WONDERWARE INTOUCH BASIC TRAINING

INDEX

Course Introduction

The purpose of this course is to give students the InTouch knowledge necessary to develop Human-Machine Interface (HMI) systems for their specific plant floor environment.

The following Chapters are included:

Day 1

       Module 1 – Introduction
           Section 1 – Wonderware Factorysuite 2000 Introduction
- Wonderware's FactorySuite
- System Requirements
- Installing InTouch
- FactorySuite Licensing
- InTouch Components
- InTouch Application Manager

       Module 2 – Development Environment
           Section 1 – WindowMaker Elements
- The WindowMaker GUI
- The Application Explorer
- WindowMaker Windows
- WindowMaker Toolbars
- WindowMaker Ruler
- WindowMaker Status Bar
- WindowMaker Color Palette

           Section 2 – Using WindowMaker
- Graphic Objects
- Lines and Outlines
- Arranging Objects
- Text Objects
- Images and Bitmaps
- Wizards

       Module 3 – Tagname Dictionary
           Section 1 – Tagname Definition
- What is the Tagname Dictionary?
- Tagname Types
- Defining a New Tagname
- Defining Tagname Details
- Printing Tagname Details

Day 2

       Module 4 – Animation Links
           Section 1 – What are Animation Links?
- Object Type Dialog Box
- Animation Link Selection Dialog Box
- Applying Color Links
- Accessing the Tag Browser
- Accessing Tagname .Fields

Section 2 – Animating Objects
- Creating Touch Links
- Assigning Key Equivalents
- Creating Display Links

Section 3 – Editing Links and Tags
- Substituting Strings
- Substituting Tagnames
- Importing and Exporting Windows
- Converting Placeholder Tagnames
- Deleting Tagnames
- InTouch Cross Reference Utility

Module 5 – InTouch QuickScripts
Section 1 – QuickScript Types
- Application Scripts
- Window Scripts
- Key Scripts
- Touch Pushbutton Action Scripts
- Condition Scripts
- Data Change Scripts
- ActiveX Event Scripts
- Quick Functions

Section 2 – Script Functions
- Using the QuickScript Editor
- IF-THEN-ELSE and Comparison Scripts
- Built-in Script Functions
- Using Local Variables
- Importing QuickScripts
- Printing Scripts

Module 6 – Alarms and Events
Section 1 – Alarms
- Alarm Types
- Alarm Priorities
- Alarm Groups
- Defining Tagname Alarm Conditions
- Standard Alarm Display
- Configuring the Standard Alarm System
- Acknowledging Alarms
- Alarm .Fields

Section 2 – Events
- Event Types
- Defining Events

Section 3 – Alarm Logging
- Logging to SQL Database
- Alarm DB Logger utility

Section 4 – Alarm Purging
- Alarm Purge/Achieve utility

Day 3

Module 7 – Real-time and Historical Trends
Section 1 – Real-time Trends
- Creating a Real-time Trend
- Configuring a Real-time Trend
- Increasing Real-time Trending Performance

Section 2 – Historical Trends
- Logging Tagnames to the Historical Log File
- Configuring Historical Logging
- Creating a Historical Trend
- Configuring a Historical Trend
- Using Historical Trend Wizards
- Configuring a Historical Trend in Runtime
- Updating a Historical Trend in Runtime
- Historical Trend .Fields
- Historical QuickScript Functions
- HistData Utility
- Historical Trending and Daylight Savings

Module 8 – I/O Communication

Section 1 – Dynamic Data Exchange (DDE)
- Wonderware SuiteLink
- InTouch I/O Address
- InTouch Access Names
- Defining an I/O Item in InTouch

Section 2 – Wonderware I/O Servers
- Configuring I/O Servers

Section 3 – Troubleshooting I/O Servers
- Communication Between InTouch and the Server
- Monitoring Communication Status Between the I/O Server and the I/O Device
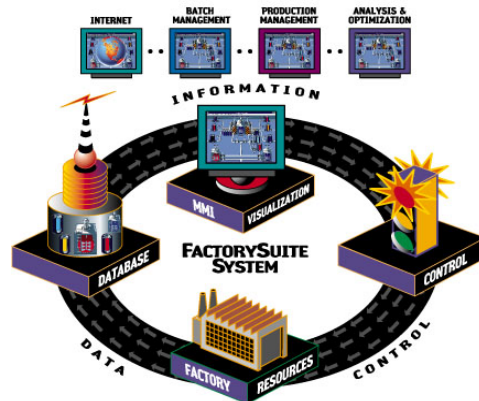- Monitoring Communication Status Between InTouch and the I/O Server

Upon completion of this course, students will be able to develop an HMI using basic InTouch elements for their plant floor environment.

Chapter : 1   Introduction

Section : 1
Introduction to Wonderware Factorysuite 2000

Wonderware's FactorySuite™

FactorySuite 2000 is the world's first integrated, component-based MMI – Manufacturing Management Information System. With FactorySuite 2000, you have access to all the information you need to run your factory. Today, it's not enough to buy just a database or just an HMI (Human Machine Interface). You need all the pieces – visualization, optimization and control, plant floor data collection, and data storage and analysis – to make your plant truly productive. FactorySuite 2000 gives you access to all your real-time plant data from which to make profitable business decisions.

Now, with the ability to deploy InTouch in a Terminal Services session you can achieve greater benefit from your existing capital investments and will have easier and more economical long-term maintenance of your applications. Then, taking advantage of our new SuiteVoyager browser/portal you can access all of your plant applications remotely, make changes in multiple applications simultaneously and respond to your customers' needs more rapidly.

FactorySuite 2000 runs on the Microsoft Windows NT4.0 operating system, and the HMI and clients also run on Windows 95/98, Windows 2000 and now Windows XP. FactorySuite 2000 sets a new standard for what you should expect in a comprehensive MMI System. We've included the following core components, providing basic system services:

- InTouch, the world's leading HMI for visualization
- InControl for Windows NT-based machine and process control
- IndustrialSQL Server, the first real-time relational database for the plant floor
- InTrackfor resource management
- InBatchfor flexible batch management

Plus all Wonderware I/O servers to connect FactorySuite 2000 to the data on the plant floor.

System Requirements
To run Intouch FactorySuite, we recommend the following hardware and software:
- Any IBM® compatible PC with a Pentium III processor or higher.
- At least 500MB of free hard disk space.

- At least 256MB of random-access memory (RAM).

- SVGA display adapter (2MB RAM recommended).
- Pointing device. For example, mouse, trackball, touch screen.
- Network adapter must be installed.
- Microsoft® Windows® 2000 Professional or Windows XP operating systems. Please refer to Compatibility Matrix for further details.

Installing InTouch

The Wonderware FactorySuite installation program is used to install InTouch.

FactorySuite Licensing

Your FactorySuite system license information can be viewed through the license viewing utility. This utility is launched from the WindowMaker Help **About** dialog box. To access the **About** dialog box, select the **About** command on the WindowMaker **Help** menu.For more information on licensing, see your online *FactorySuite System Administrator's Guide*.

InTouch Components

InTouch consists of three major programs, the InTouch Application Manager, WindowMaker and WindowViewer.

The InTouch Application Manager organizes the applications you create and establishes which application is the default application. The default application is opened when either WindowViewer of WindowMaker is launched, whether they are launched from the InTouch Application Manager or from the Windows **Start** menu. The Application Manager is also used to configure WindowViewer as an NT service, to configure Network Application Development (NAD) for client-based and server-based architectures, and to configure Dynamic Resolution Conversion (DRC) and/or distributed alarming. The DBDump and DBLoad database utilities are launched from the Application Manager.

WindowMaker is the development environment, where object-oriented graphics are used to create animated, touch-sensitive display windows. These display windows can be connected to industrial I/O systems and other Microsoft Windows applications.

WindowViewer is the runtime environment used to display the graphic windows created in WindowMaker. WindowViewer executes InTouch QuickScripts, performs historical data logging and reporting, processes alarm logging and reporting, and can function as a client and a server for both DDE and SuiteLink communication protocols.

InTouch Application Manager

You will use the InTouch Application Manager to create new applications, open existing applications in either WindowMaker or WindowViewer, delete applications, and run the InTouch DBDump and DBLoad Tagname Dictionary utility programs.

Running the InTouch Application Manager

1. Start the InTouch program (INTOUCH.EXE). The **InTouch Application Manager** dialog box appears:
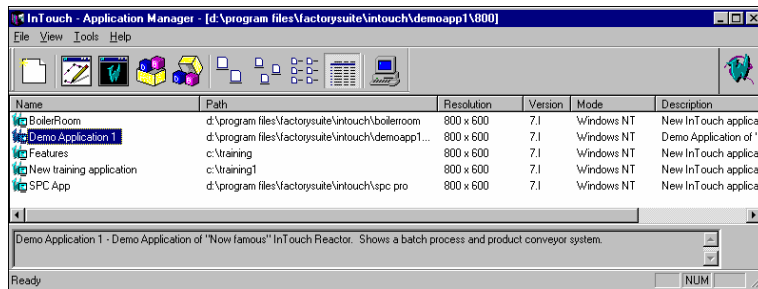
When you select an application in the list, its name and description will appear in the box at the bottom of the screen. If you right-click the description box, a menu appears displaying the commands that you can apply to the selected text.

2. To rename an application's icon, right-click the application in the list, and then click **Rename**. Enter the new name and click ENTER.

3. To delete an application's icon, right-click the application in the list and then click **Delete**. A message box appears asking you to confirm the deletion. Click **Yes** or **No**.
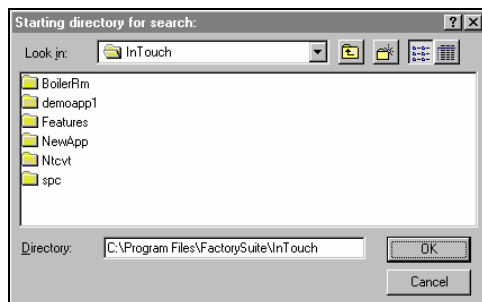
Locate the directory to search and click **OK**. The InTouch Application Manager will reappear displaying icons for all applications that were found in the selected directory.

Application Manager Tools

By default, when InTouch is initially run, the Application Manager's toolbar and status bar are displayed.

The following briefly describes each of the Application Manager's toolbar buttons:

Button   Description

Button	Description
	Executes the **New** command on the **File** menu to create a new application.
	Executes the **WindowMaker** command on the **File** menu to open the selected application in WindowMaker.
	Executes the **WindowViewer** command on the **File** menu to open the selected application in WindowViewer.

|	
	Executes the **DBLoad** command on the **File** menu to run the DBLoad utility. This utility is used to load a Tagname Dictionary input file.
	Executes the **DBDump** command on the **File** menu to run the DBDump utility program. This utility is used to extract an application's Tagname Dictionary.
	Executes the **Large Icons** command on the **View** menu to display large icons for the listed applications.
	Executes the **Small Icons** command on the **View** menu to display small icons for the listed applications.
	Executes the **List** command on the **View** menu to change the dialog box to the list view mode.
	Executes the **Details** command on the **View** menu to change the dialog box to the details view mode.
	Opens the **Node Properties** dialog box where the computer's properties are set when using Network Application Development (NAD), Dynamic Resolution Conversion (DRC) and/or distributed alarming.

Creating a New Application

1. Click the **New** tool in the toolbar or click **File/New** on the **File** menu. The **Create New Application** wizard appears.

2. Click **Next**. A second **Create New Application** wizard appears.

3. In the input box, enter the path to the directory in which you want your application to be created or click **Browse** to locate the directory.

4. Click **Next**.

5. In the **Name** box, enter a unique name for the new application's icon that appears when the application is listed in the **InTouch Application Manager** window.

6. In the **Description** box, enter a description of the application.

7. Click **Finish**. The **InTouch - Application Manager** reappears displaying an icon with the name you specified for the new application. For example:
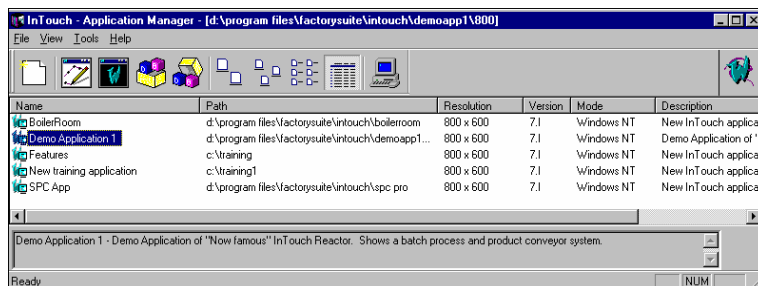
8. To open an application, click the right mouse button as you select it. Then click the name of the program you want to use for the application in the **File** menu or select the application in the list. Click the **WindowMaker** tool in the toolbar. (WindowViewer cannot be executed for a new application.)

Chapter : 2   Development Environment

Section : 1

WindowMaker Elements

WindowMaker is the development environment for InTouch. InTouch's Application Explorer provides you with a powerful, graphical method for navigating and configuring your InTouch applications. It provides easy access to WindowMaker's most commonly used commands and functions such as; all windows commands, all configuration commands, and all InTouch QuickScript editors. Additionally, the Application Explorer will display all installed add-on programs such as SQL Access Manager, SPC Pro and Recipe Manager. It also provides you with a customizable application launcher.

The WindowMaker GUI

The WindowMaker graphical user interface (GUI) adheres to Windows GUI standards. WindowMaker supports right-click mouse support, floating and docking toolbars, pull down menus, context-sensitive help, and so on. Right-click menus provide quick access to frequently used commands and a customizable color palette that provides 16.7 million color support. (The color support is limited only by your video card capability.)

The WindowMaker development environment is configurable. By default, when you initially open WindowMaker, most of the available elements are automatically displayed including all toolbars, the Application Explorer and the status bar. However, you can show or hide any or all of these elements and you can move the toolbars and the Application Explorer to any location within the WindowMaker window. You can also display the optional ruler and turn the visible grid on and off.

The Application Explorer

WindowMaker's Application Explorer is a hierarchical graphical view of your application. It displays the items you have configured in your application and provides easy access to them. It also provides quick access to many of WindowMaker's most commonly used commands and functions.

To show or hide the Application Explorer, click on the **Application Explorer** tool on the **View** toolbar or the **Application Explorer** command in the **View**.

Like all WindowMaker toolbars, the Application Explorer can be "docked" to any edge of the WindowMaker window or, "floated" anywhere within the WindowMaker window. When the Application Explorer is docked to an edge of the WindowMaker window, it automatically sizes itself accordingly and, if required, scroll bars are displayed. When the Application Explorer is floated within the WindowMaker window, its title bar is displayed and its size can be changed. For example:
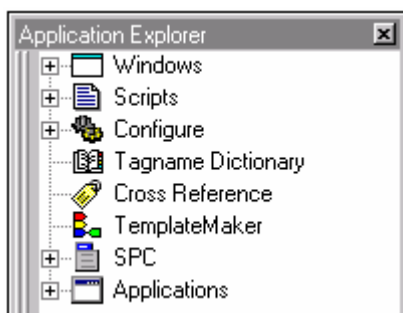
Right-click the Application Explorer's title bar to display the following options:
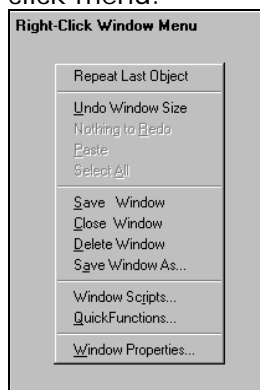**Move/Close.**
You can expand or collapse the groups listed in the Application Explorer hierarchical graphical view. For example, if you double-click on a group, the view will expand and display the group's members. If you double-click on a member, it will open that member. For example, in the **Windows** group, if you double-click on a member window name, the window will open. If you double-click on **Tagname Dictionary**, the **Tagname Dictionary** dialog box will appear, etc.

WindowMaker Windows

Your InTouch application will, more than likely, be comprised of numerous windows that you create to hold your graphics and text objects. When you create a new window in WindowMaker, you will be required to define certain properties for that window such as, its background color, title, screen position and so on. You can also create QuickScripts that execute based upon whether the window is opening, showing or closing.
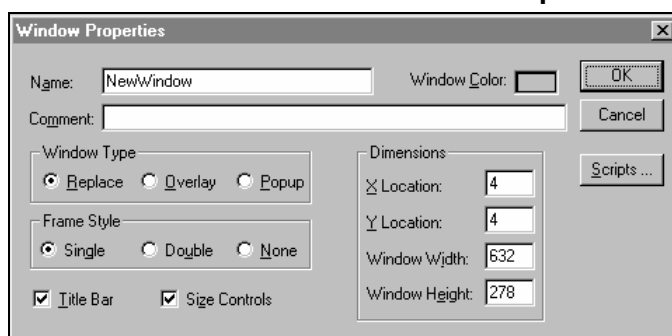
Right-Click Window Menus
To quickly access the various commands that can be applied to a window, right-click a blank area of the open window and then click the appropriate command on the right-click menu.

Right-Click Window Menu

- Repeat Last Object
- Undo Window Size
- Nothing to Redo
- Paste
- Select All
- Save Window
- Close Window
- Delete Window
- Save Window As...
- Window Scripts...
- QuickFunctions...
- Window Properties...

Creating a New Window

1. Click the New Window tool [ ] on the **General Toolbar** or click on the **File/New Window** command. The **Window Properties** dialog box appears:

Window Properties

Name: NewWindow   Window Color: [ ]   OK
Comment: [ ]   Cancel

Window Type
○ Replace ○ Overlay ○ Popup

Frame Style
● Single ○ Double ○ None

☑ Title Bar   ☑ Size Controls

Dimensions
X Location: 4
Y Location: 4
Window Width: 632
Window Height: 278

Scripts ...

2. In the **Name** box, enter the name to appear in the new window's title bar. The name can be up to 32 characters long. It can include embedded spaces, punctuation marks and any other character on the keyboard except quotation marks (").

3. In the **Comment** box, enter any miscellaneous comments that you want associated with the window (optional). This information is for documentation purposes only and is not used by the application.

4. Click the **Window Color** box to select the background color for the window. The color palette will appear:

If no change is desired, click on the current color selection or press the ESC key to close the palette.

5. Select a window background color.

6. Select a **Window Type**. There are three options:

Replace	Automatically closes any window(s) it intersects when it appears on the screen, including popup and other replace type windows.
Overlay	Appears on top of currently displayed window(s) and can be larger than the window(s) it is overlaying. When an overlay window is closed, any window(s) that were hidden behind it will reappear. Clicking on any visible portion of a window behind an overlay window will bring that window to the foreground as the active window.
Popup	Similar to an overlay window,but always stays on top of all other open windows (even if another window is clicked). Popup windows usually require a response from the user in order to be removed.

7. Select the **Frame Style** for the window. There are three styles:

Single	3-D bordered window that can have a title bar and **Size Controls**.
Double	3-D bordered window that has no title bar and cannot be sized without **Size Controls**.
None	A window without a border that cannot be sized without **Size Controls**. (With **Size Controls** it becomes a 3-D bordered window that can be sized.)

8. Select **Title Bar** to allow the window to have a title bar. The title bar is also used to move the window by clicking and dragging.

9. Select **Size Controls** to allow the user to resize the window in WindowViewer.

10. In the **Dimensions** group, enter the pixel location for each of the window's coordinates:

X Location	The number of pixels between the left edge of the WindowMaker design area and the left edge of the window being defined.
Y Location	The number of pixels between the top edge of the WindowMaker design area and the top edge of the window being defined.
Window Width	The window's width in pixels.
Window Height	The window's height in pixels.

11. Click **Scripts** to access the **Window Script** editor. There are three types of scripts that can be applied to a window:

On Show	Executes one time when the window is initially shown.
While Showing	Executes continuously at the specified frequency while the window is showing.
On Hide	Executes one time when the window is hidden.

Opening, Saving, Closing, Deleting and Duplicating Windows

Opening Windows
When developing your application, you can open as many windows as your computer memory will support.

1. Click the Open Window tool 📂 on the **General Toolbar** or click on the **File/Open Window** command. The **Windows to Open** dialog box appears listing the names of all windows in your application.

2. Click the check box next to the name of the window(s) to open.

3. Click **OK** to close the dialog box and open the selected window(s).

Saving Windows
Once you have created a window, you will need to save it before closing or exiting the application. All graphics, QuickScripts, properties, etc. associated with the window are also saved.

1. Click the Save Window tool 💾 on the **General Toolbar** or click on the **File/Save Window** command. The **Windows to Save** dialog box appears listing the names of all windows that need to be saved.

2. Click the check box next to the name of the window(s) to save.

3. Click **OK** to close the dialog box and save the selected window(s).

Closing Windows
If you attempt to close a window that has been modified since it was last saved, you will be prompted to save your changes before WindowMaker will close the window.

1. Click the Close Window tool 📁 on the **General Toolbar** or the **File/Close Window** command. The **Windows to Close** dialog box appears listing the names of all currently open windows.

2. Click the check box next to the name of the window(s) to close.

3. Click **OK** to close the dialog box and close the selected window(s).

Deleting Windows
Deleted windows cannot be restored unless you have backed them up. You will be prompted to confirm the deletion of each window name you select.

1. Click on the **File/Delete Window** command. The **Windows to Delete** dialog box appears listing the names of all currently open windows.

2. Click the check box next to the name of the window(s) to delete.

3. Click **OK** to close the dialog box and delete the selected window(s).

Duplicating Windows
To create a duplicate copy of an existing window, the window to be duplicated must be open.

1. Click on the **File/Save Window As** command. The **Window to save under new name** dialog box appears, listing the names of all currently open windows.

2. Click the check box next to the name of window that you want to duplicate. (Only one window name can be selected.) The **Save "*Window Name*" As** dialog box appears:
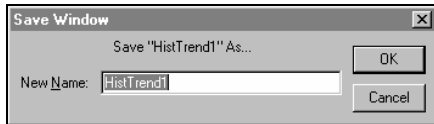
3. In the **New Name** box, enter a valid name for the new window.

4. Click **OK** to close the dialog box and create the duplicate window.

Creating a Window to Hide the Title and Menu Bars

The WindowMaker design area is the entire area below the title and menu bars and within the window frame. The design area becomes the display area in WindowViewer. The specific location X=0 and Y=0 is always the upper left corner just below the title and menu Bars.

The title and menu bars are each 20 pixels high and are above the design area. For example, if WindowMaker is maximized and you are using a 1024x768 video display, the visible design area would be 1024x730 (768 less 19 pixels for the title and 19 pixels for the menu bar equals 730 pixels in the visible design area). If WindowViewer is configured to show its title bar and menu bar, the display area in WindowViewer will fill the screen below the title bar and menu bar exactly as seen in WindowMaker.

To take advantage of the additional space used by the title and menu bars, you can design an application with the title bar and menu bars hidden. When the title bar and menu bar are hidden, the upper left corner of the window references a new position on the screen. This increases the visible display area and provides you with more display area. If you configure WindowViewer in this manner, all of your windows will automatically appear to move up and a gap will appear at the bottom of the window. To fill this gap, you need to increase the window height by setting the Y location of the window to a negative value. This places the window under the title bar and menu bar in WindowMaker and on top of them in WindowViewer.

You can use this technique with a popup window to hide the title bar and menu bar in WindowViewer. You can also create a touch link pushbutton or QuickScript to hide this popup window when the title bar and menu bar need to appear to the user. In addition, by applying security and a password, you can restrict certain users from hiding the window and gaining access to the menus (including the ability to exit WindowViewer).

WindowMaker Toolbars

The tools on the WindowMaker toolbars are grouped by common functionality. For example, the **Arrange** toolbar contains tools that you can use to quickly apply most of the commands found on the **Arrange** menu.

If you rest the cursor on a tool, a tool tips box will appear displaying the name of the tool:

General Toolbar

The **General** toolbar is grouped with tools that execute most of the window commands found on the **File** menu and the Microsoft Windows Clipboard tools found on the **Edit** menu:

Button Description

	Executes the **New Window** command on the **File** menu and opens the **Windows Properties** dialog box to create a new window.

	Executes the **Open Window** command on the **File** menu to open the **Windows to Open** dialog box listing the names of existing windows that you can select to open.
	Executes the **Close Window** command on the **File** menu to open the **Windows to Close** dialog box listing the names of all currently open windows that you can select to close.
	Executes the **Save Window** command on the **File** menu to open the **Windows to Save** dialog box listing the names of all currently open windows that have been modified since they were last saved.
	Automatically saves all currently open windows that have been modified since they were last saved. This tool does not ask for confirmation on a per window basis. It saves all modified windows automatically.
	Executes the **Cut** command on the **Edit** menu to cut the currently selected objects(s) from the window and copies them to the Windows Clipboard.
	Executes the **Duplicate** command on the **Edit** menu to duplicate the currently selected object(s) in the window.
	Executes the **Copy** command on the **Edit** menu to copy the currently selected objects(s) and copies them to the Windows Clipboard. (Copied objects are not erased from the window.)
	Executes the **Paste** command on the **Edit** menu to paste any object that has been cut or copied to the Windows Clipboard. (The cursor changes to the paste mode. Click in the window to paste the copied or cut object.)
	Executes the **Undo** command on the **Edit** menu to reverse (undo) the last action or command applied to an object.
	Executes the **Redo** command on the **Edit** menu to reverse (redo) the last undo action or command applied to an object.
	Executes the **Print** command on the **File** menu to open the **WindowMaker Printout** dialog box used to print database and window information and QuickScripts.

Wizards/ActiveX Toolbar

The **Wizards/ActiveX** toolbar, by default, only contains the wizard tool used to access the wizard **Selection Dialog** box. However, you can add any installed wizard or ActiveX control to the toolbar:

Button Description

	Displays the **Wizard Selection** dialog box used for selecting wizards to paste into your windows.

Format Toolbar

The **Format** toolbar is grouped with tools that execute most of the text object formatting commands found on the **Text** menu. It also contains the tools used to access the color palette to select line, text, fill, window background and transparent object color.

Button   Description

	Executes the **Font** command on the **Text** menu to open the **Font** dialog box used to select the font, its style and size.
	Executes the **Bold** command on the **Text** menu to apply **bold** styling to single or multiple text string selections and numeric value fields.
	Executes the **Italic** command on the **Text** menu to apply *italic* styling to single or multiple text string selections and numeric value fields.
	Executes the **Underline** command on the **Text** menu to apply <u>underline</u> styling to single or multiple text string selections and numeric value fields.

Button	Description
	Executes the **Reduce Font** command on the **Text** menu to reduce the point size of any font. This command can be applied by selecting the text string(s) and clicking on the tool.
	Executes the **Enlarge Font** command on the **Text** menu to enlarge point size of any font. This command can be applied by selecting the text string(s) and clicking on the tool.
	Executes the **Left Justified** command on the **Text** menu to align the left edge of single or multiple text string selections and numeric value fields.
	Executes the **Centered** command on the **Text** menu to center single or multiple text string selections and numeric value fields.
	Executes the **Right Justified** command on the **Text** menu to align the right edge of single or multiple text string selections and numeric value fields.
	Opens the color palette used to select the color for a line object or an object's outline.
	Opens the color palette used to select an object's fill color.
	Opens the color palette used to select the color for a text object.
	Opens the color palette to select a window's background color.
	Opens the color palette to select a transparent color for a bitmap object.

Draw Object Toolbar

The **Draw Object** toolbar is grouped with all the tools used to draw both simple graphic objects (rectangles, ellipses, lines or text objects) and complex objects (real-time trends, historical trends, bitmaps and 3-dimensional buttons with labels):

Button   Description

Button	Description
	Selector mode used to select objects in the window.
	Rectangle tool used to draw rectangles or squares.
	Rounded rectangle tool used to draw rectangles or squares with rounded corners.
	Ellipse tool used to draw ellipses or circles.
	Line tool used to draw lines at any angle.
	Line tool used to draw horizontal or vertical lines.
	Line tool used to draw polylines.
	Text tool used to enter text objects.
	Bitmap tool used to draw a bitmap container for pasting a bitmap directly from the Windows Clipboard or one of the following file types: .BMP, .JPG, .PCX or .TGA.
	Real time trend tool used to draw real time trend objects.
	Historical trend tool used to draw historical trend objects.
	Button tool used to draw a 3-dimensional button with a label.

View Toolbar

The **View** toolbar is grouped with tools that execute most of the window commands found on the **View** menu. These commands are used to control the state of the WindowMaker window.

Button   Description

Turns the **Application Explorer** command on the **View** menu on and off to show/hide the Application Explorer.

Toggles the **Hide All** command on the **View** menu on and off to hide/show all docked toolbars.

When the hide all mode is active, the overall size of WindowMaker remains the same. To return to normal mode, click the Hide/Restore All tool on the floating **View Toolbar** or click on the **View/Hide All** command.

In the hide all mode, all floating toolbars remain visible and the **View Toolbar** automatically floats on top of WindowMaker. If any of the floating toolbars are docked in the hide all mode, the mode is automatically terminated.

Toggles **Full Screen** command on the **View** menu on and off to switch the display mode from normal view to full screen.

To return to normal mode, click the Full Screen tool on the floating **View Toolbar** or click on the **View/Full Screen** command.

In the full screen mode, all WindowMaker program elements are hidden except any open windows and floating toolbars. The **View Toolbar** automatically floats on top of WindowMaker.

In the full screen mode, the coordinates of the client area will remain the same. For example, the top left is 0,0. The full screen mode automatically sets the coordinates after it maximizes the client area, hides the Title Bar and menu bar and adjusts the client area to mimic View's full screen mode.

Toggles the **Snap to Grid** command on the **Arrange** menu on and off to show/hide the visible grid used to align objects. It works with the **Arrange/Snap to Grid** command.

If the **Snap to Grid** option in the **WindowMaker Properties** dialog box is not selected, this tool will have no effect.

Turns the **Ruler** command on the **View** menu on and off to show/hide the ruler.

Arrange Toolbar

The **Arrange** toolbar is grouped with tools that execute most of the object arranging commands found on the **Arrange** menu:

Button   Description

Executes the **Align Left** command on the **Arrange/Align** submenu. Aligns the left edge of all selected objects with the left edge of the left most selected object.

Executes the **Align Center** command on the **Arrange/Align** submenu. Aligns the vertical centerline of all selected objects with the centerline of the group of objects selected.

Executes the **Align Right** command on the **Arrange/Align** submenu. Aligns the right edge of all selected objects with the right edge of the right most selected object.

Executes the **Align Top** command on the **Arrange/Align** submenu. Aligns the top edge of all selected objects with the top edge of the top most selected object.

Executes the **Align Middle** command on the **Arrange/Align** submenu. Aligns the middle of all selected objects with the middle of the group of objects.

Executes the **Align Bottom** command on the **Arrange/Align** submenu. Aligns the bottom edge of all selected objects with the bottom edge of the lowest selected object.

Executes the **Align Centerpoints** command on the **Arrange/Align** submenu. Aligns the centerpoint of all the selected objects with the centerpoint of the group of selected objects.

Executes the **Send to Back** command on the **Arrange** menu to place all selected objects behind all objects that are not selected.

Executes the **Bring to Front** command on the **Arrange** menu to place all selected objects in front of all objects that are not selected.

Executes the **Space Horizontal** command on the **Arrange** menu to evenly space all selected objects horizontally between the left most and right most selected objects.

Executes the **Space Vertical** command on the **Arrange** menu to evenly space all selected objects vertically between the top most and bottom most selected objects.

Executes the **Make Symbol** command on the **Arrange** menu to combine multiple objects into a single unit called a symbol.

Executes the **Break Symbol** command on the **Arrange** menu to break a symbol into its individual components.

Executes the **Make Cell** command on the **Arrange** menu to combine multiple selected objects into a single unit called a cell. When combining cells, each cell will be retained. When the combined cell is broken, the original cells are restored.

Executes the **Break Cell** command on the **Arrange** menu to break a selected cell. When combining cells, each cell will be retained. When the combined cell is broken, the original cells are restored.

Executes the **Rotate Clockwise** command on the **Arrange** menu to rotate selected objects clockwise 90 degrees.

Executes the **Rotate CounterClockwise** command on the **Arrange** menu to rotate selected objects counter clockwise 90 degrees.

Executes the **Flip Horizontal** command on the **Arrange** menu to flip selected objects horizontally.

Executes the **Flip Vertical** command on the **Arrange** menu to flip selected objects vertically.

Executes the **Reshape Object** command on the **Edit** menu to reshape a polygon or polyline.

WindowMaker Ruler

The WindowMaker ruler can be used to do precision alignment of the objects in your windows, and when determining the movement of an object during an animation sequence.

The small tick marks are spaced 5 pixels apart. The medium tick marks are spaced 10 pixels apart. The numbered large tick marks are spaced 50 pixels apart.

The ruler's 10 and 50 pixel spacing increments are equivalent to the distance in pixels that a selected object is moved when you hold down the SHIFT or CTRL key and press an up, down, right or left arrow key.

For example, to move an object 10 pixels at a time, hold down the SHIFT key while pressing an arrow key. To move an object 50 pixels at a time, hold down the CTRL key while pressing an arrow key. Use just the arrow key to move the object 1 pixel at a time.

These features can be useful when making fine alignment and location adjustments.

WindowMaker Status Bar

When an object is selected in a window, the WindowMaker status bar displays the object's upper left X and Y pixel coordinates and the object's pixel height and width:
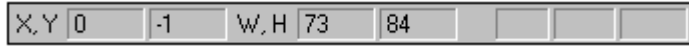
When multiple objects are selected, the status bar displays the coordinate for the left edge of the left most object (X) and the coordinate for the top edge of the top most object (Y). The width and height are also shown for the entire group:
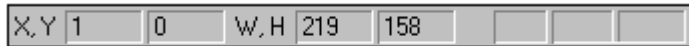
When a blank area of a window is clicked, the status bar displays the X and Y coordinates for the current location of the cursor in the window:

WindowMaker Color Palette

The WindowMaker palette provides 16.7 million color support. (The color support is limited only by your video card capability.) By default, the palette offers a wide range of color selections. However, you can create your own custom palettes. Your custom palettes can be loaded into and exported from the WindowMaker color palette.

Standard Color Palette

The WindowMaker color palette is used to apply color to static and dynamic properties of lines, rectangles, round rectangles, ellipses, polylines, polygons and text. It is also used to select a window background color and the transparent color for bitmaps (allowing objects to be viewed behind bitmaps).

The color palette appears when a colored square in a dialog box is clicked or when one of the color tools is used to apply line, fill or text color to a selected object.

1. To select a standard color, click a color in the **Standard Palette** section. (The color palette will close and the color you selected will be applied.)

2. To select one of InTouch's 32 classic colors (palette colors prior to InTouch Version 7.0), click the >> in the right corner. The **Classic Colors** will appear:

Custom Color Palette

The WindowMaker color palette allows you to define custom colors and add them to your palette. Palettes created in other windows applications can be imported and added to

the standard palette. You can also export custom palettes to other windows applications.

Creating a Custom Color

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. The following menu appears:

3. Click **Edit Custom Color**. The **Add a Color** dialog box appears:
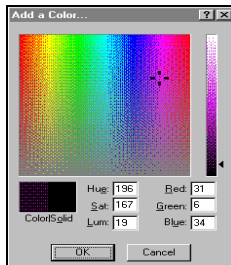
4. Click anywhere in the matrix and then use the slider at the right of the dialog box to adjust the color's attributes.

5. Click **OK**. The color you selected will be added to the square that you originally clicked in the color palette.

Selecting a Custom Color with the Blotter 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. The following menu appears:
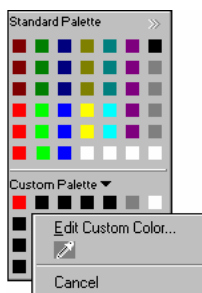
3. Click the blotter tool and then click the color to add to the **Custom Palette** section of the color palette. You can select any color anywhere within the WindowMaker window or outside of WindowMaker. This feature is primarily intended to be used when creating transparent bitmaps.

Section 2

Using WindowMaker

This section covers the creation and manipulation of graphic objects, lines and outlines, text objects, images and bitmaps. It also includes wizard installation/configuration, ActiveX controls, and MS Paint.

Graphic Objects

Once you have created a new window, it can be populated with graphic objects. WindowMaker provides you with numerous tools for editing and arranging the various graphic objects.

Selecting and Sizing Objects

When an object is clicked on, several small boxes will surround it. These boxes are called "handles" and are used to resize and/or reshape the object.

The notion of "selected" is a key concept of WindowMaker graphics editing. The presence of "handles" around an object indicates that it is "selected." Clicking directly on an object selects it. Clicking on a blank area of the window deselects any currently selected object(s) in that window.

In general, any command that you execute is applied to all selected objects (assuming that the command is valid for the object).

Selecting All Objects in a Window

To select all objects in the active window, click the **Edit/Select All** command, right-click a blank area of the open window and then click **Select All**, or press F2.

Selecting Multiple Objects

To select multiple objects, select your first object. Then hold down the SHIFT key and click the other objects to be selected. To deselect a specific object from a group of selected objects, while all objects are selected, hold down the SHIFT key and click the object to deselect.

Selecting a Group of Objects

Move the cursor to a blank area of your window. Click the left mouse button and drag the mouse. A dotted selection rectangle with a small hand cursor will appear. Drag the mouse until you have completely surrounded all of the objects that you want to select. Release the mouse button. All of the objects that were completely within the rectangle will be selected.

Deselecting a Group of Selected Objects

If you hold down the SHIFT key while drawing a selection rectangle, all enclosed selected objects will become deselected. This technique may also be used to start a selection rectangle on top of another object.

If you hold down the SHIFT key while clicking the left mouse button, the object under the cursor will not be dragged when the cursor is moved. Instead, a selection rectangle will be drawn.

Duplicating Objects

1. Select the object(s) to duplicate.

2. Click the Duplicate Selection(s) tool 🔲 on the **General Toolbar** or click on the **Edit/Duplicate** command.

Cutting Objects to the Windows Clipboard

1. Select the object(s) to cut.

SSM Infotech Solution Pvt. Ltd.

2. Click the Cut to Clipboard tool ✂ on the **General Toolbar** or Click on the **Edit/Cut** command.

Copying Objects to the Windows Clipboard

1. Select the object(s) to copy.

2. Click the Copy to Clipboard tool 📋 on the **General Toolbar** or click on the **Edit/Copy** command.

Pasting Objects from the Windows Clipboard

1. Copy or cut the object.

2. Click the Paste from Clipboard tool 📋 on the **General Toolbar** or click on the **Edit/Paste** command. The cursor will change to a corner symbol.

3. Hold down the left mouse button. A dotted line rectangle the size of the copied object will appear. Drag the rectangle to the location in the window where you want to paste the object.

4. Release the mouse button to paste the object.

Cutting, Copying, Pasting and Clearing Object Links
WindowMaker's link paste buffer is a temporary storage area, storing links that can be cut or copied from an object. (The buffer only stores the links for your most recent cut or copy action.) The links stored in the link paste buffer can be pasted to any object or symbol. If multiple objects are selected, the links are pasted to each individual object.

If a pasted link has no apparent value to the object (e.g., a line color link on a text object), the link is not pasted.

1. Select the object to apply the links command to.

2. Click on the **Edit/Links** command and then click **Cut**, **Copy**, **Paste** or **Clear**.

Deleting Objects

1. Select the object(s) to delete.

2. Click on the **Edit/Erase** command or select the object and press the DEL key.

Lines and Outlines

The style and width of a line object can be changed, including the outlines around ellipses, rectangles, polygons, bitmaps or images. A line style or width change can be applied to a single selected object or multiple selected objects.

The **Line** menu is divided into two sections. The top section contains line widths and the bottom section contains line styles:
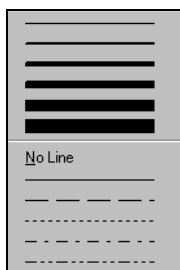
Applying a Line Command
Select the object and then click the desired line style or width on the **Line** menu.

Removing an Object's Outline
Select the object and then click **No Line** on the **Line** menu. The object's outline will be removed.

Arranging Objects

WindowMaker provides numerous tools to arrange objects in your windows. The **Arrange Toolbar** contains tools that can be used to quickly apply most of the commands found on the **Arrange** menu:
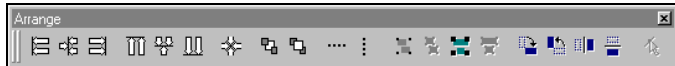
Aligning Objects
Objects can be aligned by their left or right edges, centers, centerpoints, tops, middles or bottoms.

1. Select the objects(s).

2. Click on the **Arrange/Align** command and then click the appropriate align command in the submenu (or click on the appropriate tool on the **Arrange Toolbar**). Object(s) will be aligned according to your selection:

Layering Objects
Objects can be layered in your window by positioning them in front or behind each other.

Positioning an Object Behind Another Object

1. Select the objects(s).

2. Click the Send to Back tool on the **Arrange Toolbar** or click on the **Arrange/Send to Back** command. The selected object(s) will be redrawn behind the object(s) not selected in your window:

Positioning an Object in Front of Another Object

1. Select the objects(s).

2. Click the Bring to Front tool on the **Arrange Toolbar** or click on the **Arrange/Bring to Front** command. The selected object(s) will be redrawn in front of the object(s) not selected in your window:

Rotating Objects
Most objects (including bitmaps, JPEG, PCX, TGA images and text objects) can be rotated in WindowMaker utilizing either **MS Paint** or the **Arrange Toolbar**. Objects can be rotated clockwise or counter-clockwise 360 degrees in 90 degree increments (90 degrees, 180 degrees, 270 degrees, and 360 degrees). Any links attached to the object are rotated with the object. Cells cannot be rotated, while symbols can be rotated.

Rotating Objects and Text using the Arrange Toolbar

1. Select the object(s) or text.

2. Click on either the Rotate Clockwise tool or the Rotate CounterClockwise tool on the **Arrange Toolbar**. Or, click on the **Arrange/Rotate Clockwise** or

CounterClockwise command. The selected object(s) will be rotated 90 degrees in the direction you choose.

Rotating Objects using MS Paint

Right-click on the bitmap and select **Rotate/Flip**.

Mirroring Objects

Most WindowMaker objects can be flipped horizontally or vertically (including bitmaps, JPEG, PCX, and TGA images - text cannot be flipped, but can be rotated). When an object is flipped, it is transformed into its horizontal or vertical mirror image. Any links attached to the object are flipped with the object.

1. Select the object(s).

2. Click the Flip Horizontal tool ▯ or the Flip Vertical tool ▭ on the **Arrange Toolbar** or click on the **Arrange/Flip Horizontal** (**Flip Vertical**) command. The selected object(s) will flip.

Creating Cells and Symbols

Multiple objects can be combined into two different types of single units: cells and symbols. Multiple cells can be combined into a single cell. Cells are objects that maintain a fixed spatial relationship between individual graphic elements. The individual components within a cell can be animated (with the exception of another cell). Cells cannot be resized, nor can animation links be attached to cells. However, animation links can be attached to symbols and symbols can be included in a cell. All animation links associated with a symbol or an object(s) within a cell are unchanged. The attributes of objects such as text, font, line width, radius and relative positions within a cell cannot be sized until the cell is broken into its individual components.

Symbol   **Cell**
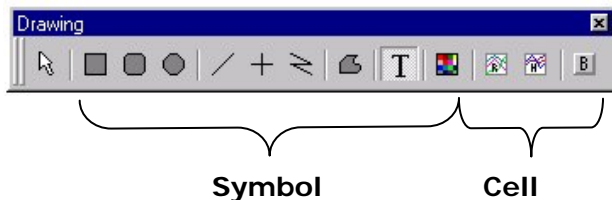
-Group of individual object with   -Group of individual objects
 one an animation link   with individual animation
-Create a symbol, then assign   objects
 the animation link
-Only contains 'Simple Objects'   -Can contain any object
-Symbols can be resized and have   including bitmaps, trends,
 normal sized handles   buttons, and wizards
                                        -Cells cannot be resized
                                         and have large handles

When combining cells, each cell is retained. When the combined cell is broken, the original cells are restored.
A symbol's function is to create one object from a number of simple objects and/or multiple symbols. If one of the objects selected has animation links attached to it, the links will be attached to the new symbol. (If the link paste buffer has links in it, you will be asked if you want to paste the links on the new symbol.)

Creating a Cell or Symbol

1. Select the objects that to include in the cell or symbol.

2. Click on the Make Cell tool [icon] or the Make Symbol tool [icon] on the **Arrange Toolbar** or click on the **Arrange/Make Cell** (**Make Symbol**) command.

Breaking a Cell or Symbol

1. Select the symbol or cell.

2. Click on the Break Cell tool [icon] or the Break Symbol tool [icon] on the **Arrange Toolbar** or click on the **Arrange/Break Cell** (**Break Symbol**) command.

Snapping Objects to the Grid

When you are arranging objects in your windows, turning on the grid will cause your graphic to snap at the upper left pixel interval on the grid. If you select multiple objects, the snapping will be applied to the upper left corner of the first object selected in the group.

Configuring the Grid

1. Click on the **Special/Configure/WindowMaker** command. The **WindowMaker Properties** dialog box appears.

2. In the **Spacing** box, enter the number of pixels that you want spaced between the snap to grid's coordinates.

3. Select **Show Grid** if you want a visible grid in your windows when you turn on WindowMaker's "snap to grid" functionality.

Text Objects

The font, font style, font size, justification and rotation of any selected text object can be changed in WindowMaker. Text objects can also be rotated 360 degrees by 90 degree increments (90 degrees, 180 degrees, 270 degrees and 360 degrees).

The **Format Toolbar** contains tools used to quickly apply most of the commands found on the **Text** menu to selected objects:
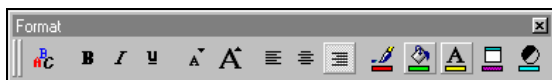
Formatting Text Objects

All WindowMaker text commands operate on single or multiple text string selections and numeric value fields. If no text object is selected when a command on the **Text** menu is executed, the command is automatically applied to the respective text tool's default setting on the **Format Toolbar** and the default setting of the Text tool on the **Draw Object Toolbar**.

The text justification attribute settings are particularly important for text objects used for outputting dynamic values. The justification determines how fields of varying length will be displayed in runtime.

For example, if you are displaying a numeric value at the end of a text string that is centered or is right justified, the entire text string (including the value) will be centered again or justified again each time there is a change in the number of displayed digits.

Displaying Numeric Values

Text objects are also used to display static or dynamic numeric values. By attaching a **Touch Links User Inputs** - **Analog** or **Value Display - Analog** animation link to a text object you can display the value of an analog (integer or real) tagname.

To determine the display format of the analog value, the following four characters are used:

0 - zero
- number or pound sign
, - comma
. - period or decimal point

The following illustrates field formatting for analog values:

Displays any whole number, e.g., **1234** would display as **1234**

0.0 Forces one leading zero and one decimal place,
e.g., **.1** would display as **0.1** and **77.1** would display as **77.1**

00000 Forces leading zeros as required, e.g., **123** would
display as **00123** and **1234** would display as **01234**

#,##0.0 Inserts comma and leading zero if required and one decimal place, e.g.,
1234.56 would be displayed as **1,234.6**

0,000.0 Forces comma, leading zeros and one decimal place,
e.g., **12.3** would display as **0,012.3**

Creating a Text Object

1. Click the Text tool $\boxed{T}$ in the **Draw Object Toolbar**.

2. Enter the text string in your window.

Displaying a Numeric Value in a Text String

1. Click the Text tool and then enter a text object in the window using one of the previously described valid numeric formats (e.g., **Current Seconds = #**).

2. Select the object and then click on the **Special/Animation Links** command or double-click the text object. The animation links selection dialog box appears.

3. In the **Value Display** section, click **Analog**. The **Output -> Analog Expression** dialog box appears:
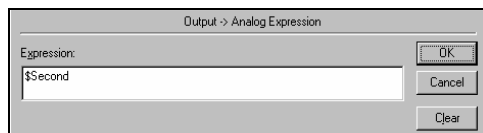
4. In the **Expression** box, enter an analog tagname or expression. (In this example, the system tagname **$Second** is being used.)Click **OK**.

5. Click the **Runtime** fast switch in the upper right hand corner of the menu bar (or use the short cut keys ALT + !) to switch to WindowViewer.

6. If you used this example, you will see the current system seconds (a value between 0-59) displayed in place of the pound (#) sign in the text string.

7. Click the **Development** fast switch in the upper right hand corner of the menu bar (or use the short cut keys ALT + !) to return to WindowMaker.

Changing a String's Font, Font Style and Font Size

1. Select the text string and then click the Fonts tool $\boxed{\text{ᵇᴄ}}$ on the **Format Toolbar** or click on the **Text/Fonts** command. The standard Windows **Font** dialog box appears:
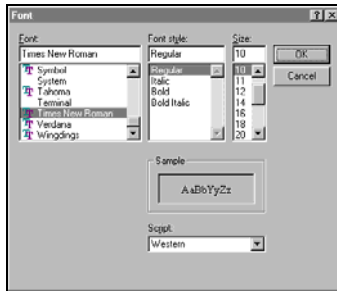
2. Select the desired font from the **Font** list (the font name will appear in the **Font** field). Once a font is selected, the styles and sizes available for it will appear in the **Font Style** and **Size** fields. When a font size is selected, a sample of the font in the selected style and size will appear in the **Sample** field (see above example).

3. Click **OK**.

Editing Text Objects

1. Select the object or button with the text.

2. Click on the **Special/Substitute Strings** command, or right-click the text object, and point to **Substitute**. The **Substitute Strings** dialog box appears:
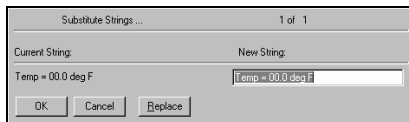
3. In the **New String** box, enter the new string and then click **OK**.

Replacing a Portion of a Text Object

If you change a portion of a text object's text, InTouch will automatically make the change to all selected text objects using the same text.

1. Select all text objects.

2. Click on the **Special/Substitute Strings** command, or right-click the text object, and point to **Substitute**. The **Substitute Strings** dialog box appears.

3. Click **Replace**. The **Replace Text** dialog box appears:
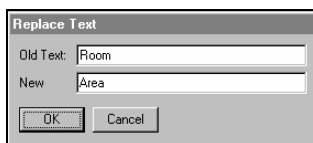
4. In the **Old Text** box, enter the portion of the string to replace.

5. In the **New** box, enter the replacement text.

6. Click **OK**. The **Substitute Strings** dialog box reappears showing the change made to the selected text strings:
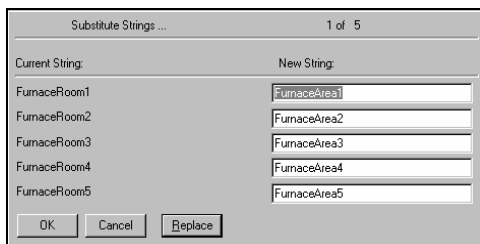
7. Click **OK**. The selected text objects will automatically be modified.

Images and Bitmaps

All graphic objects such as pictures, screen captures, AutoCad drawings, JPEG, PCX and TGA file types created in other Windows programs must be pasted into a bitmap container in WindowMaker.

WindowMaker sees a bitmap as a single object. You cannot animate the individual elements of a bitmap, nor can you include bitmaps in symbols. You can include them in a cell.

You can edit a bitmap using the Microsoft Paint program included with Windows without leaving WindowMaker.

MS Paint can now be activated automatically by right-clicking on a bitmap or empty bitmap container. While in MS Paint, the bitmap can be edited.

You can also define a bitmap with a transparent color so it can float over other objects. In doing so, the window background color or any objects behind the bitmap will show through in the areas where the transparent color is used. (Only one transparent color may be used per bitmap.)

Importing a Bitmap, JPEG, PCX or TGA File utilizing MS Paint

1. Click on the Bitmap tool . The cursor turns into a cross-hair. Draw a bitmap container in your window (any size).

2. **Right-click on the bitmap container** or click on the **Edit/EditImage** command. A drop down menu appears:
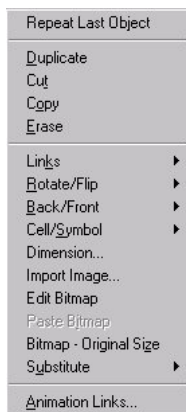
3. Click on Import Image to paste an image into the container. The Windows **Select Image File** dialog box appears.
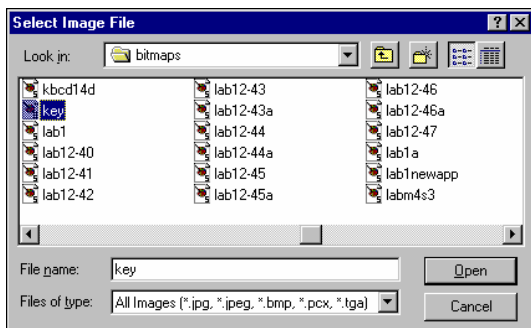
4. Select the .JPG, .JPEG, .BMP, .PCX, .TGA file to import as a bitmap. Then click **Open** or double-click the image filename. The image will be pasted into your bitmap container.

5. To return the bitmap to its original size, select it and then click on the **Bitmap - Original Size** command on the dialog box. The bitmap will be redrawn to its original size.

6. To save the bitmap while in MS Paint and return to WindowMaker, click on the **File/Save** or **File/Exit** commands. Both commands serve the same function.

Pasting a Bitmap from the Windows Clipboard

1. Copy the graphic to the Windows Clipboard. (For example, display the graphic and then hold down the ALT key while pressing the PRINT SCRN key to copy it to the Windows Clipboard.)

2. Click the Bitmap tool . The cursor turns into a cross-hair. Draw a bitmap container in your window (any size).

3. Select the bitmap container.

4. Click on the **Edit/Paste Bitmap** command, either on the toolbar or the **Paste Bitmap** command on the MS Paint dialog box. The bitmap from the Windows Clipboard will be pasted into the bitmap container.

5. To make the bitmap its original size, select it and then click on the **Edit/Bitmap - Original Size** command or **Bitmap – Original Size** on the dialog box. The bitmap will be redrawn to its original size.

Resizing a Bitmap

Objects can be resized in WindowMaker using X,Y location or W,H parameters.

Right click on the bitmap to display the drop down menu:
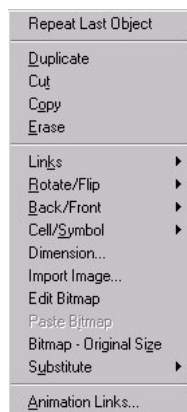
Click on **Dimension** to display the **Set Object Dimension** dialog box.
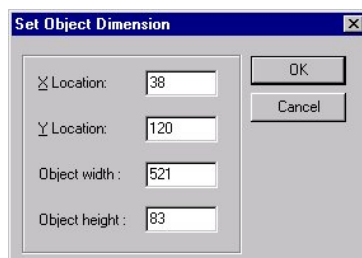
Change the dimensions to your specifications and click **OK**. The bitmap is now resized.

Creating a Transparent Bitmap

A bitmap or image can be defined with a transparent color so it can float over other objects. In doing so, the window's background or any objects behind the bitmap will show through in the areas where the transparent color is used.

1. Click the Bitmap tool [icon]. The cursor turns into a cross-hair. Draw a bitmap container in your window (any size).

2. Select the bitmap container.

3. Select the bitmap container and then click **Edit/Paste Bitmap** (if you have copied the graphic to the Windows Clipboard). Otherwise, click **Import Image** (to locate and select the .BMP, .PCX, .TGA or .JPG file to open). The bitmap image will be pasted into the bitmap container:

4. Select the bitmap and then click **Edit/Bitmap - Original Size** to return the bitmap to its original size.

5. With the bitmap selected, click the Transparent Color tool [icon] on the **Format** toolbar to open the transparent color palette.

6. Right-click a blank color square in the **Custom Palette** section at the bottom of the color palette:

7. Click the Blotter tool (the **Edit Custom Color** dialog box will close).

8. Click the color in the bitmap to make transparent. The color will be copied to the color square selected in the transparent color palette.

9. Click the color square to apply the transparent color to the bitmap.

Wizards

Wizards are easy to use and easy to configure and save a considerable amount of development time. To configure a wizard, install it, select it in the **Wizard Selection** dialog box, paste it into your window and then double-click it. Its respective configuration dialog box will appear (assuming the wizard is configurable).

For example, to use a slider wizard, you would need to configure items such as the tagname effect, the minimum and maximum range labels for the slider, the fill color, etc. Development time is saved using Wizards because you don't have to draw the individual components for the object, set the value ranges for the object or animate the object.

Installing or Removing Wizards

1. Click on the **Special/Configure/Wizard/ActiveX Installation** command, or right-click on **Installation/Open**. The **Wizard/ActiveX Installation** dialog box appears with the **Wizard Installation** property sheet active:
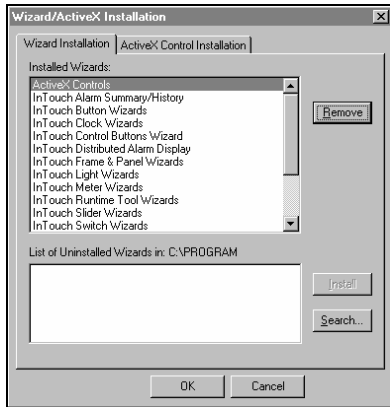
2. In the **Installed Wizards** list, select the wizard(s) to remove from your application and then click **Remove**. A message box will appear asking you to confirm the deletion.

3. Click **Yes** to remove the wizard. The removed wizard is moved to the **List of Uninstalled Wizards** list.

4. To install wizards, select them in the **List of Uninstalled Wizards** and then click **Install**.

5. Click **Search** to install wizards from another directory. The **Search for Wizard files** dialog box appears

6. Locate the directory containing the wizards to install and then click **OK**. The wizard installation dialog box reappears.

7. Any Wizards that were found will appear in the **List of Uninstalled Wizards** and can now be installed.

Placing a Wizard in a Window

1. Click the Wizard tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears:
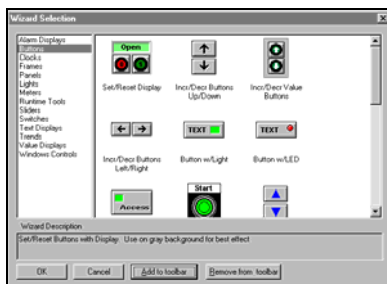
2. In the list of wizards, click the desired wizard category.

3. Select the desired wizard and then click **OK** or double-click the wizard. The dialog box will close and your window will reappear.

4. The cursor will change to a corner symbol when you return to the window. Click the location in the window where you want to paste the wizard.

5. Double-click the wizard to configure it (if applicable).

Chapter : 3   Tagname Dictionary

Section : 1

Tagname Definition

This section describes the Tagname Dictionary, the different tagname types and how to define a new tagname.

What is the Tagname Dictionary?

The Tagname Dictionary (runtime database) is the heart of InTouch. At runtime, the database contains the current value of all of the items in the database. In order to create the runtime database, InTouch requires information about all of the variables being created. Each variable must be assigned a tagname and type. InTouch also requires additional information for some variable types. For instance, for I/O type tagnames, InTouch requires more information in order to be able to acquire the value and convert it for internal use. The Tagname Dictionary is the mechanism used to enter this information.

Tagname Types

When you are defining tagnames in the InTouch database, you must assign a specific type to each tagname according to its usage. For example, if the tagname is to read or write values coming to or from another Windows application, such as an I/O Server, it must be an I/O type tagname. The following describes each InTouch tagname type and its usage.

Memory Type Tagnames
Memory tagname types exist internally within your InTouch application. They are used to create system constants and simulations. They can also be used to create calculated variables that are accessed by other Windows programs. In simulations, you can use memory tagnames to control the actions of a background QuickScript. For example, you can define a memory tagname "COUNT" that is changed in an action QuickScript to cause various animation effects to occur for the current STEP of a process. There are four Memory types:

Memory Discrete
Internal discrete tagname with a value of either 0 (False, Off) or 1 (True, On).

Memory Integer
A 32-bit signed integer value between -2,147,483,648 and 2,147,483,647.

Memory Real
Floating (decimal) point memory tagname. The floating point value may be between -$3.4e^{38}$ and $3.4e^{38}$. All floating point calculations are performed with 64-bit resolution, but the result is stored in 32-bit.

Memory Message
Text string tagname that can be up to 131 characters long.

I/O Type Tagnames
All tagnames that read or write their values to or from another Windows program are I/O type tagnames. This includes all inputs and outputs from programmable controllers, process computers and data from network nodes. I/O tagnames are accessed either through the Microsoft Dynamic Data Exchange (DDE) or Wonderware SuiteLink communication protocols.

When the value of a read/write I/O type tagname changes, it is immediately written to the remote application. The tagname may also be updated from the remote application whenever the item to which the tagname is linked changes in the remote application. By default, all I/O tagnames are set to Read/Write. However, you can restrict them to read only by selecting the Read Only option in the **Tagname Dictionary** dialog box. There are four I/O types:

I/O Discrete
Discrete input/output tagname with a value of either 0 (False, Off) or 1 (True, On).

I/O Integer
A 32-bit signed integer value between -2,147,483,648 and 2,147,483,647.

I/O Real
Floating (decimal) point tagname. The floating point value may be between +3.4e38. All floating point calculations are performed with 64-bit resolution, but the result is stored in 32-bit.

I/O Message
Text string input/output tagname that can be up to 131 characters long.

Miscellaneous Type Tagnames
There are several special tagname types that you can assign to tagnames to perform complex functions, such as creating dynamic alarm displays, historical trends, monitoring or controlling the tagname each historical trend pen is plotting, etc. There are also indirect tagname types that you can use to reassign a tagname to multiple sources. These special tagname types are described below:

Group Var
The Group Var type is used for a tagname with an assigned Alarm Group to create dynamic alarm displays, disk logs and print logs. Group Var type tagnames are used to create alarm windows or alarm logs that display all alarms associated with a specific group variable. You can also control the alarms that are displayed or logged by assigning a different Alarm Group to the Group Var tagname.

You can also use a Group Var type tagname to create buttons the operator can click to selectively display alarms for different areas of a plant in the same alarm window. All of the .fields associated with Alarm Groups can be applied to Group Var tagnames.

Hist Trend
InTouch requires a Hist Trend type tagname when you create a historical trend. All of the .fields associated with historical trends can be applied to Hist Trend tagnames.

Tag ID
This is a special type that is used with historical trend objects. You use Tag ID type tagnames to retrieve information about tagnames being plotted in a historical trend. In most cases, you would use Tag ID tagnames to display the name of the tagname assigned to a specific pen or to change the tagname assigned to the pen.
You can process a statement in a QuickScript to assign a new tagname to any pen in any historical trend. For example, the following statement could be used in your QuickScript:
MyHistTrendTag.Pen1=MyLoggedTag.TagID;
When this QuickScript is executed, Pen1 in the historical trend associated with the Hist Trend tagname "MyHistTrendTag" would begin trending the historically logged data for the "MyLoggedTag."

Indirect Discrete, Indirect Analog, Indirect Message
Indirect type tagnames allow you to create one window and reassign the tagnames in that window to multiple sources. For example, you could create a Data Change

QuickScript that would modify the source for all tagnames in a window, based on a value that has changed.

When you equate an indirect tagname to another source tagname, both the indirect tagname and the source tagname become exact duplicates of each other in every aspect including .fields, scripts, etc. If the value of the source tagname changes, the indirect tagname reflects the change. If the indirect tagname's value changes, the source tagname changes accordingly. You can define indirect tagname values in the database as retentive and reset them to take on their last tagname assignment on startup.

Indirect tagnames are assigned by using the .Name field. For example, if you created an indirect analog tagname called "Setpoint" and used the expression below in a QuickScript, "Setpoint1" would become the source for the value of "Setpoint" and vice versa:

Setpoint.Name = "Setpoint1";

Setpoint.Name = Setpoint1.Name;

You can also concatenate tagnames for use in indirect tags. For example, if you created a Data Change QuickScript that executes each time the value of the tagname "Number" changes, the indirect tagname, "Setpoint," would change accordingly:

Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#");
When this QuickScript executes, the value of the analog tagname "Number" is converted to text and added to the analog tagname "Setpoint," making "Setpoint.Name" equal to "Setpoint1." (Indirect analog type tagnames are used for both integer (whole numbers) and real (floating point) tagnames.

Defining a New Tagname

Tagnames can be up to 32 characters long and must begin with an alpha character (A-Z or a-z). The remaining characters can be A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ and &.

Tagnames are also auto-indexed. For example, if you enter and save tagname R4001 and then click New, the tagname will automatically be indexed to R4002. If a tagname contains a character separating numbers, it is auto-indexed by the first whole number InTouch finds. For example, N7-0 would be indexed as
N7-1. Positive changes only are permitted. For example, R4002 to R4003, R4003 to R4004, etc.

You need to be careful when you use dashes (-) in tagnames. They are valid for use in tagnames, but they are also used as the negation and subtraction "operator" in expressions or logic. Therefore, some ambiguity arises.

For example, if you use A=B-C in an expression, do you mean that A=B minus C or do you want to assign a tagname named B-C to a tagname named A? InTouch will assume the latter. You can prevent this by separating the tagnames from the operators with blank space(s). For example, A = B - C.

Consider this example: X-101=FT-101*SP-101

Can you see where FT-101 is being multiplied by SP-101 and assigned to X-101 due to the fact that no spaces were used?

The first time you access the Tagname Dictionary, the definition for the internal system tagname $AccessLevel is displayed. Once you define tagnames in the Tagname Dictionary, when you access it again, the last edited tagname's definition is displayed.

Click << or >> to browse through the tagname definitions currently stored in your Tagname Dictionary. (The browse buttons will be inactive when there are no previous or next tagnames to display.)

Intouch Basic Training Manual

Click Select to quickly locate a specific tagname definition. The Select Tag dialog box will appear in the selection mode.

1. Click on the Special/Tagname Dictionary command or in the Application Explorer, double-click Tagname Dictionary. The Tagname Dictionary dialog box appears:
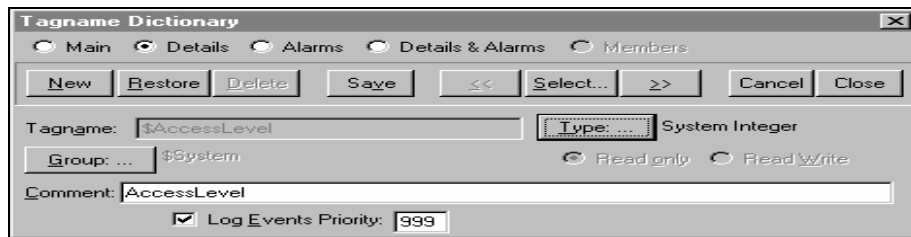
2. Click New. (The Tagname box clears.)

3. In the Tagname box, enter the name you want to use for the new tagname.

4. Click Type. The Choose tag type dialog box appears:

5. Select the type for the tagname and click OK. The respective details dialog box for the selected type will appear.

    The names of any SuperTags created in the TemplateMaker will also appear in this dialog box and can be selected as the tag type. For example, ColdRoom and EvapUnit above. For SuperTags not created using the TemplateMaker, the name "SuperTag" will appear. For example, SuperTags created in an animation link tagname or expression input box, a QuickScript, or created in an external file and then loaded the DBLoad utility.

6. Click Group to assign the tagname to a specific Alarm Group. The Alarm Group dialog box will appear. Select the name of the Alarm Group to assign to the tagname. Click Done.

7. For I/O type tagnames, select Read Only to restrict the tagname to read only capabilities in runtime.

8. For I/O type tagnames, select Read Write to grant the tagname read and write capabilities in runtime.

9. In the Comment box, enter any miscellaneous comment you want the system to store regarding your tagname (up to 50 characters). You can configure your alarm windows to display these comments whenever the tagname is in alarm.

10. Select Log Data to log the tagname to the historical log file during runtime whenever its engineering unit value changes more than the specified Log Deadband value or, by default, once an hour, regardless of change.

11. Select Log Events to log all data value changes to the tagname that are initiated by the operator, I/O, a QuickScript or by the system.

    When you define a tagname to do event monitoring, an event message is logged to the alarm system each time the tagname's value changes. The event message logs how the value changed and whether the change was initiated by the operator, I/O, scripts or the system.

When you select Log Events, the Priority field becomes active. The value you type for the Priority determines the event priority level for the tagname. Valid entries in this field are 1 to 999, where 1 is the highest and 999 is the lowest priority.

12. Select Retentive Value to retain the current value of the tagname whenever WindowViewer is exited. This value will be used as the initial value for the tagname whenever WindowViewer is restarted.

Retentive values cannot be selected or cleared for new or existing tagnames if WindowViewer is running.

13. Select Retentive Parameters to retain any changes the operator makes to the value of any alarm limit fields for the tagname. This value will be used as the initial value for the alarms when WindowViewer is restarted.

14. Define the details for the type of tagname.

15. Click Done.

Defining Tagname Details

The initially displayed Tagname Dictionary dialog box is used to input basic tagname information. Many points, especially inputs and outputs, require greater detail to be properly handled. For each type of tagname specified, a specific details dialog box is used to define the details for the tagname type.

Most of the tagname types have their own specific detail level dialog boxes and alarm condition dialog boxes. By default, when you select the type for your tagname, its respective details level dialog box will appear.

Once you have defined the basic tagname, you will need to define the details for the tagname and, if required, the alarm conditions.

Printing Tagname Details

In addition to printing a saved cross reference .csv file, you can print listings of the Tagname Dictionary details, alarm information, link details and scripts. Printing the Tagname Dictionary details can help determine tagname usage.

Click on the File/Print command. The WindowMaker Printout dialog box appears:
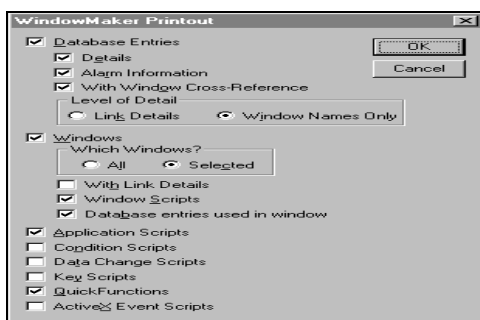
1. Select **Database Entries** to print all database information. If **Database Entries** is selected, the following three options become active:

- Select **Details** to include the database details in your report.
- Select **Alarm Information** to include the database alarm information in your report.
- Select **With Window Cross Reference** to print all database entries with window cross-references. Selecting this option will activate **Level of Detail** options:
  - Select **Link Details** to print the location and animation link details where the tagname was used.

— Select **Window Names Only** to print only the name of the cross-referenced windows(s).

2. Select **Windows** to print a listing of the database entries used in the application windows. If you select **Windows**, the following three options become active:

- Select **All** to print the database entries for all windows in the application.
- Select **Selected** to print only the database entries for specific windows. The **Windows to Print** dialog box appears:
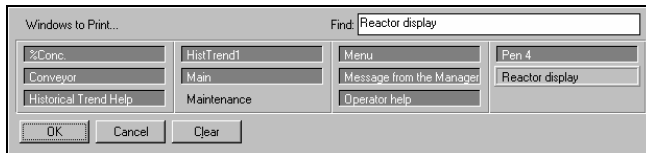
3. Select the windows you want to print and click **OK**. (By default, all window names will be selected when the dialog box appears.)

- Select **With Link Details** to print the link details for the window(s).
- Select **Window Scripts** to print the scripts associated with the window(s).
- Select **Database entries used in window** to print the tagnames used in the window(s).
- Select **Application Scripts** to print the application scripts.
- Select **Condition Scripts** to print the condition scripts associated with the window(s).
- Select **Data Change Scripts** to print the data change scripts associated with the window(s).
- Select **Key Scripts** to print the key scripts associated with the window(s).
- Select **Quick Functions** to print your QuickFunctions.

4. Click **OK** to begin printing your report.

Tag Browser

The Tag Browser is your primary tool for viewing and selecting local and remote tagnames and tagname **.fields** from FactorySuite applications or any other tag source that supports the InTouch Tagname Dictionary interface. It allows you to select existing tagnames, add new tagnames and view basic Tagname Dictionary information. The Tag Browser is also used to access the dialog boxes that allow you to perform tagname editing, tagname replication and to select tagnames (remote references) in remote tag sources.

The first time the Tag Browser is accessed, by default, **<local>** will be selected for the tag source. This means that the tagnames in the local application's Tagname Dictionary will be displayed. Thereafter, the last accessed tag source's tagnames will be displayed.

The Tag Browser operates in two modes: "Filtered Selection Mode" and "Unlimited Selection Mode." The mode for the Tag Browser is determined by the method used to access it. The following lists the primary methods that can be used to access the Tag Browser in each mode:

Filtered Selection Mode

- Clicking the **Select** button in the Tagname Dictionary.

- When WindowMaker is running, double-clicking a cell in the **Unit#** column in a Recipe Manager **Unit Template** definition.

- In runtime, clicking any **Pen#** button in the **Historical Trend Setup** dialog box. In this instance, the Tag Browser will only display the tagnames that are defined with the **Log Data** option selected in the Tagname Dictionary.

> This functionality is only supported when the **Allow Runtime Changes** option has been selected for the historical trend during development.

- In runtime, clicking any object linked to the **HTSelectTag()** function.

Unlimited Selection Mode

- Double-clicking an animation link tagname or expression input box.

- Double-clicking an ActiveX or wizard tagname or expression input box.

- Double-clicking a blank area in any InTouch QuickScript window.

- In the InTouch QuickScript editor, selecting the **Tagname** command on the **Insert** menu.

- Pressing the ALT + N keys in the InTouch QuickScript editor.

- Double-clicking a blank **New Name** box in the **Substitute Tagnames** dialog box.

- Double-clicking the **Tagname.FieldName** input box in the SQL Access **Bind List Configuration** dialog box.

The Tag Browser's status bar provides status on the following items for the currently displayed tag source:

- Total number of items in the application.

- The name of the currently selected item.

- Tagname **.field** selected, if any.

- The Access Name associated with the tag source.

Tag Browser Selection Modes

The Tag Browser operates in two selection modes: Filtered Selection Mode and Unlimited Selection Mode.

Filtered Selection Mode

If you click Select in the Tagname Dictionary dialog box, the tagnames displayed (and available for selecting) will be limited to the current InTouch application. The same is true if, during runtime, an operator selects a new tagname for a historical trend pen. For example:
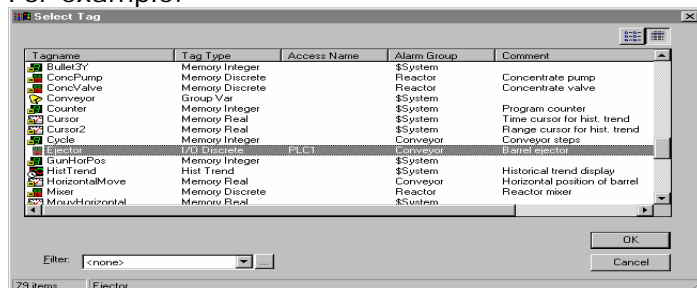
When the Tag Browser is accessed from the Tagname Dictionary and a tagname is selected in this view, its Tagname Dictionary definition will appear after you click OK.

Unlimited Selection Mode

The unlimited selection mode is accessed by double-clicking in a blank area in any InTouch QuickScript window, animation link tagname or expression box or, a blank New Name box in the Substitute Tagnames dialog box. The tagnames defined in a local or remote tag source can be displayed and selected in this mode.

Tagname .fields can also be selected for the tagname in this mode. When a tagname and/or tagname.field is selected in this mode, it is automatically entered into the

InTouch QuickScript, animation link tagname or expression box or other location from which you accessed the Tag Browser.
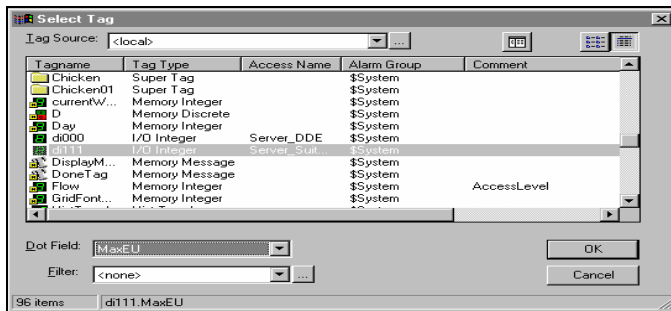
Selecting a .Field

1. Click the **Dot Field** arrow to open the list of **.fields** that can be associated with the type of tagname currently selected.

    By default, **<none>** will initially be displayed for all types of tagnames.

2. Click the **.field** in the list to append to the selected tagname.

Defining Tag Browser Filters

You will use the procedures described in this section to define the filters (search criteria) to use to populate the Tag Browser. By creating filters, you can sort any tagname list and display only the tagnames that meet the specified criteria. Tagnames can be sorted based on **Tagname**, **Tag Type**, **Access Name**, **Alarm Groups** and tagname **Comments**. You can use one or a combination of any of these items to set the criteria for your display. You can also save each filter instance and reuse it at any time.

For example, if you have 40,000 tagnames defined in your Tagname Dictionary and you only need to deal with 20 that are assigned to a particular Access Name or Alarm Group, you can create a filter and specify the Access Name and/or Alarm Group as the criteria that the tagnames must meet in order to be displayed in the Tag Browser.

1. Click the Define Filter button ⟦ ... ⟧. The **Define Tag Filter** dialog box appears:
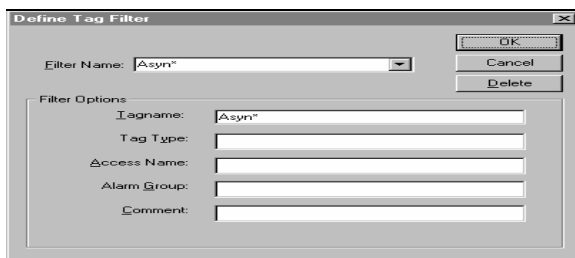
2. In the **Filter Name** box, enter a unique name to identify the filter that you are defining or click the **Filter Name** arrow to select a previously defined filter name from the list. (As you define filters, the **Filter Name** you enter is added to the list.)

3. In the **Tagname** box, enter the tagname expression to use as a filter. If left blank, the system will ignore this field in the filter definition.

4. In the **Access Name** box, enter the local Access Name expression to use as a filter. If left blank, the system will ignore this field in the filter definition.

5. In the **Alarm Group** box, enter the name of the Alarm Group expression to use as a filter. If left blank, the system will ignore this field in the filter definition.

6. In the **Comment** box, enter the comment expression to use as a filter. If left blank, the system will ignore this field in the filter definition.

7. Click **OK** to close dialog box.

Chapter : 4   Animation Link

Section : 1

What are Animation Link ?

Graphic objects or symbols can be "brought to life" using animation links. Animation links can make an object or symbol change in appearance to reflect changes in the value of a tagname or an expression. For example, you can create a pump symbol that is red when it is off and green when it is on. You can also make the pump symbol a touch-sensitive pushbutton that the operator can click with the mouse to turn the pump on and off. You can use these and many other special effects by defining animation links for your objects or symbols.

InTouch supports two basic types of links: Touch Links and Display Links. Touch Links allow operator input into the system. Display Links allow output to the operator. Value sliders or pushbuttons are examples of Touch Links. Color fill, location or blink links are examples of Display Links.

Most animation links share the following common controls:

- Object Type Dialog Box
- Common Color Palette
- Quick access to the Tag Browser
- Quick access to the Tagname .Fields
- Support for Key Equivalents
- Right-click mouse support in the **Tagname** or **Expression** input boxes (displays a menu with commands that can be applied to the selected text).

Object Type Dialog Box

The **Object Type** dialog box appears at the top of the screen above the link selection dialog box. It is the header dialog box common to all links created. It displays the description of the type of object that you have selected for animation link attachment. For example, **Button**.

If multiple links have been attached to an object, you can click **Prev Link** and **Next Link** to quickly page forward or backwards through the link dialog boxes for each link attached to the object.

Animation Link Selection Dialog Box

Multiple links can be defined for your objects or symbols. By combining various links, you can create almost any screen animation effect imaginable. You can make objects change color, size, location, fill level, visibility, etc.
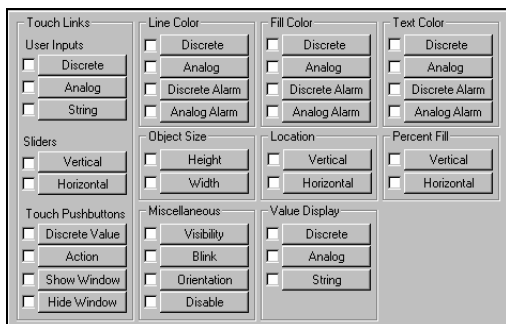
Applying Color Links

Color can be applied to the dynamic properties of lines, rectangles, round rectangles, ellipses, polylines, polygons and text. The color palette is used to select the colors to be linked to the value of the tagname, the tagname's alarm state, etc.

You must use solid colors for line and text color links. If a dithered (mixed) color is selected, WindowMaker, by default, will select the closest solid color. Custom color palettes can be created and added the standard WindowMaker color palette.

When a color link is attached to an object or symbol and then a color box is clicked in a link's dialog box, the color palette appears.

Click the desired color. The color palette automatically closes and the selected color will appear in the color box.

Accessing the Tag Browser

You can quickly view all of the tagnames defined in your application when you are creating animation links by accessing the Tag Browser. If you select the tagname to be assigned to your link from the Tag Browser, it is automatically inserted into the **Tagname** or **Expression** box.

The following steps describe how to access the tag browser:

1. Double-click any blank animation link **Tagname** or **Expression** input box. The Tag Browser will appear.

2. Click the Tree View tool to display the tree view pane and all defined tag sources:
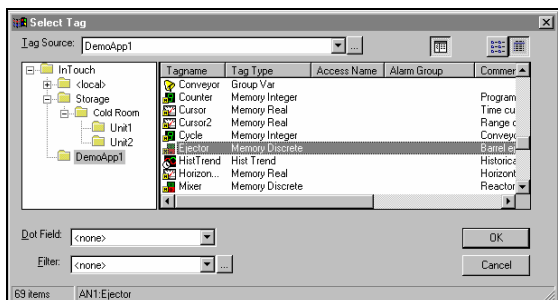
If you are not using the tree view mode, click the **Tag Source** arrow and select the name for the tag source to display in the list. The Tag Browser will refresh and the selected tag source's tagnames will be displayed.

3. Select the tagname for the link and click **OK** (or double-click the tagname to simultaneously select it, close the Tag Browser and insert it into the **Tagname** or **Expression** box).

    To use a **.field** with the selected tagname, click the **Dot Field** arrow, select the **.field** and then click **OK**.

    To display a tagname's database definition, enter the tagname in the **Tagname** or **Expression** box and double-click it. The **Tagname Dictionary** dialog box will appear displaying the tagname's definition.

Accessing Tagname .Fields

There are two methods that can be used to access tagname **.fields** from an animation link **Tagname** or **Expression** input box:

Accessing Tagname .Fields Through the Tag Browser

1. Double-click a blank **Tagname** or **Expression** input box. The Tag Browser appears displaying the tagnames defined for the current tag source:
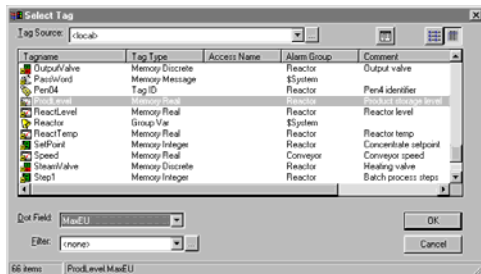
2. Click the **Dot Field** arrow to open the list of **.fields** you can associate with the currently selected tagname.

   By default, **<none>** will initially be displayed for all types of tagnames.

3. Click the **.field** in the list to append to the selected tagname.

Accessing Tagname .Fields Through the Choose Field Name Dialog Box

1. In the **Tagname** or **Expression** input box, enter a tagname plus a period (**tagname.**) and then double-click to the right (or enter a period only and then double-click to the right). The **Choose field name** dialog box appears displaying all tagname .fields:
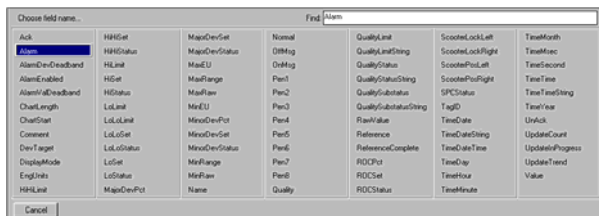
2. Select the .field. The dialog box will close and the selected .field will automatically be inserted into the **Tagname** or **Expression** input box.

Section : 2

Animating Objects

The following steps describe a simple animation procedure:

1. Create and select the object (line, filled shape, text, button or symbol) to animate.

2. Click on the **Special/Animation Links** command or double-click on the object. The dialog box containing all animation links appears:
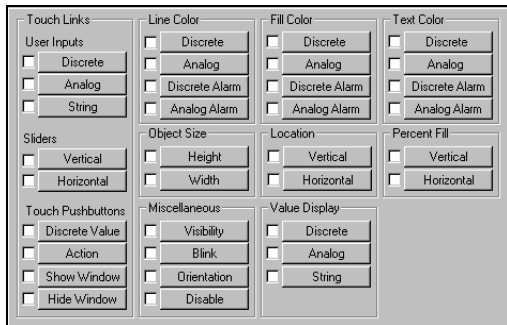
3. Click the appropriate link button to attach your link to the selected object.

   If a link is not applicable for the selected object, its button will not be active.

   Clicking the check box only selects the link. Clicking the link name button selects the link and opens its detail definition dialog box. The check box will automatically be selected when you click the link name button and accept the input. However, if you clear a link's check box, the animation link is removed from the selected object.

4. Enter the details for the link definition and then click **OK**. The **Link Selection** dialog box will reappear and another link can be created for the object.

5. Click **OK** to accept all links for the object and close the dialog box.

   When creating animation links, the tagname entered in the animation link's tagname or expression box must be defined in the Tagname Dictionary first. If it is not defined, a message box will appear asking you to define the tagname now. If you click **Yes**, the Tagname Dictionary will appear and the tagname can be defined.

Creating Touch Links

Touch Links are used on objects or symbols to make them "touch-sensitive" in runtime. They allow the operator to input data into the system. For example, the operator may turn a valve on or off, enter a new alarm setpoint, run a complex logic script, log on using text strings, etc.

Touch links are easily identified in runtime. A "frame" surrounds a touch-sensitive object when the cursor passes over it. Also the TAB key can be pressed to move from object to object. If a touch link object or symbol contains text objects that are placed on top of each other, the top text object will be used to display the data value.

The operator activates a touch-sensitive pushbutton by clicking it, touching the object (when using a touch screen), pressing an assigned key equivalent or pressing the ENTER key when the object is "framed." There are nine types of touch links:

Touch Link   Types

User Inputs   Discrete, Analog, String

| Sliders | Vertical, Horizontal |

Touch Pushbuttons Discrete Value, Action, Show Window, Hide Window

User Input Touch Links

User Input Touch Links are used to create touch-sensitive objects that allow operator input into the system. For example, pushbuttons to change discrete states, analog values or security logons. There are three types of **User Input** touch links:

User Input	Description
Discrete	Used to control the value of a discrete tagname. When this link is activated, a dialog box will appear prompting the operator to make a selection.
Analog	Used to input the value of an analog (integer or real) tagname. When the link is activated, an input box will appear and the value may be entered from the standard keyboard or an optional on-screen keypad.
String	Used to create an object into which a string message may be input. When the link is activated, an input box for entering the message value or an optional on screen keyboard will appear.

Creating a Discrete Input Link

1. Double-click the object (or select it and then click on the **Special/Animation Links** command). The link selection dialog box will appear.

2. In the **User Inputs** section, click **Discrete**. The **Input -> Discrete Tagname** dialog box appears:
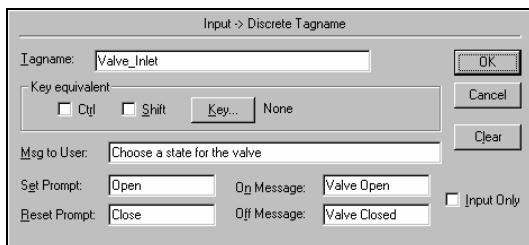
3. In the **Tagname** box, enter a discrete type tagname.

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

5. In the **Msg to User** box, enter the message to appear in the input dialog box when the input link is activated.

6. In the **Set Prompt** and **Reset Prompt** boxes, enter the messages the operator will see on the discrete value on and off buttons.

7. In the **On Message** and **Off Message** boxes, enter the messages to appear in the text field (if any) associated with the object when the object is on or off.

8. Select **Input Only** to prevent the input from being displayed in a text field associated with the object. (This option only applies to an object that has a text field associated with it.)

9. Click **OK** to attach the link to the object and return to the animation links dialog box. You can now attach another link to the object if desired.

Creating an Analog Input Link

1. Double-click the object. The link selection dialog box appears.

2. In the **User Inputs** section, click **Analog**. The **Input -> Analog Tagname** dialog box appears:

```
                    Input -> Analog Tagname
  Tagname:    Analog_Input                        [  OK  ]
  Key equivalent
    □ Ctrl    □ Shift    [Key...]  None           [Cancel]

                                                  [ Clear ]
  Msg to User:  Enter a new value
  Keypad?        Min Value: 0.
   ○ Yes ● No    Max Value: 9999.    □ Input Only
```

3. In the **Tagname** box, enter an analog (integer or real) type tagname.

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

5. In the **Msg to User** box, enter the message to appear in the input dialog box when the input link is activated.

6. Select **Yes** in the **Keypad** section to display an on-screen numeric keypad for inputting the new value of the string.

7. In the **Min Value** box, enter the minimum input value for the tagname.

8. In the **Max Value** box, enter the maximum input value for the tagname.

9. Select **Input Only** to prevent the input from being displayed in a text field associated with the object. (This option only applies to an object that has a text field associated with it, such as a 3 dimensional button.)

10. Click **OK** to attach the link to the object and return to the animation links dialog box. You can now attach another link to the object.

Creating a String Input Link

1. Double-click the object. The link selection dialog box appears.

2. In the **User Inputs** section, click **String**. The **Input -> String Tagname** dialog box appears:

```
                    Input -> String Tagname
  Tagname:    String_Input                        [  OK  ]
  Key equivalent
    □ Ctrl    □ Shift    [Key...]  None           [Cancel]

                                                  [ Clear ]
  Msg to User:  Enter a new string value
  Echo Characters?   Keypad?
   ● Yes ○ No        ○ Yes ● No     □ Input Only
```
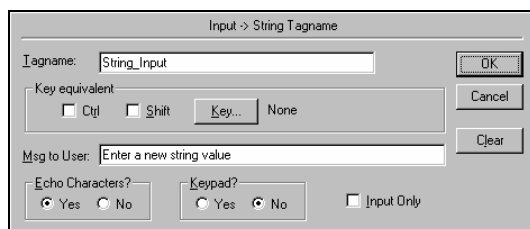
3. In the **Tagname** box, enter a message type tagname.

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

5. If you are displaying the optional keypad when this link is activated, enter the prompt message to appear in the keyboard in the **Msg to User** box.

6. Select **Yes** in the **Echo Characters** section to see the input string on the screen as it is entered. If the data is sensitive (e.g., a password) and should not be visible, select **No**.

7. In the **Keypad** section, select **Yes** to display an on-screen keyboard for inputting the new value of the string.

8. Select **Input Only** to prevent the input from being displayed in a text field associated with the object. (This option only applies to an object that has a text field associated with it, such as a 3 dimensional button.)

9. Click **OK** to attach the link to the object and return to the animation links dialog box. You can now attach another link to the object.
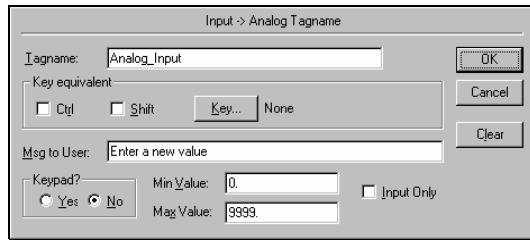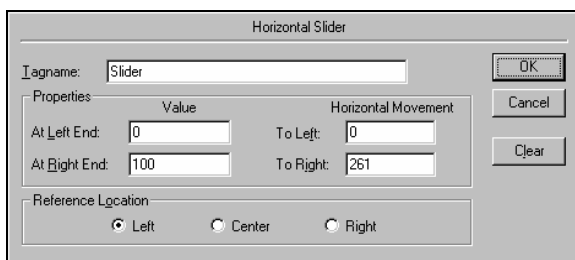
Slider Touch Links

Slider Touch Links are used to create objects or symbols that can be moved around the window with a mouse or other pointing devices (e.g., a finger on a touch screen). As the object or symbol is moved, it alters the value of the tagname linked to it. This provides the ability to create devices for setting values in the system.

An object may have a horizontal, a vertical slider touch link or both. By using both links on a single object, the value of two analog tagnames can be altered simultaneously.

Creating a Horizontal (or Vertical) Slider Link

1. Double-click the object. The link selection dialog box appears.
2. In the **Slider** section, click **Horizontal**. The **Horizontal Slider** dialog box appears:

3. In the **Tagname** box, enter an analog (integer or real) type tagname.
4. In the **At Left End** box, enter the value for the tagname when the slider is in its farthest left position.
5. In the **At Right End** box, enter the value for the tagname when the slider is in its farthest right position.
6. In the **To Left** box, enter the number of pixels the slider can move to the left.

   At the far left position, the tagname's value will be equal to the value entered in the **At Left End** field.
7. In the **To Right** box, enter the number of pixels the slider can move to the right.

   At the far right position, the tagname's value will be equal to the value entered in the **At Right End** field.
8. Select the **Reference Location** on the object that the cursor will lock when moving the object.
9. Click **OK** to attach the link to the object and return to the animation links dialog box.

Touch Pushbutton Touch Links

Touch Pushbutton Touch Links are used to create object links that immediately perform operations when clicked with the mouse or touched (touch screen). These operations can be **Discrete Value Changes**, **Action Script** executions, **Show** or **Hide Window** commands. There are four types of **Touch Pushbutton** links:

Touch Pushbuttons	Description
Discrete Value	Used to make any object or symbol into a pushbutton that controls the state of a discrete tagname. Pushbutton actions include set, reset, toggle, momentary on (direct) and momentary off (reverse) types.

Action	Allows any object, symbol or button to have up to three different action scripts linked to it; **On Down**, **While Down** and **On Up**. Action scripts can be used to set tagnames to specific values, show and/or hide windows, start and control other applications, execute functions, etc.
Show Window	Used to make an object or symbol into a button that opens one or more windows when it is clicked or touched.
Hide Window	Used to make an object or symbol into a button that closes one or more windows when it is clicked or touched.

Creating a Discrete Value Touch Pushbutton Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Touch Pushbutton** section, click **Discrete Value**. The **Pushbutton -> Discrete Value** dialog box appears:

3. In the **Tagname** box, enter a discrete type tagname.

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

5. Select the **Action** option to use for the pushbutton as follows:

   Direct Sets the value equal to 1 (True, On, Yes) as long as the pushbutton is pressed and held down. The value automatically resets to 0 (False, Off, No) when the button is released.

   Reverse Sets the value equal to 0 (False, Off, No) when the pushbutton is pressed and held down. The value automatically resets to 1 (True, On, Yes) when the button is released.

   Toggle Reverses the state of the discrete tagname when it is pressed, e.g., if the tagname is equal to 1 and the button is pressed, it is reset to 0 and vice-versa.

   Reset Sets the value equal to 0 (False, Off, No) when the pushbutton is pressed.

   Set Sets the value equal to 1 (True, On, Yes) when the pushbutton is pressed.

6. Click **OK** to attach the link to the object and return to the animation links dialog box.
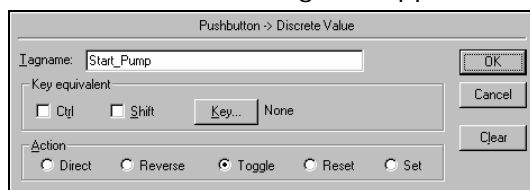
Assigning Key Equivalents

You can assign a specific key on the keyboard to activate certain animation 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 invisible or disabled.

You can define the same key in multiple windows. However, the definition in the most recently opened window will be the active one. In the case of overlay windows, the key will be active in the window on top.
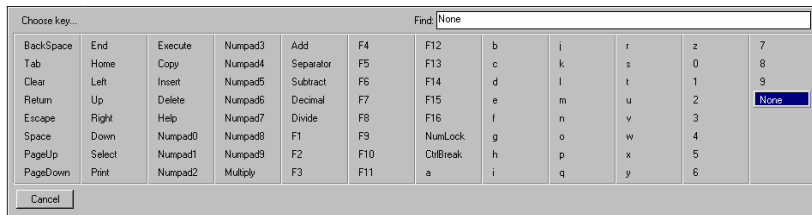
The animation links that support key equivalents will display the **Key Equivalent** group in their link dialog boxes. For example:

The following steps describe how to assign a key equivalent to an animation link:

1. Select CTRL and/or SHIFT if the operator needs to hold down either or both of these keys when pressing the key equivalent.
2. Click **Key**. The **Choose key** dialog box appears:

Choose key...						Find: None					
BackSpace	End	Execute	Numpad3	Add	F4	F12	b	j	r	z	7
Tab	Home	Copy	Numpad4	Separator	F5	F13	c	k	s	0	8
Clear	Left	Insert	Numpad5	Subtract	F6	F14	d	l	t	1	9
Return	Up	Delete	Numpad6	Decimal	F7	F15	e	m	u	2	None
Escape	Right	Help	Numpad7	Divide	F8	F16	f	n	v	3	
Space	Down	Numpad0	Numpad8	F1	F9	NumLock	g	o	w	4	
PageUp	Select	Numpad1	Numpad9	F2	F10	CtlBreak	h	p	x	5	
PageDown	Print	Numpad2	Multiply	F3	F11	a	i	q	y	6	

Cancel

3. Click the key to assign to the link. The dialog box will close and the link dialog box will reappear, displaying the name of the selected key next to the **Key** button.

Creating Display Links

Display Links provide output to the operator. There are eight types of display links:

Display Link	Types
Line, Fill & Text Color	Discrete, Analog, Discrete Alarm, Analog Alarm
Object Size	Height, Width
Location	Horizontal, Vertical
Percent Fill	Horizontal, Vertical
Miscellaneous	Visibility, Orientation, Blink, Disable
Value Display	Discrete, Analog, String

Color Links

Color links are used to animate the **Line Color**, **Fill Color**, and **Text Color** attributes of an object.

Each of these color attributes may be made dynamic by defining a color link for the attribute. The color attribute may be linked to the value of a discrete expression, analog expression, discrete alarm status or analog alarm status. There are four types of line, fill and text color links:

Color Link	Description
Discrete	Used to control the fill, line and text color attributes of an object or symbol that is linked to the value of a discrete expression.
Analog	Line, fill, and text color of an object or symbol can be linked to the value of an analog tagname (integer or real) or an analog expression. Five value ranges are defined by specifying four breakpoints. Five different colors can be selected which will be displayed as the value range changes.
Discrete Alarm	Text, line, and fill color of an object can be linked to the alarm state of a tagname, Alarm Group or Group Variable. This color link allows a choice of two colors: one for the normal state and one for the alarm state of the tagname. This link can be used for both analog and discrete tagnames. If used with an analog tagname, it responds to any alarm condition of the tagname.
Analog Alarm	Text, line, and fill color of an object can be linked to the alarm state of an analog tagname, Alarm Group or Group Variable. Allows a specific color to be set for the normal state and a separate color for each alarm condition defined for the tagname.

Creating a Discrete Fill Color Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Fill Color** section, click **Discrete**. The **Fill Color -> Discrete Expression** dialog box appears:
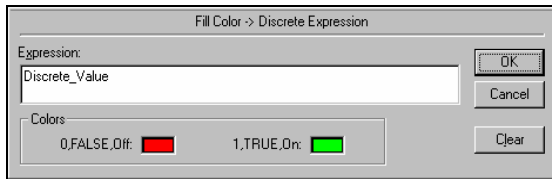
3. In the **Expression** box, enter a discrete tagname or an expression that equates to true or false.

   An expression can be up to 256 characters. If you need to use a larger expression, create a QuickFunction and call it in your expression.

   Discrete expressions can also 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 fill color of the object will change.

   Right-click the **Expression** box to access the commands that can be applied to the selected text.

4. In the **Colors** group, click each color box and select a color for each tagname state.

5. Click **OK** to attach the link to the object and return to the animation links dialog box.

Creating an Analog Expression Color Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Fill Color** section, click **Analog**. The **Fill Color -> Analog Expression** dialog box appears:
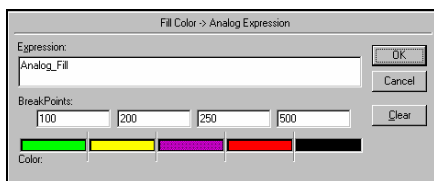
3. In the **Expression** box, enter an analog (integer or real) tagname or an expression that equates to an analog value.

4. In each **Break Points** box, specify the breakpoint values (decimals are valid for real type tagnames) where the object will change color.

   It is not necessary to use four different values. For example, if you only want the object to change color three times, enter three values and then use the same color for the third and fourth values.

5. In the **Colors** group, click each color box to select a color for each breakpoint.

6. Click **OK** to attach the link to the object and return to the animation links dialog box.

Creating a Discrete Alarm Status Color Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Fill Color** (**Line Color** or **Text Color**) section, click **Discrete Alarm**. The **Fill Color -> Discrete Tagname Alarm Status** dialog box appears:
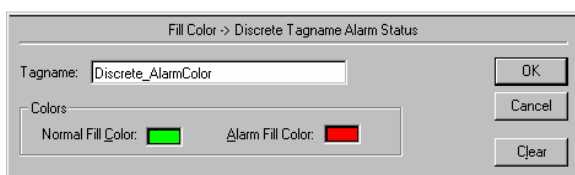
3. In the **Tagname** box, enter the discrete tagname whose alarm status is to be associated with the object.

4. In the **Colors** group, click each color box to select a color for each state.

5. Click **OK** to attach the link to the object and return to the animation links dialog box.

Creating an Analog Alarm Status Color Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Fill Color** section, click **Analog Alarm**. The **Fill Color -> Analog Tagname Alarm Status** dialog box appears:
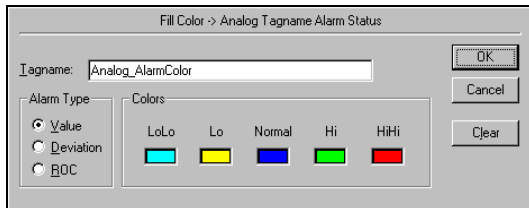
3. In the **Tagname** box, enter the analog (integer or real) tagname whose alarm status is to be associated with the object.

4. In the **Alarm Type** group, select the type of alarm to associate with the object. There are three mutually exclusive types of analog color links available:

    Value Alarm - Up to five different colors can be selected, depending on the status of the value alarms defined for the tagname.

    Deviation - Up to three different colors can be selected, depending on the status of the deviation alarms defined for the tagname.

    ROC (Rate-of-Change) - Two different colors can be selected, depending on the status of the rate-of-change alarm defined for the tagname.

5. In the **Colors** group, click each color box to select a color for each state.

6. Click **OK** to attach the link to the object and return to the animation links dialog box.

Object Size Links
Object Size links are used to vary the height and/or width of an object according to the value of an analog (integer or real) tagname or analog expression. Size links provide the ability to control the direction in which the object enlarges in height and/or width by setting the "anchor" for the link. Both height and width links can be attached to the same object.

Creating a Height (or Width) Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Object Size** section, click **Height**. The **Object Height -> Analog Value** dialog box appears:
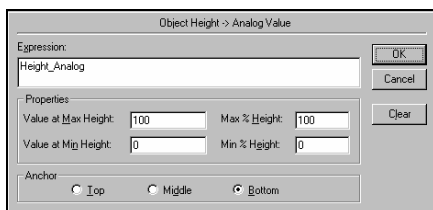
3. In the **Expression** box, enter an analog (integer or real) tagname or an expression that equates to an analog value.

4. In the **Value at Max Height** box, enter the value of the tagname or expression that will result in the object reaching its maximum height.

5. In the **Value at Min Height** box, enter the value of the tagname or expression that will result in the object reaching its minimum height.

6. In the **Max % Height** box, enter the percentage (0-100) of its height that the object will be when the tagname or expression reaches the value set in the **Value at Max Height** box.

7. In the **Min % Height** box, enter the percentage (0-100) of its height that the object will be when the tagname or expression reaches the value set in the **Value at Min Height** box. The percent height figures are expressed as a percentage of the actual "drawn size" of the object, which is 100%.

8. Select the **Anchor** point from which the object will enlarge in height.

   Selecting **Top** will cause the object to be enlarged from its top downward. Selecting **Middle** will cause the object to be enlarged from its centerpoint outwards in both directions. Selecting **Bottom** will cause the object to be enlarged from its bottom upwards.

9. Click **OK** to attach the link to the object and return to the animation links dialog box.

Location Links
Location Links are used to make an object automatically move horizontally, vertically or in both directions in response to changes in the value of an analog tagname or expression.

Creating a Horizontal Location Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Location** section, click **Horizontal**. The **Horizontal Location** dialog box appears:

3. In the **Expression** box, enter an analog (integer or real) tagname or an expression that equates to an analog value.

4. In the **At Left End** box, enter the value for the tagname when the object is located at its farthest left position.

5. In the **At Right End** box, enter the value for the tagname when the object is located at its farthest right position.

6. In the **To Left** box, enter the number of pixels the object can move to the left of its drawn position.

   At the far left position, the tagname's value will be equal to the value entered in the **At Left End** field.

7. In the **To Right** box, enter the number of pixels the object can move to the right of its drawn position.

   At the far right position, the tagname's value will be equal to the value entered in the **At Right End** field.

8. Click **OK** to attach the link to the object and return to the animation links dialog box.

Percent Fill Links

Percent Fill Links are used to vary the fill level of a filled shape (or a symbol containing filled shapes) according to the value of an analog tagname or an expression that computes to an analog value. For example, this link may be used to show the level of liquids in a vessel. An object or symbol may have a horizontal fill link, a vertical fill link or both.

Creating a Vertical Percent Fill Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Percent Fill** section, click **Vertical**. The **Vertical Fill -> Analog Value** dialog box appears:
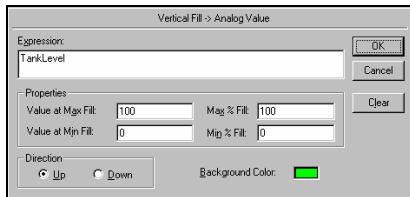

3. In the **Expression** box, enter an analog (integer or real) tagname or an expression that equates to an analog value.

4. In the **Value at Max Fill**, enter the value that will result in the object being filled to its maximum level.

5. In the **Value at Min Fill**, enter the value that will result in the object being filled to its minimum level.

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

    If the value of the expression is greater than this number, it will be ignored.

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

    If the value of the expression is greater than this number, it will be ignored.

8. Select the **Direction** the object is to fill from.

    If **Up** is selected, it will be filled from the bottom to the top. If **Down** is selected, it will be filled from the top to the bottom.

9. Click the **Background Color** box and select a color. This **Background Color** selection is for the color of the **unfilled** portion of the object. The actual fill color is the color selected for the object when it is drawn. If both **Vertical Percent Fill** and **Horizontal Percent Fill** links are linked to the same object, the last color selected in either of their link dialog boxes will be used as the background color.

10. Click **OK** to attach the link to the object and return to the animation links dialog box.

Miscellaneous Links

There are four types of miscellaneous links:

| Misc Link	Description
Visibility	Used to control the visibility of an object based on the value of a discrete tagname or expression.
Blink	Used to make an object blink based on the value of a discrete tagname or expression.
Orientation	Used to make an object rotate based on the value of a tagname or expression.

Disable   Used to disable the touch functionality of objects based on the value of a tagname or expression.

Creating a Visibility Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Miscellaneous** section, click **Visibility**. The **Object Visibility -> Discrete Value** dialog box appears:
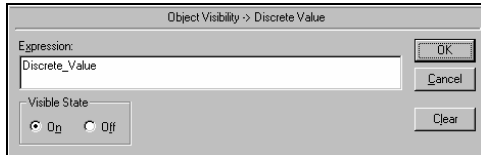
3. In the **Expression** box, enter a discrete tagname or an expression that equates to a discrete value.

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

5. Click **OK** to attach the link to the object and return to the animation links dialog box.

Creating a Blink Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Miscellaneous** section, click **Blink**. The **Object Blinking -> Discrete Value** dialog box appears:
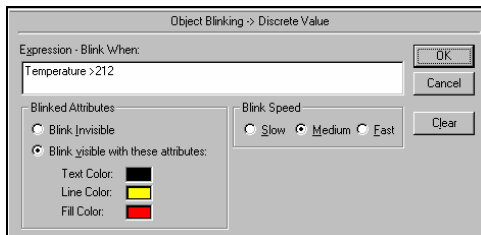
3. In the **Expression - Blink When** box, enter a discrete tagname or an expression that equates to a discrete value.

4. Select the **Blinked Attributes** for the object.

   If you select **Blink Invisible**, the object/symbol blinks by disappearing and reappearing in the window. If you select **Blink visible with these attributes**, the object/symbol remains visible in the window and the change in the color attributes selected creates the blinking effect.

   Click the **Text Color**, **Line Color** and **Fill Color** boxes to select a color.

   Choosing a "fill" blink color that is the same as the object's "fill" color will not allow the object to "blink."

5. Select the **Blink Speed** of the object.

6. Click **OK** to attach the link to the object and return to the animation links dialog box.
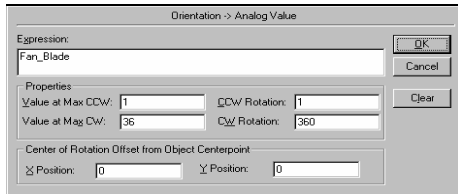
Creating an Orientation Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Miscellaneous** section, click **Orientation**. The **Orientation -> Analog Value** dialog box appears:

3. In the **Expression** box, enter an analog (integer or real) tagname or an expression that equates to an analog value.

4. In the **Value at Max CCW** box, enter the value the expression must be for the object to be rotated to its maximum counter-clockwise position.

   If the value of the expression is greater than this number, it will be ignored.

5. In the **Value at Max CW** box, enter the value the expression must be for the object to be rotated to its maximum clockwise position.

   If the value of the expression is greater than this number, it will be ignored.

6. In the **CCW Rotation** box, enter the degrees the object will rotate counter-clockwise when the **Value at Max CCW** is reached.

7. In the **CW Rotation** box, enter the degrees the object will rotate clockwise when the **Value at Max CW** is reached.

   Objects are rotated clockwise or counter-clockwise based on the original position in which they were drawn in WindowMaker.

   To force a text object to a specific angle, simply set **Value at Max CCW** to 360, **Value at Max CW** to 0, **CCW Rotation** to 360 and **CW Rotation** to 0. Then enter the angle value (e.g., 90) in the **Expression** box. Remember, without a tagname, this expression will never change and the object will always hold its 90 degree position.

8. In the **X Position** box, enter the number of pixels the rotation centerpoint is to be moved horizontally from the centerpoint of the object. (Positive values are to the right of centerpoint.)

   The orientation link uses the center of the object or symbol as the center of rotation.

9. In the **Y Position** box, enter the number of pixels the rotation centerpoint is to be moved vertically from the centerpoint of the object. (Positive values are to the left of centerpoint.)

10. Click **OK** to attach the link to the object and return to the animation links dialog box.

Creating a Disable Link

The disable link is very useful when you are applying security to your application. For example, you can disable objects based on the logged on operator's access level or name.

1. Double-click the object. The link selection dialog box appears.

2. In the **Miscellaneous** section, click **Disable**. The **Object Disabled -> Discrete Value** dialog box appears:
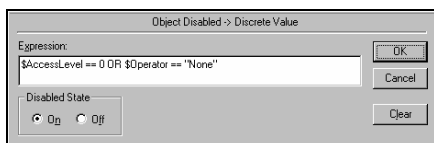
3. In the **Expression** box, enter a discrete tagname or an expression that equates to a discrete value.

By using the above expression, if no one is logged on, the object or button is secured from tampering.

4. Select the **Disabled State** that will turn the functionality of the object on or off when the discrete tagname or expression is true.

    A disabled state of **On** means the touch functionality of the object or button are turned off and cannot be clicked as long as the expression is true.

5. Click **OK** to attach the link to the object and return to the animation links dialog box.

Value Display Links

Value Display Links allow text objects to display the value of a discrete, analog, or string tagname. There are three types:

Value Display Type	Description
Discrete	Uses the value of a discrete expression to display an On or Off user defined message in a text object.
Analog	Displays the value of an analog expression in a text object.
String	Displays the value of a string expression in a text object.

Creating a Discrete Value Display Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Value Display** section, click **Discrete**. The **Output -> Discrete Expression** dialog box appears:
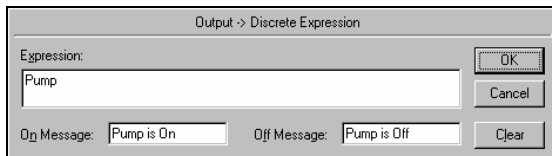
3. In the **Expression** box, enter a discrete tagname or an expression that equates to a discrete value.

4. In the **On Message** box, enter the message that will be displayed when the value of the discrete expression equals 1 (True, On, Yes).

5. In the **Off Message** box, enter the message that will be displayed when the value of the discrete expression equals 0 (False, Off, No).

    The messages will be displayed 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 displayed message at runtime.

6. Click **OK** to attach the link to the object and return to the animation links dialog box.

    You can also use a **Value Display Output -> String Expression** link to display the on and off messages for a discrete tagname. For the link, you would enter the following expression:

    DText (Pump, Pump.OnMsg, Pump.OffMsg);

    In this expression, the **.OnMsg** and **.OffMsg** strings will be extracted from the InTouch Tagname Dictionary definition for the discrete tagname, Pump.

Creating an Analog Value Display Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Value Display** section, click **Analog**. The **Output -> Analog Expression** dialog box appears:
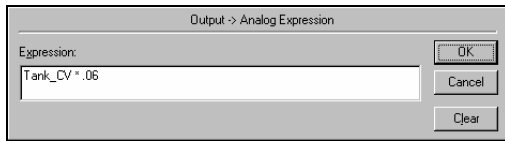
3. In the **Expression** box, enter an analog (integer or real) tagname or an expression that equates to an analog value. (You can also use a discrete type tagname in this expression. It will simply display a 1 or 0.)

4. Click **OK** to attach the link to the object and return to the animation links dialog box.

Creating a String Value Display Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Value Display** section, click **String**. The **Output -> String Expression** dialog box appears:
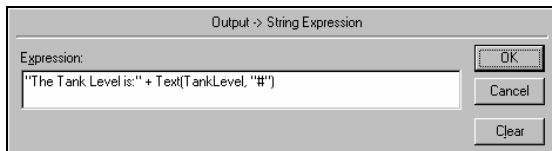
3. In the **Expression** box, enter a message tagname or an expression that equates to a message tagname.

   In the above expression, the function **Text()** is used to convert the value of the integer tagname, TankLevel, to a string.

4. Click **OK** to attach the link to the object and return to the animation links dialog box.

Section : 3

Editing Links and Tags

There are several ways to edit existing links and tags. This section includes substituting, converting, and deleting tagnames as well as importing/exporting windows and the InTouch Cross Reference Utility.

Substituting Strings

1. Select the object or button with the text.

2. Click on the **Special/Substitute Strings** command. The **Substitute Strings** dialog box appears

Substituting Tagnames

When an object is duplicated, it becomes an exact replica of the original including links, animation, scripts, etc. However, if you need to use a different tagname for an object that has been duplicated, you must change the tagname. In WindowMaker, this is called "substituting a tagname." Tagnames for objects can be selected and changed at any time and tagnames for multiple objects can be changed at the same time.

1. Select the object(s) whose tagname you want to change. Click on the **Special/Substitute Tags** command. You can also right-click the selected object, point to **Substitute/Substitute Tags**. The **Substitute Tagnames** dialog box appears:

2. In the **New Name** box, enter a new tagname and then click **OK**. The tagname associated with the selected object(s) will automatically be changed.

Importing and Exporting Windows

Importing Windows

Importing windows from one InTouch application to your current application can save development time. It also provides you with a quick and easy method for creating remote tagname references. It allows you to reuse your previously created windows, objects and window scripts. When you move windows from one InTouch application to another, you <u>must</u> use the **File/Import** command.

1. Close all windows in your current application.

2. Click on the **File/Import** command. The **Browse for Folder** dialog box appears:
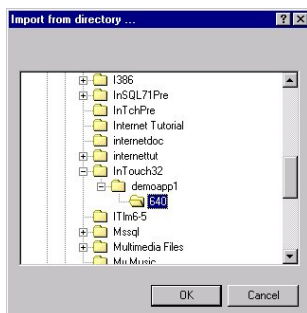
3. Select the application directory (folder) containing the windows that you want to import.

4. Click **OK**. The following dialog box appears:
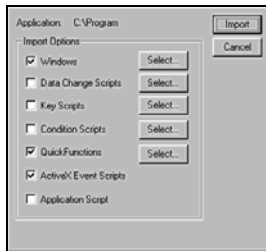
5. Select the item(s) to import and then click **Select**. A dialog box will appear to select the window or QuickScript(s) to import.

Exporting Windows

Exporting windows is very useful when creating or maintaining a library application or to quickly create remote tagname references in another application. To move windows from one InTouch application to another, you <u>must</u> use the **File/Export Window** command.

1. Close all windows in your current application.

2. Click on the **File/Export Window** command. The **Browse for Folder** dialog box appears:
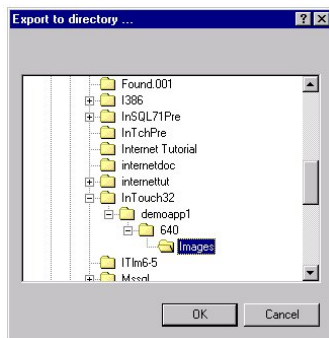
   Select the application directory (folder) to which you want to export the window(s). The **Windows to Export** dialog box appears:
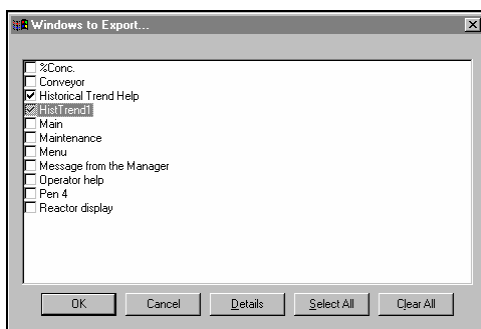
3. Select the window(s) to export and click on **OK**.

Converting Placeholder Tagnames

When tagnames are indexed (to take them out of service) or windows are imported/exported to/from your current application, all tagnames associated with that window are transferred with the window. They are not, however, added to your new application's database. Instead, tagnames are automatically marked as "placeholder"

(index) tagnames. These placeholder tagnames must be converted and, if required, defined in your new application Tagname Dictionary. For example:

In this example, to convert the placeholder tagnames to local tagnames, click **Convert**.

The placeholders **?d:**, **?i:**, **?m:** and **?r:** preceding the tagnames. indicate the type the tagname was originally defined as:

d   Discrete type
i   Integer type
m   Message type
r   Real type

Deleting Tagnames

To delete an unused tagname from the Tagname Dictionary, complete the following steps:

1. Close WindowViewer if it is running.

2. Click on the **Special/Tagname Dictionary** command. The **Tagname Dictionary** dialog box appears.

3. Click **Select**. The **Select Tag** (Tag Browser) appears.

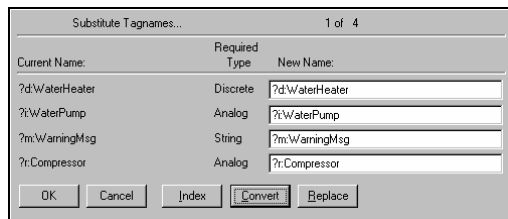4. Select the tagname to delete and then click **OK**. The **Tagname Dictionary** dialog box appears, displaying the selected tagname's definition.

5. Click **Delete**.

The **Delete** button will not be available if WindowViewer is running or if InTouch considers the tagname as being used in the application.

Updating Use Counts

InTouch maintains a use count for each item in the database. This count is not updated automatically for certain operations, such as deleting a window, changing tagnames in links or scripts, etc. In these cases, InTouch continues to consider the tagname as being used in the application and will not allow it to be deleted. Therefore, you may need to update your use count in order to delete one tagname, or multiple tagnames.

Since InTouch maintains a use count for each item in the database, you may need to update the use counts to set all unused tagnames to zero before InTouch will allow you to delete them:

1. Close all windows.

2. Click on the **Special/Update Use Counts** command.

3. Click **Yes** to update the use counts. When the system has completed updating, the following dialog box appears:
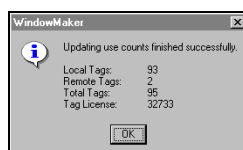
InTouch Cross Reference Utility

The InTouch Cross Reference Utility allows you to determine your tagname and SuperTag usage in animation links, wizards, InTouch QuickScripts, QuickFunctions, ActiveX controls, scripts and the following InTouch add-on programs: SPC Pro, SQL Access Manager and Recipe Manager. For all objects such as wizards, ActiveX controls and animation links, it displays the window name and the coordinates of all objects linked to the tagname. It also allows you to view any QuickScript or QuickFunction where a tagname is found. The InTouch Cross Reference Utility can remain open in WindowMaker while other tasks are performed.

Using the InTouch Cross Reference Utility

1. Click on the **Special/Cross Reference** command. The **InTouch Cross Reference Search Criteria** dialog box appears:

2. The **Search Criteria** group allows you to limit the scope of your search. The scope can be determined by selecting only the options required:
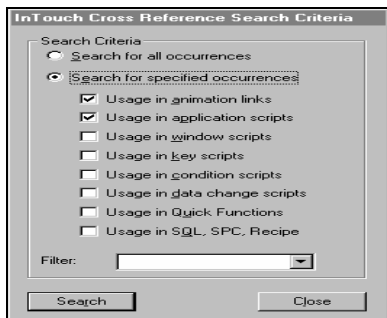
Search for all occurrences	Search for all uses of the tagname or SuperTag in animation links, InTouch QuickScripts and all add-on programs such as SPC, SQL Access Manager, Recipe Manager, etc.
Search for specific occurrences	Search for only the tagname or SuperTag in the specified options. For example, to search for the usage in window scripts, select **Usage in window scripts**.

3. In the **Filter** box, enter a unique name to identify the filter that you are defining or click the **Filter** arrow to select a previously defined filter from the list. (As you define filters, the name you enter is added to the **Filter** list.)

   The filter editor control allows you to enter a wildcard expression to limit the scope of the tagnames in your search. If no filter is used, the information for all tagnames in the current application will be acquired.

   The multiple wildcard is the asterisk symbol (*****). For example, "Asyn*" would search for all tagnames beginning with the characters "Asyn".

   The single character wildcard is the question mark (?). For example, the filter "Tag?" would search for all four character tagnames that begin with "Tag". The filter "Tag??" would search for all five character tagnames that begin with "Tag". Any sequence of valid InTouch tagname characters, together with the two wildcard characters, is acceptable in a filter. The valid tagname characters are: A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ and &.

4. Click **Search** to begin the cross reference search based on the specified view criteria.

Viewing the Cross Reference Search Results

When you perform a cross reference search, the **InTouch Cross Reference Utility** dialog box appears listing all instances of usage found for the **Filter** that you specified.

If no filter is used, all tagnames defined in the current application's Tagname Dictionary are displayed. For example:
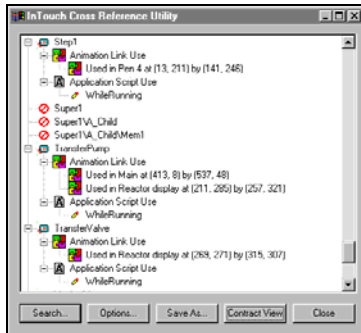
Cross Reference Utility Icons

The following briefly describes the various icons that may appear in the InTouch Cross Reference Utility:

Icon	Description
⊞	Tagname or SuperTag is assigned to an InTouch object or used to store a value in an InTouch QuickScript, wizard or add-on program. Click to expand the level's view.
⊟	Click to collapse an expanded level's view.
⊘	Displayed tagname or SuperTag is defined in the application's Tagname Dictionary, but it is not assigned to an object.
	Displayed tagname or SuperTag is used in either an animation link or InTouch QuickScript. Double-click or click ⊞ to expand the view.
	Displayed tagname or SuperTag is assigned to an animation link. Double-click, or click ⊞ to display the window name and the coordinates for object(s) in the window assigned to the animation link.
Ⓐ	Displayed tagname or SuperTag is used in an Application script. Double-click or click ⊞ to expand the view and display the type of Application script that uses the tagname or SuperTag.
✏	Displayed for all Application **On Startup**, **While Running**, and **On Shutdown** scripts; Window **On Show**, **While Showing**, and **On Hide** scripts, and Key **On Key Down**, **While Down**, and **On Key Up** scripts. Double-click the script to view it.
	Displayed tagname or SuperTag is used in a Window script. Double-click or click ⊞ to expand the view to display the name of the window with the script. Double-click any listed window name to view the script.
	Displayed tagname or SuperTag is used in a Data Change script. Double-click or click ⊞ to expand the view and then double-click any listed script to view it.
	Displayed tagname or SuperTag is used in a Condition script. Double-click, or click ⊞ to expand the view to display the script's condition and its type. For example, **$Hour==12 On True**. Double-click any listed script to view it.
	Displayed tagname or SuperTag is used in a Key script. Double-click or click ⊞ to expand the view and display the key assigned to the script and the script's type. For example, **F2 On Key Down**. Double-click any listed script to view it.
UDF	Displayed tagname or SuperTag is used in a QuickFunction. Double-click or click ⊞ to expand the view and display the QuickFunction that uses the tagname or SuperTag. Click to expand the view to display the name(s) of the QuickFunctions in which the tagname or SuperTag is used. Double-click any listed script to view it.
Ax	Displayed tagname or SuperTag is used in an ActiveX Event script. Double-click or click ⊞ to expand the view and display the ActiveX Event script.
	When cross referencing by **Window**, this icon precedes the window name in which the displayed tagname or SuperTag is used. Double-click or click ⊞ to view all tagnames used in the window.
SPC	Displayed tagname or SuperTag is used in a SPC Pro application. Double-click or click ⊞ to view the name of the SPC Dataset in which the tagname or SuperTag is used.
SQL	Displayed tagname or SuperTag is used in a SQL application. Double-click or click ⊞ to view the name of the SQL Bind List in which the tagname or SuperTag is used.

 Displayed tagname or SuperTag is used in a Recipe Manager application.

Changing the Cross Reference Search Criteria

If desired, after the initial cross reference search has been performed, you can narrow the search by modifying your original search options.

1. In the **InTouch Cross Reference Utility** dialog box, (displayed after you have performed your initial search), click **Options**. The **InTouch Cross Reference View Options** dialog box appears:
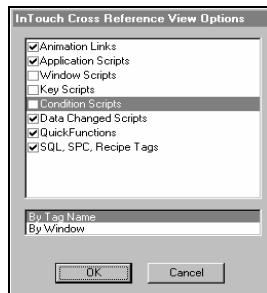
2. Select the search criteria options to modify for the new search. The options available here are based on the **Search Criteria** originally selected in the **InTouch Cross Reference Search Criteria** dialog box. If you selected **Search for all occurrences**, all search criteria options will be available. If you selected **Search for specific occurrences**, only the specific occurrences you originally selected will be available. To change the **Search Criteria** selection, click **Cancel**. The **InTouch Cross Reference Utility** dialog box reappears. Click **Search** and select the new **Search Criteria** option.

3. In the list at the bottom of the dialog box, select if the tree view is to be populated by tagname or window name and click **OK**.

Cross Referencing By Tagname

Cross referencing by tagname alphabetically lists all tagnames found for the specified search criteria (default view). Based on your specified search criteria, this view allows you to view the usage of all tagnames found in windows, animation links, scripts and add-on applications.

You can double-click a displayed tagname and then double-click **Animation Link Use** to expand the view. When the view is expanded, the window name and the location (coordinates) of the object(s) linked to the tagname are displayed.

You can double-click a tagname and then double-click any of its associated scripts to open it in the **Script usage for _Tagname_** dialog box:
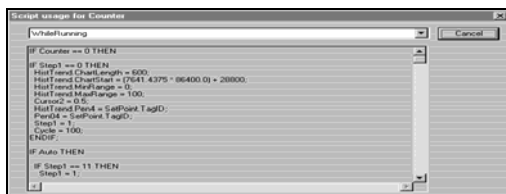
The list box at the top of the screen displays all scripts associated with the selected tagname. Click the arrow to open the list and select another script for viewing. For Application, Window, Key and Condition scripts, the list will contain the names of all scripts that use this tagname. For Data Change scripts, only the tagname is listed. For QuickFunctions, the list will contain the names of all QuickFunctions. Click **Cancel** to close the dialog box dialog box.

Cross Referencing By Window Name

Cross referencing by window name sorts the display by window name and then the tagnames used in the window. For example:
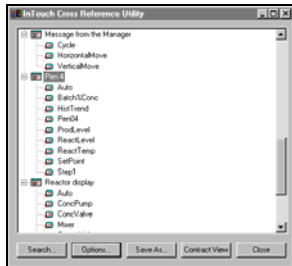
Click **Expand View** to display all view levels available for the displayed tagnames or windows. For example:
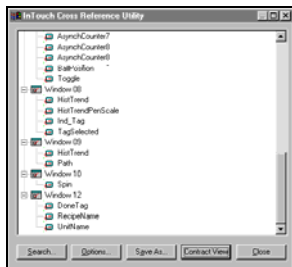
Click **Contract View** to return the dialog box to its default mode. Click **Close** to exit the InTouch Cross Reference Utility.

Saving Cross Reference Files

Cross reference files can be saved and viewed later in any text editor program that supports the comma separated variable (.csv) file format. The information stored in a cross reference file corresponds to the information currently displayed in the **InTouch Cross Reference Utility** dialog box.

1. In the **InTouch Cross Reference Utility** dialog box, click **Save As**. The **Save As** dialog box appears:
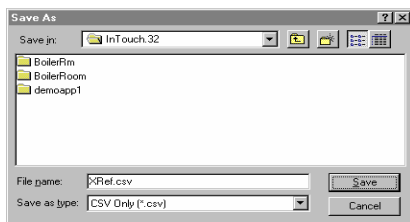
2. In the **File name** box, enter the name to save the cross reference file under. The file must be saved as a .csv file.

3. Click **Save**.

Printing Cross Reference Files

A cross reference .csv file can be opened in any text editor program that supports the .csv file format and then printed as a report.

For example, in Notepad, the file would appear as follows:
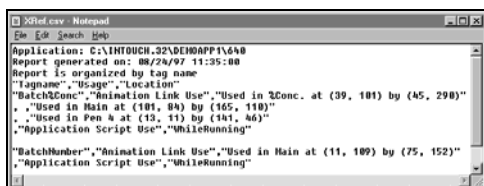
To print the file, click on the **File/Print** command.

Chapter : 5   Intouch QuickScript

Section : 1

QuickScript Types

InTouch QuickScript capabilities allow you to execute commands and logical operations based on specified criteria being met. For example, a key being pressed, a window being opened, a value changing, etc.

QuickFunctions are scripts that can be called from other scripts and animation link expressions. The reused code is stored in one script and in one location, allowing all script instances to be updated with one edit session.

All InTouch QuickScripts are event driven. The event may be a data change, condition, mouse click, timer, etc. The order of processing is <u>application specific</u>. While it may appear that there is some inherent order in the way multiple scripts initiated by the same event are scheduled, there is no guarantee of any specific order. Therefore, you should not build any dependency on the order of processing.

The following briefly describes the types of scripts you can create:

Script Type	Description
Application	Scripts linked to the entire application.
Window	Scripts linked to a specific window.
Key	Scripts linked to a specific key or key combination on the keyboard.
Action Pushbutton	Scripts associated with an object linked to a **Touch Link - Action Pushbutton**.
Data Change	Scripts linked to a tagname and/or **tagname.field** only.
Condition	Scripts linked to a discrete tagname or expression.
ActiveX Event	Scripts that execute ActiveX control events in runtime.
QuickFunctions	Scripts you create that can be called from other InTouch QuickScripts or animation link expressions. QuickFunctions can be both synchronous and asynchronous, while all other script types are synchronous only.

Application Scripts

Application Scripts are linked to the entire application. They can be used to start other applications, create process simulations, calculate variables, etc. There are three types of Application Scripts that can be applied to an application:

On Startup	Executes one time when the application is initially started up.
While Running	Executes continuously at the specified frequency while the application is running.
On Shutdown	Executes one time when the application is exited.

Accessing the Application Script Editor
Click on the **Special/Scripts/Application Scripts** command. In the Application Explorer under **Scripts**, double-click **Application**. Or, under **Scripts**, right-click **Application**, and then click **Open**. The **Application Script** editor appears:
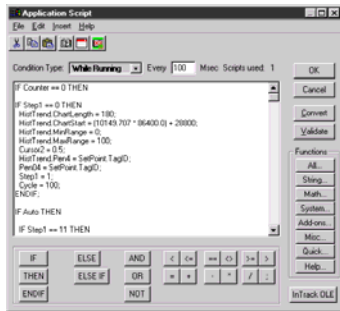
When a **While Running** script is selected, the **Every Msec** box becomes active. Enter the number of milliseconds that must elapse before the script executes. To execute the script immediately, create an identical **On Startup** script. As long as the condition or event for the **While Running** script is met, the script will repeatedly execute at the specified frequency.

Window Scripts

Window Scripts are linked to a specific window. There are three types of scripts that can be applied to a window:

On Show   Executes one time when the window is initially shown.

While Showing   Executes continuously at the specified frequency while the window is showing.

On Hide   Executes one time when the window is hidden.

Accessing the Window Script Editor

Click on the **Special/Scripts/Window Scripts** command. Or to quickly access the Window Script editor for a specific window, in the Application Explorer, under **Windows**, right-click the window name, and then click **Window Scripts**.
The **Window Script** editor appears:

When **While Showing** is selected, the **Every Msec** box becomes active. Enter the number of milliseconds that must elapse before the script executes. To execute the script immediately, create an identical **On Show** script. As long as the condition or event for the **While Showing** script is met, the script will repeatedly execute at the specified frequency.

Key Scripts

Key Scripts are linked to a specific key or key combinations on the keyboard. They can be used to create global keys for the application. For example, returning to a main menu window, logging off the operator, etc. There are three types of Key Scripts that can be applied to a key:

On Key Down   Executes one time when the key is initially pushed down.

While Down   Executes continuously at the specified frequency while the key is held down.

On Key Up   Executes one time when the key is released.

Accessing the Key Script Editor

Click on the **Special/Scripts/Key Scripts** command. The **Key Script** editor appears. Or, in the Application Explorer, under **Scripts**, you can also right-click **Key**, and then click **Open**.

When **While Down** is selected, the **Every Msec** box becomes active. Enter the number of milliseconds that must elapse before the script executes. To execute the script immediately, create an identical **On Key Down** script. As long as the condition or event for the **While Down** script is met, the script will repeatedly execute at the specified frequency.

Touch Pushbutton Action Scripts

Touch Pushbutton Action Scripts are similar to Key Scripts, except they are associated with an object linked to a **Touch Link- Action Pushbutton**. (The script editor is accessed through the animation link selection dialog box.) They are executed when the operator clicks or presses the object or button assigned to the link. There are three types of Touch Action Scripts that you can apply to an object:

On Key Down   Executes one time when the key is initially pushed down.

While Down   Executes continuously at the specified frequency while the key is held down.

On Key Up   Executes one time when the key is released.

Creating as Action Pushbutton Script

1. Draw the object or button to be linked to the script.

2. Double-click or select the object and then click on the **Special/Animation Links** command. The animation link selection dialog box appears.

3. In the **Touch Pushbutton** section, click **Action**. The **InTouch -> Action Script** editor appears:

Creating an Action Touch Pushbutton Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Touch Pushbutton** section, click **Action**. The **InTouch -> Action Script** editor appears:
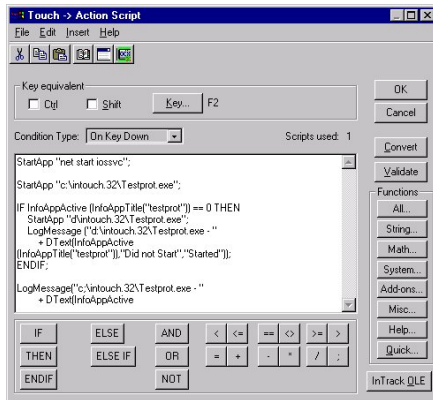
3. Click the **Condition Type** arrow and select the script type to apply to the object. You can apply all three script types to the same key:

    On Key Down Executes the script one time when the key is initially pressed.

    While Down Executes the script continuously on a time interval as long as the key is held down.

    On Key Up Executes the script one time when the key is released.

    A **While Down** script will begin executing after the specified number of milliseconds has elapsed. To cause immediate execution, create a duplicate **On Key Down** script.

4. Click in the script editor's window and enter the script to execute when the object is activated.

5. Click **OK** to attach the script to the object and return to the animation links dialog box.

Creating a Show (or Hide) Window Touch Pushbutton Link

1. Double-click the object. The link selection dialog box appears.

2. In the **Touch Pushbutton** section, click **Show Window**. The **Windows to Show when touched** dialog box appears:
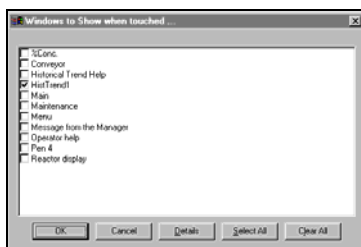
3. Select the window(s) to open when the object is clicked or touched.

4. Click **OK** to attach the link to the object and return to the animation links dialog box.

To change a window's type, right-click a blank area of the open window and then click **Window Properties**. The **Window Properties** dialog box appears and the type can be changed. (You cannot change the window's type if WindowViewer is running.)

Assigning a Key Equivalent to a Script

The Key Script and the Touch Action Script editors are different from the other QuickScript editors. Since you are creating scripts that apply to a key, you must specify the key(s) the operator will press to execute the script.

Assigning a Key to a Key Script

1. Select CTRL and/or SHIFT if the operator will need to hold down the CTRL and/or SHIFT keys and press the key to execute the script.

2. Click **Key** to assign a key to the script. The **Choose key** dialog box appears:
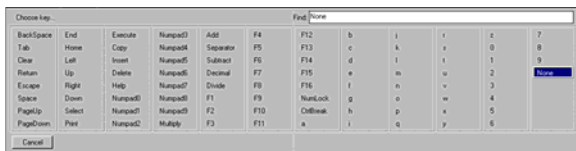
3. Click the desired key. The dialog box automatically closes and your selection is automatically entered in the **Key** box.

Condition Scripts

Condition Scripts are linked to a discrete tagname or expression that equates to true or false. Discrete expressions containing analog tagnames can also be used (see example below). There are four types of scripts that can be applied to a condition:

On True   Executes one time when the condition transitions to true.
On False   Executes one time when the condition transitions to false.
While True   Executes continuously while the condition is true.
While False   Executes continuously while the condition is false.

Accessing the Condition Script Editor

1. Click on the **Special/Scripts/Condition Scripts** command. The **Condition Script** editor appears:

2. Since Condition Scripts are executed based on a condition being met, you must specify the condition (a discrete tagname or expression) in the **Condition** box.

   All four script types can be applied to the same condition. Both **While True** and **While False** scripts will begin executing <u>after</u> the specified number of milliseconds have elapsed. To cause immediate execution, create duplicate **On True** and/or **On False** scripts. For example:
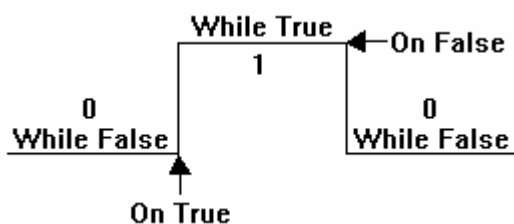
3. Enter any miscellaneous comments in the **Comment** box.

Data Change Scripts

Data Change Scripts are linked to a tagname and/or **tagname.field** only. They are executed one time when the value of the tagname or **tagname.field** changes by a value greater than the deadband defined in the Tagname Dictionary.

Accessing the Data Change Script Editor

1. Click on the **Special/Scripts/Data Change Scripts** command. The **Data Change Script** editor appears:
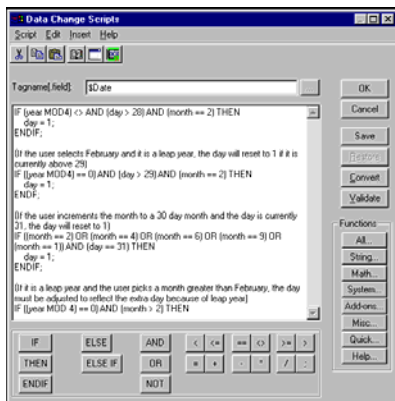
2. Since Data Change Scripts are executed based on a change in a data value, you must specify a tagname or **tagname.field** in the **Tagname[.field]** box.

3. Click on the **Insert/Tagname** command or double-click the script window. The **Select Tag** dialog box appears.

ActiveX Event Scripts

Most ActiveX controls have events associated with them. For example, click, double-click, mouse down, and key press are typical events used in many ActiveX controls. InTouch ActiveX Event scripts are provided to support event actions. One ActiveX Event script can be associated with each event. ActiveX control events are executed in runtime (WindowViewer).

QuickFunctions

QuickFunctions are scripts that can be called from other scripts or expressions. They are stored in the application in which they are created. Calling QuickFunctions from other scripts or expressions allows you to create a script one time and then reuse it. Reusing these scripts decreases your application maintenance by reducing the amount of duplicate code that is copied and pasted over and over into scripts. The reused code is stored in one script and in one location, thereby supporting updates of all script instances in one edit session.

Section : 2

Script Functions
Using the InTouch QuickScript Editor

The InTouch QuickScript editor is basically the same for all script types. Therefore, to avoid redundancy, its common functions and features are described in this section. The items that are unique to a script type are described in that script type's respective section later in this section.
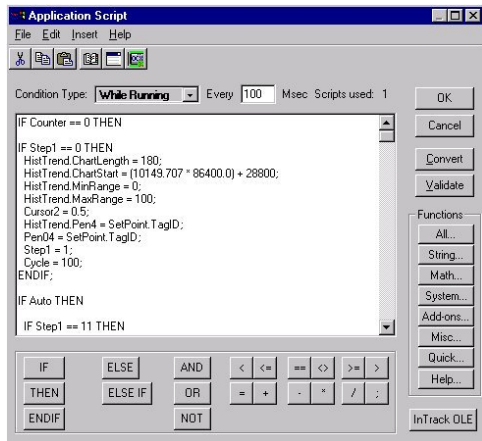
The QuickScript Toolbar

The QuickScript toolbar gives you quick access to editor functions:

Tool	Description
	Click to cut selected text from the script
	Click to copy selected text from the script
	Click to paste text into the script from the clipboard
	Click to view the Tagname Dictionary
	Click to Insert a window name from the current application
	Click to view the ActiveX Control Browser

All script dialogs can now be enlarged to increase the area for typing in scripts. Resize the dialogs' borders by dragging the borders, or click the maximize button on the toolbar.

QuickScript Editor Common Procedures
This section describes the generic procedures that you will use when writing scripts in the various InTouch QuickScript editor dialog boxes.

There are text, equivalency and mathematical operator buttons at the bottom of the QuickScript editor that can be used to quickly insert the displayed keyword, function or symbol into your script at the cursor location.

Indent/Unindent Script Text
Position the cursor at the beginning of the line to indent and then press the TAB key. To remove the indent, hold down the SHIFT key and press the TAB key.

Create a New Script
Click on the **Script/New** command.

Delete a Script

Select the text to delete and click on the **Script/Erase** command. The entire script is deleted from your application.

Undo Your Last Action

Click on the **Edit/Undo** command. Your last editing operation is reversed.

Select the Entire Script

Click on the **Edit/Select All** command. The entire script is selected.

Cut Selected Text from a Script

Select the text to remove and click on the **Edit/Cut** command. The cut text is deleted from the script and copied to the Windows Clipboard. You can now paste the cut text into another location in this script or into another script.

Copy Selected Text from a Script

Select the text to be removed and click on the **Edit/Copy** command. The text is copied to the Windows Clipboard. You can now paste the copied text into another location in this script or into another script.

Paste Text into a Script

Click on the **Edit/Paste** command. The contents of the Windows Clipboard is pasted into your script at the cursor location.

Clear Text in a Script

Click on the **Edit/Clear** command. All text in the script is erased. However, the script is not deleted from your application. If you select this command, close the script editor and then reopen it, the script will reappear.

To completely delete the script, you must use the **Script/Erase** command or select the entire script, right-click a blank area of the script window and click **Delete**.

Insert a Function into a Script

1. Click on the **Insert/Functions** command and then click the name of the function category. The respective **Choose function** dialog box appears.

2. Click the desired function. The dialog box will close and the function will automatically be inserted into your script at the cursor location.
   The types of functions available are:

Function	Description
All	The **Choose function** dialog box appears displaying all available functions including the functions for each installed add-on program (Recipe Manager, SPC Pro and SQL Access Manager).

String	The **Choose function** dialog box appears displaying all available string functions.
Math	The **Choose function** dialog box appears displaying all available mathematical functions.
System	The **Choose function** dialog box appears displaying all available system functions. For example, the functions to start and/or activate another application, read and/or write file and disk information, etc.
Add-ons	The **Choose function** dialog box appears displaying all available functions for each installed add-on program (Recipe Manager, SPC Pro and SQL Access Manager).
Misc	The **Choose function** dialog box appears displaying all available miscellaneous functions. For example, the functions for alarms, historical trending, windows controls, ActiveX controls, etc.
Quick Functions	The **Choose function** dialog box appears listing the names of all the QuickFunctions available for calling from the current script.
Help	The **Choose function to Obtain Help for** dialog box appears listing all available functions. Click a function to open its respective Help topic.

Insert a Tagname into a Script

1. Click on the **Insert/Tagname** command. The Tag Browser will appear in the unlimited selection mode.
2. Double-click the tagname or select it and click **OK**. The Tag Browser will close and the tagname will automatically be inserted into your QuickScript at the cursor location.

   To quickly access the Tag Browser, double-click a blank area in the QuickScript window.

   To access a specific tagname's definition in the Tagname Dictionary, enter the tagname in the QuickScript window and then double-click it.

Insert a Tagname .Field into a Script

1. Click on the **Insert/Tagname** command. The Tag Browser will appear in the unlimited selection mode.

2. Select the tagname and then click the **Dot Field** arrow. Select the **.field** to use with the tagname in the list.

3. Click **OK**. The selected tagname.**field** will be inserted into your QuickScript at the cursor location.

Find or Replace a Tagname in a Script

1. Click on the **Edit/Find** command. The **Replace** dialog box appears:
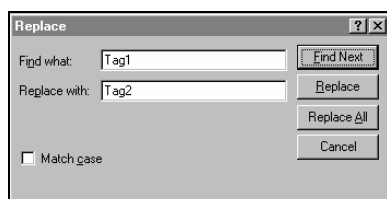
2. In the **Find what** box, enter the tagname to find (or replace) and click **Find Next**.

3. In the **Replace with** box, enter the new tagname to replace the old tagname and click **Replace** or **Replace All**.

4. Select the **Match case** option if you find specific upper or lowercase instances of the tagname.

5. Click **Cancel** to close the dialog box.

Insert a Window Name into a Script
1. Click on the **Insert/Window** command. The **Window Name to Insert** dialog box appears, displaying the names of all the windows in your application.

2. Click the desired window name. The dialog box will close and the window name will automatically be inserted into your script at the cursor location.

Validate a Script
Click **Validate** to verify that your script syntax is accurate. This can be done at any time.

Validation is automatically performed when you click **OK** or **Save**. If the system encounters errors when validating a script, a corresponding error message box will appear.

Save a Script

When writing multiple scripts, click **Save** to save the first one. Then click on **Script/New** command to write a new script.

Restore a Script
If you change a script and then decide to clear your changes and restore the original script, click **Restore**.

Exit the Script Editor
Click on the **Script/Exit** command. The script editor will close and the script will be saved unless an error is encountered.

You can also close the script editor by clicking **OK** once you have completed writing your script.

Specify a Script's Execution Frequency

In the **While *Running/Showing/Down* Every 0 Milliseconds** boxes, enter the number of milliseconds that must elapse before the script executes.
When creating an Application **While Running** script, Window **While Showing** scripts, Condition **While On True/On False** scripts or Key and Touch Pushbutton Action **While Down** scripts, you must specify the frequency (in milliseconds) that they will be executed.

IF-THEN-ELSE and Comparisons in Scripts

The IF-THEN-ELSE statement is used to conditionally execute various instructions based on the state of an expression. The following comparison operators are used to set up the conditions in an IF-THEN-ELSE statement:

< Less than
> Greater than
<= Less than or Equal to
>= Greater than or Equal to
== Equivalency ("is equivalent to")
<> Not Equal to

Below are some examples of complex scripts:

IF-THEN statement with no ELSE clause:

```
IF a <> 0 THEN
    a = a + 100;
ENDIF;
```

IF-THEN-ELSE statement with one ELSE clause:

```
IF temp > 500 THEN
    Disc = 1;
    Real = 43.7;
ELSE
    Disc = 0;
    Real = 93.4;
ENDIF;
```

IF-THEN-ELSE statement with one ELSE IF clause and no ELSE clause:

```
IF temp > 500 THEN
    Disc = Disc * 10;
ELSE
    IF temp > 250 THEN
        x = y / z;
        a = abc + def;
    ENDIF;
ENDIF;
```

IF-THEN-ELSE statement with one ELSE IF clause and one ELSE clause:

```
IF temp > 500 THEN
    Disc = Disc - 10;
ELSE
    IF temp < 250 THEN
        Disc = Disc + 10;
    ELSE
        Disc = Disc + 50;
        Real = 100;
    ENDIF;
ENDIF;
```

IF-THEN-ELSE statement with multiple ELSE IF clauses and one ELSE clause:

```
IF temp > 100 THEN
    temphihi = 1
    Disc = 50;
ELSE
    IF temp > 80 THEN
        temphi = 1;
ELSE
    IF temp < 10 THEN
        templo = 1;
ELSE
    IF temp < 30 THEN
        templolo = 1;
ELSE
        tempok = I;
    ENDIF;
ENDIF;
ENDIF;
ENDIF;
```

IF-THEN-ELSE statement that tests for Condition 1 *or* Condition 2:

```
IF (pump1 < 50.0) OR (pump2 < 50.0) THEN
    alarm-1 = 1;
ELSE
    alarm-1 = 0;
ENDIF;
```

IF-THEN-ELSE statement that tests for Condition 1 *and* Condition 2:

```
IF (pump1 < 50.0) AND (pump2 < 50.0) THEN
    alarm-2 = 1;
ELSE
```

```
    alarm-2 = 0;
ENDIF;
```

IF-THEN-ELSE statement that tests for equivalency:

```
IF a > 50 THEN
    IF b == 100 THEN
        c = 0;
    ENDIF;
ENDIF;
```

Built-in Script Functions

InTouch provides numerous built-in functions that can be linked to objects or pushbuttons or used in scripts to perform a multitude of tasks. For example, acknowledging alarms, hiding windows, changing the tagname being trended by a pen, etc.

These functions are accessible through the **Insert** menu or by clicking the various buttons in the **Functions** section of the Script Editor. Once you select a function in its respective dialog box, the function and its required arguments are automatically pasted into your script at the cursor location. After the function is pasted into your script, highlight the argument to modify and enter the new value.

Using Local Variables

Local variables can be declared within a script to store temporary results and create complex calculations with intermediate scripting values. This can be done without impacting your licensed tagname count.

Local variables or tagnames can be used interchangeably within the same script. However, local variables lose their value and meaning once the script ends. Tagnames are global and retain their values. Unlike tagnames, local variables are declared within the body of the script. The number of local script variables can be declared within a given script body is limited only by your available memory. Once you have declared a local variable, it can be included in one or more expressions within the same script body. The expression and syntax rules for the placement of local variable names within a script body are the same as those for tagnames, with one exception. Local variables do not support **.field** references.

Like tagnames, local variables can be used on both the left and right side of statements and expressions that include other local variables and tagnames of different data-types.

Valid Local Variable Syntax
Each local variable must be declared within the script as a separate **DIM** statement. (One per line - cascading is not permitted.) The **DIM** statement syntax and format are as follows:

DIM LocalVarName [AS data-type];

Where:

DIM　　　　Required keyword.

LocalVarName　　Variable name that conforms to tagname format and restrictions. Variable names can be up to 32 characters long and must begin with A-Z or a-z. The remaining characters can be: A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ and &.

AS　　　　Optional keyword.

If you omit the AS clause from the DIM statement, by default, the variable will be declared as an integer data-type.
For example:

DIM LocVar1;

is equivalent to:

DIM LocVar1 AS Integer;

data-type   Can be any one of the following keywords:

Integer	**DIM LocVar1 AS Integer;**
Real	**DIM LocVar2 AS Real;**
Discrete	**DIM LocVar3 AS Discrete;**
Message	**DIM LocVar4 AS Message;**

The InTouch DIM statement cannot be cascaded. For example, the following examples are invalid and cannot be used:

DIM LocVar1 AS Integer, LocVar2 AS Real;

DIM LocVar3, LocVar4, LocVar5, AS Message;

To declare the multiple variables in InTouch, you must enter a separate DIM statement for each variable. For example, the following examples are valid:

DIM LocVar1 AS Integer;

DIM LocVar2 AS Real;

Importing QuickScripts

Importing QuickScripts from one InTouch application to another can save development time. It allows you to reuse your previously created QuickScripts. To move QuickScripts from one InTouch application to another, you <u>must</u> use the **File/Import** command.

1. Close all windows in your current application.

2. Click on the **File/Import** command. The **Browse for Folder** dialog box appears:
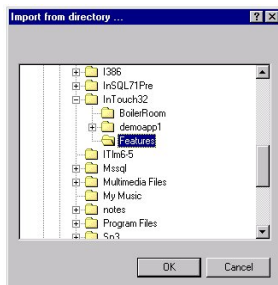
3. Select the application directory (folder) containing the QuickScript(s) to import.

4. Click **OK**. The following dialog box appears:
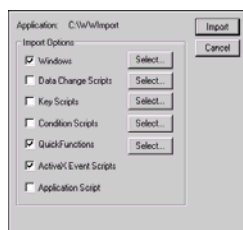
5. Select the QuickScript type(s) to import.
6. Click **Select**. The **Select a *ScriptType* Script** dialog box appears:

7. Select the QuickScript(s) to import and then click **OK** to close the dialog box.

8. Click **Import**. The system will automatically begin to import the selected QuickScript(s) into your current application.

Converting Placeholder Tagnames

When a QuickScript is imported into a new application, all of the tagnames in the QuickScript are imported with it, but they are not added to your Tagname Dictionary. Instead, they are automatically converted to "placeholder" tagnames. You must convert the placeholder tagnames in order to use them. If they are not currently defined in the new application's Tagname Dictionary, you will be asked to define each of them.

When the tagnames in an imported QuickScript are converted to placeholder tagnames, three index characters are added to the beginning of each tagname. For example, when a discrete tagname is imported, the tagname is prefixed with the three characters **?d:**. When a tagname of 30, 31 or 32 characters in length is imported, the three indexing characters will still be added to the beginning of each tagname. However, the addition of these three characters will <u>not</u> truncate the length of your existing tagname. For example, for placeholder tagnames only, a 32 character tagname is increased to 35 characters. These three additional spaces are allotted <u>only</u> for placeholder tagnames. This increase in tagname length is not supported for standard tagnames.

1. Click on the **Special/Scripts** command and then click the type of QuickScript you imported. The QuickScript editor will appear displaying the first QuickScript on file for the type of script you selected. For example, if you imported QuickScript functions, the **QuickFunctions** script editor appears:
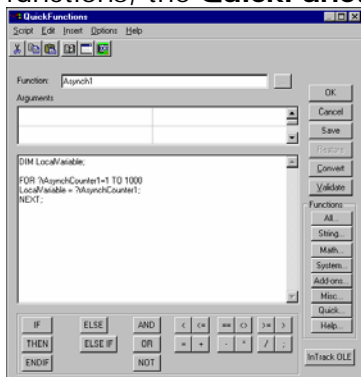
2. Click **Convert**. The **Convert** dialog box appears:
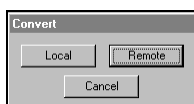
3. Click **Local** to convert the tagnames in the QuickScript to local tagnames.

4. After the tagnames are converted, click **OK** in the QuickScript editor.

Printing Scripts

All scripts in each InTouch QuickScript category can be printed.

1. Click on the **File/Print** command. The **WindowMaker Printout** dialog box appears.

2. To print window scripts, select **Windows**, and then select **Window Scripts**. In the **Which Windows?** group, select **All** to print the scripts for all windows in the application. To print a specific window's script, select **Selected**. The **Windows to Print** dialog box appears. Select the window(s) whose script you want to print and then click **OK**.

3. To print all scripts for a QuickScript type, select the QuickScript type and then click **OK**.

Chapter : 6   Alarms and Events

Section : 1

Alarms

InTouch has two types of notifications to inform operators of process activity: Alarms and Events. Alarms represent warnings of process conditions that could cause problems and require an operator response. A typical alarm is triggered when a process value exceeds a user-defined limit, such as an analog value exceeding a hi-limit threshold. This triggers an *unacknowledged* alarm state, which can be used to notify the operator of a problem. If configured to do so, InTouch can also log this alarm to a disk-based file and send it to a printer. Once the operator acknowledges the alarm, the system returns to an *acknowledged* state.

Any tagname can be configured to perform event monitoring while it is being defined in the Tagname Dictionary. An event message is logged to the alarm system each time the tagname's value changes. The event message logs how the value changed and whether the change was initiated by the operator, I/O, scripts or the system.

Alarm Types

InTouch classifies alarms into several general categories based on their characteristics. These categories are known as *Type* and *Class*. The standard alarm system categorizes all alarms into five general *Types*: Discrete, Deviation, Rate-of-Change, Value, and SPC. The table below summarizes the classification for the standard alarm system:

Alarm Condition	Standard Type
Discrete	DISC
Deviation - Major	LDEV
Deviation - Minor	SDEV
Rate-of-Change	ROC
SPC	SPC
Value - LoLo	LOLO
Value - Low	LO
Value - High	HI
Value - HiHi	HIHI

Each alarm can be associated with an InTouch tagname. Depending on a tagname's type, one or more of the alarm classes or types can be defined for it. Alarm conditions are defined in the Tagname Dictionary.

Alarm logging, printing and a standard display can also be configured to show the alarm *Type* field.

Alarm Priorities

Each alarm configured in InTouch has a priority value associated with it. This value represents the severity of the alarm and can range from 1 to 999 with 1 being the most severe. By creating alarm ranges using these priorities and assigning alarms to each, you can easily filter out critical alarms from non-critical alarms. Animation links, acknowledgment scripts and filtered viewing/printing can also be created, all based on the priority range.

For example, if a process plant has determined that they need four levels of severity, the following ranges could be configured:

Alarm Severity	Priority Range
Critical	1 - 249
Major	250 - 499
Minor	500 - 749
Advisory	750 - 999

As the plant engineers create InTouch tagnames and alarm conditions, each alarm will be assigned to one of these severity levels by choosing a priority number within that range. With these ranges configured, the plant operators can now easily display/print certain severity levels.

Alarm Groups

Each InTouch alarm is assigned to a logical Alarm Group. These groups are user-definable and can be arranged in a hierarchy up to eight levels deep. The groups provide a way of categorizing alarms based on an organization, plant layout or any other metric. Alarm Groups are useful for filtering alarm displays, alarm printers and acknowledgment scripts.

Each tagname is associated with an Alarm Group. If an Alarm Group name is not associated with a tagname, by default, InTouch automatically associates with the root group, **$System**. Any Alarm Group can have both tagnames and other Alarm Group names associated with it. Alarm Groups are organized into a hierarchical tree structure with the root group **$System** at the top of the tree. All defined Alarm Groups automatically become descendants of the root group.

This tree may have up to eight levels. Each Alarm Group may have a maximum of 16 subgroups. Each subgroup may have a maximum of 16 subgroups, etc., until the maximum of 8 levels is reached.
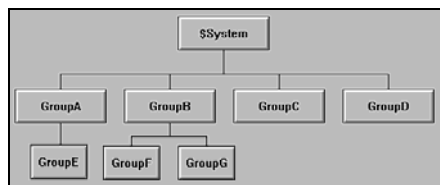
The previous illustration displays only Alarm Groups, not the tagnames within each group. The tree concept is analogous to the MS-DOS directory structure, where a directory may contain other sub-directories (analogous to groups) and file names (analogous to tagnames).

The distributed alarm system also uses these groups as the basis for its Alarm Group Lists.

Creating an Alarm Group

1. Click on the **Special/Alarm Groups** command. The **Alarm Group Definition** dialog box appears:

   You can also create Alarm Groups and associate tagnames with them while you are defining your tagnames in the Tagname Dictionary.

2. Click **Add**. The **Add Alarm Group** dialog box appears:

The **Modify** and **Delete** buttons are not available until an Alarm Group is defined. The **$System** Alarm Group cannot be modified or deleted.
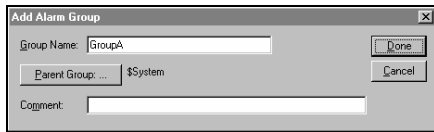
3. In the **Group Name** box, enter the name for the new Alarm Group. Since this is the first Alarm Group you have created, it is automatically assigned to the **$System** Parent Group.

   After you have created an Alarm Group, it can be used as a Parent Group.

4. Click **Parent Group** to assign your Alarm Groups to a different Parent Group. The **Alarm Group Selection** dialog box appears:
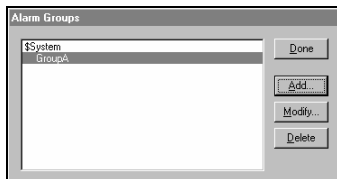
5. In the **Select an Alarm Group** list, double-click the name of the Alarm Group to use as the Parent Group for the new Alarm Group (or select it and then click **Done**). The **Add Alarm Group** dialog box reappears displaying the selected Parent Group. For example:

6. In the **Comment** box, enter any comment for the new Alarm Group.

7. Click **Close**. The **Alarm Group Definition** dialog box appears displaying your Alarm Group hierarchy:
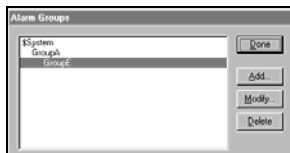
8. Click **Done**.

Deleting an Alarm Group

1. Click on the **Special/Alarm Groups** command. The **Alarm Group Definition** dialog box appears:

   You can also delete Alarm Groups while you are defining your tagnames in the Tagname Dictionary.
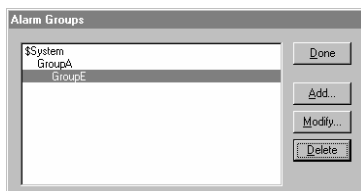
2. Select the Alarm Group to delete in the list and then click **Delete**.   A message box will appear asking you to confirm the deletion. Click **Yes** to delete the Alarm Group.

Modifying an Alarm Group

1. Click on the **Special/Alarm Groups** command. The **Alarm Group Definition** dialog box appears:

2. Select the Alarm Group to modify in the list and then click **Modify**. The **Modify Alarm Group** dialog box appears:
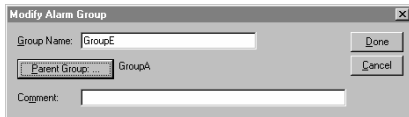
3. Make the required changes to the Alarm Group. Click **Parent Group** to change the parent group for the Alarm Group. The **Alarm Groups** dialog box appears:
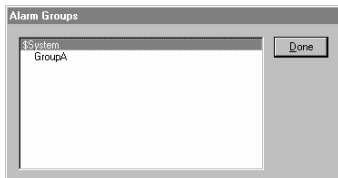
4. Select the new parent group and then click **Done**. The **Modify Alarm Group** dialog box reappears displaying the new parent group.
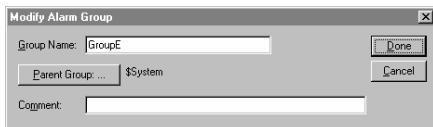
5. Click **Done**.

Defining Tagname Alarm Conditions

Alarm conditions for tagnames can be defined at the same time that tagnames are defined. There are two types alarm detail dialog boxes: one for discrete type tagnames and one for analog (integer or real) type tagnames.

Defining Discrete Tagname Alarm Conditions
An alarm condition can be defined for a discrete type tagname's **On** state or **Off** state:

1. Click on the **Special/Tagname Dictionary** command. The **Tagname Dictionary** dialog box appears.

2. Click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog box to display the discrete alarm details dialog box:
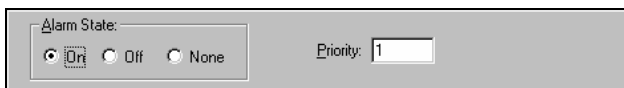
3. Click the **Alarm State** the tagname is to be in when in alarm.

4. In the **Priority** box, enter a value between 1 and 999 (1 is the highest priority and 999 is the lowest). You can use this priority value to select the alarms to be displayed in a window, logged to disk or printed.

5. Click **Close** (in the **Tagname Dictionary** dialog box) to save your tagname
    definition and close the tagname dialog boxes.

Defining Analog Tagname Alarm Conditions

1. Click on the **Special/Tagname Dictionary** command. The **Tagname Dictionary**
    dialog box appears.

2. Click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary**
    dialog box to display the analog alarm details dialog box:

3. Select the alarm types (**LoLo**, **Low**, **High**, **HiHi**) to detect when the value of an
    analog type tagname is beyond an absolute limit.
4. In the **Alarm Value** box, enter the limit value for the alarm.

    For example, in the case of **LoLo** and **Low** alarms, an alarm condition exists
    whenever the value of the tagname is less than the **Alarm Value**. In the case of
    High and **HiHi** alarms, an alarm condition exists whenever the value of the tagname
    is greater than the **Alarm Value**. These fields support the use of real numbers
    (e.g., 100.75).

5. In any of the **Pri** (priority) boxes, enter a number between 1 and 999 (1 is the
    highest priority and 999 is the lowest). You can use the priority value to select the
    alarms to be displayed in a window, logged to disk or printed.

6. In the **Value Deadband** box, enter the number of engineering units the tagname
    value must drop below the alarm value before it is taken out of alarm.

    For example, to return-to-normal from an alarm condition, a tagname value must
    not only return inside its alarm limit, but also return through your specified **Value
    Deadband**. The **Value Deadband** prevents "nuisance" alarms caused by repetitive
    re-annunciation of an alarm (where the tagname value 'hovers' around the limit,
    continually hopping in and out of alarm).

7. Select the deviation (**Minor** and **Major Deviation**) alarm types to detect when the
    value of an analog type tagname is in a major or minor deviation from the specified
    Target value.

8. In the **%Deviation** box, enter the percentage that the analog tagname can deviate
    from the **Target** value to produce a minor or major deviation alarm condition. It is
    expressed as a percentage of the range of the tagname. The range is defined by the
    Min EU and **Max EU** values entered in the tagname's details dialog box.

9. In the **Target** box, enter the desired or reference value of the tagname from which
    minor and/or major deviation percentages are based.

10. In the **Deviation Deadband %** box, enter the deviation percentage the tagname
    value must drop below the target before it is taken out of alarm. For example, let's
    assume the following setup for an integer tagname:

    Minimum Value = -1000
    Maximum Value = 1000
    Minor Deviation % = 10
    Major Deviation % = 15
    Target = 500

    To calculate at what value the Minor or Major Deviation alarm will take place if the
    total range of the tagname is **-1000 to +1000 or 2000**, multiply **2000** by either

the Minor or Major Deviation percentage **(2000 x .10 (Minor) =200)**. If the Target is **500**, a Minor Deviation will occur whenever the tagname's value drops below **300** or rises above **700**.

11. Select **Rate of Change** to detect when the value of an alarm changes an excessive amount for a specified time interval. The tagname is tested for a **Rate of Change** alarm whenever its value changes. At this time, the change rate is calculated using the previous value, the time of the last update, the current value, and the current time. This is compared to the rate of change allowed in the alarm definition. If the rate of change is greater than the alarm limit, the **Rate of Change** alarm condition is set for the tagname. A **Rate of Change** alarm remains in effect until the next change in the tagname is less than the excessive change amount for the time interval.

12. In the **% per** box, enter the maximum allowable percentage change.

13. Select **Sec**, **Min**, or **Hr** for the time interval units of the change.

Standard Alarm Display

The standard alarm system provides a unique display object to show locally generated alarms. The standard alarm display uses two predefined display types: Alarm Summary and Alarm History. The Alarm Summary only displays the current unacknowledged and acknowledged alarms. If an alarm returns to normal (RTN), the entry is removed from the display (if you have configured it to do so). No events are displayed with an Alarm Summary. The Alarm History object displays all of the alarm and events that have occurred. The Alarm History display shows the occurrence of the alarm, the time of acknowledgment (if any) and the time the alarm condition returned to normal.

In both the Alarm Summary and the Alarm History display objects, each entry is shown as a separate line. The number of entries displayed is determined by the size you have drawn the object and the size of the font that you are using. The standard alarm display lists all active alarms or subsets of active alarms as determined by the current value of the Alarm Group and priority expression associated with the particular alarm display.

InTouch allows you to configure the number of alarms stored for the Alarm History object and the appearance of the alarm displays (including the information that is displayed, logged and printed). You can also select the colors used for the title bar, title text, the background of the alarm display and the colors used to display the various alarm conditions in the window. In addition, the Alarm Group and alarm priority levels displayed can be dynamically controlled at runtime.

Creating a Standard Alarm Display

1. Click the wizard tool [icon] in the **Wizard/ActiveX Toolbar**.

2. Select the **Alarm Displays** category in the list of wizards to display both alarm wizards.

3. Double-click the **Standard Alarm Display** wizard or select it and then click **OK**. The dialog box closes and your window reappears with the cursor in the "paste" mode. Click in the window to paste the alarm display:
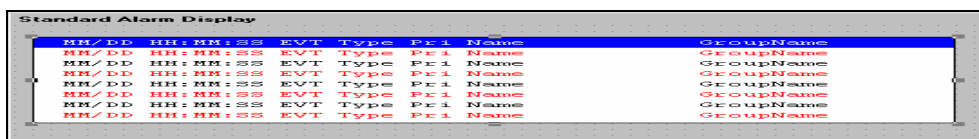
4. The alarm display object can now be configured.

Configuring a Standard Alarm Display

The first time an alarm object is pasted, the system default configuration settings are used. Once an alarm object has been configured, the next one created will, by default, be configured with the same settings.

1. Double-click the alarm display or, with the alarm display selected, click on the **Special/Animation Links** command. The **Alarm Configuration** dialog box appears:
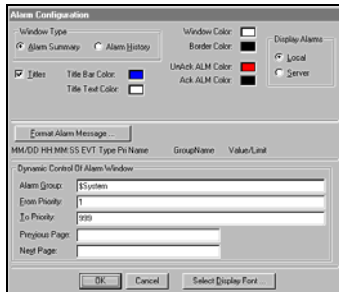
2. Select the **Window Type** for the alarm display to create:

    Alarm Summary   Displays a summary of all of the currently active alarms
    Alarm History   Displays a history of alarm events.

3. Select **Titles** to display a title bar with labels for each column on the alarm display. (Selecting this option activates the **Title Bar Color** and **Title Text Color** selection boxes.)

4. Click the **Title Bar Color** box to display the InTouch color palette. Select a color for the object's title bar. Repeat this process for all displayed color selections.

5. Select the **Display Alarms** type:

    Local   To display locally generated alarms and events.
    Server   To display the alarms/events collected by the server node.

    The server node is defined in the **WindowViewer Properties - General** property sheet.

6. To define a specific group of alarms to be logged, enter an Alarm Group name or the name of a Group Variable in the **Alarm Group** box. To log all Alarm Groups, enter **$System**.

    To control the choice of alarms logged at runtime, create a Group Variable type tagname (e.g., **ALARMGRP**). Then configure a key or action button script to assign a specific Group Name to the Group Variable. For example, the following would be entered in the QuickScript:

    ALARMGRP.Name="AlarmGroupName";

7. In the **From Priority** and **To Priority** boxes, enter the highest and lowest alarm priority level for the range of priorities to show in the alarm display.

    An analog tagname or an expression can be entered in either of the priority boxes if the tagname's value is to determine the priority level to be logged. The alarm priority level can be logged by assigning a value to this tagname through an analog input link or QuickScript.

8. In both the **Previous Page** and **Next Page** boxes, enter the discrete tagname to use to page up and down through the list of alarm messages when there are more alarms than the window can display.

9. Click **Select Display Font** to open the **Font** dialog box. Choose the font, style and size for alarm display object messages.

    The standard alarm display requires the use of fixed-pitch fonts (as opposed to variable-pitch or proportionally spaced fonts). This allows the entries to maintain

their column format appearance in the display. Therefore, only the fixed-pitch fonts will appear as possible selections for the alarm window.

10. Click **Format Alarm Message** to configure the various items to be shown for each alarm message in the alarm display.

   After your alarm messages have been formatted, check your display object to make sure that you have drawn it large enough to show all of your choices. If it is not large enough, the text at the right side of the object will be truncated.

11. Click **OK** to save your settings and close the **Alarm Configuration** dialog box.

Formatting Standard Alarm/Event Messages

The information shown in an alarm display object, logged to disk or printed is configurable. This configuration process involves selecting the information to display and, in some cases, the number characters to display.

The order of appearance of the items in the message is fixed and cannot be altered.

1. Double-click the alarm display. The **Alarm Configuration** dialog box appears.

2. Click **Format Alarm Message**. The **Format Alarm Message** dialog box appears:
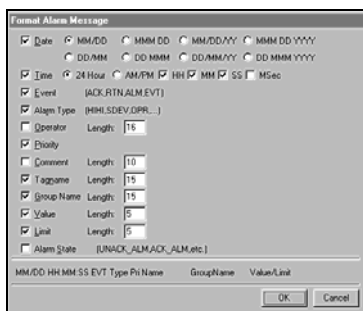
3. Select **Date** to display the date in the alarm message. Select the format for the date as follows:

Selection	Display	Selection	Display
MM/DD	02/28	**MM/DD/YY**	02/28/97
DD/MM	28/02	**DD/MM/YY**	28/02/97
MMM DD	Feb 28	**MMM/DD/YYYY**	Feb 28 1997
DD MMM	28 Feb	**DD/MMM/YYYY**	28 Feb 1997

4. Select **Time** to display the time in the alarm message. Select the format for the time as follows:

24 Hour	Selects the 24 hour military time format (e.g., three o'clock in the afternoon is displayed as 15:00).
AM/PM	Selects the AM/PM format (e.g., three o'clock in the afternoon is displayed as 3:00 PM).
HH	Displays the hour the alarm/event occurred.
MM	Displays the minute the alarm/event occurred.
SS	Displays the second the alarm/event occurred.
MSec	Displays the millisecond the alarm/event occurred.

5. Select **Event** to display the event type. Event types include:

ACK	Displays when the alarm has been acknowledged.
RTN	Displays when the alarm condition returns to normal.
ALM	Displays when the tagname is in alarm state.
EVT	Displays when the tagname's value is changed more than the deadband by either the operator, I/O, a QuickScript or the system.

6. Select **Alarm Type** to display the alarm type. Alarm types include:

HIHI, etc.	Displays for Alarm Value conditions.

SDEV	Displays for Minor Deviation Alarm conditions.
LDEV	Displays for Major Deviation Alarm conditions.
OPR	Displays if an operator change caused the alarm condition.

7. Select **Operator** to display the logged-on operator's ID associated with the alarm condition. Enter a value in the **Length** box to control the number of characters displayed (16 characters maximum).

8. Select **Priority** to display the alarm priority.

9. Select **Comment** to display the tagname's comments. These are the comments that were entered in the **Comments** box when the tagname was defined in the database. In the **Length** box, enter the number of characters to display (50 characters maximum).

10. Select **TagName** to display the tagname. In the **Length** box, enter the number of characters to display (32 characters maximum).

11. Select **Group Name** to display the Alarm Group name. In the **Length** box, enter the number of characters to display (32 characters maximum).

12. Select **Value** to display the value of the tagname when the alarm occurred. In the **Length** box, enter the number of characters to display.

    The value should be large enough to provide the desired level of precision (15 characters maximum).

13. Select **Limit** to display the alarm limit value of the tagname. In the **Length** box, enter the number of characters to display.

    The size of this field should be large enough to provide the desired level of precision (32 characters maximum).

14. Select **Alarm State** to display the state (unacknowledged, acknowledged, etc.) of the alarm.

Configuring the Standard Alarm System

Various parameters for the alarm system can be configured, such as printer and logger buffer size, event enabling, new alarm positioning in the alarm display, etc.
Alarm/event logging and printing properties can also be controlled.

Configuring Alarm/Event General Properties

1. Click on the **Special/Configure/Alarms** command. The **Alarm Properties** dialog box appears with the **General** properties sheet active:
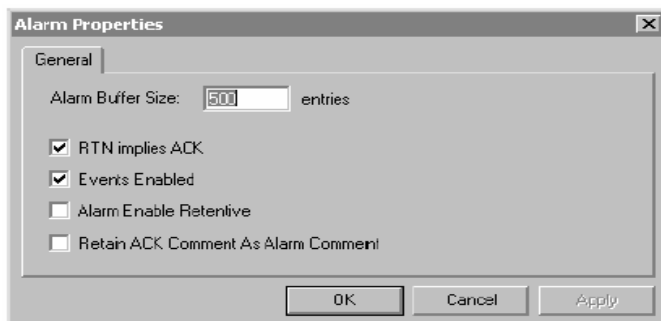
2. In the **Alarm Buffer Size** box, enter the number of "in-memory" alarm events WindowViewer needs to maintain. (The maximum number of alarms that the node can store for summary or history queries.)

Only "in-memory" alarm events can be displayed in alarm display objects. If alarms are not being used, this value may be set to 1 to conserve memory.

3. In the Printer Buffer Size box, enter the number of buffer bytes to be used by WindowViewer for parallel printers.If you are using a serial printer for alarm printing, this entry will have no effect. Increase the default number (2048) only when experiencing problems with printer overflow.

4. In the **Update Frequency of Printer/Logger** box, enter the number of milliseconds that WindowViewer should wait before trying to print alarm messages again when a printer is offline.

5. Select **RTN implies Ack** to automatically acknowledge (ACK) alarmed tagnames that return to the "normal" state (RTN). Do not select this option if the operator is to acknowledge an alarm after it returns to normal.

6. Select **Position New Alarms as End of Alarm Window** to display new alarms at the end of the alarm display object. Alarm windows provide the ability to page back and forth through the alarm queue. Enabling this option will cause the alarm display object to automatically scroll forward to show the new alarm. If this option is not enabled, the new alarm will be added to the bottom of the list, but the alarm display object will only scroll forward by one line.

7. Select **Events Enabled** to enable event logging of all data changes initiated by the operator, I/O, QuickScripts or the system. (Only tagnames with **Log Events** selected will be effected.)

8. Select **AlarmEnable Retentive** to retain the state of the **.AlarmEnable** variable when WindowViewer is closed.

9. Select **Use Tag Comment Field for Alarm Comments** if you are using distributed alarming and want the distributed alarm system to use the comments in the Tagname Dictionary for alarm acknowledgment comments.

10. Select **Expanded Summary Alarms** to generate a record of an alarm. When an alarm occurs it generates a record in the alarm display object showing that an alarm condition has occurred with the date and time stamp of the alarm. This record does not leave the display until an operator has acknowledged the alarm and a RTN has occurred.

11. Click **OK** to save your settings and close the dialog box.

Acknowledging Alarms

Local alarms can be acknowledged by using the **.Ack** (**.field**) in an action or key script.

Creating an Alarm Acknowledge Button

1. Create a 3-D button or any other object to which an action or key script can be linked.

2. Double-click the object to display the animation link selection dialog box.

3. In the **Touch Pushbuttons** section, click **Action**. The QuickScript editor appears.

4. Enter any of the following statements for the QuickScript:

Ack $System;	Acknowledges all local alarms in the system.
Ack *Group Name*;	Acknowledges all local alarms in a specific Alarm Group.
Ack *Group Var*;	Acknowledges all local alarms in a group indicated by the value of the Group Variable, an indirect Alarm Group tagname type.
Ack *Tagname*;	Acknowledges a specific tagname's alarms.

$System.Ack=1;   Acknowledges all local alarms in the system.

GroupName.**Ack=1;**   Acknowledges all local alarms in a specific Alarm Group.

GroupVar.**Ack=1;**   Acknowledges all local alarms in a group indicated by the value of the Group Variable, an indirect Alarm Group tagname type.

Tagname.**Ack=1;**   Acknowledges a specific tagname's alarms.

5. Click **OK**.

Alarm .Fields

InTouch provides various alarm **.fields** (dot fields) that allow you to dynamically control and/or monitor various alarm conditions. Many of these .fields are accessible using I/O, expressions and/or scripts. I/O access provides the ability to monitor and/or control a specific tagname's alarm information using other Windows applications (e.g., Excel or a remote View application).

If you create an analog alarmed tagname called **Analog_Tagname**, it will have "attributes" associated with it such as its name, its **HiHi** setpoint, etc. Some of these "attributes" are accessible through logic scripts, expressions and user inputs and are known as **.fields**.

The syntax required to access these data fields associated with a tagname is **Tagname.field**. The following briefly describes how to use each alarm **.field**.

.Field	Description
.Ack	Monitors/controls the alarm acknowledgment status.
.Alarm	Signals that an alarm condition exists.
.AlarmClass	Returns the class of the alarm.
.AlarmComment	Returns the comment information for the alarm.
.AlarmDate	Returns the date of the alarm.
.AlarmDevDeadband	Monitors/controls the deviation percentage deadband for both minor and major deviation alarms.
.AlarmEnable	Disables/enables events and alarms.
.AlarmGroupSel	Returns the alarm group of the alarm.
.AlarmLimit	Returns the limit of the alarm.
.AlarmName	Returns the name of the alarm.
.AlarmOprName	Returns the operator for the alarm.
.AlarmOprNode	Returns the operator node for the alarm.
.AlarmPri	Returns the priority value (1-999) for the alarm.
.AlarmProv	Returns the provider for the alarm.
.AlarmState	Returns the state of the alarm.
.AlarmTime	Returns the time of the alarm.
.AlarmType	Returns the type of the alarm.
.AlarmValDeadband	Monitors/controls the value of an alarm's deadband.
.Alarm Value	Returns the value of the alarm.

.DevTarget	Monitors/controls the target for minor and major deviation alarms.
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	Read/write analog tagname .fields that monitor/control the limits for value alarm checks. These .fields are only valid for integer and real tags.
.HiStatus, .HiHiStatus, .LoStatus, .LoLoStatus	Read only discrete tagname .fields that determine whether an alarm of a specified type exists.
.MajorDevPct	Read/write integer tagname .field that monitors or controls the major percentage of deviation for alarm checking.
.MajorDevStatus	Read only discrete tagname .field that determines whether a major deviation alarm exists for the specified tagname.
.MinorDevPct	Read/write integer tagname .field used to monitor and/or control the minor percent of deviation for alarm checking.
.MinorDevStatus	Read only discrete tagname .field used to determine whether a minor deviation alarm exists for the specified tagname.
.Name	Read/write message tagname .field used to display the actual name of the tagname. For example, it can be used to determine the name of an Alarm Group that a Group Variable is pointing to or the name of a TagID tagname. It can also be written to in order to change the Alarm Group that a Group Variable is pointing to.
.Normal	Read only discrete tagname .field that is equal to 1 when there are no alarms for the specified name. This .field is valid for Alarm Groups and Group Variables as well as ordinary tagnames.
.ROCPct	Read/write .field used to monitor and/or control the rate of change for alarm checking.
.ROCStatus	Read only discrete .field used to determine whether a Rate-of-Change alarm exists for the specified tagname.

Section : 2

Events

Events represent normal system status messages and do not require an operator response. A typical event is triggered when a certain system condition takes place, such as an operator logging into InTouch.

Events provide a way to track infrequent changes to the value of a tag (logons, setpoint changes, start times etc.). Events are grouped with alarms in that they can be viewed in an alarm window object, stored in an ASCII file or sent to a printer. Like alarms, they can also be sorted by group or priority. The difference between events and alarms is that when a tag reaches an alarm state, the .alarm field (along with certain other .fields) for that tag is set to true. The .alarm field for the group that the tag is a member of or a sub-member of is also set to true. An event does not affect its own .alarm field or the .alarm field of any other tag. It is simply an event that is recorded. Discrete, analog and string tags can be event logged. If configured to do so, InTouch can log an event to a disk-based file and send it to a printer.

Event Types

InTouch classifies events into general categories based on their characteristics. These categories are known as *Event Types*. The table below summarizes the classifications:

Event	Condition
ACK	Alarm was acknowledged
ALM	Alarm has occurred
EVT	Alarm event occurred
RTN	Tagname returned from an alarm state to a normal state
SYS	System event occurred
USER	$Operator changed
DDE	Tagname value was poked from a DDE client
LGC	QuickScript modified the tagname value
OPR	Operator modified the tagname value using the Value Input

The first six events listed above are configured automatically when event logging is enabled. The remaining three must be defined for each tagname in the Tagname Dictionary.

Defining Events

1. In the Tagname Dictionary, check the **Log Event** box for all of the tags to be event logged.

2. Configure the alarm group the tag is a member of and the priority of the event.

3. Click on the **Special/Configure/Alarms** command and select the **General** tab. Verify that the **Event Enabled** box is checked.

4. Verify that the group and priority settings are correctly set to allow the alarm window, alarm file or alarm printer to show the event.

Section : 3

Logging to SQL Database

The Alarm DB Logger logs alarm data into the database. If the OLEDB Provider is SQL Server, you will need to specify the SQL Server machine in the Alarm DB Logger Manager. Alarm DB Logger automatically creates the necessary data structures, if they do not already exist in the database.

Using the Alarm DB Logger Utility
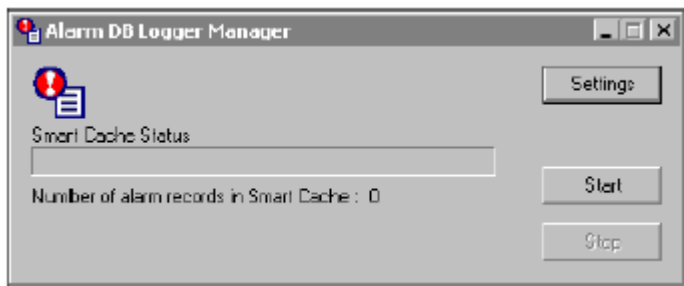
The Alarm DB Logger utility is a separate executable that takes care of the logging work. It is launched and starts working either as a service or a normal application (depending upon the running mode you select when you configure the Alarm DB Logger). The logging utility retrieves the setting information from the registry and performs the logging. The Alarm DB Logger is an Alarm Consumer.

**To use the Alarm DB Logger utility**

1. Start up the Alarm DB Logger Manager. The **Alarm DB Logger Manager** dialog box appears. **Tip** When minimized, the Alarm DB Logger Manager appears as an icon in the system tray. When you right-click the icon, a popup menu appears displaying the following commands: **Start** - Begins the alarm logging process. **Stop** - Ends the alarm logging process. **Settings** - Opens the **Alarm DB Logger Manager – Configuration** dialog box. **Hide Window** -Minimizes the Alarm DB Logger Manager dialog box to an icon in the system tray. **Show Window** - Opens and maximizes the Alarm DB Logger Manager dialog box. **Close** - Exits the Alarm DB Logger Manager Utility. The **Smart Cache Status** indicates the percentage fill of the in-memory buffer with alarm records.

2. Click **Settings** to configure the Alarm DB Logger. The **Alarm DB Logger Manager - Configuration** dialog box appears. For more information on configuring the Alarm DB Logger, see "Alarm DB Logger Configuration."

3. Click **Start** to begin the alarm logging process.

4. Click **Stop** to end the alarm logging process.

Alarm DB Logger Configuration

Before you begin using the Alarm DB Logger, you must configure a few items, such as your database connection, your query list, the logging mode and so on. This information is configured via the Alarm DB Logger Manager. The first Alarm DB Logger Manager Utility dialog box deals with your database connection. You can either select **SQL Server** or **Microsoft Data Engine (MSDE)** for storage of data. The Alarm DB Logger Configuration Utility also allows you to create necessary data structures (tables, views, and stored procedures) and to test your database connection.

To configure Alarm DB Logger

1. Start up the Alarm DB Logger Manager. The **Alarm DB Logger Manager** dialog box appears.
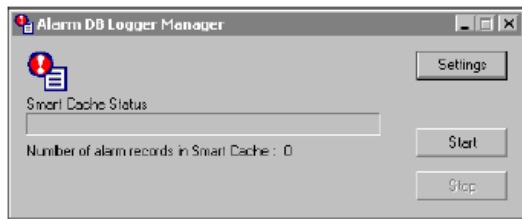
2. Click **Settings**. The **Alarm DB Logger Manager - Configuration** dialog box appears.
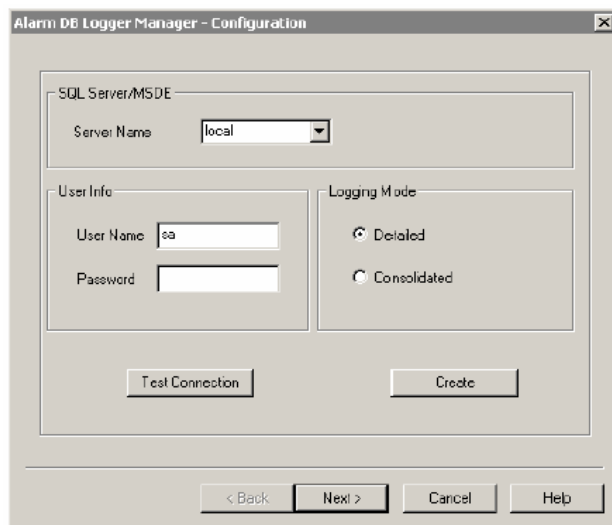
3. In the **SQL Server/MSDE** group, click the **Server Name** arrow to open the listing of available SQL/MSDE Servers and then select the name of the database server that you want to use.

4. In the **User Info** group, type in your **User Name** and **Password** in the respective entry boxes. (All **Password** characters are displayed as ** for security reasons).
5. In the **Logging Mode** group, click the **Detailed** or **Consolidated** radio button.

6. Click **Test Connection** to test your connection to the target database.

7. Click **Create** to create new database/tables on the selected Server.

8. Click **Next** to access the **Alarm DB Logger Manager - Query Selection** dialog box.

Alarm DB Logger Query Configuration

The second Alarm DB Logger Manager dialog box refers to the query details that need to be selected for alarm logging. It provides you with a multi-line text entry box to key in the query. It also allows you to select the **from** and **to** alarm priority values.

To configure query details

1. If it is not already running, start up the Alarm DB Logger Manager Utility. The **Alarm DB Logger Manager Configuration** dialog box appears.

2. Click **Next**. The **Alarm DB Logger Manager - Query Selection** dialog box appears.
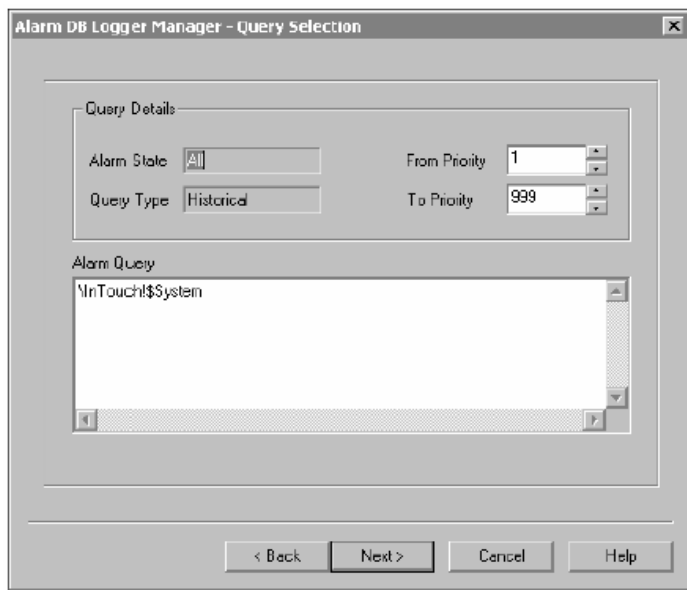
3. Click the **From Priority** arrows to select the starting of the alarm priority range.

4. Click the **To Priority** arrows to select the ending of the alarm priority range.

5. In the multi-line **Alarm Query** entry box, type the sets of InTouch alarm queries that you want to perform.

6. Click **Next**. The **Alarm DB Logger Manager - Advanced Setting** dialog box appears.

Alarm DB Logger Advanced Setting Configuration

The third Alarm DB Logger Manager dialog box refers to the advanced query settings, such as the running mode (either as a Windows Service or a normal application). It also provides you with the option for logging of events. You can tune the performance of the alarm logging by setting the frequency at
which the alarms are to be flushed from the memory buffer to the database.

To configure advanced setting details

1. If it is not already running, start up the Alarm DB Logger Manager. The **Alarm DB Logger Manager Configuration** dialog box appears.

2. Click **Next**. The **Alarm DB Logger Manager - Query Selection** dialog box appears.

3. Click **Next**. The **Alarm DB Logger Manager - Advanced Setting** dialog box appears.
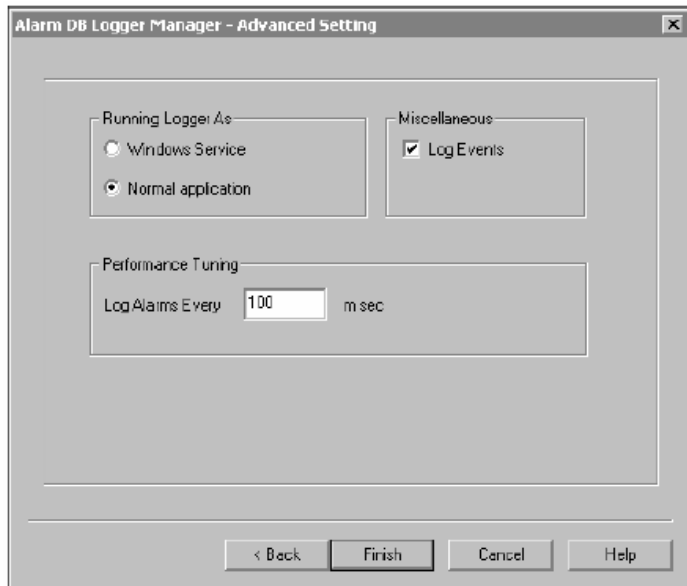
4. In the **Running Logger As** group, select the option that you want to use for the logger:

- **Windows Service** - Configures the logger to function as a Windows service
- **Normal Application** - Configures the logger to function as a normal application

5. Select the **Log Events** option to log all Events to the events table.

6. In the **Log Alarms Every # msec** entry box, type the interval at which the alarms are to be logged in milliseconds

7. Click **Finish** to close the Alarm DB Logger Manager Utility and save all logger configuration settings into the registry.

Section : 4

Alarm DB Purge/Archive Utility

The Alarm DB Purge/Archive utility is a separate application installed along with InTouch. The Alarm DB Purge/Archive utility takes care of configuring your alarm database for Purge/Archive operations and showing the status of any purging activity. The utility also provides you with the ability to specify
timer-based purging and archiving. You will use the Alarm DB Purge/Archive Utility to:

- Select the type of records that need to be purged and/or archived (consolidated alarm records or alarm detail records).

- Specify automatic timer-based purging/archiving based on your configuration.

- Purge the whole database, if necessary.

- Set the time of day that the purging/archiving activity will be performed.

- Specify the time interval at which the purging/archiving will be performed. It can be daily, weekly or monthly.

- Display the status of the purging/archiving.

- Store the status in the log file. (You can use this log file in the future to check the state of your purging/archiving activity.)

Purge/Archive General Properties Configuration

The Alarm DB Purge/Archive Utility **General** property sheet refers to the Purge/Archive properties. You can select the type of table that needs to be purged - either **AlarmDetail** or **AlarmConsolidated** tables.

You are given the option to archive the data that you want to purge. You can choose the path to the folder in which you want to archive the data. Archive files are created for each table that is purged/archived. The archived file names are automatically created by a unique combination of the table name, date and time.

For example, the name of the archive file for **AlarmMaster** that was archived on November 10, 2000 at 5:30 p.m. would be **AlarmMaster_11102000_1730.txt**.
For more information, see "Distributed Alarm Database Views."

To purge/archive your alarm database

1. Start up the Alarm DB Purge/Archive utility. The **Purge/Archive** dialog box appear
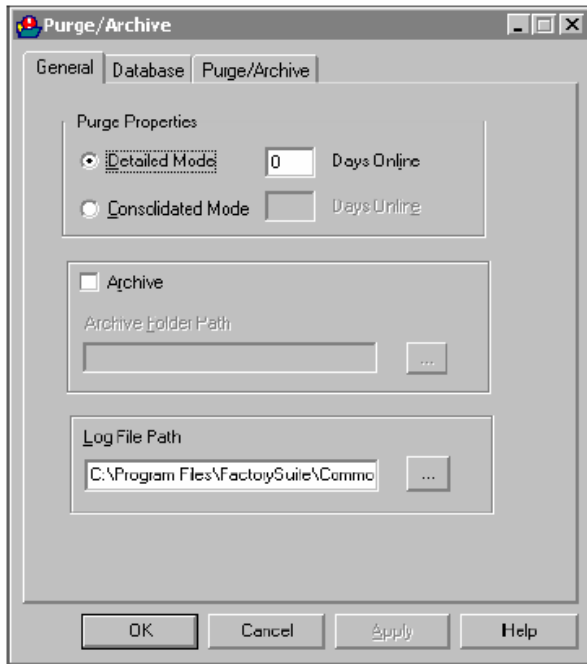
When minimized, the Alarm DB Purge/Archive Utility appears as an icon in the system tray. When you right-click the icon, a popup menu appears displaying the following commands: -

Command	Desciption
Purge All Now	Purges the entire database.
Purge Now	Starts purging now with the present settings.
Cancel Purge	Stops the current purging process. (This command is only active when purging is going on.) When you stop purging, the database is rolled back to its original state.
Activate	Activates timer-based purging.
DeActivate	Deactivates the timer for purging.
Test Now	Perform a test purge to verify your connection to the database and target locations
Clear Status	Clears the status window.

2. In the **Purge Properties** group, select the option that you want to use for purging:

- **Detailed Mode** to purge alarms that have been logged in the Detailed mode.
- **Consolidated Mode** to purge alarms that have been logged in Consolidated mode.

3. The respective **Days Online** entry box becomes active after you select a **Purge Properties** mode. Type in the number of days you would like to retain the data. All data from the day previous to the number specified will be purged. (Valid entries are 0-9999.) If the number is 0, all records are purged except the records logged on the same day.

4. Select the **Archive** option and then specify the full path (up to 255 alphanumeric characters) to the database file that you want to archive either by typing the full path in the **Archive Folder Path** entry box, or by clicking the respective Browse (…) button to locate and select the file. (Once selected, the path will automatically appear in the entry box.)

5. In the **Log File Path** entry box, specify the full path (up to 255 alphanumeric characters) to the folder where you want to store your status file, or click the respective Browse (…) button to locate and select the
folder. (Once selected, the path will automatically appear in the entry box.)

6. Click **Apply** to save the settings to the registry, or click **OK** to save the settings and close the Alarm DB Purge/Archive utility.

Purge/Archive Database Configuration

The Alarm DB Purge/Archive **Database** property sheet allows you to choose which server the purge/archive will access. **To configure the Alarm DB Purge/Archive Utility database**
1. Start up the Alarm DB Purge/Archive utility. The **Purge/Archive** dialog box appears.

2. Click the **Database** tab to activate the **Database** property.
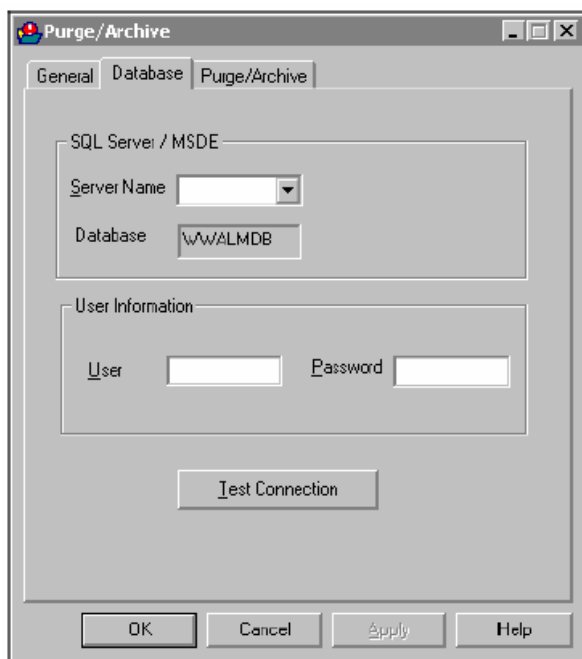
3. In the **SQL Server / MSDE** group, click the **Server Name** arrow to open the listing of available SQL/MSDE Servers, then select the name of the database server that you want to use.

4. The **Database** box displays the database name. (This is a read-only field that defaults to **WWALMDB**.)

5. In the **User Information** group, type in your **User** name and **Password** in the respective entry boxes. (All **Password** characters are displayed as ** for security reasons.)

6. Click **Test Connection** to test your connection to the database.

7. Click **Apply** to save the settings to the registry, or click **OK** to save the settings and close the Alarm DB Purge/Archive utility.

Purge/Archive Configuration

The Alarm DB Purge/Archive Utility **Purge/Archive** property sheet allows you to set the time at which automatic purging and archiving should trigger. It also allows you to perform a test purge to verify your connection to the database and target locations and to start and stop purging.

To configure the Alarm DB Purge/Archive Utility time interval

1. Start up the Alarm DB Purge/Archive utility. The **Purge/Archive** dialog box appears.

2. Click the **Purge/Archive** tab to activate the **Purge/Archive** property sheet.
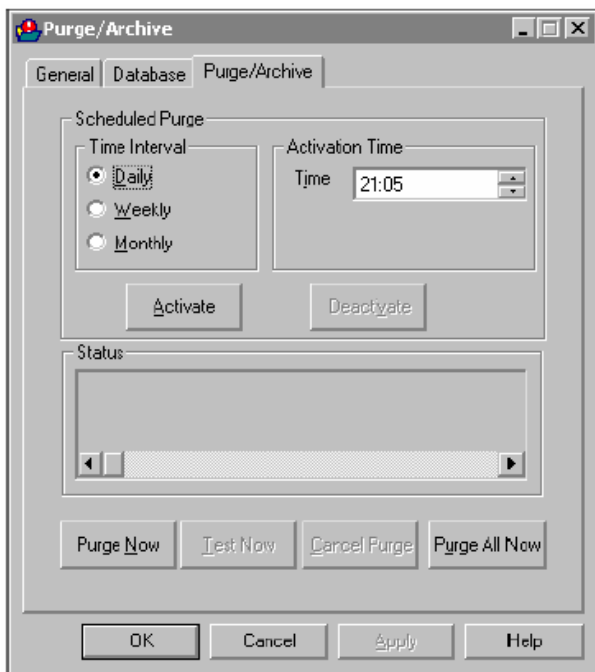
3. Select the **Time Interval** option that you want to use:

• **Daily** - Starts purging daily at the time selected in **Time** field.

• **Weekly** - Starts purging weekly on the day selected in the **Day** field and at the time selected in Time field.

• **Monthly** - Starts purging every month on the day selected in the **Day** field and at the time selected in the **Time** field.

4. After you have selected the **Time Interval** option, click the **Time** up/down arrows to select the time of day that you want the purge/archive activity to begin.

5. After you have selected the **Time Interval** option (for **Weekly** and **Monthly** only), click the **Day** arrow to select the day on which you want to perform the purge-archive activity. (Day of the week when **Time Interval** is **Weekly**, or day of the month when **Time Interval** is **Monthly**.)

6. The status of the purging activity is displayed in the **Status** area.

7. Click **Purge Now** to start purging now with the present settings.

8. Click **Test Now** to perform a test purge to verify your connection to the database and target locations. (This button is only active if the **Archive** option in the General tab has been selected. It creates blank archive files in the specified archive path.)

9. Click **Cancel Purge** to stop purging. (This button is active only when purging is going on.) When you stop purging, the database is rolled back to its original state.

10. Click **Purge All Now** to purge the entire database.

11. Click **Apply** to save the settings to the registry, or click **OK** to save the settings and close the Alarm DB Purge/Archive utility.

Chapter : 7   RealTime and Historical Trending

## Section : 1

RealTime Trends

InTouch provides two types of trend display objects: Real-time and Historical. Both trend objects can be configured to display graphical representations of multiple tagnames over time.

Real-time trends allow you to chart up to four pens (data values). Historical trends allow you to chart up to eight. Both types of trends are created using special tools in WindowMaker. InTouch also provides you with complete trend configuration control. For example, you can specify the time span, value range, grid resolution, location of time stamps, location of value stamps, number of pens, and color attributes.

InTouch also supports a distributed history system that allows you to retrieve historical data from any InTouch historical log file, even those across a network.

Creating a Real-time Trend

1. Select the **Real-time Trend** tool in the **Draw Object Toolbar**.
2. Click in the window and drag the mouse diagonally to draw a rectangle the size you want your trend to be. The trend can be any size and there is no limit to the number of trends that can be placed on a screen.
3. Release the mouse. The real-time trend object appears in the window:
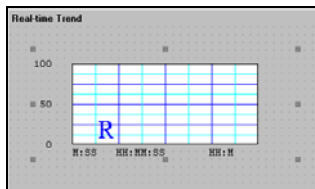
    In runtime, data is written in the trend from right left.
4. Double-click the trend to open its configuration dialog box. A trend object is like any other object drawn in WindowMaker. It can be moved by grabbing it with the mouse or resized by grabbing one of the object "handles."

Configuring a Real-time Trend

The first time a real-time trend object is pasted into a window, the system default configuration settings are used. Once a real-time trend has been configured, the next trend created will, by default, have the same settings.

1. Double-click the trend. The **Real Time Trend Configuration** dialog box appears:
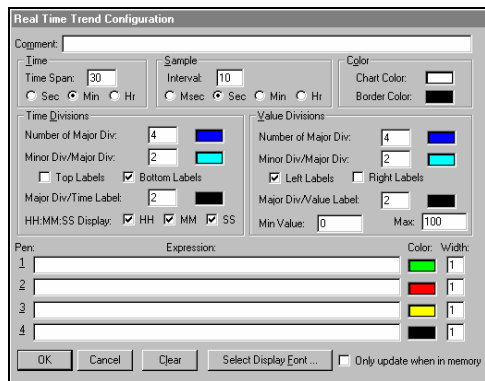
2. In the **Time Span** box, enter the length of time to be displayed horizontally (x-axis) on the trend. Then select a time increment option for the length of time.

   For example, if you enter 30 for the **Time Span** and then select **Min**, the horizontal time span of the chart will be 30 minutes long.

3. In the **Sample Interval** box, enter the frequency at which the trend expression will be evaluated and the chart updated. Then select the option for the time increment to which the number will relate.

   For example, if you enter 10 for the **Interval** and select **Sec** for the time increment, the expression will be evaluated every 10 seconds.

4. In the **Color** group, click the **Chart Color** and select a background color.

5. In the **Color** group, click the **Border Color** box and select a trend border color. Repeat this process for all color selections.

6. In the **Number of Major Div** box of the **Time Divisions** group, enter the number of major time divisions for the trend. Then select a color for the division lines. The number of major time divisions must be an even multiple of the number of **Minor Div/Major Div**.

7. In the **Minor Div/Major Div** box of the **Time Divisions** group, enter the number of minor time divisions to be visible within each major time division. Then select the color for the division lines.

8. In the **Time Divisions** group, select **Top Labels** to display time labels at the top of the trend.

9. In the **Time Divisions** group, select **Bottom Labels** to display time labels at the bottom of the trend. Trends can have both top and bottom labels or no labels at all.

10. If using time labels, enter the number of time labels per major time division line in the **Major Div/Time Label** box of the **Time Divisions** group.

11. In the **Time Divisions** group, select a color for the major time division lines.

12. The settings in the **Value Divisions** group are configured the same way as the settings in the **Time Divisions** group, except the minor and major value divisions set the vertical value (y-axis) range for the trend. This range uses Engineering Units and is the same for all trended tagnames. To display decimal points for the minor and major value divisions at runtime, they must be formatted here. For example, 0.00 to 100.00.

13. In the **Expression** box, enter the local tagname or expression each **Pen** will trend.

    Up to four pens can be visible in a trend. The pens can be used to display any local tagname or an expression that contains one or more local tagnames. (Message type tags cannot be logged or trended.) The ability to trend expressions is useful in creating custom displays to show tagnames with widely different ranges.

14. Click the color box to select the color for each pen to plot each tagname in the trend.

15. In the **Width** box, enter the number of pixels wide each pen is to be. Selecting a pen width greater than 1 significantly reduces performance in trend updating and printing of the trend.

16. Click **Select Display Font** to access the **Font** dialog box. Select the font, style and size to use when printing the trend.

17. Select **Only update when in memory** to update your trend only when it is displayed in the active window.

    If this option is not selected, the trend will be updated constantly, even if it is not in an open window. This may result in slightly slower system performance.

18. Click **OK**.

Increasing Real-time Trending Performance

1. Set the pen width to '1'.

2. Be sure no other objects are placed on top of the Real-time trend.

3. Lower the number of "samples" being taken.

    For example, if the **Time Span** is set to 30 minutes and the **Sample Interval** is set to 2 seconds, the number of samples taken during the 30 minutes will be calculated as:

    30* 60/2 = 900

    If the **Time Span** is set to 30 minutes and the **Sample Interval** is set to 5 seconds, the number of samples taken during the 30 minutes will be calculated as:

    30* 60/5 = 360

Lab – Real-time Trends

Summary Tasks

1. Configure 3 Additional Pens for the **Real Time Trend** window.

Section : 2

Historical Trends

Historical trends provide "snapshots" of data from a time and date in the past. They are not dynamic. Unlike real-time trends, historical trends are only updated when they are instructed to do so, either through the execution of a QuickScript or an action by the operator (e.g., clicking a button).

Up to eight tagnames (pens) can be trended at one time with no limit to the number of trends displayed. You have complete flexibility in designing the interface to your trend. "Scooters" can be created that the operator "slides" over the trend to access a variety of data based on the scooter's current location. For example, when the operator positions the scooter over an area on the trend that has visible data, the time and values at that location for all database values being trended is returned.

You can also create buttons to zoom in and out between the scooters or to data, such as the maximum to minimum value. Average and standard deviation can be displayed for the complete chart or for the area between the scooters. Historical trends can also be scrolled by any amount of time. Custom scales can be created and linked to the **.MinEU** and **.MaxEU** tagname **.fields** to display the minimum and maximum Engineering Units.

The distributed history system extends the retrieval capabilities of historical trends to include remote log databases. This system allows information from multiple historical log databases to be displayed in a single trend.

In addition to its trending capabilities, InTouch includes the HistData Utility, designed to work with InTouch historical log files. The HistData utility converts encrypted historical log files (.LGH) to comma separated variable files (.CSV) for use in spreadsheet or text editing environments such as Microsoft Excel.

Logging Tagnames to the Historical Log File

In WindowViewer, the values of logged tagnames are written to the historical log file each time they change more than the specified **Log Deadband** and, by default, once an hour regardless of change. For a tagname's value to be written to the historical log file, it must be configured to be logged in the Tagname Dictionary.

For integer and real (floating point) tagname types, you can set the **Log Deadband** in their respective details dialog boxes. The **Log Deadband** controls how many Engineering Units a tagname's value must change before it is logged to disk.

Configuring a Tagname for Logging

1. Click on the **Special/Tagname Dictionary** command. The **Tagname Dictionary** dialog box appears:
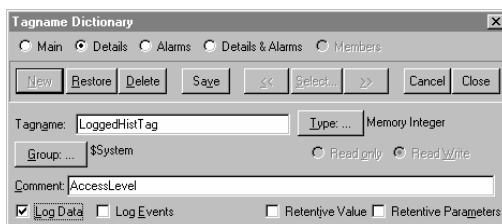
2. Open the desired tagname's definition and select **Log Data**.

    To actually log tagnames, enable logging as described in the following section.

If a tagname is changed from logged to not logged, the data already logged for the tagname will not be accessible.

Any changes made in WindowMaker to logging while WindowViewer is running are ignored until WindowViewer is restarted.

Configuring Historical Logging

In order for tagnames configured with the **Log Data** option to be written to the historical log file, the global logging function must be enabled.

1. Click on **Special/Configure/Historical Logging**. The **Historical Logging Properties** dialog box appears:
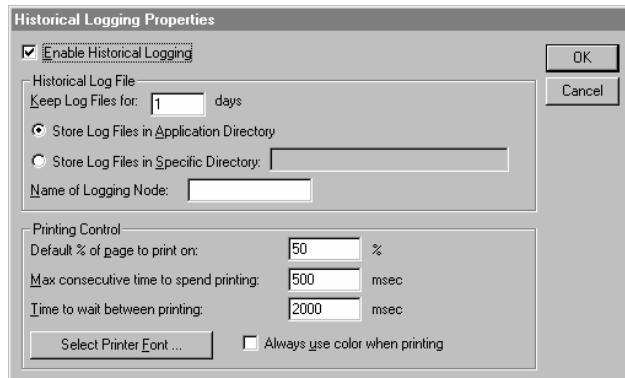
2. Select **Enable Historical Logging** to turn on global tagname logging.

3. In the **Keep Log Files for** box, enter the number of days (prior to today) of log files to keep saved to disk.

   For example, if you enter 10 and today is the 12th day of the month, the log files for the 2nd through the 12th (10 days plus today) will be saved to disk. The file for the 1st will automatically be deleted. If you enter a zero, the log files will be kept indefinitely.

4. Select **Store Log Files in Application Directory** to save the historical log file in your application directory. Select **Store Log Files in specific Directory** to save files to another directory. Enter the complete path to the directory you want to use.

   This entry must be either a DOS path such as C:\HISTLOG.
   By default, historical log files are named as follows:

   YYMMDD00.LGH and YYMMDD00.IDX

   where: **YY** equals the year the file was created
            MM equals the month the file was created (01-12)
            DD equals the day the file was created (01-31)
            00 always displays zeros

   For example, if the files were created on October 31, 1997, they would be named as follows:

   97103100.LGH and **97103100.IDX**

5. In the **Name of Logging Node** box, enter the NetDDE node name (not the computer name) for the node that will be logging to the history log file.

6. In the **Default % of page to print on** box, enter the percentage ratio for the page size to trend.

   Example: If you use 50 for the percentage, when you print a historical trend, it will fill half of the page (vertically and horizontally). A printout this size would take roughly one quarter of the time to prepare as a full page printout.

8. In the **Max consecutive time to spend printing** box, enter the number of milliseconds (processor time slice) the historical trend print module will spend consecutively printing.

9. In the **Time to wait between printing** box, enter the number of milliseconds the historical trend print module will wait between printouts.

10. Select **Always use colors when printing** if you are using a color printer or plotter.

11. Click **Select Printer Font** to access the Windows **Font** dialog box.

12. Click **OK** to save your settings and close the dialog box.
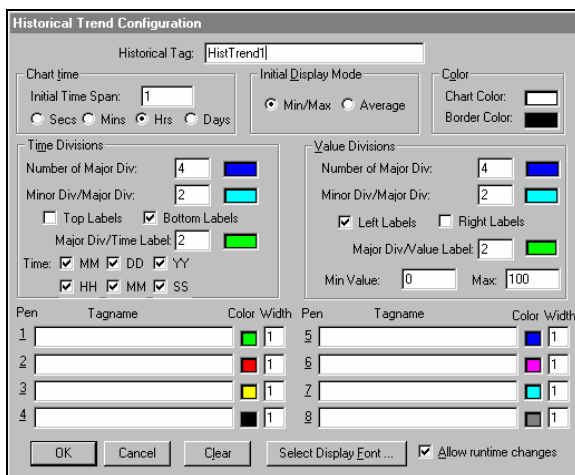
Creating a Historical Trend

1. Select the **Historical Trend** tool 🖼 in the **Draw Object Toolbar**.

2. Click in the window and drag the mouse diagonally to draw a rectangle the size you want your trend to be.

3. Release the mouse. The historical trend appears in the window:
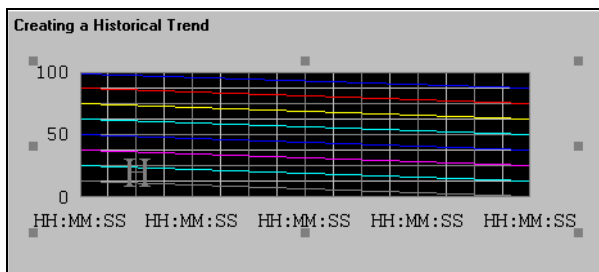
Configuring a Historical Trend

The first time a historical trend object is pasted into a window, the system default configuration settings are used. Once a historical trend has been configured, the next trend created will, by default, have the same settings.

1. Double-click the trend. The **Historical Trend Configuration** dialog box appears:

2. In the **Historical Tag** box, enter the tagname to use for the trend.

   If the tagname entered is not currently defined in the Tagname Dictionary, you will be asked to define it now. If you select **Yes** to define the tagname now, InTouch will automatically display the **Tagname Dictionary** dialog box and default the tagname type to **Hist Trend**. (The tagname must be defined as a **Hist Trend** type.) You must use a different tagname for each historical trend.

3. In the **Initial Time Span** box, enter the length of time to display horizontally (x-axis) on the trend and then select the time increment option for the length of time.

4. Select the **Initial Display Mode** to use for the trend as follows:

   Min/Max - Each pixel on the chart will display the minimum to maximum range that the point covered in the time represented by that pixel.

   Average - Displays the average value for each pixel (e.g., time segment).

5. In the **Color** group, click the **Chart Color** box and select a background color.

6. In the **Color** group, click the **Border Color** box and select a trend border color.

7. In the **Number of Major Div** box of the **Time Divisions** group, enter the number of major time divisions in the trend. Then select a color for the division lines.

8. In the **Minor Div/Major Div** box of the **Time Divisions** group, enter the number of minor time divisions to be visible within each major time division. Then select a color for the division lines.

9. In the **Time Divisions** group, select **Top Labels** to display time labels at the top of the trend.

10. In the **Time Divisions** group, select **Bottom Labels** to display time labels at the bottom of the trend.

11. If using time labels, enter the number of time labels per major time division line in the **Major Div/Time Label** box of the **Time Divisions** group.

12. In the **Time Divisions** group, select a color for the major time division lines.

13. The settings in the **Value Divisions** group are configured the same way as the settings in the **Time Divisions** group. The minor and major value divisions set the vertical value (y-axis) range for the trend. This range uses Engineering Units and is the same for all trended tagnames.

14. In the **Expression** box, enter the local tagname or expression each **Pen** will trend.

    Up to eight pens can be visible in a trend. (Message type tags cannot be logged or trended.)

15. Click the color box to select the color for each pen to plot each tagname in the trend.

16. In the **Width** box, enter the number of pixels wide each pen is to be.

    Selecting a pen width greater than 1 significantly reduces performance in screen updating and printing.

17. Click **Select Display Font** to access the **Font** dialog box. Select the font, style and size for the trend display.

18. Select **Allow runtime changes** to allow the operator to make changes to the trend's configuration in runtime. These changes include changing pen assignments, start date, time, etc.

    If this option is selected, when the operator clicks on the trend in runtime, the **Runtime Setup** dialog box will appear and changes to the trend can be made.
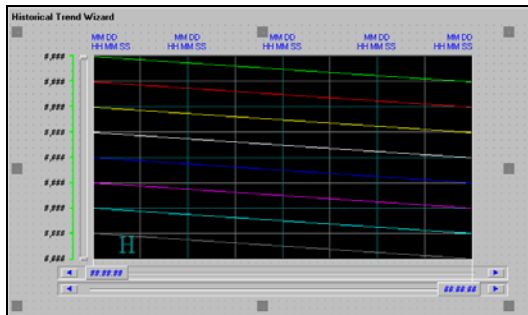
19. Click **OK**.

Using Historical Trend Wizards

The InTouch trend wizard provides a quick and easy way to create a historical trend. The trend wizard allows you to configure a full-featured historical trend with scooters, zooming, etc., with just a few mouse clicks.

Intouch Basic Training Manual

1. Click the wizard tool [icon] in the **Wizard Toolbar**. The **Wizard Selection** dialog box appears:
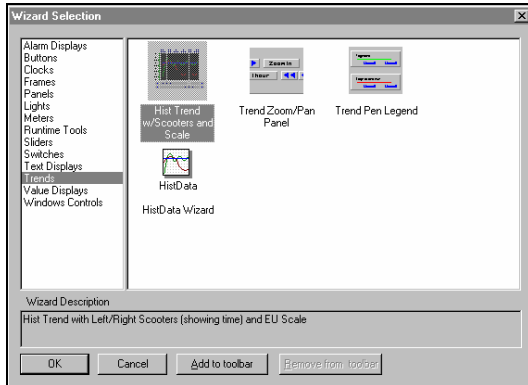
2. Select **Trends** in the list of wizards to display the available trend wizards.

3. Select the **Hist Trend with Scooters** wizard and click **OK**. The dialog box closes and your window reappears with the cursor in the "paste" mode.

4. Click in the window to paste the trend wizard:

5. Double-click on the trend wizard to open the **Historical Trend Chart Wizard** configuration dialog box:
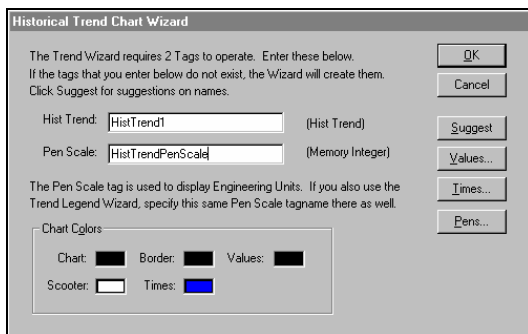
6. Enter the required information to configure the trend and click **OK**.

   Click **Suggest** for the wizard to automatically enter the configuration settings. The settings configured for a historical trend wizard are the same as those configured when the historical trend object is drawn using WindowMaker's trend tool in the **Draw Object Toolbar**.

7. To add zoom and movement functions or pen controls to your trend, use the trend **Zoom/Pan Panel** and **Trend Pen Legend** wizards, respectively. For these components to work together, the **Hist Trend** tagname must be used.

Breaking a Wizard

1. Select the historical trend wizard.

2. Click the break cell tool in the **Arrange Toolbar** or click on the **Arrange/Break Cell** command.

Configuring a Historical Trend in Runtime

If the **Allow runtime changes** option was selected when you configured your historical trend, the trend will be "touch-sensitive" in WindowViewer and the operator will be able to change the pen assignments, change the start date and time, etc.

1. Click the trend in WindowViewer, the **Historical Trend Setup** dialog box appears:
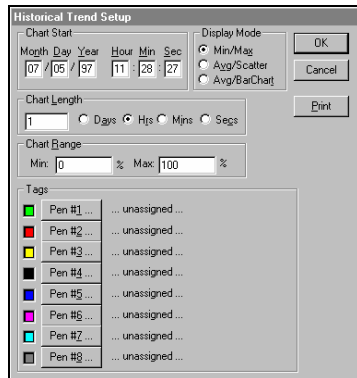
2. In the **Chart Start** group, enter the starting date and time for the chart.

3. Select the **Display Mode** for your chart. There are three modes as described in the following examples:

    Min/Max Historical Trend
    This mode displays the trends or changes in the percentage of Engineering Units scale as a vertical line over the time span, with emphasis on time flow and rate-of-change and not on the amount of change.

    Average/Scatter Historical Trend
    This mode displays the average value of the point during the time intervals.

    Average/Bar Chart Historical Trend
    This mode displays the average value of the point during the time intervals in bar form.

4. In the **Chart Length** box, enter the horizontal (x-axis) length of time to be displayed on the trend and then select the time increment for the length.

    If you enter a 1 and select **Hrs**, the trend will be 1 hour long.

5. In the **Chart Range** boxes, enter the percentage of Engineering Units scale that the trend is to zoom in/out (vertical (y-axis) range to be displayed on the trend).

    The units for the range are a "percentage" of Engineering Units scale. These values should be from 0 to 100.

6. Click each **Pen#** to select the tagname to trend. The Tag Browser appears in the filtered selection mode:
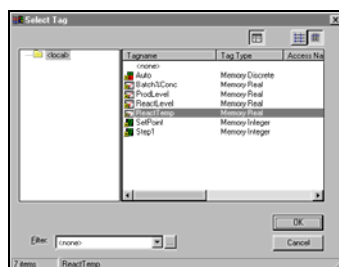
Only the tagnames that are defined with the **Log Data** option selected will be displayed for the selected tag source.

7. Double-click the tagname the selected pen is to plot on the trend or select the tagname and click **OK**. The **Historical Trend Setup** dialog box reappears displaying the selected tagname next to the **Pen#** button you originally clicked.

   You can click the **Filter** arrow to open the list of defined filters that can be used to populate the Tag Browser. The first entry of this list is **<none>**, which means that no filter is being used. Only the tagnames that are defined with the **Log Data** option selected will be displayed for the selected tag source.

8. Click **Print** to print the historical trend.

Updating a Historical Trend in Runtime

When a historical trend is first shown in WindowViewer, it will display data for the specified configurations. Unlike real-time trends, historical trends do not update themselves continuously. A **change must be made to the trend** for it to update. Any of the following methods can be used to update the trend:

1. Select **Allow runtime changes** in the **Historical Trend Configuration** dialog box (in WindowMaker) to allow the operator to manually change the trend's time and/or date and force the update.

2. Use the following in a QuickScript on a button to allow the operator to update the chart:

   Hist_TrendTag.UpdateTrend = 1
   HTUpdateToCurrentTime(*Hist_Tag*);
   HTScrollLeft(*Hist_Tag,Percent*);
   HTScrollRight(*Hist_Tag,Percent*);
   HTZoomIn(*Hist_Tag,LockString*);
   HTZoomOut(*Hist_Tag,LockString*);
   HTSetPenName(*Hist_Tag,PenNum,Tagname*);

3. Change any of the following trend tagname **.fields**:

   .ChartStart
   .ChartLength
   .MaxRange
   .MinRange
   .Pen1-.Pen8

Historical Trend .Fields

For a given historical trend tagname there are many **.field**s that only apply to historical trend tagnames. Each historical trend **.field** is described briefly below

.Field	Description
.ChartLength	Read/write integer tagname .field used to control the length of time displayed in a Historical trend graph. **.ChartLength** displays the length of the chart in seconds.
.ChartStart	Read/write integer tagname .field used to control the starting time and/or to scroll the corresponding historical trend. **.ChartStart** displays the number of elapsed seconds since 12:00 a.m., 1/1/70.
.DisplayMode	Read/write analog tagname .field used to determine the method to be used in displaying values on the trend.
.MaxRange, .MinRange	Read/write real tagname .fields used to represent the percentage of the tagname's Engineering Unit

	range that should be displayed for each tagname being trended. The limits for **.MaxRange** and **.MinRange** are from 0 to 100 and **.MinRange** should always be less than **.MaxRange**. If a value less than 0 or greater than 100 is assigned to either of these fields, the value will be clamped at 0 or 100. If **.MinRange** is greater than or equal to **.MaxRange**, the trend will not display any data.
.Pen1 - .Pen8	Read/write TagID type tagname .fields used to control the tagname being historically trended by each pen. A TagID type tagname can <u>only</u> be equated to another TagID tagname. It <u>cannot</u> be mixed with any other tagname type unless the **.TagID** extension is added to the other tagname. **.TagID** cannot be used for remote history provider tagnames.
.ScooterLockLeft	Read/write discrete field. When the value of this field is TRUE, the RIGHT scooter cannot move to the left of the left scooter's position. (0=FALSE, 1=TRUE).
.ScooterLockRight	Read/write discrete field. When the value of this field is TRUE, the LEFT scooter cannot move to the right of the right scooter's position. (0=FALSE, 1=TRUE).
.ScooterPosLeft	Read/write real field which represents the position of the left scooter (range 0.0 to 1.0).
.ScooterPosRight	Read/write real field which represents the position of the right scooter (range 0.0 to 1.0).
.TagID	Read/write TagID tagname .field used in conjunction with the Historical Trend .Pen1 - .Pen8. TagID tagnames monitor and/or control the tagname being trended by a pen.
.UpdateCount	Read-only integer .field that is incremented when a retrieval is complete for the trend
.UpdateInProgress	Read-only discrete .field that shows historical data retrieval status (0=no retrieval in progress, 1=retrieval in progress).
.UpdateTrend	Read/write discrete tagname .field that can be set to 1 to cause a Historical trend to update using all current values.

Historical QuickScript Functions

There are several internal functions that can be used to specify the tagname to be trended by each pen, display the value at a scooter location, scroll the trend by a percentage, etc. These functions are described briefly below:

Function	Description
HTGetLastError	Determines if there was an error during the last retrieval of a specified pen.
HTGetPenName	Returns the pen name of the tagname currently used for the specified pen # of the specified trend.
HTGetTimeAtScooter	Returns the time in seconds since 00:00:00 hours GMT, January 1, 1970 for the sample at the scooter location specified by *ScootNum* and *ScootLoc*. *UpdateCount*, *ScootNum*, and *ScootLoc* cause the expression to be evaluated when any of these parameters change. This ensures that the expression is evaluated after new retrievals or after a scooter moves.

HTGetTimeStringAtScooter Returns the string containing the time/date for the sample at the scooter location specified by *ScootNum* and *ScootLoc*. *UpdateCount*, *ScootNum*, and *ScootLoc* cause the expression to be evaluated when any of these parameters change. This ensures that values are updated after new retrievals or after a scooter moves. The format of the string determines the contents of the return value.

HTGetValue Returns a value of the requested type for the entire trend's specified pen.

HTGetValueAtScooter Returns a value of the requested type for the sample at the specified scooter position, trend and pen #. The *UpdateCount* parameter will cause the expression to be evaluated after a retrieval is complete.

HTGetValueAtZone Returns a value of the requested time for the data contained between the right and left scooter positions for a trend's specified pen.

HTScrollLeft Sets the start time of the trend to a value older than the current start time by a percentage of the trend's width. The effect is to scroll the date/time of the chart to the left by a given percent.

HTScrollRight Sets the start time of the trend to a value newer than the current start time by a percentage of the trend's width. The effect is to scroll the date/time of chart to the right by a given percent.

HTSetPenName Assigns a different tagname to a trend's pen.

HTUpdateToCurrentTime Causes the data to be retrieved and displayed with an end time equal to the current time. The start time will be equal to EndTime minus the Width of the chart.

HTZoomIn Calculates a new chart width and start time. If the trend's .ScooterPosLeft is 0.0 and the .ScooterPosRight is 1.0, the new chart width equals the old chart width divided by two. The new start time will be calculated based on the value of *LockString*.

HTZoomOut Calculates a new chart width and start time. The new chart width is the old chart width multiplied by two. The new start time will be calculated based on the value of *LockString*.

HistData Utility

The HistData utility program provides access to the historical data files created by InTouch. It is used to move selected historical data into a requesting program such as Microsoft Excel. HistData provides you with the ability to immediately view historical data or create a file for later access. Access to the historical data may be accomplished via macro functions in a requesting program or from within InTouch.

The HistData program should be started (then reduced to an icon) prior to starting any program that will be using it.

HistData Database

The HistData program contains its own I/O. The items are used to specify start period, duration and sampling interval, etc., for the historical data to be accessed. The following lists the items defined in the HistData program:

Item	Type	Description
DATADIR	Message	Pathname of the directory containing the historical data files (e.g., C:\InTouch\App).
DBDIR	Message	Pathname of the directory containing the InTouch Tagname Dictionary (e.g.,C:\InTouch\App).
STARTDATE	Message	Data sample start date in the format MM/DD/YY.
STARTTIME the	Message	Data sample start time in the format HH:MM:SS using 24-hour clock.
DURATION	Message	The length of time for which data is to be returned. **DURATION** can be expressed in weeks, days, hours, minutes and seconds. The following are the valid characters: **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). Fractional values are also permitted (e.g., .5s for 500 milliseconds). To request a single sample, set **DURATION** to 0 (zero).
INTERVAL	Message	The length of time between samples. **INTERVAL** can be expressed in weeks, days, hours, minutes and seconds (e.g., 1w represents 1 week). Fractional values are also allowed (e.g., .25d represents 6 hours). (The valid characters are the same as those for **DURATION**.)
TAGS	Message	The list of tagnames to return historical data for. **TAGS** is entered in the form "TagA,TagB,TagZ". In addition, the date and/or time for a sample can be requested by using the internal system tagnames **$Date** and **$Time**. For example: "$Date,TagA,TagB" "$Time,TagA,TagB" "$Date,$Time,TagA,TagB"
TAGS1, TAGS2,...		Message The **TAGS** string can be 131characters in WindowViewer and 255 characters in Excel. The string can be appended for longer requests by adding tagname items named "Tags1""Tags2", etc. If a tagname needs additional tagname text appended to it, a plus (**+**) is entered at the end of the string. For example: TAGS="$Date,ProdLevel,ProdTemp,+" TAGS1="ReactLevel,Temp,GasLevel,+" TAGS2="MotorStatus"
PRINTTAGNAMES		Discrete This item defaults to 1 and causes HistData to print the tagnames on the first line of the output file above the associated column of values. If the tagnames are not to be printed, this item's value must be changed to 0 (zero).
DATA HistData	Message	This item is used to hold the requested data in the

		program in comma separated variable format. It is used by other applications to **ADVISE** or **REQUEST** data via DDE.
SENDDATA	Integer	When set to 1, HistData will update the **DATA** item with the requested data. When the update is complete, S **SENDDATA** is automatically reset to 0 (zero).
FILENAME (e.g.,	Message	Complete pathname of file towrite the requested data C:\INTOUCH\HDFILE.CSV).
WRITEFILE the	Integer	When set to 1, HistData will write the requested data to file specified by the **FILENAME** item name. When the file update is complete, **WRITEFILE** is automatically reset to 0 (zero).
STATUS	Discrete	Displays the status of the last operation. 1 indicates success and 0 (zero) indicates that an error occurred.
ERROR will be	Message	A string containing a description of the last error. It "None" when **STATUS** is 1 and will contain an error message string when **STATUS** is 0 (zero).

Lab – Historical Trends

Summary Tasks

1. Create a new window called **Historical Trend**.
2. Use the **Trends** wizards to create a historical trend that will plot the existing tags, **RealTag** and **IntTag**.

Chapter : 8   I/O Communication

Section : 1

Dynamic Data Exchange (DDE)

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The *server* application provides the data and accepts requests from any other application interested in its data. Requesting applications are called *clients*. Some applications such as InTouch and Microsoft Excel can simultaneously be both a *client* and a *server*.

FastDDE provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between *client* and *server*. Although Wonderware's FastDDE has extended the usefulness of DDE for our industry, this extension is being pushed to its performance constraints in distributed environments.

NetDDE extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM PCs connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

Wonderware SuiteLink

Wonderware SuiteLink uses a TCP/IP based protocol. SuiteLink is designed specifically to meet industrial needs, such as data integrity, high-throughput, and easier diagnostics. This protocol standard is only supported on Microsoft Windows NT 4.0 or higher.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. Each connection between a client and a server depends on your network situation. SuiteLink was designed specifically for high speed industrial applications and provides the following features:

- Consistent high data volumes can be maintained between applications, regardless of whether the applications are on a single node or distributed over a large node count.

- Value Time Quality(VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.

- Extensive diagnostics of the data throughput, the server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT operating system performance monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.

- The network transport protocol is TCP/IP using Microsoft's standard Winsock interface.

InTouch I/O Address

InTouch identifies an element of data in an I/O Server program by using a three-part naming convention that includes the *application name, topic name* and *item name*. To obtain data from another application, the *client* program (InTouch) opens a channel to the *server* program by specifying these three items.

In order for InTouch to acquire a data value from another application, it must also know the name of the *application* providing the data value, the name of the *topic* within the application that contains the data value, and the name of the specific *item* within the *topic*. In addition, InTouch needs to know the data's type: discrete, integer, real (floating point), or message (string). This information determines the I/O type for the tagname when it is defined in the InTouch database. Now, when WindowViewer is running, it will automatically perform all of the actions required to acquire and maintain the value of this *item*.

For example, in the case of Excel, the *application name* is "Excel," the *topic name* is the name of the specific spreadsheet that contains the data and the *item name* is the identification of the cell on the spreadsheet to/from which the data is to be read/written.

When another Windows application requests a data value from InTouch, it also must know the three I/O address items. The following describes the I/O address convention for InTouch:

1. **VIEW** *(application name)* identifies the InTouch runtime program that contains the data element.

2. **TAGNAME** *(topic name)* is the word <u>always</u> used when reading/writing to a tagname in the InTouch database.

3. **ActualTagname** *(item name)* is the actual tagname defined for the item in the InTouch Tagname Dictionary.

For example, to access a data value in InTouch from Excel, a DDE Remote Reference formula would be entered in the cell into which the data value is to be written:

**=VIEW|TAGNAME!'ActualTagname'**

InTouch Access Names

When you create I/O type tagnames or remote tagname references, they must be associated with an Access Name. Access Names contain the information that is used to communicate with other I/O data sources including the node name, application name and topic name.

Creating an Access Name

1. Click on the **Special/Access Names** command. The **Access Names** dialog box appears:
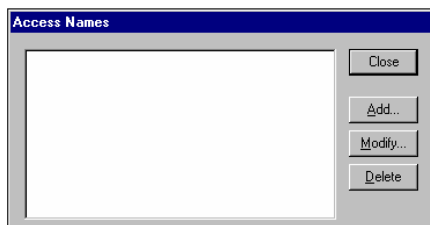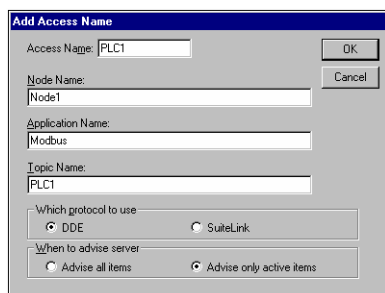

Access Names can be created in the Application Explorer, under **Configure/Access Names**. Or, right-click **Access Names**, and then click **Open**.
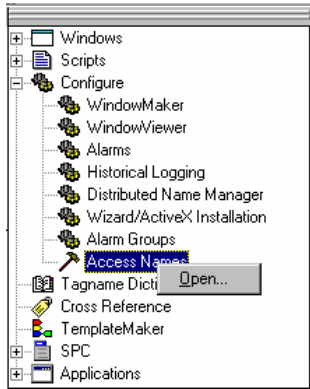
Access Names can also be created while you are defining an I/O type tagname in the Tagname dictionary.

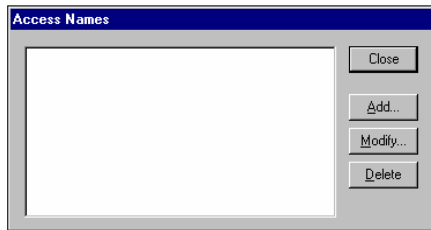2. Click **Add**. The **Add Access Name** dialog box appears:

3. In the **Access Name** box, enter the name you want InTouch to use for the Access Name. (For simplicity, use the same name that you will use for the *topic name*.)

   InTouch uses Access Names to reference real-time I/O data. Each Access Name equates to an I/O address, which can contain a Node, Application, and Topic. In a distributed application, I/O references can be set up as global addresses to a network I/O Server or local addresses to a local I/O Server.

4. If the data resides in a network I/O Server, enter the remote node's name in the **Node Name** box.

5. In the **Application Name** box, enter the actual program name for the I/O Server program from which the data value will be acquired. In this case the value is coming from the Wonderware Modbus I/O Server. Therefore, **MODBUS** is used. **Do** **not** enter the **.exe** extension portion of the program name.

6. In the **Topic Name** box, enter the *topic name* to access.

   The **Topic Name** is an application-specific sub-group of data elements. In the case of data coming from a Wonderware I/O Server program, the *topic name* is the **exact** same name configured for the *topic* in the I/O Server program. When communicating with Microsoft Excel, the *topic name* must be the name given to the spreadsheet when it was saved. For example, Book1.xls.

7. Select the protocol you are using.

8. Select **Advise all items** if the server program is to poll for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. Selecting this option will impact performance and is not recommended.

9. Select **Advise only active items** if the server program is to poll only points in visible windows and points that are alarmed, logged, trended or used in any script. A touch pushbutton action script will not be polled unless it appears in a visible window.

10. Click **OK** to accept the new Access Name and close the dialog box. The **Access Names** dialog box reappears displaying the new Access Name selected in the list:

11. Click **Close** to close the dialog box and return to your tagname definition.

Modifying or Deleting an Access Name

1. Click on the **Special/Access Names** command. The **Access Names** dialog box appears:
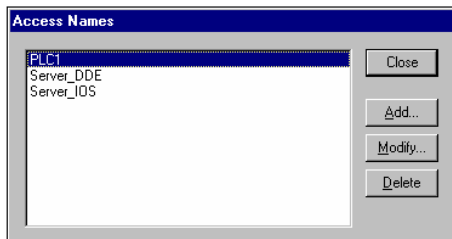
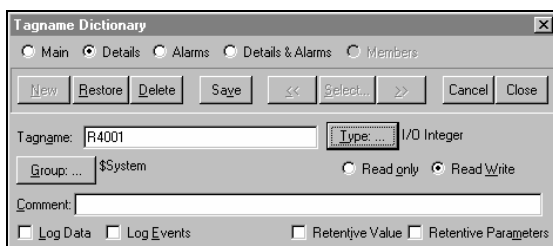Or, in the Application Explorer; **Configure/Access Names** or right-click **Access Names** then **Open**.

2. To change an Access Name's definition, select it in the list and then click **Modify**. The **Modify Access Name** dialog box appears. Make the required changes and then click **OK**. The **Access Names** dialog box reappears. Click **Close** or repeat this procedure if you need to modify other defined Access Names.

3. To delete an Access Name, select it in the list and then click **Delete**. A message box will appear asking you to confirm the deletion. Click **Yes**. Click **Close** or repeat this procedure to delete other defined Access Names.

Defining an I/O Item in InTouch

InTouch can receive data from other local or remote Windows applications when I/O type tagnames are defined in the Tagname Dictionary. Each I/O type tagname references a valid *item* in the I/O Server program.

Defining an I/O Type Tagname

1. Click on the **Special/Tagname Dictionary** command. The **Tagname Dictionary** dialog box appears.

2. Click **New**. The **Tagname** box clears.

In the Application Explorer; double-click **Tagname Dictionary**.

The first time you access the Tagname Dictionary, the definition for the internal system tagname **$AccessLevel** is displayed. When tagnames have been defined in the Tagname Dictionary, the last edited tagname's definition is displayed.

3. In the **Tagname** box, enter a name for the new tagname.

4. Click **Type**. The **Choose tag type** dialog box appears:
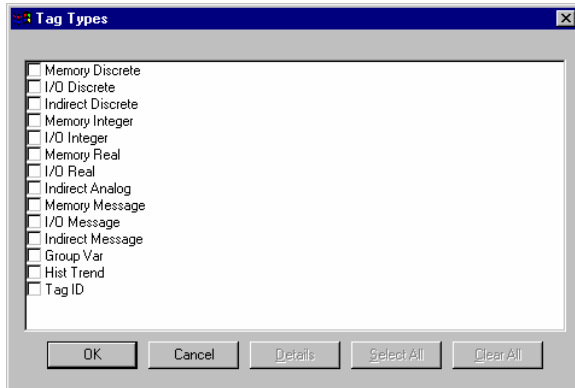
5. Select the I/O type for the tagname.

6. Once the I/O type is selected, click **OK**. The respective "details" dialog box will appear. For example, if I/O Integer is selected, the following dialog box appears:
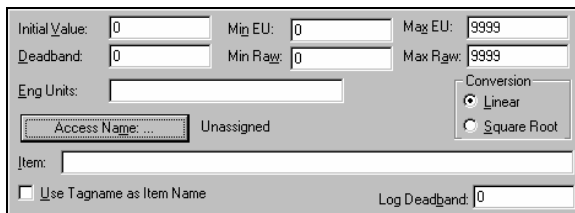
If the "Details" dialog box does not appear, click **Details** at the top of the screen.

7. Specify all the required details for defining the *item*.

8. Click **Access Name**. The **Access Names** dialog box appears:
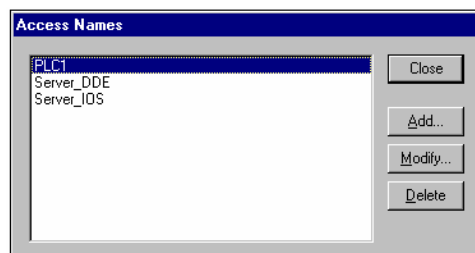
9. Double-click the Access Name to use or select it and click **Close**.

10. The selected Access Name (now associated with this tagname definition) appears adjacent to the **Access Name** button in the details dialog box. For example:
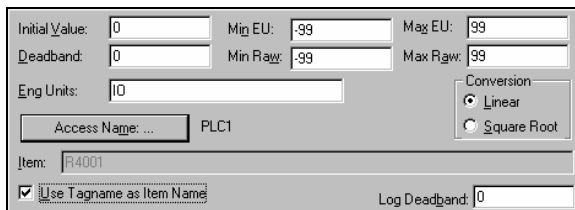
11. In the **Item** box, enter the *item name* for the data value in the I/O Server program.

12. Click **Close**.

Section : 2

Wonderware I/O

Wonderware servers are Microsoft Windows application programs that act as I/O Servers and allow other DDE aware Windows applications (InTouch, Excel, etc.) access to data in the real world (PLCs, RTUs, etc.). Wonderware servers are primarily intended for use with Wonderware's InTouch program, however, they can be used by any Microsoft Windows program capable of acting as a DDE client.

In this section, we will examine the start-up, configuration and use of a Wonderware I/O Server. Because Wonderware's servers are Windows applications, they will all have the same basic appearance and capabilities. You should keep in mind; however, that depending on which server you require, additional hardware (network, etc.) may be necessary and the configuration screens may require additional information. We will use the Modbus I/O Server as our reference for this section and will implement the Modbus Server as a point-to-point server using the RS-232 serial port to access one Modicon 984 PLC via its programming port.

Configuring I/O Servers

Once the I/O Server has been installed, a small amount of configuration is required. Configuring the server automatically creates a configuration file named, MODBUSDV.CFG. This file stores the configuration information about communication ports and all of the topic definitions (described in detail later).

The configuration file is automatically saved to the directory in which the I/O Server is installed unless a different directory is specified via the **Configure/I/O Server Settings...** command.

1. Start the Modbus server. The following window will appear:
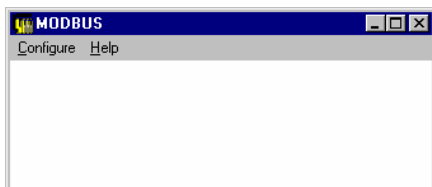
2. Click on the **Configure/Com Port Settings** command to configure the communication port that will be used to communicate with the PLC equipment. The **Communication Port Settings** dialog box appears:
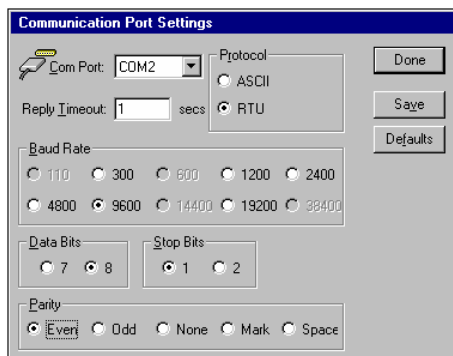
3. Click on the **Com Port** down arrow and select the communication port that is connected to the PLC equipment.

4. In the **Reply Timeout** box, enter the amount of time (in seconds) that all PLCs connected via this serial communications port will be given to reply to commands from the I/O Server.

5. In the **Protocol** section, select the protocol configured for the equipment attached to this communication port. **RTU** is recommended.

6. In the **Baud Rate** section, select the baud rate (serial bit rate) setting that matches the equipment connected to this communication port.

7. In the **Data Bits** section, select the option for the number of data bits that corresponds to the configuration of the equipment on this communication port. If **ASCII** is selected for the protocol, use **7**. If **RTU** is selected, use **8**.

8. In the **Stop Bits** section, select the appropriate number of stop bits for the communication port. If the baud rate is greater than 300, the stop bits should be set to **1**.

9. In the **Parity** section, select the setting that corresponds to the configuration of the equipment on this communication port.

10. Click **Save** to save the current settings entered for the selected communication port. The **Communication Port Settings** dialog box will remain displayed and another communication port can be configured.

11. When all configurations have been made, click on **Done**.

Creating Topic Definitions

The **Configure/Topic** command is used to create, modify or delete topic definitions. One or more topic definitions must exist for each PLC that the I/O Server will communicate with. Each topic definition must contain a unique name for the PLC associated with it. This unique name is then used as the *topic name* portion of the DDE Address for all DDE conversations to that PLC.

1. Click on the **Configure/Topic** command to access the initial **Topic Definition** dialog box:
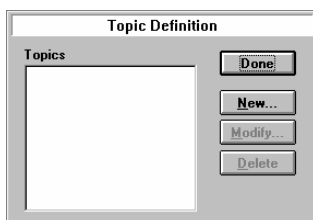
2. Click on **New** to add a new topic definition. The **MODBUS Topic Definition** dialog box appears:
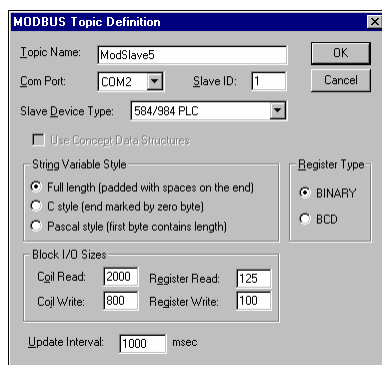
3. Enter a unique name (up to 32-characters long) for the PLC in the **Topic Name** box, e.g., ModSlave5.

4. To select a com port other than the default, click on the **Com Port** down arrow. Select the communications port to be associated with this topic.

5. Enter the Slave ID of the PLC in the **Slave ID** box.

6. Click on the **Slave Device Type** down arrow to select a slave device type other than the default.

7. Select the appropriate **String Variable Style** option that the PLC will use to store ASCII strings in its registers.

8. Select the appropriate **Register Type**.

9. In the **Coil Read** box, enter the maximum number of consecutive coils to be read at one time. In this example, the valid coil read values can be between 8 and 2000 and must be an even multiple of 8.

10. In the **Coil Write** box, enter the maximum number of consecutive coils that can be written to at one time. In this example, the valid coil write values can be between 8 and 800 and must be an even multiple of 8.

11. In the **Register Read** box, enter the maximum number of consecutive registers to be read at one time. In this example, the valid register read values can be between 1 and 125.

12. In the **Register Write** box, enter the maximum number of consecutive registers that can be written to at one time. In this example, the valid register write values can be between 1 and 100.

13. In the Update Interval box, enter the frequency (in milliseconds) that the I/O Server will read (poll) the items/points associated with this topic. (Different items/points can be polled at different rates by defining multiple topic names for the same PLC and setting different updates rates for each topic.)

14. When all entries have been made, click on the **OK** button to accept the entries and close the dialog box. The **Topic Definition** dialog box will reappear with the new topic listed:
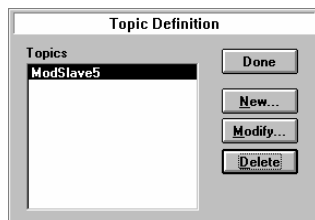
15. Click on the **Done** button to close this dialog box and return to the server's program window.

16. Click on **Modify** to change an existing topic definition. Click on **Delete** to delete an existing topic definition.

Lab – I/O Communication

Summary Tasks

1. Start and configure the Modbus server.
2. Test the server.
3. Create a new window called **Main**, and create new tags. Animate the tags and use them in a factory floor application that you setup yourself.

Section : 3

Troubleshooting I/O Server

This section explains how to troubleshoot I/O Servers, including monitoring communication between the I/O Server and the I/O device and between the I/O Server and InTouch.

Communication Between InTouch and the Server

Some very common error situations can occur when attempting to establish a DDE conversation between InTouch and a I/O Server.

When WindowViewer requires the status of a DDE item, it opens a channel with the I/O Server and requests it to advise WindowViewer whenever the DDE item changes.

The I/O Server automatically handles all of the messages to and from the PLC. The *client* application simply tells the I/O Server what register, coil number or I/O point to read or write. The I/O Server then automatically updates the *client* upon any change of that data value.

It is highly recommended that you always start the I/O Server before starting WindowViewer. If WindowViewer attempts to establish a channel with a I/O Server that is not running, the **Initiating DDE Conversation** message box will appear for each uninitiated DDE conversation:
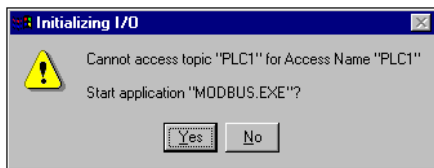
When this dialog box appears, note the information appearing on the second line (for example, **MODBUS|ModSlave5** above). This information indicates that at least one DDE tagname in WindowMaker is linked to a DDE Access Name that defines **MODBUS** as the *application name* and **ModSlave5** as the *topic name*.

In this case, when the WindowViewer program was started, it immediately looked for a I/O Server application named **MODBUS** containing a topic defined as **ModSlave5**. It could not find this application and/or topic and is notifying you that communications cannot be established.
The following describes the three situations that will cause the previous dialog box to appear:

1. The server is not running.

   The I/O Server application (**MODBUS** in this case) is not running. You can verify this by opening the Windows **Task List** (CTRL+ALT) and checking the list of currently running applications for the **MODBUS** server.

2. The server's program name is misspelled in the DDE Access Name definition.

   The server is running, but its name is misspelled in one or more DDE Access Name definitions. The name entered in the DDE Access Name definition **must be the I/O Server's actual program name** (less the .EXE extension) as seen in the Windows File Manager.

3. The topic is not defined in the server or it is misspelled.

Let's assume that the server's name is spelled correctly and is running. WindowViewer is now looking for a topic defined in the server as **ModSlave5**. To check the *topic name*, close WindowViewer (the server cannot be configured if WindowViewer is running) and open the server's program window. Click on the **Configure/Topic Definition** command. If there is a topic defined as **ModSlave5** listed in the dialog box, is it spelled __exactly the same__ (including spaces, etc.) as it is in the DDE Access Name definition?

When you have checked all of the above, restart WindowViewer and switch to the server's program window. Data should now appear in the window. This data indicates that WindowViewer and the server are communicating. The data also indicates read and write messages that the server is attempting to send to the PLC. These are not error messages, only **status** messages are written to this window. If no data appears in the server's window, check the WWLogger for error messages. The most common error message is:

**Error for DDE: MODBUS|ModSlave5!<null> ("item") Advise failed**

This error message indicates that the *item* used on one or more tagnames is unrecognized by the server. **InTouch** tagnames use specific naming conventions when accessing data from a server.

Monitoring Communication Status Between the I/O Server and the I/O Device

For each *topic name* being used, there is a built-in discrete *item*, **Status**, that can be used to monitor the state of your communications with the I/O Servers program. **Status** is set to "0" when communication with the device fails (cable disconnected, PLC is powered down, etc.) and is set to "1" when communication is successful.

From InTouch, you can read the status of server communication by defining a tagname and associating it with the *topic* configured for the device. **Status** must be used as the *item name*. For example, if WindowViewer is communicating with a PLC using the Wonderware Modbus I/O Server, the Access Name definition would be:
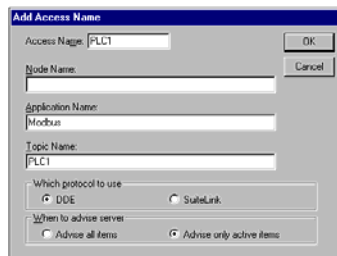
To monitor the status of all communication to the *topic*, PLC1, you would create the following tagname definition:
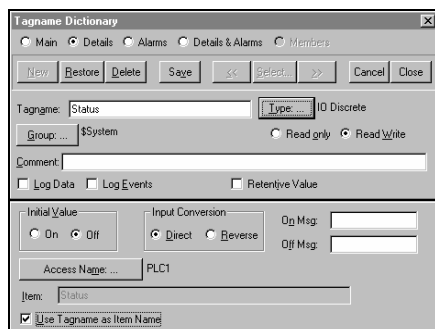
From Excel, you can read the status of PLC communication by entering the following formula in a cell:

=MODBUS|PLC1!'STATUS'

Monitoring Communication Status Between InTouch and the I/O Server

WindowViewer supports a built-in *topic name* called **IOStatus** (**DDEStatus** in versions prior to InTouch 7.0) that can be used to monitor the status of specific I/O conversations.
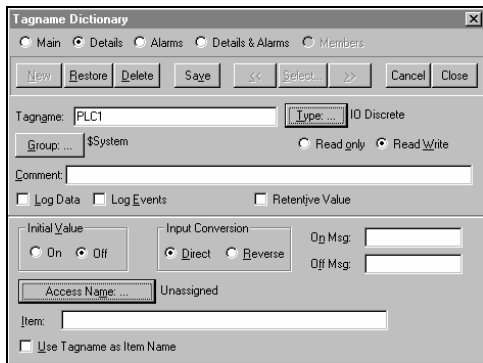
Using IOStatus Topic Name

Let's assume that WindowViewer (View) is communicating with the Wonderware Modbus I/O Server to a PLC that has been defined in the I/O Server with **PLC1** for its *topic name*.

Monitoring the Status of I/O Communications

1. Click on the **Special/Tagname Dictionary** command. The **Tagname Dictionary** dialog box appears.

2. Create an **I/O Discrete** type tagname. (In this example, the tagname is the same as the *topic name* we need to monitor):

    When you are monitoring an I/O conversation using **IOStatus**, you must define at least one I/O type tagname for the Access Name being monitored.

3. Click **Access Name** to assign the tagname to an Access Name definition that defines **IOStatus** for its *topic name*. The **Access Name Definition** dialog box appears:
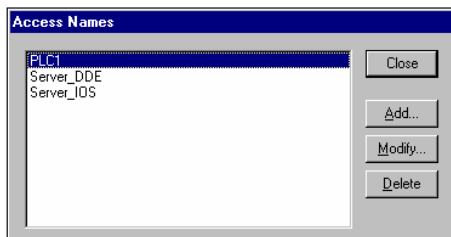
    Notice that an Access Name definition called **PLC1** (the topic we need to monitor) currently exists. To be sure that this is the correct Access Name (its **Topic Name** is **PLC1**), click **Modify** to view the definition:
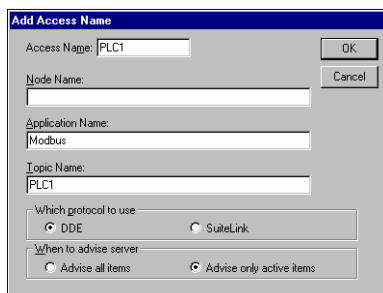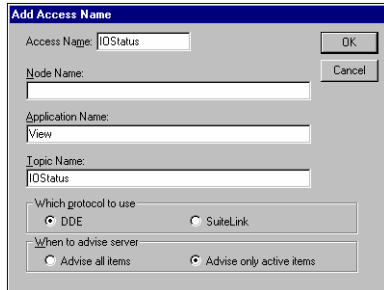
Finding the Access Name containing the right *topic name* in this example is easy because we kept the tagname and the **Topic Name** the same.

4. Click **Cancel** to close the dialog box and return to the initial **Access Name Definition** dialog box.

5. Click **Add**. The **Add Access Name** dialog box appears:

6. In the **Access Name** box, enter **IOStatus**.

7. Since you are going to monitor the status in WindowViewer, in the **Application Name** box, enter "View."

8. In the **Topic Name** box, enter the InTouch internal *topic*, **IOStatus**.

9. Select **Advise only active items**.

10. Click **OK** to close the dialog box. The initial **Access Name Definition** dialog box reappears displaying your new **Access Name**, **IOStatus**, in the list:
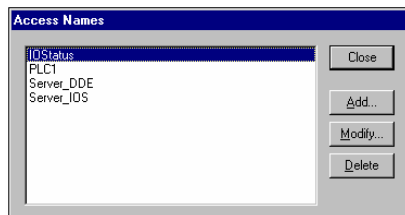
11. Click **Close** to close the dialog box and associate the new **Access Name** with your **I/O Discrete** tagname:
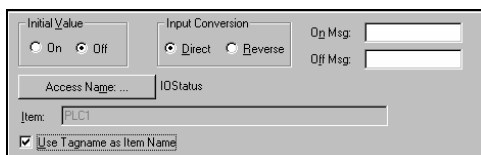
12. In the **Item** box, enter the actual **Access Name** to monitor. In this case, **PLC1**.

   Since your tagname is the same as the **Access Name**, you can select **Use Tagname as Item Name** and automatically enter it for the **Item**.

Using IOStatus Topic Name in Excel

Excel can also be used to perform this same type of monitoring by entering the same information in a formula in a spreadsheet cell. For example, to monitor the same topic as above, the following would be entered:

=view|IOStatus!'PLC1'