

wcInsertItem() Function

Inserts the specified string into the list of a list box or combo box at the specified position. Unlike the wcAddItem() function, the wcInsertItem() function does not sort a list, even if it is created as a sorted list box or combo box.

Category

windows control

Syntax

```
[ErrorNumber=]wcInsertItem("ControlName", ItemPosition, "Message");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

ItemPosition
A number corresponding to the position of the item to be added. If this parameter is -1, the string is added to the end of the list. Any number or Integer tagname.

Message
Contains the string to insert at the position indicated by ItemPosition. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement inserts a new item called "Blueberry" into a list box at fourth position from the top when an action script runs.

```
wcInsertItem("ListBox_1", 4, "Blueberry");
```

See Also

wcAddItem()

wcDeleteItem() Function

Deletes the item at a specified position from the list of either a list box or combo box.

Category

windows control

Syntax

```
[ErrorNumber=]wcDeleteItem("ControlName", ItemPosition);
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

ItemPosition
A number corresponding to the position of the item. Any number or Integer tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement deletes the third item in a list when an action script runs:
```
wcDeleteItem("ListBox_1", 3);
```

wcDeleteSelection() Function

Deletes the currently selected item from the list.

Category

windows control

Syntax
```
[ErrorNumber =]wcDeleteSelection("ControlName");
```

Parameter

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement deletes the currently selected item in a list box list when an action script runs:
```
wcDeleteSelection("ListBox_1");
```

wcClear() Function

Removes all items from the list box or combo box.

Category

windows control

Syntax
```
[ErrorNumber=]wcClear("ControlName");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement clears all items in a list box when an action script runs:
```
wcClear("ListBox_1");
```

Loading and Saving List Items From or To a File

Use the following script functions to load and or save items in a combo box or list from or to a file.

Script function	Effect
wcLoadList()	Loads the contents of a list box or combo box with new items from a file.
wcSaveList()	Saves the contents of a list box or combo box to a file.

wcLoadList() Function

Loads a list box or combo box with new items from a file.

Category

windows control

Syntax
```
[ErrorNumber=]wcLoadList("ControlName", "Filename");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

Filename
Contains the name of a file. If you do not supply a complete path name as part of the message parameter, the function checks the application directory for the message file. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

If you use external files to fill the list and combo boxes, they must follow specific formatting and contain specific information. Format:

ControlType, ListCount

ListItem, ItemIndex

ListItem, ItemIndex

:   :

:   :

ListItem, ItemIndex

The ControlType is either COMBOBOX or LISTBOX.

For example, you want to load a list file to a combo box and it contains three items to select from and those items have no item data assigned. The format of the file appears as:

COMBOBOX, 3

Chocolate, 0

Vanilla, 0

Strawberry, 0

The COMBOBOX is the control type. The list count is 3 for Chocolate, Vanilla, and Strawberry. Chocolate is then listed as the first item or position 1. Vanilla as position 2, and Strawberry as position 3. Each of these items has a data value of 0.

For more information on item data, see *wcSetItemData() Function* on page 110.

Applies To

List boxes and combo boxes.

Example(s)

The following statement loads a properly formatted list (located in c:\wclist.txt.) into a combo box:
```
wcLoadList("Combobox_1", "c:\wclist.txt");
```

See Also

wcAddItem(), wcSaveList()

wcSaveList() Function

Saves the items of a list box or combo box to a file.

Category

windows control

Syntax
```
[ErrorNumber=]wcSaveList("ControlName","Filename");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

Filename
Contains the name of a file. If the file does not exist, it is created. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement saves the current items in a list box in a file (c:\newlist.txt) when an action script runs:
```
wcSaveList("ListBox_1", "c:\newlist.txt");
```

See Also

wcLoadList(), wcSetItemData()

Finding Items In a Combo Box or List

Use the wcFindItem() function to search for a specified item in a list box or combo box. If the item is found, this function returns the corresponding position to an integer tagname as the fourth parameter.

wcFindItem() Function

Determines the corresponding position of the first item in the list box or combo box that matches the supplied message string.

Category

windows control

Syntax

```
[ErrorNumber=]wcFindItem ("ControlName", "MessageTag", CaseSens, Tagname);
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

MessageTag
The message string to be compared. Actual string or message tagname.

CaseSens
Determines the type of the string comparison. It can either be a discrete value or tagname. The following values are valid:

0 = case-insensitive.

1 = case-sensitive.

Tagname
Integer tag into which the position of the matching item is returned. If no matching item is found, -1 is returned.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes, combo boxes

Example(s)

Assuming that ListBox_1 is a list box that contains "ItemA", "ItemB", and "ItemC", the function returns the following values into Result:
```
wcFindItem("ListBox_1", "ItemB", 0, Result);
```

returns 2
```
wcFindItem("ListBox_1", "Itemb", 1, Result);
```

returns -1
```
wcFindItem("ListBox_1", "itemc", 0, Result);
```

returns 3
```
wcFindItem("ListBox_1", "XYZ", 0, Result);
```

returns -1

Working with Item Indexes in a Combo Box or List

Use the following dotfields to work with the item index of a list box or combo box.

Dot Field	Effect
.TopIndex	The integer index of the topmost item in the list box.
.NewIndex	The integer index (tagname) of the last item added to the list box or combo box through the wcAddItem() or wcInsertItem() functions.
.ListIndex	The index (tagname or number) of the currently selected item in the list.

.TopIndex Dotfield

Sets or reads the integer index of the top-most item in a list box.

Category

windows control

Usage

```
[ErrorNumber=]GetPropertyI("ControlName.TopIndex", Tagname);
[ErrorNumber=]SetPropertyI("ControlName.TopIndex", Number);
```

Parameters

ControlName
Name of the windows control. For example, ListBox_1.

Tagname
An Integer tagname that holds the property value when the function is processed.

Number
The index number that defines the top-most item in the list box. Can be a literal integer value or a integer tagname or expression that provides an integer value.

Remarks

This property is available only in run time.

Data Type

Integer (read/write)

Applies To

List boxes.

Example(s)

The following statement sets the TopIndex of the list box object "ListBox_1" to a value of 14:

```
SetPropertyI("ListBox_1.TopIndex",14);
```

See Also

GetPropertyI(), SetPropertyI(), .ListIndex, .NewIndex

.NewIndex Dotfield

Returns the integer index (Tagname) of the last item added to the list box or combo box via the wcAddItem() or wcInsertItem().

Category

windows control

Usage

```
[ErrorNumber=]GetPropertyI("ControlName.NewIndex", Tagname);
```

Parameters

ControlName
Name of the windows control. For example, ListBox_4.

Tagname
A tagname containing the integer index of the last item added to the list or combo box. For empty lists, a value of -1 is returned.

Remarks

This property is only available in run time.

Data Type

Integer (read-only)

Applies To

List boxes and combo boxes.

Example

The following statement retrieves the index of the most recently added item in the list box named "ListBox_1" and writes that value to the memory integer tagname NewItemIndex.

```
GetPropertyI("ListBox_1.NewIndex", NewItemIndex);
```

See Also

GetPropertyI(), wcAddItem(), wcInsertItem(), .ListIndex, .TopIndex

.ListIndex Dotfield

Sets or reads the index (Tagname or Number) of the currently selected item in the list.

When using a list box, an index of -1 indicates that no item is currently selected.

When using a combo box, an index of -1 indicates that the user has typed new text into the text entry field of the control.

Syntax

```
[ErrorNumber=]GetPropertyI("ControlName.ListIndex", Tagname);
[ErrorNumber=]SetPropertyI("ControlName.ListIndex", Number);
```

Parameter

ControlName
Name of the windows control. For example, ListBox_4.

Tagname
The tagname to which the index of the currently selected item is written.

Number
The index number that defines a specific item in the list.

Remarks

The index number defines a specific item in a list. Use the .ListIndex dotfield to set or determine the index of the currently selected item in a list or combo box.

This property is available only in run time.

Data type

Integer (read or write)

Applies to

List boxes and combo boxes.

Example

This statement retrieves the index of the currently selected item in the list box named "ListBox_1" and writes that value to the memory integer tagname MyListBoxIndex.

```
GetPropertyI( "ListBox_1.ListIndex", MyListBoxIndex );
```

See Also

GetPropertyI(), SetPropertyI(), .NewIndex, .TopIndex

Counting List Box or Combo Box Items

The .ListCount dotfield contains the number of items in a list box or combo box.

.ListCount Dotfield

Reads the number of items in the list box or combo box.

Category

Windows control

Syntax

```
[ErrorNumber=]GetPropertyI("ControlName.ListCount", Tagname);
```

Parameter

ControlName
Name of the windows control.

Tagname
A valid tagname that contains the integer count of the items in the list.

Remarks

This property is available only in run time.

Data Type

Integer (read-only)

Applies To

List boxes and combo boxes.

Example(s)

The following statement retrieves the number of items in the list box named *ListBox_1* and writes that value to the memory integer tag *MyListBoxCount*.

```
GetPropertyI("ListBox_1.ListCount", MyListBoxCount);
```

See Also

GetPropertyI(), .ListIndex

Getting or Setting the Value of a List Item

Use the wcGetItemData() function to find the integer value associated with the list item identified by the item index.

Use the wcSetItemData() function to assign an integer value to the item in the list specified by item index. This assigns a number to a string.

wcGetItemData() Function

Reads the integer value associated with the list item identified by the ItemIndex parameter.

Category

windows control

Syntax

```
[ErrorNumber=]wcGetItemData("ControlName", ItemIndex, Tagname);
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

ItemIndex
A number corresponding to the position of the item. Any number or integer tagname.

Tagname
Actual name of a real or integer tagname. The wcGetItemData() function places the numeric value corresponding to the item into this tagname upon return from the function.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement retrieves the numeric value associated with the fifth item in a list box and returns it to the ItemValue Integer tagname when an action script runs:

```
wcGetItemData("ListBox_1", 5, ItemValue);
```

If the fifth item in the list is assigned the integer value 4500, the ItemValue tagname contains 4500.

See Also

wcSetItemData()

wcSetItemData() Function

Assigns an integer value of the item (the Number parameter) to the item in the list specified by the ItemIndex parameter. This function enables the assignment of a number to a string.

Category

windows control

Syntax

```
[ErrorNumber=]wcSetItemData("ControlName", ItemIndex, Number);
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

ItemIndex
An integer value specifying the list item you want to edit. Any number or Integer tagname.

Number
An integer value representing the item data. Any number or Integer tagname.

Remarks

You can create complete lists containing the items using a program like Notepad and then load them using one function call. Format the list as required by the wcSaveList() function.

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Use the wcGetItemData() function to return the value (item data) associated with the list item. The tagname parameter contains the returned numeric value. This parameter could be an I/O Integer tagname that writes directly to the real world device.

Example(s)

A recipe has three ingredients; flour, sugar and salt. The quantity of flour is 4500 grams, sugar is 1500 and salt is 325 grams. The values are assigned to each of the list box items by using a data change script triggered by what recipe (tagname, RecipeName)   is selected:

```
wcSetItemData("ListBox_1", 1, 4500); {set 1st item in the list (flour)=4500}
wcSetItemData("ListBox_1", 2, 1500); {set 2nd item in the list (sugar)=1500}
wcSetItemData("ListBox_1", 3, 325);  {set 3rd item in the list (salt)=325}
```

See Also

wcLoadList(), wcSaveList(), wcGetItemData()

Getting the Name of a List Item

Use the wcGetItem() function to return the item string associated with the corresponding item index in the list box or combo box.

wcGetItem() Function

Returns a string containing the contents of the item corresponding to the ItemIndex in the list box or combo box.

Category

windows control

Syntax

```
[ErrorNumber=]wcGetItem("ControlName", ItemIndex, Tagname);
```

Parameters

> *ControlName*
> The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

> *ItemIndex*
> A number corresponding to the position of the item. Any number or Integer tagname.

> *Tagname*
> Message tagname. The wcGetItem function will place the data corresponding to the item index into this tagname upon return from the function.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes and combo boxes.

Example(s)

The following statement returns the string value of the tenth item in a combo box to the ListSelection message tag when an action script runs:

```
wcGetItem("Combobox_1", 10, ListSelection);
```

If item ten in the list is "Vanilla," then the ListSelection tag contains the string "Vanilla".

Loading the Contents of a Text Box

Use the wcLoadText() function to load the contents of a text box from a file. Use the wcSaveText() function to save the contents of a text box to a text file.

Note: If the tag name has been defined with a maximum length, only that number of characters can be assigned from the text box contents to the tag. If no tag is assigned to the text box, its contents can be up to 65,535 characters.

wcLoadText() Function

Replaces the contents of the text box with the contents of the file.

Category

windows control

Syntax

```
[ErrorNumber=]wcLoadText("ControlName", "Filename");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

Filename
Contains the name of a file. If a complete path name is not supplied as part of the message parameter, the function will check the application directory for the file. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

Text boxes.

Example(s)

The following statement loads a text file (c:\InTouch.32\readme.txt.) into a text box when the window (On Show Window script) containing the text box opens:

```
wcLoadText("Textbox_1", "c:\InTouch.32\readme.txt");
```

wcSaveText() Function

Saves the text contained in the text box to the specified file. If the file doesn't exist, it is created. If it does exist, it must be read/write.

Category

windows control

Syntax

```
[ErrorNumber=]wcSaveText("ControlName", "Filename");
```

Parameters

ControlName
The name of the windows control object. For example, ListBox_1. Actual string or message tagname.

Filename
Contains the name of the destination file. If you do not supply a complete path name, the file is saved in the application directory. If the file exists, it is overwritten. If the file does not exist, it is created. The resulting file can subsequently be loaded into a text box object using the wcLoadText() function. Actual string or message tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

Text boxes.

Example(s)

The following statement saves the current information entered in a text box to a file in c:\InTouch.32\newtext.txt when an action script runs:

```
wcSaveText("Textbox_1", "c:\InTouch.32\newtext.txt");
```

See Also

wcLoadText()

Checking If a Text Box is Read-Only

Use the .ReadOnly dotfield to determine whether the content of the text box is read-only or read/write.

.ReadOnly Dotfield

Determines whether the content of the text box is read-only or read/write.

Category

windows control

Usage

```
[ErrorNumber=]GetPropertyD("ControlName.ReadOnly", Tagname);
```

Parameters

ControlName
Name of the windows control. For example, Textbox_1.

Tagname
A discrete tagname that holds the property value when the function is processed.

0 = Contents of the text box is read/write

1 = Contents of the text box is read-only

Remarks

This property is available in both development and run time.

Data Type

Discrete (read-only)

Applies To

Text boxes.

Example(s)

The following statement retrieves the read-only status of the Text box named "TextBox_1":
```
GetPropertyD("TextBox_1.ReadOnly",A_Tagname);
```

See Also

GetPropertyD(), SetPropertyD()

Getting or Setting the Label of a Check Box

The .Caption dotfield defines the message text of a check box.

.Caption Dotfield

Determines the message to be displayed with the check box.

Category

windows control

Usage
```
[ErrorNumber=]GetPropertyM ("ControlName.Caption", Tagname);
[ErrorNumber=]SetPropertyM ("ControlName.Caption", "Message");
```

Parameters

ControlName
Name of the windows control. For example, ChkBox_4.

Tagname
A message tagname that holds the property requested.

Message
A message string surrounded in quotes.

Remarks

This property is read/write during both development and run time.

Data Type

Message (read/write)

Applies To

Check boxes.

Example

This statement sets the caption of the check box object "CheckBox_1" to "Blue Paint Option."
```
SetPropertyM("CheckBox_1.Caption","Blue Paint Option");
```

See Also

GetPropertyM(), SetPropertyM()

Understanding Windows Controls Error Messages

Given an error number, wcErrorMessage(), returns a string message describing the error. It applies to list boxes, text boxes, combo boxes, radio buttons and check boxes.

The Window Controls functions return values based on the result of processing QuickScript functions. The return value is used for error diagnostics. You can assign these values to integer tag names. For example:

```
ErrorNumber = wcGetItem("ControlName", Number, Tagname);
```

In this script, `ErrorNumber` is an integer tag that contains the returned error value. The returned value of the function can be passed to the wcErrorMessage(). The wcErrorMessage() will return a string description of the error. For example:

```
ErrorMsg = wcErrorMessage(ErrorNumber);
```

In this script, `ErrorMsg` is a message type tag that contains the text of the returned error. The following table identifies numeric error values and their definitions.

Error Message	Definition
0	Success
-1	General failure
-2	Insufficient memory available
-3	Property is read-only
-4	Specified item already present
-5	Object name unknown
-6	Property name unknown
-x	Unknown error.

wcErrorMessage() Function

Returns a message string describing the error.

Category

windows control

Syntax

```
ErrorMessage=wcErrorMessage(ErrorNumber);
```

Parameters

ErrorMessage
Message tagname.

ErrorNumber
Number returned by all windows control functions. Any number or Integer tagname.

Remarks

For a list of returned error numbers, see *Understanding Windows Controls Error Messages* on page 114.

Applies To

List boxes, text boxes, combo boxes, check boxes and radio buttons.

Example(s)

If an error occurs while a list is being loaded, show the text description of the error into the ErrorDescription message tagname. In this example, a Value Display animation link is assigned to the ErrorDescription tagname to show the error message.

In the "On Show" window QuickScript:

```
ErrorNumber=wcLoadList("ListBox_1","c:\recipe.txt");
ErrorDescription=wcErrorMessage(errornumber);
```

You can use this function with all windows control functions to show error messages:

```
ErrorNumber=wcAddItem("ListBox_1","AM_4A4356");
ErrorMsg=wcErrorMessage(ErrorNumber);
```

ActiveX Controls

About ActiveX Controls

ActiveX controls are stand-alone software components that bring additional functionality to InTouch applications. ActiveX controls have properties, methods, and events that can be modified while an application is running to change the behavior of the control.

You can use several types of ActiveX controls in your InTouch application.

- The InTouch HMI includes ActiveX controls for alarming.
- Other products, such as ActiveFactory, provide controls for manipulating and analyzing data.
- You can use third-party ActiveX controls.
- You can develop your own ActiveX controls in Visual Basic or C.

You can use any number of ActiveX controls in your InTouch application. You can:

- Select and paste an ActiveX control into any application window.
- Adjust the size of the control, if sizing is supported by the control.
- Duplicate, cut, copy, paste and delete ActiveX controls.
- Align ActiveX controls: left, right, top, bottom, and center point.
- Add ActiveX controls to the Wizards/ActiveX Toolbar.
- Combine ActiveX controls with other objects when creating a cell.
- Use the properties, methods, and events supported by the particular ActiveX control properties.

InTouch applications do not support the following types of ActiveX controls:

- Windowless controls
- Simple frame site or area box
- Containers
- Data controls
- Dispatch objects

InTouch applications only support these data types: discrete (Boolean), integer (32-bit numbers), real (floating point IEEE notation with 32 bits), and message (strings up to 131 characters). Unsupported data types include variant, pointers, arrays, structures, and parameterized properties.

ActiveX controls cannot overlap other InTouch objects, such as window controls or graphic objects. Too many ActiveX controls on one window can reduce system performance.

Using ActiveX Controls

You select an ActiveX control and paste it into a window and then double-click it. Its configuration dialog box appears.

When you configure an ActiveX control, you can give it a unique control name. The control name can then be referenced in scripts.

To use an ActiveX control in an InTouch application

1. Install the ActiveX control. See *Installing and Removing ActiveX Controls* on page 120.

2. Select and paste the ActiveX control into an InTouch window.

3. Configure the ActiveX control properties and bind them to tags.

4. Associate ActiveX events to ActiveX Event scripts.

5. Call ActiveX methods and set ActiveX control properties in ActiveX Event scripts, or other InTouch QuickScripts.

To place an ActiveX control in a window

1. Click the Wizard Dialog tool in the Wizards/ActiveX Toolbar. The Wizard Selection dialog box appears.
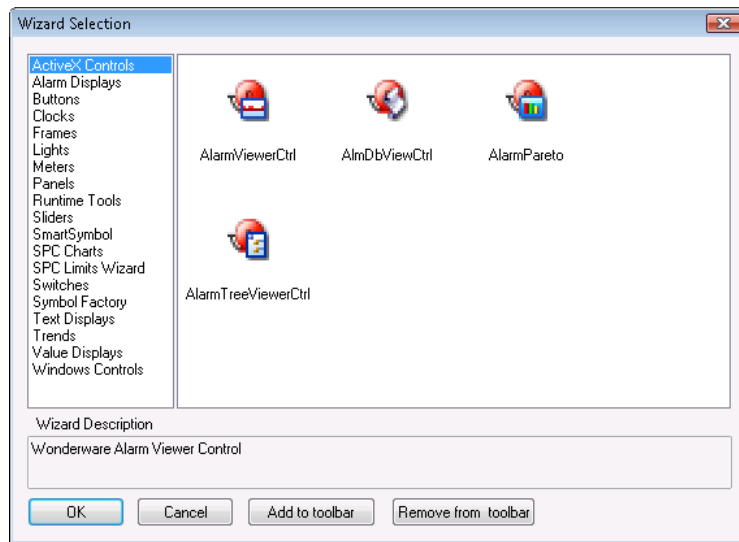
2. In the list of wizards, click the **ActiveX Controls** category. All available ActiveX controls appear in the display area.

3. Double-click the ActiveX control to use. The dialog box closes and the cursor changes to the corner symbol.

4. Click the location to paste the ActiveX control.

To add ActiveX controls to the toolbar

1. Click the **Wizard Dialog** tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears.

2. Select the ActiveX control to add.

3. Click **Add to toolbar**.

To remove ActiveX controls from the toolbar

1. Click the **Wizard Dialog** tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears.

2. Click **Remove from toolbar**. The **Remove Wizard from Toolbar** dialog box appears.

3. Select the ActiveX control to remove.

4. Click **OK**.

Configuring ActiveX Controls

The properties that you can configure for a particular ActiveX control are determined by the ActiveX control itself. All properties have a default value.

You must paste an ActiveX control into an InTouch window before you can configure its properties.

- A default control name, such as Calendar1, is generated when you paste the ActiveX control in a window. You reference the control name in scripts. An ActiveX control must be running in an open application window for any script to run against it.

- You can assign the ActiveX control properties to InTouch tags. You must assign each property type to a corresponding InTouch tag type.

ActiveX controls have three standard tabs: **Control Name**, **Properties**, and **Events**.
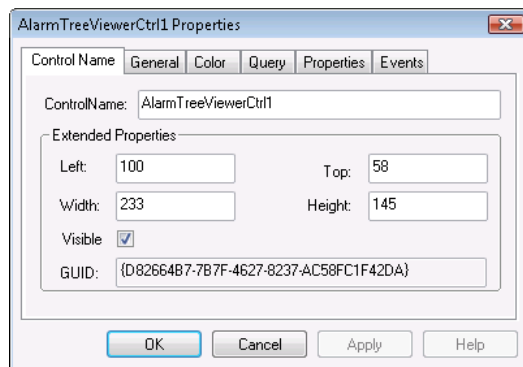
Use the **Events** tab to assign scripts to available control events, such as when the user double-clicks the mouse.

Any other tabs and their configurations are unique to the control and depend upon its properties. For example, some controls may require you to configure colors and fonts, while others may not have these properties.

Naming ActiveX Controls

A new instance of a control is created and given a unique name when you:

- Select **Duplicate** on the **Edit** menu.

- Select **Cut** or **Copy** and then **Paste** on the **Edit** menu.

- Select **Save Window As** on the **File** menu.

- Click **Undo** and then **Redo.**

- Import a window that contains a control.

ActiveX controls must have unique names. If you try to use a name that already exists for a control, an error message appears.

To name an ActiveX control

1. Paste the ActiveX control into your development window.

2. Double-click the control. The control **Properties** dialog box appears.

3. Click the **Control Name** tab and then type a name for the ActiveX control in the **Control Name** box.

Standard Operations on ActiveX Controls

You can perform standard operations on ActiveX controls just like any other InTouc h object. For more information, see *Special Manipulations for All Objects* on page 47.
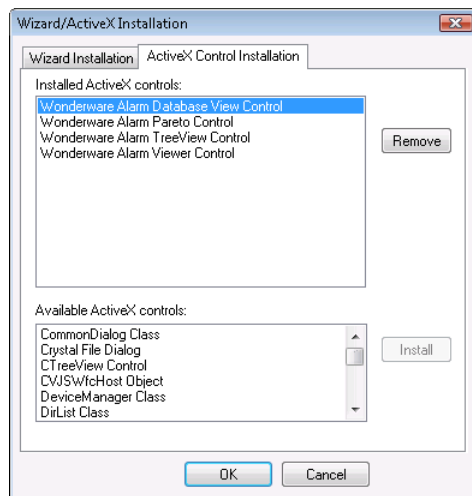
Installing and Removing ActiveX Controls

Even though you have already installed an ActiveX control on your computer, you must also make it known to the InTouch HMI by installing it to WindowMaker.

When you remove a control from WindowMaker, it is not deleted from your computer. It is simply removed from memory and does not function.

To install an ActiveX control

1. On the **Special** menu, point to **Configure** and then click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.

2. Click the **ActiveX Control Installation** tab. The **ActiveX Control Installation** property sheet appears.

3. In the **Installed ActiveX controls** list, select the control to install in the **Available ActiveX controls** list and then click **Install**.

Tip: To select multiple controls, use the SHIFT key or CTRL key.

---

4. Click **OK**.

To remove an ActiveX control

1. On the **Special** menu, point to **Configure** and then click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.

2. Click the **ActiveX Control Installation** tab. The **ActiveX Control Installation** property sheet appears.

3. In the **Installed ActiveX controls** list, select the control to remove from your application and then click **Remove**. A message appears asking you to confirm the deletion.

Tip: To select multiple controls, use the SHIFT key or CTRL key.

---

4. Click **Yes** to remove the control. The control is moved to the **Available ActiveX controls** list.

5. Click **OK**.

CHAPTER 7

Working with the Industrial Graphic Editor

Managing Industrial Graphics

Industrial Graphics are graphics you can create to visualize data in an HMI/SCADA system.

You use the Industrial Graphic Editor to create Industrial Graphics from basic elements, such as rectangles, lines, and text elements. You can also use the Industrial Graphic Editor to embed and configure an Industrial Graphic from the Graphic Toolbox library of graphics.

After you create an Industrial Graphic, you can embed it into another graphic or an HMI system window and use it at run time.

You can embed an industrial graphic in a template or instance of an object providing several ways to visualize object-specific information quickly and easily. Embedding a graphic in a template means that you can update one graphic and cascade the changes throughout your application.

Depending on your development requirements, you can select where and how to store industrial graphics.

- Create and store graphics as a standard set that you can re-use, such as a generic valve graphic.

- Store graphics as templates if you want to use the graphics in multiple instances at run time.

- Store graphics for use in a specific application.

Industrial Graphic Editor Workflows for Use with HMI/SCADA Applications

The Industrial Graphic Editor is the standard graphic editor used across various HMI/SCADA applications. It can be used to create graphics that can be embedded into HMI/SCADA applications to represent assets on the plant floor. The overall workflow is described in the following sections.
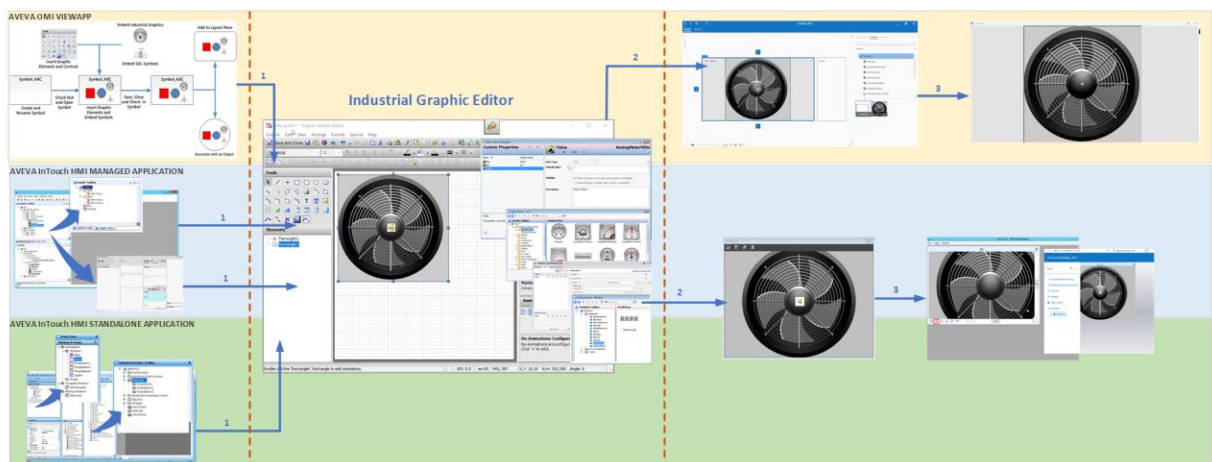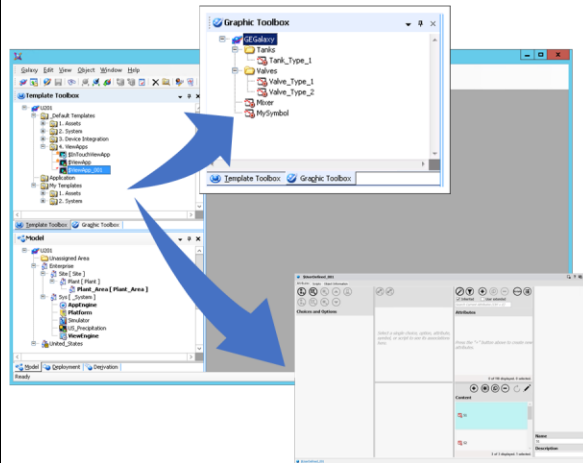
InTouch Managed Applications

The workflow for InTouch HMI managed applications starts with the IDE.
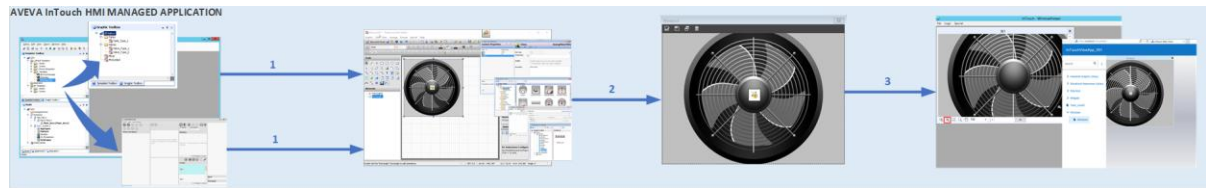
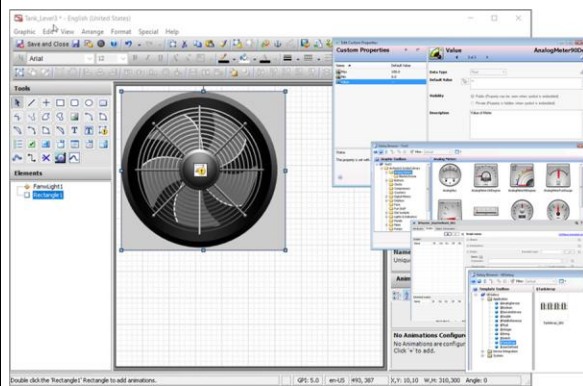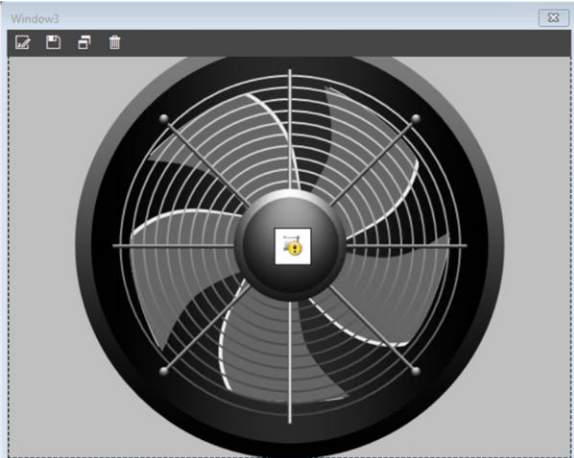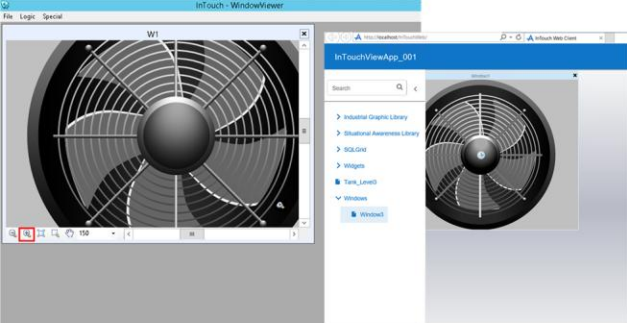| Step 1	**In the IDE:**
	From the Graphic Toolbox, create a new symbol, or select an existing graphic. • Open the graphic to launch the Industrial Graphic Editor. - or - Open a $UserDefined object in the Template Toolbox. 1. From the Content pane, click the **+** button to add a graphic to the object. 2. Click the **Edit Content** button to launch the Industrial Graphic Editor.
Step 2	Edit the graphic in the Industrial Graphic Editor. Modify graphical elements as needed. • Insert graphic elements and embed symbols. • Add/modify scripts, references, custom properties, and animations to enable the graphic to represent the asset, and to respond to commands and data changes at run time.

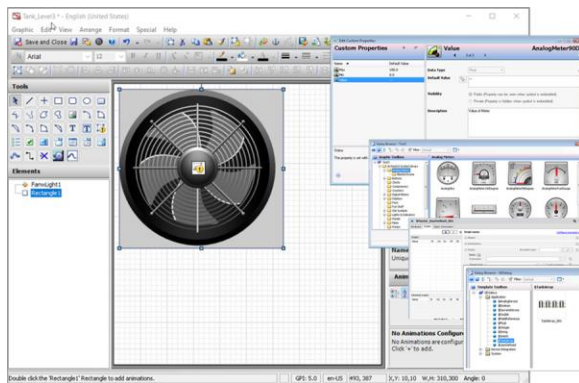|	
Step 3	Embed the graphic in a Window.
Step 4	To view the graphic in runtime, launch WindowViewer or the Web Client.

InTouch Standalone Applications

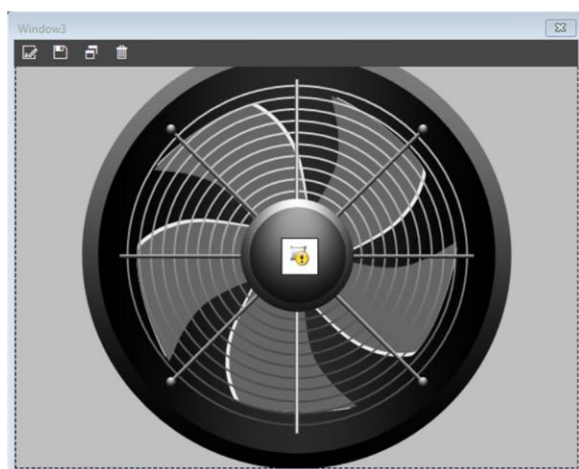The workflow for InTouch HMI standalone applications starts with WindowMaker.
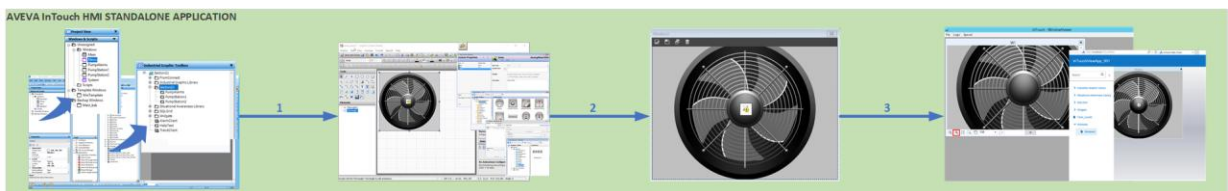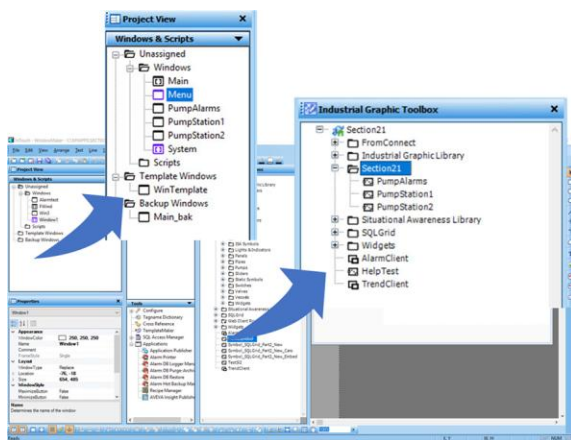
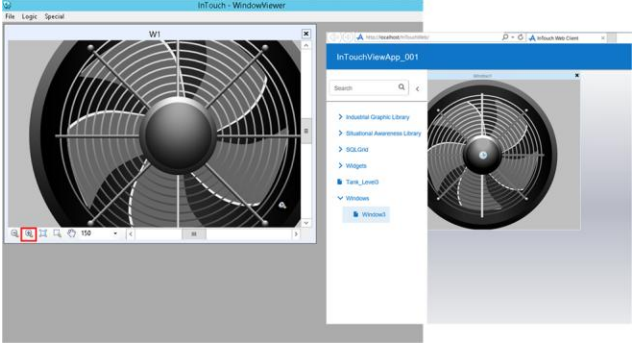| **Step 1**	In InTouch HMI WindowMaker create symbols using the Industrial Graphic Editor. Edit the graphic to launch the Industrial Graphic Editor.
Step 2	In the Industrial Graphic Editor use scripting, custom properties, tag and object references, styles, tools, animations and formatting to build the graphics.
Step 3	Embed the graphic in a Window.

Step 4	To view the graphic in runtime, launch WindowViewer or the Web Client.

Industrial Graphic Editor

The Industrial Graphic Editor is the tool you use to create an industrial graphic. Refer to the Industrial Graphic Editor User Guide for instructions on using the editor. The following sections will list any InTouch HMI specific configurations or settings to be used with the Industrial Graphic editor.

Opening Graphics for Editing

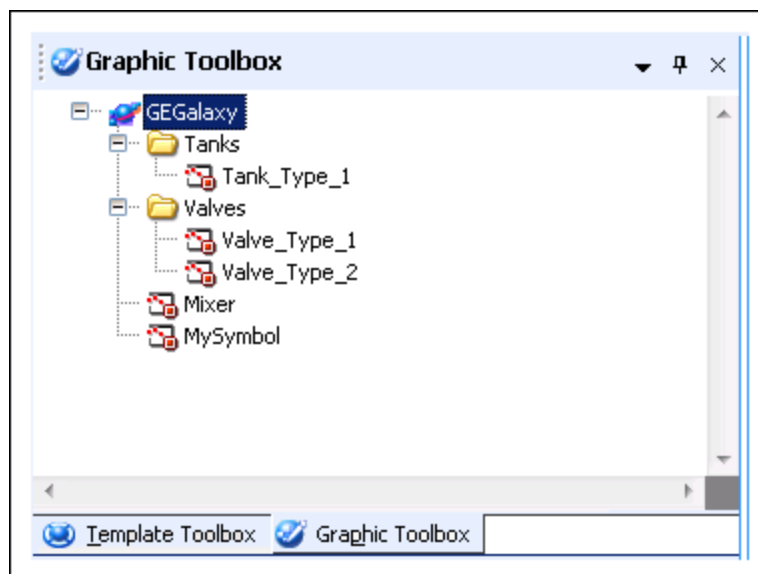You can start the Industrial Graphic Editor from a graphic:

- Contained in the Graphic Toolbox.

- Contained in an AutomationObject template or instance.

- Embedded in an InTouch window.

You check out a graphic when you open it for editing with the Industrial Graphic Editor. No other user can edit the graphic while you have it checked out.

You can open multiple instances of the Industrial Graphic Editor at the same time. However, you cannot edit the same graphic in multiple instances of the Industrial Graphic Editor.

To edit a graphic in the Graphic Toolbox

1. Open the Graphic Toolbox.

2. Browse to the graphic you want to edit.

3. Double-click the graphic. The Industrial Graphic Editor appears.

4. Edit the graphic.

5. Click **Save** and **Close**. The Industrial Graphic Editor closes and the updated symbol is checked in.

To edit a symbol contained in an AutomationObject

1. Open the AutomationObject.

2. Click the **Graphics** tab.

3. Select the symbol to edit and click **Open**. The Industrial Graphic Editor appears.

4. Edit the symbol.

5. Click **Save** and **Close**. The Industrial Graphic Editor closes and the updated graphic is checked in. Depending on the AutomationObject, a confirmation message may appear. Click **Yes** to save.

To edit a graphic that is embedded into an InTouch window

1. In WindowMaker, open the InTouch window that contains the embedded graphic.

2. Right-click the embedded graphic to edit, point to **Industrial Graphic "graphicname"**, and then click **Edit Symbol**. The Industrial Graphic Editor appears.

3. Edit the graphic.

4. Click **Save** and **Close**. The Industrial Graphic Editor is closed and the updated graphic is checked in.

Note: To leave the graphic checked out, click **Keep Checked Out** in the Industrial Graphic Editor. This ensures that no other user can check out your graphic for editing.

Connecting Animations with InTouch Tags

You can connect an element animation and appearance to an InTouch tag. The InTouch tag provides values at run time that control the animation and appearance of the element.

You can connect an element animation to an InTouch tag by:

* Configuring a reference with the **intouch:tag** syntax. This syntax is not required for referencing tags in an InTouch HMI stand-alone application using Industrial Graphics.

* Using a custom property and configuring the custom property in the embedded Industrial graphic in InTouch to reference an InTouch tag. For more information, see the *AVEVA InTouch HMI* and *ArchestrA Integration Guide*.

* Configuring an ArchestrA attribute reference to the managed InTouchViewApp object that contains the InTouch tags as attributes. The InTouchViewApp object uses the functionality of an InTouchProxy object.

* Configuring an ArchestrA attribute reference to an InTouchProxy object that contains the InTouch tags as items. This is a special case of configuring an ArchestrA attribute reference.

Important: The History Summary data type only works with ArchestrA object attributes intended for AVEVA OMI ViewApps or InTouch HMI managed applications. Do not attempt to use custom properties assigned the History Summary data type in InTouch HMI Modern applications.

Using the InTouch:Tagname Syntax

When you use the **intouch:tagname** syntax, the animation connects to the InTouch tag of the node the graphic is used. There are some restrictions on how you can use this syntax:

* Unlike in Application Server, you cannot use true and false as Boolean values. Use 1 and 0 instead.

- If you want to make a reference to an InTouch SuperTag, use the following syntax instead:
  ```
  attribute("intouch:SuperTag\Member")
  ```

- The "InTouch: " prefix is not needed for an InTouch Standalone application with Industrial Graphics. Tag browsing and runtime tag binding will work without the "InTouch:" prefix.

- ArchestrA custom properties referencing InTouch tags which have hyphens in their names will not work in run time. For example, "InTouch:TAG-1" will not work in run time.

Setting the Input Mode

In some boxes you can enter a value or expression that uses static and/or references to attributes and element properties. Boxes that support both input methods have an **Input Mode** selection icon.

Select:

- **Static Mode input** icon to specify literal static value or expression such as 3.141 or "Test".

- **Reference Mode input** icon to specify a reference to an attribute or element property such as: Tank_001.PV.

Note: To use static string values with or without references in Reference mode, you can enclose them with double-quotes such as: "Description: "+Tank_001.Desc

When you configure an element to reference one of its own properties in a configuration field or a script, you can just use its property name. For Industrial graphics, there are no self-referencing keywords such as "me." as used for AutomationObjects.

You can, however, use the "me." keyword to reference attributes of the AutomationObject that is hosting the Industrial graphic you are currently configuring.

Connecting Animations with InTouchViewApp Attributes

To be able to browse for InTouch tags, you must first:

- Create a managed InTouch application by deriving an InTouchViewApp template and configuring it in WindowMaker.

- Derive an instance of the InTouchViewApp derived template.

The InTouch tags are represented by attributes of the InTouchViewApp object instance.

Using the Galaxy Browser InTouch Tag Browser Tab

You can select InTouch tags directly from the Galaxy Browser when configuring a reference requiring an InTouch tag for an Industrial graphic animation or client script.

When invoked from either the animation editor or the script editor, the Galaxy Browser displays an **InTouch Tag Browser** tab in line with the **Attribute Browser** and **Element Browser t**abs.

The **InTouch Tag Browser** tab lists all InTouchViewApp instances and templates for the current Galaxy in the left pane. The right pane displays the InTouch tags for the selected InTouchViewApp. The **DotFields:** list box will display the dotfields associated with the selected tag.

The **Dotfields** list box below the right pane enables you to specify dotfields for the selected tag.

The **InTouch Tag Browser** tab behaves as follows:

- The InTouch Tag Browser functionality is available only from the animation editor or the script editor.

- The Galaxy Browser reads InTouch tags from the Tagname Dictionary.

- The Tagname Dictionary component is installed and available to the Galaxy Browser whether or not InTouch HMI is installed.

- Tags are refreshed only when the Galaxy Browser is closed and reopened.

- All tags remain in memory until you close and reopen the Galaxy Browser.

- If the InTouchViewApp is checked out by the current user, then the Galaxy Browser reads the latest Tagname Dictionary content of that InTouchViewApp.

- If the InTouchViewApp is checked in and is accessed by any user, then the Galaxy Browser always reads the checked-in version of the Tagname Dictionary.

- If the InTouchViewApp is checked out by a user other than the current user, the Galaxy Browser reads the most recently checked in Tagname Dictionary of that InTouchViewApp.

- If you select an InTouchViewApp template, the output reference string syntax is <InTouch:selectedTag>. If you select an InTouchViewApp instance, the output reference string syntax is <SelectedInTouchViewAppInstance.selectedTag>.

To connect animations to InTouch tags

1. Select the element.

2. On the **Special** menu, click **Edit Animations**. The **Edit Animations** dialog box appears.

3. Select the animation from the Animation list.

4. Select the parameter.

5. Click the browse button. The **Galaxy Browser** appears.

6. Click the **InTouch Tag Browser** tab to show the **InTouch Tag Browser** page.

7. Select the InTouchViewApp object that corresponds to the managed InTouch application. The right panel shows the InTouch tags.

8. Select a tag and click **OK**. The selected ArchestrA reference to an InTouch tag appears in the configuration box.

CHAPTER 8

Comparing WindowMaker and Industrial Graphic Editor

Differences between WindowMaker and the Industrial Graphic Editor

You can use the Industrial Graphic Editor to do most of the tasks you do in InTouch WindowMaker. You can also use many of the same shortcut keys.

The Industrial Graphic Editor has features that are not available in InTouch WindowMaker, such as:

- Additional elements.

- Additional and enhanced appearance of the elements.

- Additional and enhanced design-time functionality.

Elements

Elements are the graphical objects you use to create an Industrial Graphic. The Industrial Graphic Editor provides elements that are not available in InTouch WindowMaker, such as:

- Curves and closed curves.

- Arcs, pies, and chords defined by two or three points.

- Status elements to conditionally show an icon depending on quality and status of attribute data .

- Path graphics that you create by joining line-based elements together to form a new closed element.

- Windows common controls, such as the Calendar control and Date Time Picker control.

Appearance

The Industrial Graphic Editor extends the InTouch graphic configuration. For example, you can use:

- Gradients for line, fill, and text color.

- Patterns for line, fill, and text color.

- Textures for line, fill, and text color.

- Partial transparency.

- Fill behavior in relation to a symbol or screen.

Enhanced Functionality

The Industrial Graphic Editor provides a entire array of enhancements to make your life easier when creating visualization for your manufacturing environment.

Usability Enhancements

The Industrial Graphic Editor makes it easy to select and configure elements. You can:

- Select elements from a list as well as from the canvas. This lets you select elements beneath others without having to move them.

- View and change properties and animation (links) of an element by simply selecting it on the canvas.

- Edit elements contained in groups and path graphics without having to break the group or path graphic. This is called inline editing.

Style Replication

Using the Format Painter, you can simply apply the style of one element with one click to another element, even to an element of a different type.

Animation Replication

Using the Industrial Graphic Editor you can copy, cut, and paste animations from one element to another element, even to an element of a different type.

Element Positioning

The Industrial Graphic Editor extends the positioning feature of InTouch WindowMaker and lets you:

- Distribute elements equally in horizontal or vertical direction.

- Make elements same horizontal and/or vertical size.

- Increase or decrease horizontal or vertical space.

- Remove horizontal or vertical space between elements.

- Lock an element so that you do not accidentally move or edit it.

- Rotate any element at design time by any angle around a center of rotation.

- Apply resizing and rotating to multiple elements at the same time.

- Move the z-order of an element one level backward or forward.

- Align text within text boxes and buttons.

Group Functionality

The Industrial Graphic Editor uses the concept of groups instead of the cell and symbol concepts of InTouch WindowMaker. You can:

- Embed groups within groups.

- Edit individual elements within a group (or an embedded group) without breaking up the group.

- Easily remove elements from or add elements to existing groups.

Extensibility with Custom Properties

You can add custom properties to a symbol or embedded symbol. You can connect custom properties to AutomationObject attributes, element properties, and even InTouch tags. You can use the custom properties as you would with any pre-defined property at design time and run time.

Note: ArchestrA custom properties referencing InTouch tags that have hyphens in their names will not work in run time. For example, "InTouch:TAG-1" will not work in run time.

Element Styles

Element Styles define one or more of the fill, line, text, blink, outline, and status properties of graphic elements. Apply an Element Style to a graphic element to set the element to the preconfigured properties defined in that Element Style. The element's local properties that are defined in the Element Style are disabled.

Element Styles help drive standards for screen builders and others who are creating symbols.

Miscellaneous Enhancements

Using the Industrial Graphic Editor, you can:

* Access the properties of the elements and custom properties of the symbol through scripting.

* Set the tab order of the elements.

* Use line end styles, such as arrows.

* Dynamically disable specific animations from elements without losing the configuration information.

* Use image meta files and other image formats.

* Use anti-aliasing to improve how the symbol is shown.

Procedures for Common WindowMaker Tasks and Techniques

Most of the configuration that you do in InTouch WindowMaker can be easily done in the Industrial Graphic Editor. There are some differences between and similarities of graphics, animations, and scripts.

Using Graphics

You can use the Industrial Graphic Editor in basically the same way as you use InTouch WindowMaker. The Industrial Graphic Editor includes a drawing area on which you can place graphical objects to construct a visual representation of production processes and to provide an interface between a human and a machine.

Some objects you use in InTouch do not exist in the Industrial Graphic Editor, such as ActiveX controls and some Wizards. Their functionality is replaced other controls that are more powerful and integrate better into the ArchestrA environment.

Using the Drawing Area

The drawing area of the Industrial Graphic Editor is called the canvas. You use it like an InTouch window. Its maximum size is 2,000 by 2,000 pixels.
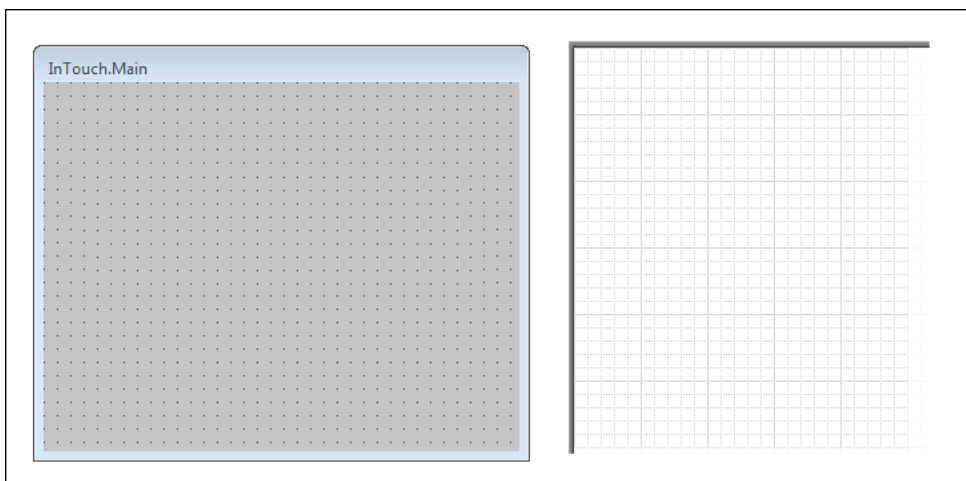
Setting the Drawing Area Color

You cannot set the drawing area color in the Industrial Graphic Editor. The drawing area color is transparent and inherits the color of the environment that the symbol is embedded into.

If you embed an Industrial graphic into an InTouch window, the area between the elements adopts the color of the InTouch window.

Using Basic Objects

InTouch basic objects such as rectangles, ellipses, and polylines can be drawn in very similar way in the Industrial Graphic Editor. The basic objects are called elements in the Industrial Graphic Editor.
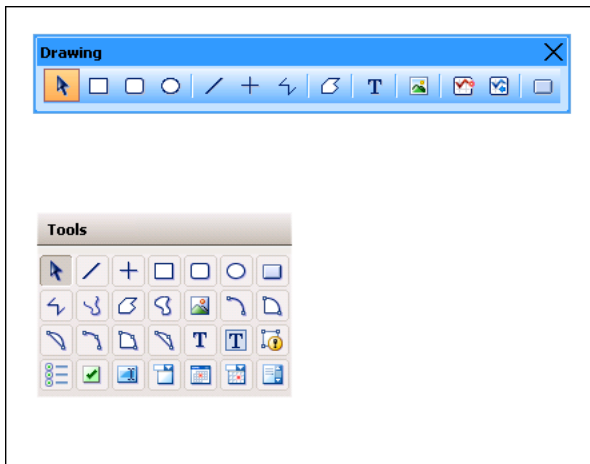
Using Complex Objects

InTouch objects such as ActiveX controls, Wizards, cells, and symbols do not exist in the Industrial Graphic Editor.

You can, however, import SmartSymbols into an Industrial Graphic. When you import a SmartSymbol, the elements and animations of a SmartSymbol are converted.

In the Industrial Graphic Editor, you can create groups of elements. Groups maintain the properties of the contained individual elements. You can set the TreatAsIcon property of a group to change the behavior of the group.

Using Wizards

You cannot import InTouch Wizards to an Industrial Graphic or into the Graphic Toolbox. Instead, use:

- The Industrial Graphic Library, which you can import into the Graphic Toolbox.

- Windows controls that are part of the Toolbox. You can use:
  o Radio button groups
  o Check boxes
  o Edit boxes
  o Combo boxes
  o Calendar control
  o DateTime picker
  o List boxes

Using Animations

You can use animations in the Industrial Graphic Editor to set run-time behavior of the symbols as you would in InTouch WindowMaker. You can configure one or more animations for an element or symbol. The data can come from various sources.

Configuring Data Sources

In InTouch WindowMaker, you use the Tagname Dictionary to define variables that hold values. In the Industrial Graphic Editor, data sources can be:

- ArchestrA AutomationObject attributes.

- Custom properties and inherited properties of the symbol.

- InTouch tags themselves. The Industrial Graphic Editor uses a special InTouch reference you can use to directly connect to InTouch tags.

Using Data Types

Industrial graphics use the ArchestrA data types, which are different than InTouch data types.

The following table shows you the data types of both and how they correspond to each other:

InTouch	ArchestrA	Description
Discrete	Boolean	Boolean value. For example: 0 or 1
Integer	Integer	Integer value. For example: -4, 7, or 22
Real	Float or Double	Float or double value with different precision. For example: 3.141, -5.332, or 1.343e+17 **Float**: 32 bit. IEEE single precision floating point standard, used when 6-7 significant digits are needed. Default is NAN. **Double**: 64 bit. IEEE double, used when 15-16 significant digits are needed. Default is NAN.
Message	String	String value. For example: "Hello World"
n/a	DateTime	Datetime value. For example: "04/13/2006 04:03:22.222 AM"
n/a	ElapsedTime	Float value that represents a time that has elapsed in seconds. It is shown often in the following format, but is stored as a float value. [–][DDDDDD] [HH:MM:]SS[.fffffff] Values are as follows: • DDDDDD is from 0 to 999999 • HH is from 0 to 23 • MM is from 0 to 59 • SS is from 0 to 59 • fffffff is fractional seconds to right of the decimal Elapsed time can be positive or negative.
n/a	InternationalizedString	A special string data type that can store special characters.

You can configure Industrial Graphics to retrieve data from the Galaxy.

When the source data type is different than the data type it is used for, the data is coerced according to the rules of ArchestrA data coercion and a string value of "-10" is coerced to "True" in the animation.

If you embed this Industrial graphic into an InTouch window, the data type of the animation link is coerced according to the InTouch data coercion. The embedded Industrial graphic shows "True" in the InTouch HMI.

However, if you directly create an discrete animation display link in the InTouch HMI that points at the original data source, the resulting value can be different.

In this example the string value "-10" is shown as "False" in the InTouch HMI.

Data Coercion – Direct and Indirect
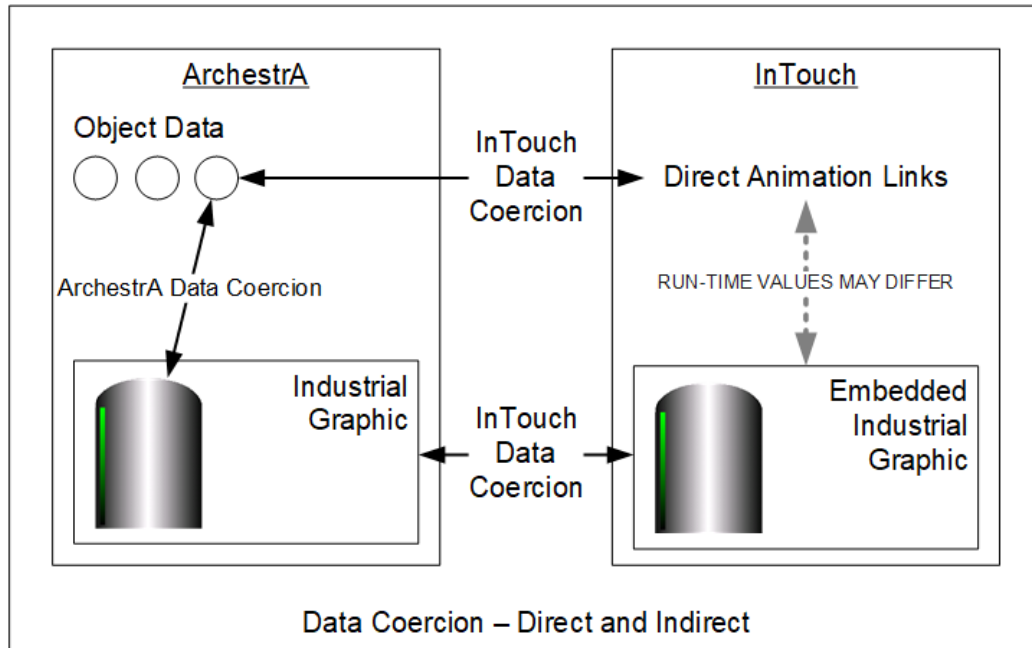
Using Animations

You configure InTouch animations using the **Animation Links** dialog box. You can open this dialog box by double-clicking an InTouch object.

You configure animations in the Industrial Graphic Editor using   the **Edit Animations** dialog box, which is normally opened by double-clicking an element.

Some of the animation types are different and others have been grouped to simplify configuration. Use the following table to find the equivalent animation type in the Industrial Graphic Editor:

InTouch Animation	Industrial Graphic Editor Animation
User Inputs - Discrete	User Input - Boolean
User Inputs - Analog	User Input - Analog
User Inputs - String	User Input - String
Sliders - Vertical	Slider Vertical
Sliders - Horizontal	Slider Horizontal
Touch Pushbuttons - Discrete Value	Pushbutton - Boolean
Action	Action Scripts

InTouch Animation	Industrial Graphic Editor Animation
Show Window	(not supported)
Hide Window	(not supported)
Line Color - Discrete	Line Style - Boolean
Line Color - Analog	Line Style - Truth Table
Line Color - Discrete Alarm	converted to Line Style
Line Color - Analog Alarm	converted to Line Style
Fill Color - Discrete	Fill Style - Boolean
Fill Color - Analog	Fill Style - Truth Table
Fill Color - Discrete Alarm	converted to Fill Style
Fill Color - Analog Alarm	converted to Fill Style
Text Color - Discrete	Text Style - Boolean
Text Color - Analog	Text Style - Truth Table
Text Color - Discrete Alarm	converted to Text Style
Text Color - Analog Alarm	converted to Text Style
Object Size - Height	Height
Object Size - Width	Width
Location - Vertical	Location Vertical
Location - Horizontal	Location Horizontal
Percent Fill - Vertical	% Fill Vertical
Percent Fill - Horizontal	% Fill Horizontal
Miscellaneous - Visibility	Visibility
Miscellaneous - Blink	Blink
Miscellaneous - Orientation	Orientation
Miscellaneous - Disable	Disable
Miscellaneous - Tooltip	Tooltip
Value Display - Discrete	Value Display - Boolean
Value Display - Analog	Value Display - Analog
Value Display - String	Value Display - String

# Using Scripts

You can configure scripts in Industrial Graphic Editor the same way as you do in InTouch WindowMaker. There are, however, some small differences:

InTouch Script	Industrial Graphic Editor Script
Application Script	(not available)
Window Script	Symbol Predefined Script
Key Script	Action Script animation with a key trigger
Condition Script	Symbol Named Script with an OnTrue, OnFalse, WhileTrue or WhileFalse trigger
Data Change Script	Symbol Named Script with a DataChange trigger
QuickFunction	(not available)
ActiveX Event Script	(not available)
Action Script	Action Script animation

Using Application Scripts

In the InTouch HMI, application scripts can be triggered:

- One time when the application starts in WindowViewer.

- Periodically when the application runs in WindowViewer.

- One time when the application shuts down in WindowViewer.

Industrial graphics correspond to InTouch applications and enable you to configure predefined scripts directly associated with symbol. Such are:

- On Show

- While Showing

- On Hide

Using Key Scripts

You cannot use key scripts in the Industrial Graphic Editor, but you can associate an element with an action script that is activated by a key combination.

Using Condition Scripts

You can configure a script that runs when a condition is fulfilled by using the Symbol Scripts feature. It lets you define triggers that run a script when a value or expression:

- Is fulfilled. (WhileTrue)

- Becomes fulfilled. (OnTrue)

- Is not fulfilled. (WhileFalse)

- Becomes no longer fulfilled. (OnFalse)

Using Data Change Scripts

You can configure a script that is run when a value or expression changes by using the Symbol Scripts feature. It lets you define a trigger that runs a script when a value or expression changes.

Using Action Scripts

You can configure Action Scripts in the Industrial Graphic Editor the same way as you would in InTouch WindowMaker. When the run-time user interacts with an element, such as with the mouse or by pressing a key, an action script can run.

You use the InTouch action script window to create action scripts.
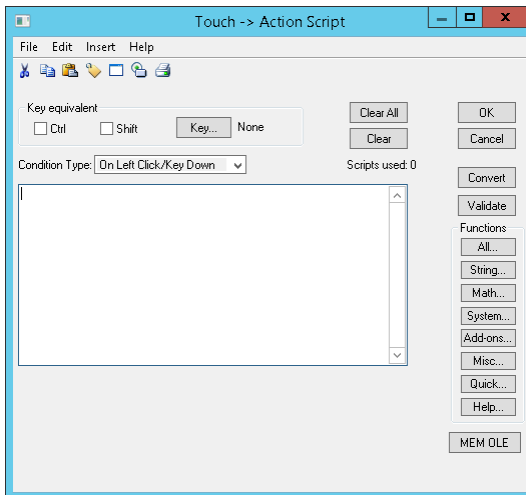
You use the Industrial Graphic Editor action script window to create action scripts.
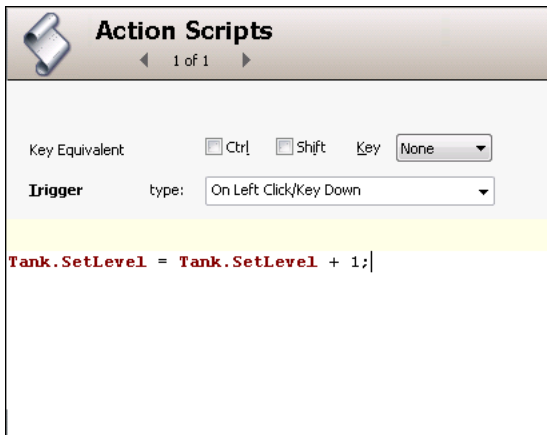
You can configure action scripts for individual elements or for the entire symbol.

You can use many of the predefined functions of InTouch WindowMaker in the Industrial Graphic Editor. For a complete list of InTouch predefined functions that can be used with Industrial Graphics, see *Importing Action Scripts* on page 143.

Other InTouch script types, such as application scripts and key scripts, can be configured with ArchestrA AutomationObjects.

CHAPTER 9

Migrating InTouch SmartSymbols

Importing InTouch SmartSymbols into an Industrial Graphic

You can use InTouch SmartSymbols in Industrial graphics by importing (migrating) them. The SmartSymbol appearance and configuration is imported and converted to animation configuration.

The imported SmartSymbol can:

- Be added to the existing elements on the canvas.

- Replace the existing elements on the canvas.

Generally, you can import any InTouch SmartSymbol into an Industrial Graphic.

Note: The SmartSymbol can contain objects that cannot be imported, or can be imported but have limited functionality. For a full list of these objects, see *Restrictions for SmartSymbol Import* on page 140.

To import an InTouch SmartSymbol into an Industrial Graphic

1. Open the Industrial Graphic Editor.

2. On the **Special** menu, click **Import InTouch SmartSymbol**. The **Find InTouch Application Wizard** appears.

3. If your InTouch application is in a different folder than the default, click the browse button to browse for the path to the InTouch application.

4. Select **Find Applications**. The **Search Root** box shows the path under which all applications are to be searched.

5. If your InTouch application is not under the specified **Search Root** path, change the **Search Root** path by typing a new start folder for the search or browsing for one.

6. Click **Find**. All InTouch applications contained in the specified **Search Root** folder are found and listed.

7. Select the application from which you want to import SmartSymbols and click **Next**. The **Select InTouch SmartSymbol** dialog box appears.

8. Browse to the location of the SmartSymbol in the SmartSymbol hierarchy, select the SmartSymbol that you want to import, and then click **OK**.

   If you already have elements on the canvas, a dialog box appears prompting if you want to replace the existing elements. Click:

   o **Yes** if you want to delete the existing elements and import the SmartSymbol on an empty canvas.

   o **No** if you want to keep the existing elements and import the SmartSymbol.

9. If the SmartSymbol contains fonts that are currently not installed on the operating system, the **Edit Font Mapping** dialog box appears.

   You can click **Continue** to accept the suggested font mapping, or change the font mapping for each individual font. To do this:

   a. Click the font name in the **Mapped Font** column. A browse button appears.

   b. Click the browse button. The **Supported Font Selection** dialog box appears.

     c. Select a font from the list and click **OK**. The selected font appears in the **Mapped Font** column.

     d. Repeat the steps for any other font you want to map to another font.

Note: If you want to save the mapping for the next time you import a SmartSymbol, check **Save mapping**.

10. The SmartSymbol is imported and appears on the canvas.

Restrictions for SmartSymbol Import

When you import an InTouch SmartSymbol, the following configuration is imported:

- InTouch graphics
- Graphical animations
- Scripts
- References

Importing InTouch Graphics

The following tables shows you InTouch graphics that:

- Can be imported without any problem.
- Can be imported but are changed in their functionality or lose some functionality in the process.
- Cannot be imported.

The following InTouch graphics can be imported without any problem:

InTouch Graphic	Industrial Graphic Element	Notes
Rectangle	Rectangle	
Rounded Rectangle	Rounded Rectangle	
Ellipse	Ellipse	
Line	Line	
H/V Line	Line	Smart Symbols convert H/V lines to Lines. Therefore, ArchestrA can only generate lines.
Polyline	Polyline	
Polygon	Polygon	
Text	Text	
Bitmap	Bitmap	
Cell	Group	ArchestrA property "Treat as Icon" = false.

InTouch Graphic	Industrial Graphic Element	Notes
Symbol	Group	ArchestrA property "Treat as Icon" = true.
Button	Button	

The following InTouch graphics can be imported, but are changed in their functionality or lose some functionality in the process:

InTouch Graphics	Industrial Graphic Element	Notes
Wizard\	Elements	When grouped in a SmartSymbol, it appears as a group of elements.
SmartSymbol	Elements	When grouped in another Smart Symbol, it is broken down into a cell, losing its SmartSymbol properties.

The following InTouch graphics cannot be imported, as they cannot be added to a SmartSymbol:

InTouch Graphic	Industrial Graphic Element	Notes
RealTime Trend	n/a	Cannot be added to a SmartSymbol.
Historical Trend	n/a	Cannot be added to a SmartSymbol.
ActiveX Controls	n/a	Cannot be added to SmartSymbol. This would include all ActiveX alarm controls (Alarm DB View, Alarm Viewer, and so on)

Importing Graphical Animation

When you import an InTouch SmartSymbol, all data configured in InTouch animations is imported to ArchestrA animations. InTouch animations and ArchestrA animations often have a different name, but perform the same function.

The following table shows you which animations correspond to each other:

InTouch Animation Link	ArchestrA animation
User Inputs - Discrete	User Input

InTouch Animation Link	ArchestrA animation
User Inputs - Analog	User Input
User Inputs - String	User Input
Sliders - Vertical Slider	Vertical Slider
Sliders - Horizontal Slider	Horizontal Slider
Touch Pushbuttons - Discrete Value	Pushbutton
Touch Pushbuttons - Action	Action Scripts
Touch Pushbuttons - Show Window	not supported
Touch Pushbuttons - Hide Window	not supported
Line Color - Discrete	Line Style
Line Color - Analog	Line Style
Line Color - Discrete Alarm	Line Style
Line Color - Analog Alarm	Line Style
Fill Color - Discrete	Fill Style
Fill Color - Analog	Fill Style
Fill Color - Discrete Alarm	Fill Style
Fill Color - Analog Alarm	Fill Style
Text Color - Discrete	Text Style
Text Color - Analog	Text Style
Text Color - Discrete Alarm	Text Style
Text Color - Analog Alarm	Text Style
Object Size - Height	Height
Object Size - Width	Width
Location - Vertical	Location Vertical
Location - Horizontal	Location Horizontal
Percent Fill - Vertical	% Fill Vertical
Percent Fill - Horizontal	% Fill Horizontal
Miscellaneous - Visibility	Visibility
Miscellaneous - Blink	Blink
Miscellaneous - Orientation	Rotation

InTouch Animation Link	ArchestrA animation
Miscellaneous - Disable	Disable
Miscellaneous - Tooltip	Tooltip
Value Display - Discrete Value	Value Display
Value Display - Analog Value	Value Display
Value Display - String Value	Value Display

Importing Action Scripts

When you import a SmartSymbol, all action scripts associated with objects in SmartSymbol are imported as well. An action script in a SmartSymbol becomes a script animation in an Industrial Graphic.

Most of the predefined InTouch functions (QuickScripts) are imported.

Mathematical Functions

The following mathematical functions in InTouch WindowMaker are supported by the Industrial Graphic Editor:
```
Abs, ArcCos, ArcSin, ArcTan, Cos, Exp, Int, Log, LogN, Pi, Round, Sgn, Sin, Sqrt,
Tan, Trunc
```

String Functions

The following string functions in InTouch WindowMaker are supported by the Industrial Graphic Editor:
```
Dtext, StringASCII, StringChar, StringCompare, StringCompareNoCase,
StringFromGMTTimeToLocal, StringFromIntg, StringFromReal, StringFromTime,
StringFromTimeLocal, StringInString, StringLeft, StringLen, StringLower,
StringMid, StringReplace, StringRight, StringSpace, StringTest, StringToIntg,
StringToReal, StringTrim, StringUpper, Text, wwStringFromTime
```

System Functions

The following system functions in InTouch WindowMaker are supported by the Industrial Graphic Editor:
```
ActivateApp
```

Miscellaneous Functions

The following miscellaneous functions in InTouch WindowMaker are supported by the Industrial Graphic Editor:
```
DateTimeGMT, LogMessage, SendKeys, WWControl
```

Importing References

When you import a SmartSymbol, the following changes are made to tags and references:

InTouch SmartSymbol	Industrial Graphic	Example
Local Tags	Prefixed with "InTouch:" keyword	Real Memory Tag "TankLevel1" is converted to "InTouch:TankLevel1"

InTouch SmartSymbol	Industrial Graphic	Example
Local Tags with dotfields	Prefixed with "InTouch:" keyword	Discrete Memory Tag "TankLevel1.InAlarm" is converted to "InTouch:TankLevel1.InAlarm"
SuperTags	Prefixed with "InTouch:" keyword. You need to manually enclose the expression by the following syntax: attribute("...");	Real Supertag member "Reactor1\Level" is converted to "InTouch:Reactor1\Level". You need to change the expression manually as follows: attribute("InTouch:Reactor1\Level");
I/O References	Prefixed with "InTouch:" keyword	Integer I/O Tag "Testprot:i00" is converted to "InTouch:Testprot:i00"
Galaxy References	"Galaxy:" prefix is removed	Galaxy Reference "galaxy:Pump1.Valve1" is converted to "Pump1.Valve1"

The following items are imported with functional change:

InTouch SmartSymbol	Industrial Graphic	Example
Galaxy:ObjectTagname. Property.#VString	"Galaxy:" prefix is removed but #VString is not supported. Applies also for #VString1, #Vstring2, #VString3 and #VString4	"Galaxy:Tank.PV.#VString4" is converted to "Tank.PV"
Galaxy:ObjectTagname. Property.#ReadSts	"Galaxy:" prefix is removed but #ReadSts is not supported	"Galaxy:Tank.PV.#ReadSts" is converted to "Tank.PV"
Galaxy:ObjectTagname. Property.#WriteSts	"Galaxy:" prefix is removed but #WriteSts is not supported	"Galaxy:Tank.PV.#WriteSts" is converted to "Tank.PV"
Galaxy:ObjectTagname. Property.#EnumOrdinal	"Galaxy:" prefix is removed but #EnumOrdinal is not supported	"Galaxy:Selection.Sel1.#EnumOrdinal" is converted to "Selection.Sel1"

Understanding Trend Pen Historical Data Retrieval

When an application containing a Trend Pen starts running in WindowViewer, a Historian query retrieves data for the entire Trend Pen period. Real-time data is plotted on the trend line during the period that historical data is being retrieved. After retrieving historical data, it is added to real-time data to back fill the trend line for the entire period.

The following procedure shows the steps to configure a Trend Pen. Typically, configuring a Trend Pen includes several steps to place the Trend Pen next to a meter graphic to visually indicate the Trend Pen plot shows the changes in the graphic's process value over time.

To configure Trend Pen

1. Click **Trend Pen** from the Graphic Toolbox.

   The cursor changes to a cross hair when placed over the Industrial Graphic Editor canvas area.

2. Place the cursor over the canvas at the position you want to set as a horizontal boundary of the Trend Pen plot.

3. Pressing the left mouse button, drag the cursor to create a rectangle.

   The horizontal and vertical boundaries of the Trend Pen graphic rectangle represent the drawing area of Trend Pen plot during run time. The horizontal axis of the graphic rectangle represents the time period of the trend. The vertical axis represents the range of the trend's possible values.
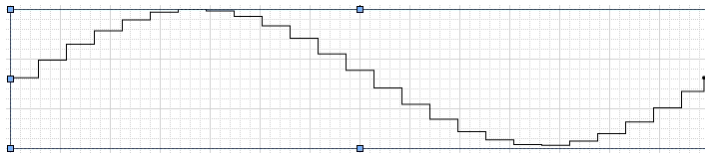
4. Release the left mouse key after sizing the rectangle to the height and width that you want to represent as the trend area.

   The **Trend Pen** dialog box appears when the mouse key is released. You can also show the **Trend Pen** dialog box by double-clicking on the Trend Pen graphic or selecting **Edit Animations** from the **Special** menu.

5. Enter a reference in the **Reference** field.

   The reference is the data source that appears as the value shown by the trend, which can be an external reference like an object's attribute, an analog tag value, or a custom property. Constants and expressions are not allowed.

6. Select **Historian** or **InTouch Log History/LGH** for the Historical Source. If you select Historian proceed with Step 8.

7. On selecting **InTouch Log History/LGH** you can use the icon against the **UNC Path** to toggle input to the field as an Expression or Static text.

   a. If you select the Expression mode, you can click the ellipsis button to launch your HMI's attribute/tag browser and select a Custom property, attribute or tag.

   b. If you select the Static Text mode, you can click the ellipsis button to launch the file browser where you can specify the LGH file name.

8. Select **Auto-Detect** or **Expression** for the method to identify the location of the Historian.

   Auto-Detect

   The Historian server is auto-detected from the AppEngine on which the reference attribute is running. For example, if the **Reference** field is set to UDO.UDA1, then **Auto-Detect** is set to the Historian server name configured for the AppEngine on which UDO is running. Auto-Detect is only valid for Application Server References.

Expression

When an expression or reference is entered in the **Server Name** field, the Trend Pen connects to the specified Historian Server.

The icon to the left of the **Server Name** field toggles input to the field as an expression or Static Text mode.

A Trend Pen only shows live data if the **Server Name** field is left blank in **Expression** mode.

9. Select **Moving** or **Fixed** as the type of trend time period.

Moving

The start time of a trend period is the current time, and the end time is the duration of the time period from the start time. The start time for the next period is set to the end time of the previous trend period.

Fixed

In a Fixed trend time period, the StartTime and EndTime properties do not change automatically. The start time of a trend period is the current time initially. The StartTime property can be changed by a script.

The EndTime property of a trend period (both Moving and Fixed) is read-only. The end time of a trend period is calculated from the specified start time and duration of the period.

10. Set the X-axis time period of the trend line in the **Duration (Minutes)** field.

The trend time period can be specified as a constant, an external reference, an expression, or custom property. If a floating point number is entered, the period is rounded up to the nearest minute.

The minimum trend period is 1 minute and the maximum period is 10080 minutes (1week).

11. Select **Auto-Range** or **Clip out of Range Values** for the scaling method to place process values on a trend line.

If **Auto-Range** is selected, the **Min Range** and **Max Range** fields are disabled. The Y-axis of the trend line is automatically adjusted to show the full range of trend values within the upper and lower boundaries of the Trend Pen graphic.

If **Clip out of Range Values** is selected, the **Min Range** and **Max Range** fields are enabled. **Min Range** and **Max Range** set the lower and upper limits of the trend's Y-axis value range. Both fields can be set to constants, external references, or custom properties.

When a value exceeds the trend's minimum or maximum limits using **Clip out of Range Values**, the trend line is truncated at the limit of the value range and appears as a horizontal line for the period when the process value is out of trend's value range.

12. From **Plot Type**, select **Step Line** or **Line** as the type of trend plot.

   o A **Step Line** plot draws a horizontal line from a trend data point to the time of the next data point on the trend's X-axis, and then draws a vertical line to the data point.

   o A **Line** plot draws a line directly to each successive point within the trend period.

Changing Trend Pen Properties During Run Time

A Trend Pen contains a collection of properties whose values can be modified during run time to change the visual and functional characteristics of a trend. The following illustration is one example of a browser or explorer typically available in HMI software to search for attributes or tags, the values of which can be changed at run time.
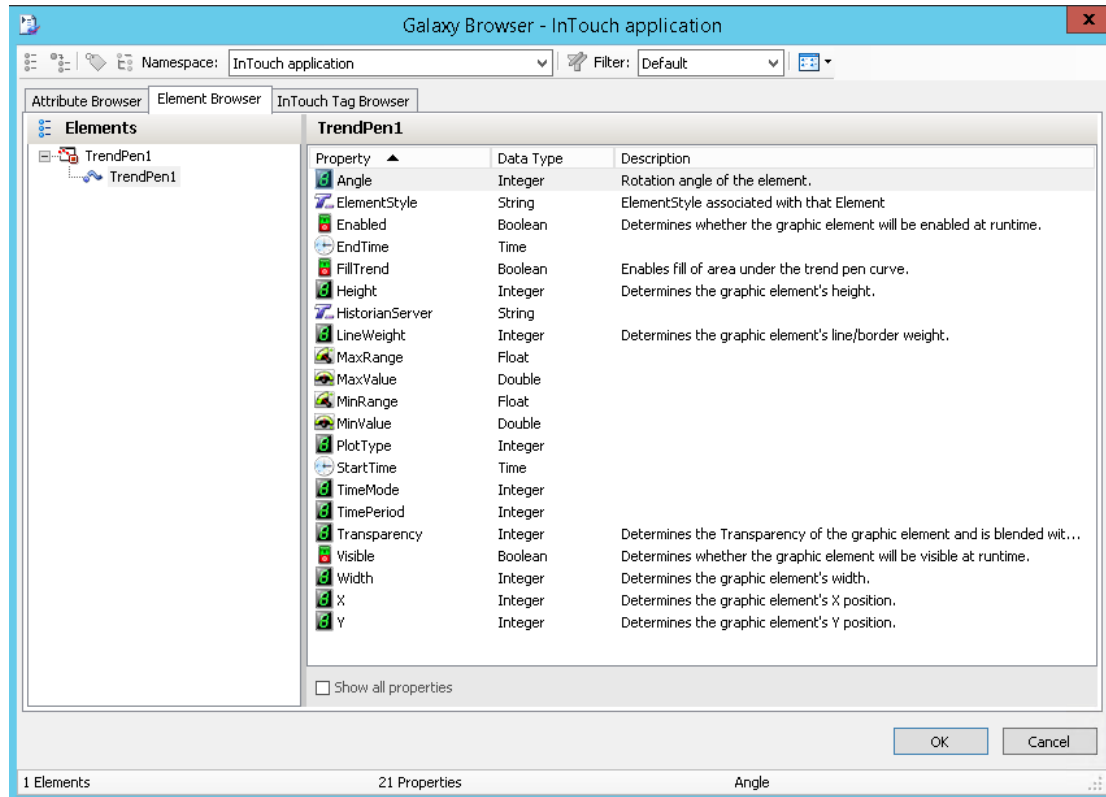
The following properties are typically modified during run time to change the visual and functional characteristics of a Trend Pen.

MinValue Property

During run time, the value of the MinValue property can be modified to change the minimum measured value shown by a trend. During run time, MinValue can be a read/write or a read only property based on the value assigned to the Trend Pen's **Y Axis Range** option during design time.

- When **Y Axis Range** is set to **Auto-Range** during design time

  The minimum measured value shown by a trend is set to the minimum value of data received from the Historian or from current data during the trend period. MinValue is read only.

- When **Y Axis Range** is set to **Clip out of range** values during design time.

  The minimum measured value shown by a trend is set to a minimum limit set from the **Min Range** option during design time.

  MinValue is read/write and can be changed during run time. When MinValue is changed during run time, the trend line is redrawn based on the values assigned to the MinValue and MaxValue properties.

  When the values assigned to MinValue and MaxValue properties are the same, the trend's Y Axis Range automatically changes to Auto-Range.

MaxValue Property

During run time, the value of the MaxValue property can be modified to change the maximum measured value shown by a trend. During run time, MaxValue can be a read/write or a read only property based on the value assigned to the Trend Pen's Y Axis Range option during design time.

- When **Y Axis Range** is set to **Auto-Range** during design time

The maximum measured value shown by a trend is set to the maximum value of data received from the Historian or from current data during the trend period. MaxValue is read only.

- When **Y Axis Range** is set to **Clip out of range** values during design time

The maximum measured value shown by a trend is set to a maximum limit set from the **Max Range** option during design time.

MaxValue is read/write and can be changed during run time. When MaxValue is changed during run time, the trend line is redrawn based on the values assigned to the MinValue and MaxValue properties.

When the values assigned to MinValue and MaxValue properties are the same, the trend's Y Axis Range automatically changes to Auto-Range.

StartTime Property

During run time, the value of the StartTime property can be modified to change the start time of a trend period based on the value set to the Trend Pen **Time Period** during design time.

- When **Time Period** is set to **Fixed** during design time

The default value assigned to the StartTime property is the time when the Trend Pen first appears in the WindowViewer, also the start time changes as time passes. StartTime is read/write and can be changed during run time. When value of StartTime changes, the Trend Pen re plots the trend using new StartTime value.

- When **Time Period** is set to **Moving** during design time

The value set to the StartTime property is the current system date and time. StartTime is read only.

PlotType Property

During run time, the value of the **PlotType** property can be modified to change the type of trend plot.

- When **PlotType** is 0, the Trend Pen plot type is Step Line. The default.

- When **PlotType** is 1, the Trend Pen plot type is Line.

The value of **PlotType** is ignored if the value is neither 0 nor 1.

When the value of **PlotType** changes in run time, trend data is retrieved again before drawing the trend.

TimeMode Property

During run time, the value of the **TimeMode** property can be modified to change the type of trend time period.

- When **TimeMode** is 0, the trend time period mode is **Moving**. The end time of the trend is the current time. The default.

- When **TimeMode** is 1, the trend time period mode is **Fixed**. The start time of the trend is when the Trend Pen first appears in the WindowViewer, also the start time changes as time passes.

The value of TimeMode is ignored if the value is neither 0 nor 1.

When the time period changes from Moving to Fixed during run time, the trend's start time and end time remain the same before switching, and the data remains as well. When the time period changes from Fixed to Moving during run time, data is retrieved again before drawing the trend. The trend's start and end times are adjusted automatically by Moving mode.

FillTrend Property

During run time, you can use the FillTrend property to change the appearance of the area below the trend pen curve.

The FillTrend property can also be modified using scripts. `For example TrendPen6.FillTrend = not TrendPen6.FillTrend;`

The Fill Property in Fill Style animation, Element Style animation, or Element Style will now apply to the FillTrend property according to the general element style precedent rules.

For more information on the Fill Style animation, see Setting Fill Style.

For more information on Element Style animation, see Configuring an Animation Using Element Styles.

Note: The FillTrend property can be enabled during design or runtime. In runtime the color will default to white, if no color is selected in design time.

<div align="right">

CHAPTER 10

</div>

Working with Industrial Graphics in AVEVA Connect

About AVEVA Connect

AVEVA Connect is the common cloud repository allowing you to reuse graphics across various nodes.

Graphics can be downloaded and uploaded on demand. Graphics are stored in 'Stores' in AVEVA Connect; there are three types of stores – global, tenant and user specific. Within each store users can configure multiple drives. You must have an AVEVA Connect user account to manage graphics in the cloud. Each drive can be configured with different access levels for different users.

Users will have read-only or read-write access depending on the access granted by the administrator. Read-Only users can only view graphics in the cloud, they cannot upload or download graphics. Multiple users can access the graphics in the same drive, but only one user can edit or save a graphic at one time.

Logging in to AVEVA Connect

You must have an AVEVA Connect account and authorization from your administration to access AVEVA Connect.

1. Launch WindowMaker.

2. Click **AVEVA > SIGN IN**.
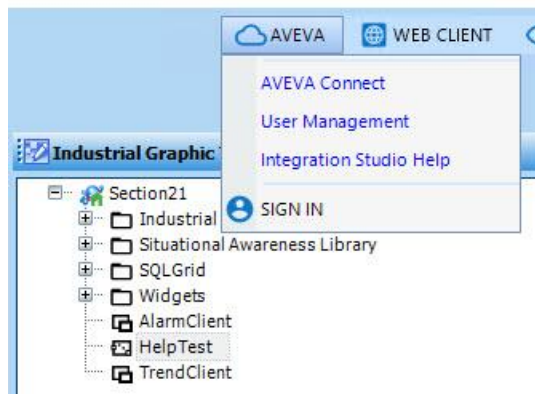
The AVEVA Menu also provides the following options:

AVEVA Connect – Links to AVEVA Connect

User Management – Links to AVEVA Connect Administrator Page

Integration Studio Help – Links to the help page

3. Enter your AVEVA Connect email address.

4. Enter your password.

If the credentials are verified, you are signed in.

5. Select the Store. For example: Pym Technologies.

6. Based on your authorization and privileges you will have access to AVEVA Connect.

7. The AVEVA Drive appears as a separate repository within the Graphic Toolbox.
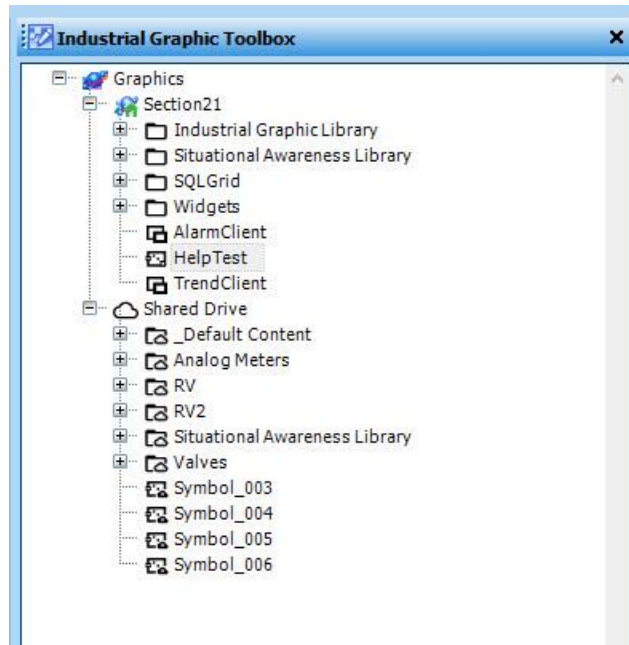
Changing the AVEVA Connect Drive

Once you are logged into a specific Store you can move between different drives and upload and download images.

1. Click **AVEVA > Shared Drive**.

2. Select a drive from the available options.
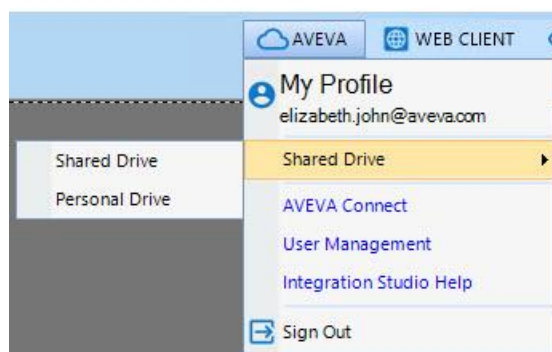
3. The graphic toolbox will refresh to display the selected drive.

Uploading Graphics to AVEVA Connect

You can create folders on the AVEVA Drive to organize your graphics, like the graphic toolbox.

1. Drag the graphic(s) from the local drive and drop it to the AVEVA Drive.

2. If the graphic already exists in the drive then a pop up dialog appears, requesting to overwrite or skip the upload. Select the required option. Click **OK**.

3. The **Upload Content** dialog appears and displays the progress of the operation. Click **View Details** to view the progress.

4. Click **Close**.

5. All embedded symbols are also uploaded, and a structure similar to the local repository is created on the drive.

You can also upload folder(s) that contains multiple graphics.

Downloading Graphics to the Local Repository

You can download graphics to the local repository and use it on your applications. You must have the appropriate permissions to download graphics.

1. Drag the graphic(s) or folder from the AVEVA Drive and drop it to the local repository.

2. If the graphic already exists, the **Content Overwrite Option** dialog box appears, select the appropriate option.
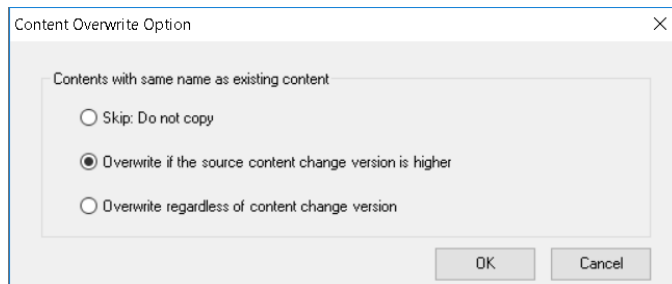
3. Before downloading the graphics, a security check verifies and lists any graphic containing any scripts or embedded controls. A security dialog appears requesting confirmation to download the content. Click **View Details** to view the list of graphic elements with details on scripts and controls. Click **Yes** to proceed.

4. The **Download Content** dialog box appears displaying the progress. The dialog box displays the client controls and scripts for each graphic. Click **Close**.

Any errors during downloading are also displayed.

5. The graphic(s) are downloaded to the local repository and embedded graphics are also copied.
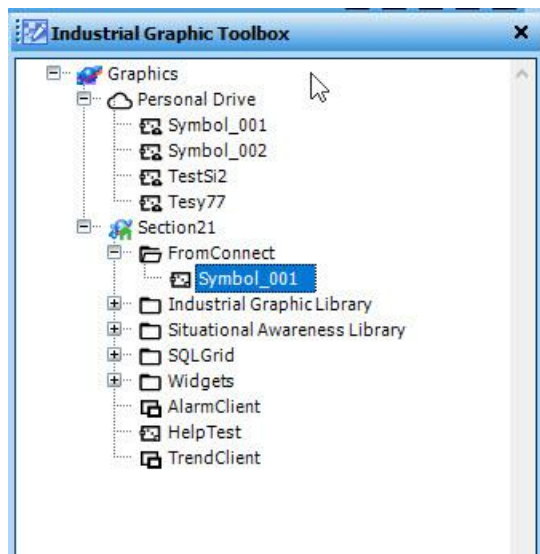
Managing Graphics on AVEVA Connect

You can also create, rename, update, duplicate and delete graphics directly on AVEVA drive.

1. Right-click the AVEVA Drive or any of the folders to create new graphic or toolset.

2. Double-click to edit graphics using the graphic editor.

3. Right-Click and select the corresponding options to rename, duplicate or delete the graphic.

4. The use is similar to managing local graphics. See the *Working with the Industrial Graphic Editor* on page 121 for more information.
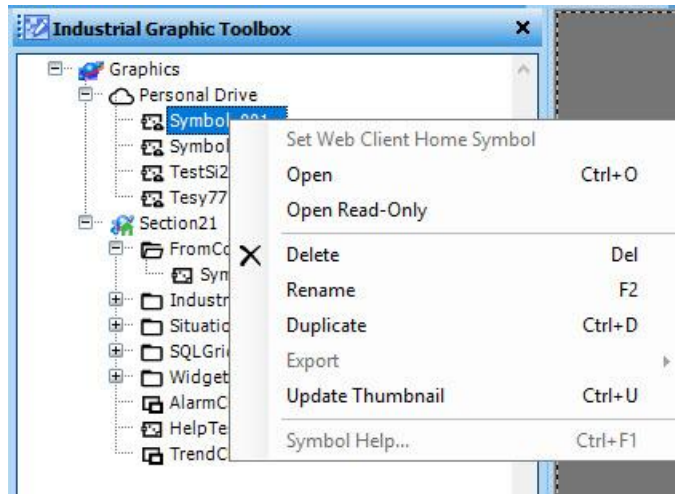
Managing Graphics with Multiple Users

Multiple users can use the graphics available on AVEVA Drive at a time. In such a scenario, the user that edits the graphic first, will cause the graphic to be locked preventing other users from making any edits. A red tick mark will appear against the graphic in the toolbox to inform the user that the graphic is checked out.

All other users can only view the graphic. A black tick mark will appear against the graphic name to signify that the graphic is being edited by another user.

After the first user saves and closes the graphic, the second user can check out and make changes to it. Once the graphic is saved, all users can view the graphic with the changes.

Index

$

.

A

B

G

grid

    show • 10, 12

    spacing • 10, 12

H

home, windows • 31

horizonal movement • 57

horizontal line button • 38

I

importing bitmaps • 49

Industrial Graphic, definition • 39

L

line button • 38

lines, select precision • 10

links

    action script • 74

    analog alarm status color, creating • 61

    analog expression color, creating • 61

    analog input • 71

    animate objects • 55

    animation • 20, 40

    animation types • 55

    blink • 64

    cannot make symbol • 40

    clipboard limitations • 48

    color • 61

    color link types • 61

    creating movement • 57

    creating rotation • 58

    cut, copy, or paste • 47

    disabling • 66

    discrete alarm status, creating • 61

    discrete color fill, creating • 61

    discrete input • 70

    discrete value touch links • 74

    display • 55