Wonderwareò FactorySuiteô InTouch

User's Guide

Revision A
December, 1997

Wonderware Corporation

Contents

Welcome to InTouch

Welcome to Wonderware® InTouch™, the quickest and easiest way to create human-machine interface (HMI) applications for the Microsoft® Windows™ 95 and Windows NT operating systems. InTouch is a component of the Wonderware FactorySuite™. InTouch applications span the globe in a multitude of vertical markets including food processing, semiconductors, oil and gas, automotive, chemical, pharmaceutical, pulp and paper, transportation, utilities, and more.

By using InTouch, you can create powerful, full-featured applications that exploit the key features of Microsoft Windows, including ActiveX controls, OLE, graphics, networking and more. InTouch can also be extended by adding custom ActiveX controls, wizards, generic objects, and creating InTouch QuickScript extensions.

InTouch consists of three major programs, the InTouch Application Manager, WindowMaker and WindowViewer. InTouch also includes the diagnostics program Wonderware Logger.

The InTouch Application Manager organizes the applications you create. It also is used to configure WindowViewer as an NT service, to configure Network Application Development (NAD) for client-based and server-based architectures, to configure Dynamic Resolution Conversion (DRC) and/or distributed alarming. The DBDump and DBLoad database utilities are also 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.

To get started quickly, read this chapter for details on how to install and start-up your InTouch system.

Contents

Special InTouch Features

InTouch includes the following features:

- **Application Explorer**
  The hierarchical Application Explorer provides you with improved navigation capabilities. For example, it displays the names of all the windows you have created, and when you double-click a window name, the window opens. When you right-click a window name, a menu appears displaying the various commands that you can execute to open the window, save the window, open the window's QuickScripts, its properties dialog box, and so on. The Application Explorer also provides you with quick access to all InTouch QuickScript types, all configuration commands, the Tagname Dictionary, the Tagname Cross Reference utility and the SuperTags TemplateMaker. The Application Explorer allows you to add short cuts to launch other FactorySuite programs or third-party applications. Display of the Application Explorer is optional.

- **Applications Run on Windows NT Operating System or Windows 95**
  The applications you create on the Windows 95 or the Windows NT operating systems are interchangeable. They can run on either operating system without requiring conversion.

- **ActiveX Container**
  InTouch is an ActiveX container. It allows you to install any third-party ActiveX control and use it in any application window. For easy access to your installed ActiveX controls, you can add them to your WindowMaker **Wizards/ActiveX Toolbar**. By using ActiveX controls, you can handle control events, control methods, and control properties all from InTouch QuickScripts. You can also associate the ActiveX control properties directly to InTouch tagnames.

- **Extended Tagname Support**
  The InTouch Tagname Dictionary supports up to 61,405 tags. The number of tagnames supported is controlled by your license.

- **Tag Browser**
  The Tag Browser allows you to select tagnames and tagname **.fields** from any FactorySuite application or any other tagname source that supports the InTouch Tagname Dictionary interface. It is your primary tool for editing your Tagname Dictionary.

- **Instrument Failure Monitoring**
  Beginning with Version 7.0, InTouch supports three tagname **.fields** (**.RawValue**, **.MinRaw** and **.MaxRaw**) that you can use in InTouch QuickScripts to monitor instrument values to determine out-of-range, out-of-calibration, or failure.

- **Remote Tagname Referencing**
  Remote tagname referencing allows you to access data from a remote data source without having to create the tagname in your local Tagname Dictionary. Remote tagnames can reference data defined in most I/O data sources using either Microsoft DDE or the Wonderware SuiteLink protocol. For example, the I/O data source may be Microsoft Excel or a remote View node. You can import graphic windows from any InTouch application, and then convert the window's placeholder tagnames to remote tagname references to create a client application that has no local Tagname Dictionary.

- **SuperTags TemplateMaker**
  The SuperTags TemplateMaker allows you to create, modify and delete custom SuperTag templates. SuperTag templates can be defined with up to 64 members. A SuperTag template can be a member of another SuperTag template for a maximum of 2 nesting levels. Members behave exactly like normal InTouch tagnames and can be used in InTouch QuickScripts and animation links. Members also support trending and alarming, as well as all tagname **.fields**.

- **QuickFunctions**
  QuickFunctions are InTouch QuickScripts you create that you can call from other QuickScripts or animation link expressions. QuickFunctions support parameters and return values. Calling QuickFunctions from other QuickScripts or expressions allows you to create a QuickFunction one time, and then reuse it over and over again. Using QuickFunctions decreases your application maintenance because regardless of how many other scripts or animation link expressions call the QuickFunction, only the QuickFunction itself needs to be maintained. By making modifications to the one QuickFunction, you automatically update dozens of other QuickScripts or expressions.

- **Asynchronous QuickFunctions**
  QuickFunctions can be configured as asynchronous. The asynchronous functionality is configured in the WindowMaker development environment and executed in the WindowViewer runtime environment. Asynchronous QuickFunctions run in the background while the main WindowViewer process is running. This allows WindowViewer to separate time consuming operations such as SQL database calls and FOR NEXT loops from the main program flow. When such time consuming operations are performed through asynchronous QuickFunctions, all animation links and other InTouch functionality simultaneously remain active.

- **Tagname Cross Referencing**
  The Tagname Cross Referencing utility allows you to determine both your tagname and SuperTag usage and, in which window or QuickScript that a specific tagname is used. For convenience, the Tagname Cross Reference utility can remain open in WindowMaker while you perform other tasks. It also allows you to view any QuickScript or QuickFunction where a tagname is found.

- **Local Variables**
  InTouch QuickScripts and QuickFunctions support the use of local variables to store temporary results and create complex calculations with intermediate scripting values. By using local variables in QuickScripts and QuickFunctions, you do not decrease your licensed tagname count.

- **WindowViewer as an NT Service**
  Beginning with InTouch 7.0, WindowViewer can be run as an NT service. This provides NT service capabilities for key InTouch components such as historical logging, providing alarms and providing I/O data. The service capabilities allow continuous operation of WindowViewer through operating system log ons and log offs such as, operator shift changes. Another functionality is automatic start up of WindowViewer following power failure or when the machine is turned off and on. This provides unmanned station startup of WindowViewer without compromising NT operating system security.

- **Distributed Alarm System**
  The new distributed system supports multiple alarm servers or "providers" concurrently, giving operators the ability to simultaneously view and acknowledge alarm information from multiple remote locations.

- **Distributed History**
  The distributed historical trending system allows you to dynamically specify a different historical file data source for each pen of a trend chart. This allows an operator to also view both native InTouch history and IndustrialSQL history in the same trend.
- **Dynamic Resolution Conversion**
  You can now develop applications in one screen resolution and run them at another, without affecting the original application. The applications can also be run at a user-defined resolution, instead of the display resolution.
- **Dynamic Reference Addressing**
  Data source references can be changed to dynamically address multiple data sources with a single tagname.
- **Network Application Development**
  New remote development features accommodate large, multi-node installations, including updating of all nodes on a network from a single development station.
- **FactoryFocus**
  FactoryFocus is a view only Runtime version of InTouch. It allows Managers and Supervisors the ability to view a HMI application process in real time. System security is increased with the view only capability since no data can be changed. You do not need to change your InTouch applications to use InTouch FactoryFocus.

  InTouch FactoryFocus functions as a client only. No data can be written out using DDE or SuiteLink, nor can data be poked to programs such as Excel. Alarms can be viewed but not acknowledged. FactoryFocus cannot act as an I/O Server to requesting clients. Features such as animation links, tagnames, real-time and historical trends are view only.
- Other InTouch features and benefits include:
  - Connectivity with more than 300 I/O Servers.
  - Low cost process viewer solution at a price significantly less than a full HMI.
  - VTQ (data Value, with associated Timestamp and Quality) of I/O type tagnames provided by an I/O Server.
  - **HTSelectTag()** function that allows the user, in runtime to select any Historically Logged Tagname.
  - Wonderware SuiteLink protocol. SuiteLink allows application commands (reads, writes and updates) and their associated data to be passed between client applications and server applications.
  - Easily networked with Wonderware NetDDE.
  - Real-time application process viewing.
  - Standard Windows 95/NT GUI format featured.
  - Right-mouse click support throughout WindowMaker for quick access to frequently used commands.
  - Floating and docking toolbars
  - Customizable color palette that provides 16.7 million color support. (The color support is limited only by your video card capability.)
  - Windows 95 and Windows NT operating systems long filename support.

System Requirements

To run InTouch, we recommend the following hardware and software:

- Any IBM® compatible PC with a Pentium 100 processor or higher.
- At least 100MB of free hard disk space.
- At least 32MB of random-access memory (RAM).

  Note We recommend 5MB of RAM per 5K tagnames. For example, 32MB of RAM for 32K tagname support and 128MB of RAM for 60K tagname support.

- SVGA display adapter (2MB RAM recommended).
- Pointing device. For example, mouse, trackball, touch screen.
- Microsoft® Windows® 95 or Windows NT™ operating systems.
- For the Windows 95 operating system to implement the distributed functionality of InTouch, Wonderware NetDDE must be installed and operational.

Note Beginning with Wonderware FactorySuite InTouch Version 7.0, InTouch no longer supports the Microsoft Windows 3.x or Microsoft Windows for Workgroups operating systems.

Installing InTouch

The Wonderware FactorySuite installation program is used to install InTouch. InTouch runs on Microsoft Windows 95 or Windows NT operating systems. The installation program creates directories as needed, copies files from the compact disk to your hard drive.

📖 For complete installation instructions, refer to your FactorySuite installation booklet or your online *FactorySuite System Administrator's Guide*.

About this Manual

This manual is divided into a series of logical building block chapters that describe the various aspects of building an InTouch application. It is written in a "procedural" format that tells you in numbered steps how to perform most functions or tasks.

If you are viewing this manual online, when you see a cross reference like this one, it is actually a "hot link" to the referenced section or chapter. Click it to "jump" to that section or chapter. When you jump to another section or chapter and you want to come back to the original section, a "back" option is provided.

These types of cross references indicate that you need to look in another FactorySuite book for more information.

These are "tips" that tell you an easier or quicker way to accomplish a function or task.

To familiarize yourself with the WindowMaker development environment and its tools, read Chapter 1 - WindowMaker Program Elements. To learn about working with windows, graphic objects, wizards, ActiveX controls and so on, read Chapter 2 - Using WindowMaker.

For details on the runtime environment (WindowViewer), see your *InTouch Runtime User's Guide*.

In addition, the *InTouch Reference Guide* provides you with an in-depth reference to the InTouch QuickScript language and functions, system tagnames, and tagname **.fields**.

The *FactorySuite Systems Administrator's Guide* also provides you with complete information on the common components in the FactorySuite, system requirements, networking considerations, product integration, technical support, and so on.

For details on the add-on program, SPC Pro, see your *SPC Pro User's Guide*.

For details on the add-on program, Recipe Manager, see your *Recipe Managers User's Guide*.

For details on the add-on program, SQLAccess Manager, see your *SQL Access Manager User's Guide*.

Online documentation is included in your FactorySuite software package for all FactorySuite components included in your package. For example, FactorySuite System Administrator's Guide, SPC Pro, SQLAccess Manager, Recipe Manager, IndustrialSQL Sever, InControl, and all Wonderware 32-bit I/O Servers. If you purchase FactorySuite Plus you also get the online documentation for the InTrack and InBatch components.

Assumptions

This manual assumes you are:

- Familiar with the Windows 95 and/or Windows NT operating system working environment.
- Knowledgeable of how to use of a mouse, Windows menus, select options, and accessing online Help.
- Experienced with a programming or macro language. For best results, you should have an understanding of programming concepts such as variables, statements, functions and methods.

Recommended Reading

For additional information on building effective human-computer interfaces, the following sources are recommended:

The Windows Interface: An Application Design Guide, Microsoft Press, 1992.

Dreyfuss, Henry. *Symbol Sourcebook: An Authoritative Guide to International Graphic Symbols.* Van Nostrand Reinhold, 1984.

Laurel, Brenda. *The Art of Human-Computer Interface Design*. Addison-Wesley, 1990.

Norman, Donald A. *The Design of Everyday Things*. Doubleday, 1990.

Tufte, Edward. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

Chappell, David. *Understanding Active X and OLE - A Guide for Developer's and Managers*. Microsoft Press, Strategic Technology Series 1996.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Prior to contacting technical support, please refer to the relevant chapter(s) in your *InTouch User's Guide* for a possible solution to any problem you may have with your InTouch system. If you find it necessary to contact technical support for assistance, please have the following information available:

1. Your software serial number.
2. The version of InTouch you are running.
3. The type and version of the operating system you are using. For example, Microsoft Windows NT Version 4.0 workstation.
4. The exact wording of system error messages encountered.
5. Any relevant output listing from the Wonderware Logger, the Microsoft Diagnostic utility (MSD), or any other diagnostic applications.
6. Details of the attempts you made to solve the problem(s) and your results.
7. Details of how to recreate the problem.
8. If known, the Wonderware Technical Support case number assigned to your problem (if this is an on-going problem).

🕮 For more information on Technical Support, see your online *FactorySuite System Administrator's Guide*.

Your FactorySuite License

Your FactorySuite system license information can be viewed through the license viewing utility that is launched from the WindowMaker Help **About** dialog box.

🖰 To access **About** dialog box, select the **About** command on the WindowMaker **Help** menu.

🕮 For more information on the licensing viewing utility, see your *FactorySuite System Administrator's Guide*.

Running InTouch for the First Time

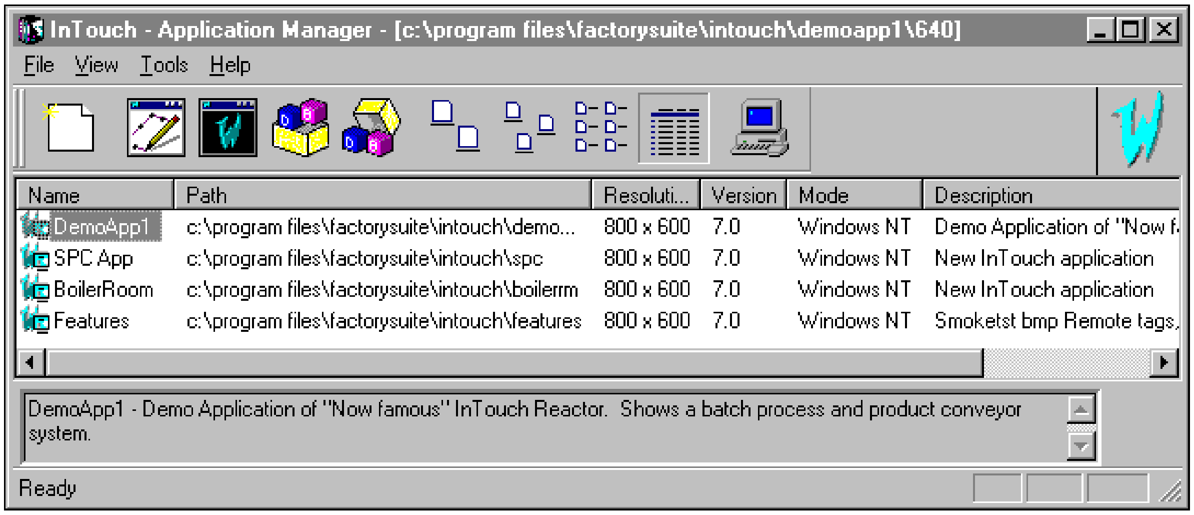
The first time you run INTOUCH.EXE, the INTOUCH.INI file is automatically created. This file contains the default configuration settings for your application. As you configure your application, your settings are written to the INTOUCH.INI file.

Once you have customized your application, when you create a new application, you can copy your customized INTOUCH.INI file to your new application's directory. This eliminates the need for you to reset your customized parameters each time you create a new application.

For more information on customizing your application, see Chapter 2 - Using WindowMaker.

➤ **To run InTouch for the first time:**

1. Start the InTouch program (INTOUCH.EXE).The **Welcome to InTouch Application Manager** dialog box will appear.
2. Click **Next**. A second **Welcome to InTouch Application Manager** dialog box will appear displaying the default path for the starting directory. For example, **C:\programfiles\factorysuite\intouch\**.
3. To specify a different directory, type the path to the directory in the input box, or click **Browse** to locate the directory.
4. Click **Finish**.
5. The **InTouch - Application Manager** will appear and automatically search your computer for any current InTouch applications. If an application(s) is found, an icon with the application's name will appear in the dialog box. For example:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\demoapp1\640]
File View Tools Help

Name	Path	Resoluti...	Version	Mode	Description
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Application of "Now f.
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTouch application
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTouch application
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst bmp Remote tags,

DemoApp1 - Demo Application of "Now famous" InTouch Reactor. Shows a batch process and product conveyor system.

Ready

➢ **To create a new application:**

1. On the **File** menu, click **New**, or click the **New** tool in the toolbar. The **Create New Application** wizard will appear.
2. Click **Next**. A second **Create New Application** wizard will appear.

   By default, the system will display the path to your InTouch directory followed by "**NewApp**."

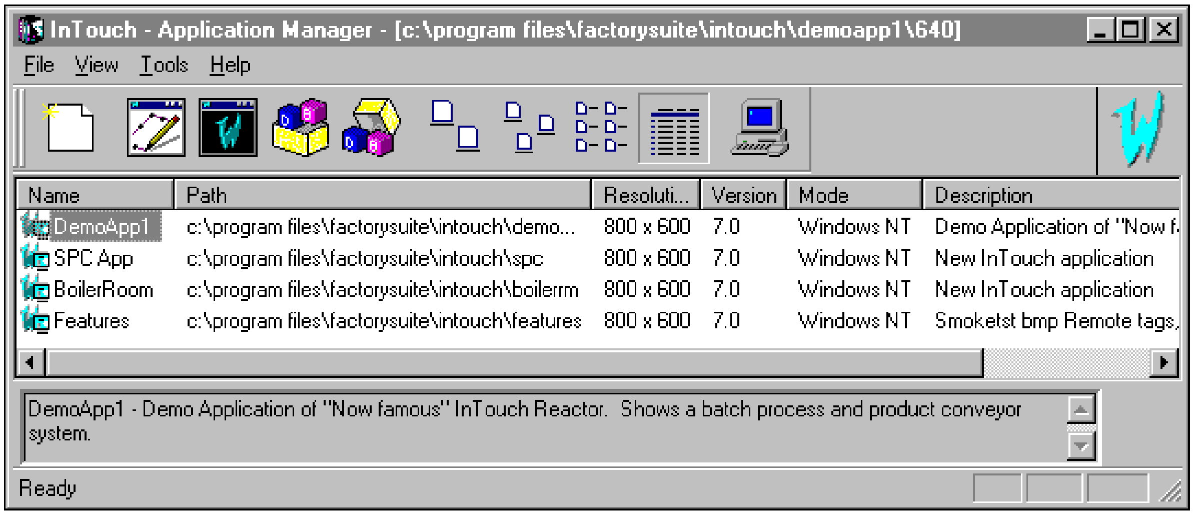
3. In the input box, type the path to the directory in which you want your application to be created or click **Browse** to locate the directory.
4. Click **Next**.

   If the directory you specify does not exist, a message dialog box will appear asking if you want to create it. Click **OK**. A third **Create New Application** wizard dialog box will appear.

5. In the **Name** box, type 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, type a description of the application.

   The description is optional. However, if you do type a description, it can be a total of 255 characters.

7. Click **Finish**. The **InTouch - Application Manager** will reappear displaying an icon with the name you specified for the new application. For example:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\demoapp1\640]
File
View
Tools
Help
Ready

Name	Path	Resoluti...	Version	Mode	Description
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Application of "Now f
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTouch application
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTouch application
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst bmp Remote tags,

DemoApp1 - Demo Application of "Now famous" InTouch Reactor. Shows a batch process and product conveyor system.

8. To open an application, click the right mouse button as you select it, and 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, and then click the WindowMaker tool in the toolbar. (WindowViewer cannot be executed for a new application.)

   To quickly open the application, double-click it's icon or select it, and then press ENTER.

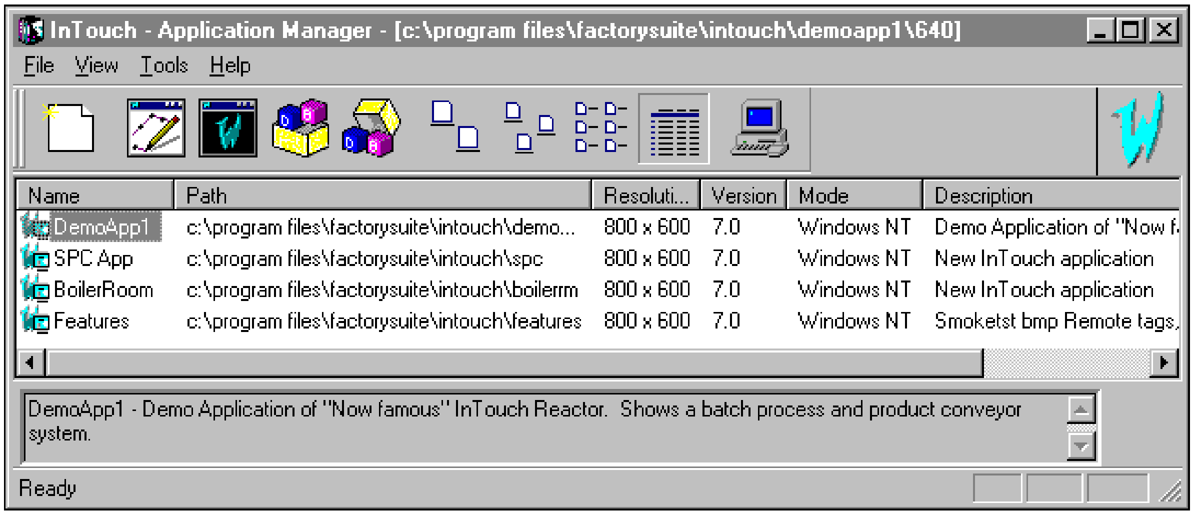
InTouch Application Manager

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.

For more information on the DBDump and DBLoad programs, see Chapter 4 - Tagname Dictionary.

➢ **To run the InTouch Application Manager:**

1. Start the InTouch program (INTOUCH.EXE). The **InTouch Application Manager** dialog box appears:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\demoapp1\640]
File View Tools Help

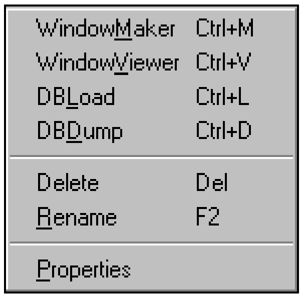
Name	Path	Resoluti...	Version	Mode	Description
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Application of "Now f
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTouch application
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTouch application
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst bmp Remote tags,

DemoApp1 - Demo Application of "Now famous" InTouch Reactor. Shows a batch process and product conveyor system.

Ready

When you select an application in the list, it's name and it's 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.

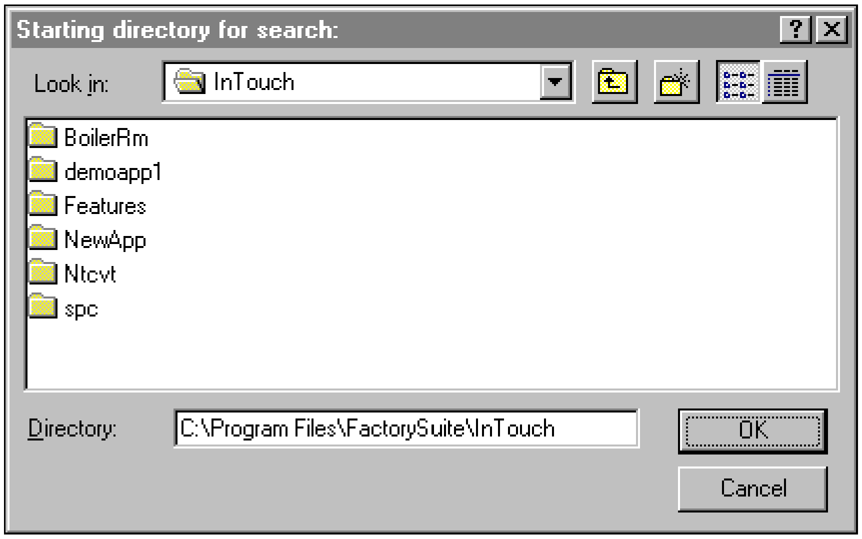
You can also execute several of the InTouch Application Manager's menu commands from the menu that appears when you click the right mouse button as you select an application. For example:



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

3. To delete an application's icon, right-click the application in the list, and then click **Delete**. A message box will appear asking you to confirm the deletion. Click **Yes** to delete the application from the window or click **No** to cancel the deletion.

   If you delete an application from the list, it does not delete your files or directory. If you need to display it later, on the **Tools** menu, click **Find Applications**. The **Starting directory for search** dialog box appears:



Locate the directory in which you want to search for applications, and then click **OK**. The InTouch Application Manager will reappear displaying icons for all applications that were found in the selected directory.

If you right-click in the window, a menu will appear displaying the commands that you can apply to a selected item.

The Application Manager's Tools

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

➢ **To hide the toolbar:**

On the **View** menu, select **Toolbar**. To show it again, repeat this step.

➢ **To hide the status bar:**

On the **View** menu select **Status Bar**. To show it again, repeat this step.

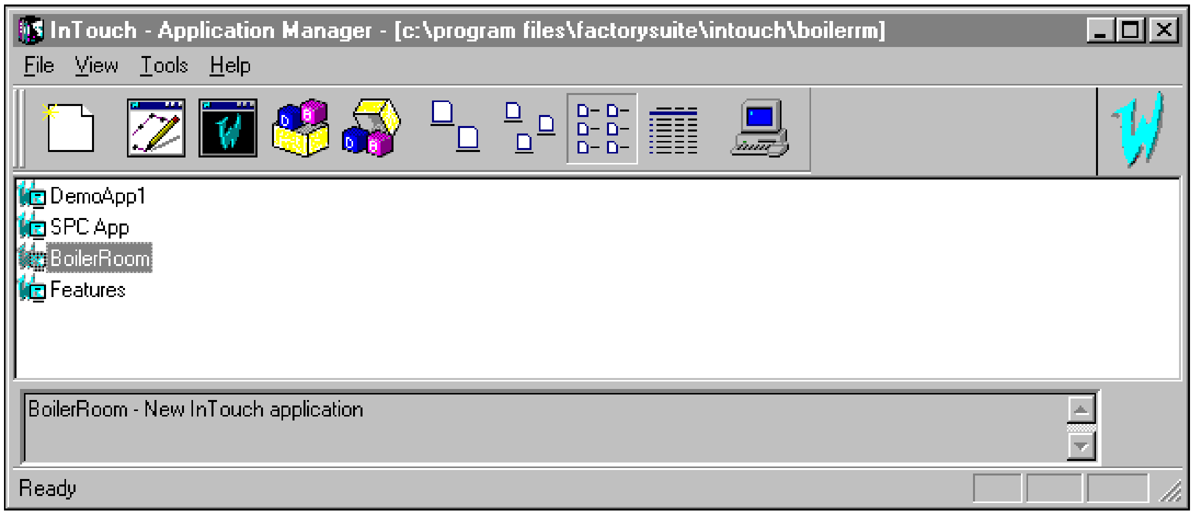
The following briefly describes each of the Application Manager's toolbar buttons:

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. To quickly open an application in WindowMaker, double-click it's icon or select it, and then press ENTER.)
☐	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 used to load a Tagname Dictionary input file.
☐	Executes the **DBDump** command on the **File** menu to run the DBDump utility program used to extract an application's Tagname Dictionary. For more information on the DBDump and DBLoad programs, see Chapter 4 - 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. For example:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\boilerrm]

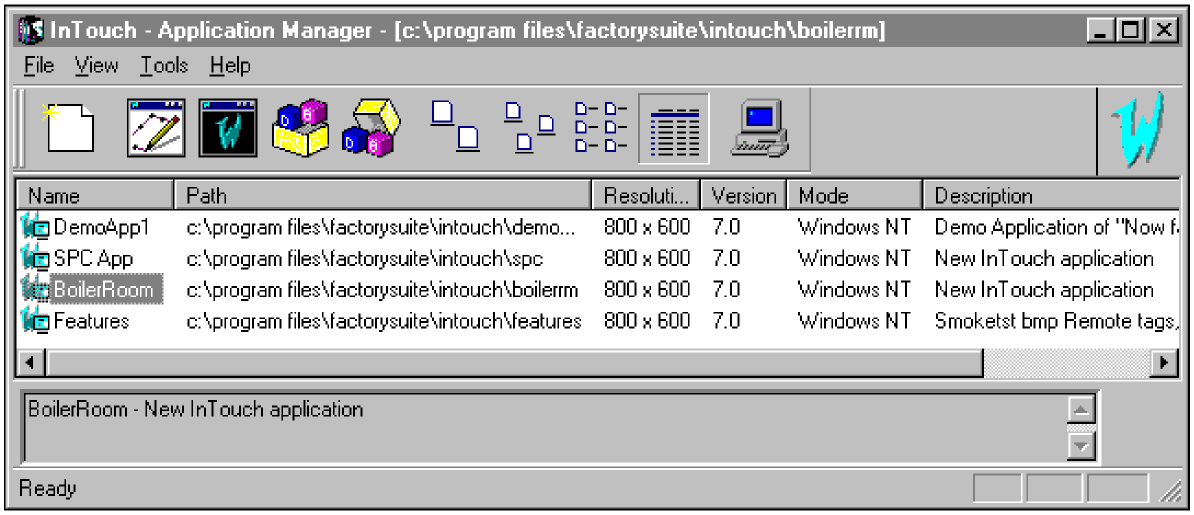
File View Tools Help

DemoApp1
SPC App
BoilerRoom
Features

BoilerRoom - New InTouch application

Ready

Executes the **Details** command on the **View** menu to change the dialog box to the details view mode. For example:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\boilerrm]

File View Tools Help

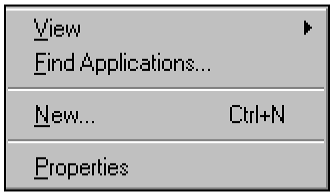
Name	Path	Resoluti...	Version	Mode	Description
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Application of "Now f.
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTouch application
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTouch application
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst bmp Remote tags,

BoilerRoom - New InTouch application

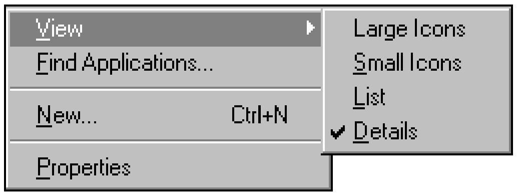
Ready

Double-clicking the right vertical separation bar of a column header will auto-size the column.

If you right-click any of the column headers, or click a blank area of the window, or click a detail (other than the application name) the following menu appears:



If you point to **View**, the following submenu appears:



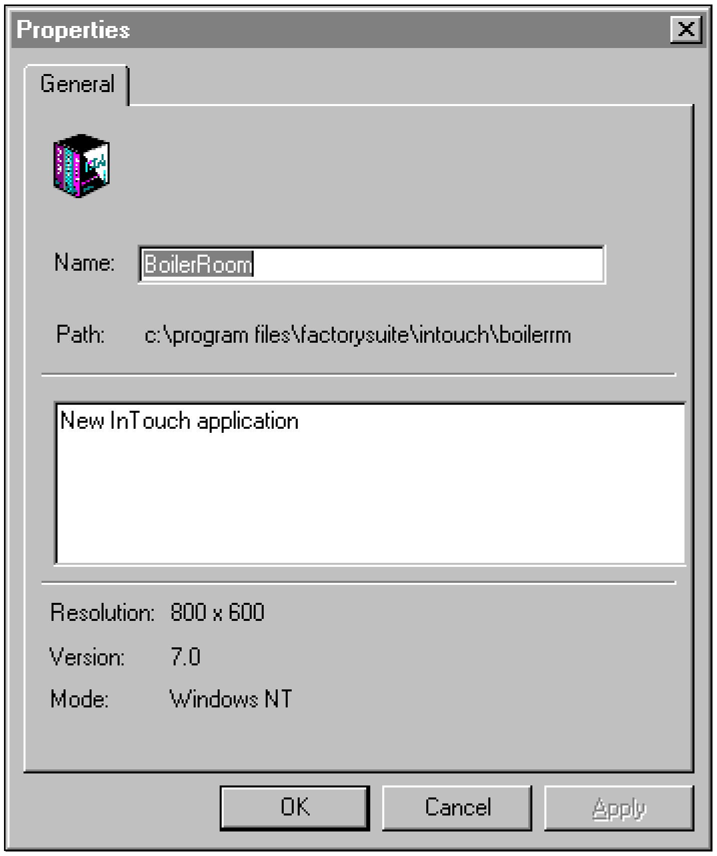
These commands are also found on the **View** menu. They control the display list in the InTouch Application Manager.

Opens the **Node Configuration** dialog box that you will use to set the computer's properties when you are using Network Application Development (NAD), Dynamic Resolution Conversion (DRC) and/or distributed alarming.

For more information on NAD and DRC, see Chapter 3 - Building a Distributed Application.

For more information on distributed alarms, see Chapter 7 - Alarms/Events.

Note When an application is selected in the Application Manager display list, selecting the **Properties** command on the **File** menu will cause the **Properties** dialog box for that application to appear:



Name: BoilerRoom

Path: c:\program files\factorysuite\intouch\boilerrm

New InTouch application

Resolution: 800 x 600

Version: 7.0

Mode: Windows NT

C H A P T E R 1

WindowMaker Program Elements

WindowMaker is the development environment for InTouch. The WindowMaker graphical user interface adheres to Windows 95 and Windows NT GUI standards. WindowMaker supports floating and docking toolbars, right-mouse click menus throughout for 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.)

WindowMaker's Application Explorer provides you with a powerful, graphical method for navigating and configuring your InTouch applications. It provides you with 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 and it provides you with a customizable application launcher.

On the Windows NT operating system, you can configure the Application Explorer to launch any other FactorySuite program or Windows program to quickly switch between HMI configuration, I/O Server configuration and control configuration.

Contents

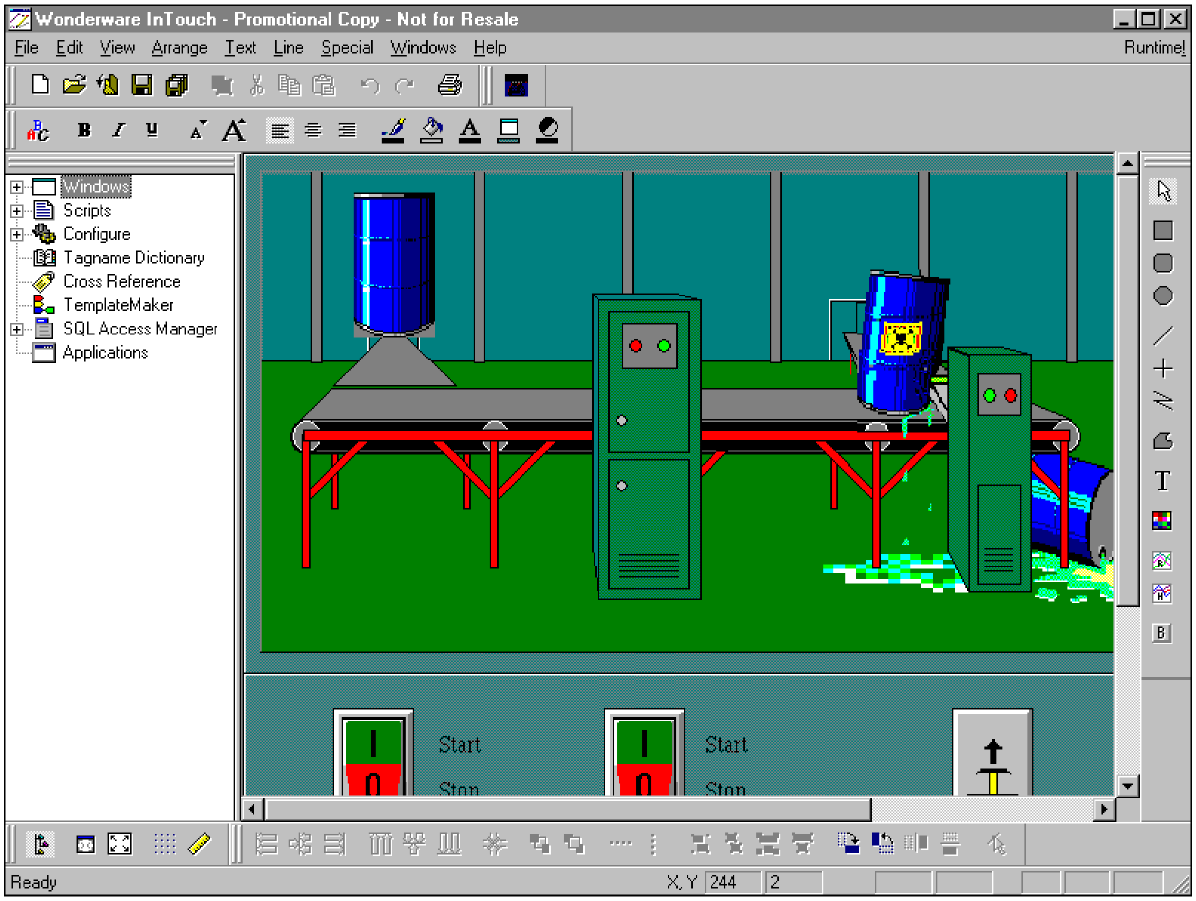
The WindowMaker GUI

WindowMaker supports the Windows 95 and Windows NT operation systems graphic user interface (GUI) standards including, right-click mouse support, floating and docking toolbars, pull down menus, context-sensitive help and so on.

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 that you desire within the WindowMaker window. You can also display the optional ruler and you can turn on and off the visible grid in your windows.

For more information on moving the toolbars see, "Working with the Floating/Docking Toolbars."

The following illustrates the elements of the WindowMaker development environment:



When you create a new application, and run WindowMaker for the first time, its program elements will automatically appear in the default configuration shown in the illustration above.

Many of the tools will not become active until a window is opened and objects are placed in the window and then selected. When a tool is not active, its functionality is not applicable for the current state of the window or the selected object.

Note When you close WindowMaker, the toolbar floating or docked positions and sizes, Application Explorer and, WindowMaker window size preferences are all saved. When you subsequently run WindowMaker they will be persistent.

The Application Explorer

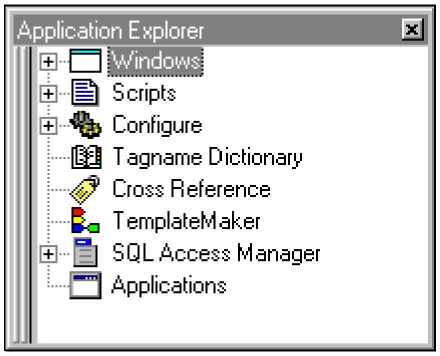
WindowMaker's Application Explorer is a hierarchical graphical view of your application. It shows you what items you have configured in your application and provides you easy access to those items. It also provides you with quick access to many of WindowMaker's most commonly used commands and functions.

Note On the Windows NT operating system, you can configure the Application Explorer to launch any other FactorySuite program or Windows program. This powerful feature allows you to quickly switch between your HMI configuration, I/O Server configuration and control configuration.

Do not add WindowViewer (VIEW.EXE) to the Application Explorer. If you add WindowViewer, new windows you create in WindowMaker may not be synchronized with the windows in WindowViewer. The proper way to launch WindowViewer is by executing the **WindowViewer** command on the **File** menu, or by clicking the **Runtime** fast switch in the WindowMaker toolbar.

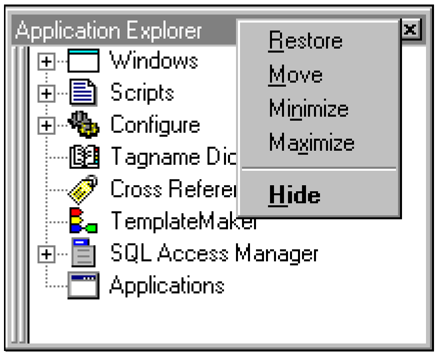
Like all WindowMaker's toolbars, the Application Explorer can be "docked" to any edge of the WindowMaker window or, "floated" anywhere within the WindowMaker window.

When you dock the Application Explorer to an edge of the WindowMaker window, it will automatically size itself accordingly and, if required, scroll bars will be displayed. When you float the Application Explorer within the WindowMaker window its title bar will be displayed. Like all WindowMaker toolbars, when the Application Explorer is floating, you can change its size. For example:



For more information on docking/floating the Application Explorer see, "Working with the Floating/Docking Toolbars."

If you right-click the Application Explorer's title bar, the following menu appears:



For more information on this menu see, "Right-Click Menus."

For more information on the right-click functionality within the Application Explorer, see "Navigating in the Application Explorer."

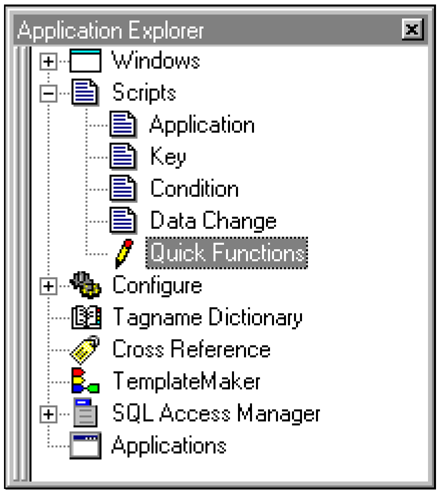
➢ **To show/hide the Application Explorer:**

1. On the **View** menu, click **Application Explorer**. (When you initially start WindowMaker, by default, the Application Explorer is displayed.)
2. Repeat step 1 to close the Application Explorer.
   - To quickly hide the Application Explorer, click the Application Explorer tool on the **View** toolbar.
   - To quickly hide the Application Explorer when it is floating in the WindowMaker window, click the ☒ button on its title bar or, right-click the title bar then, click **Hide** on the menu. When you show the Application Explorer again, it will reappear in its previous size and location in the window.

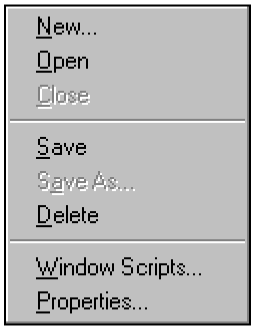
Navigating in the Application Explorer

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, and so on.

- All groups that contain members will be preceded with a ⊞. You can click the ⊞ to quickly expand the group and view its members. Likewise, you can click the ⊟ to collapse the group and hide its members. For example:



The following section briefly describes the behavior of each group listed in the Application Explorer when you perform the described action:

Item	Action	Description
Windows	Double-click, or click ⊞	Expands the view to display the names of all existing windows in your application. • Double-click a window name to open it. • Right-click a window name to display a menu of commands that you can apply to the window. For example:  New... Open Close Save Save As... Delete Window Scripts... Properties...
	Double-click, or click ⊟	Collapses the group to hide its members.
	Right-click	Displays a **New** button. Click **New** to open the **Windows Properties** dialog box to create a new window.
Scripts	Double-click, or click ⊞	Expands the view to display all InTouch QuickScript types: Application Key Condition Data Change QuickFunctions • Double-click a QuickScript type to open it. • Right-click a QuickScript type, an **Open** button appears. Click **Open** to open the QuickScript.
	Double-click, or click ⊟	Collapses the group to hide its members.

Configure	Double-click, or click ⊞	Expands the view to display many of WindowMaker's configuration commands and the **Wizard/ActiveX Installation** command. • Double-click a configuration item to open its respective dialog box. • Right-click a configuration item, an **Open** button appears. Click **Open** to open the item's respective dialog box.
	Double-click, or click ⊟	Collapses the group to hide its members.
Tagname Dictionary	Double-click	Opens the **Tagname Dictionary** dialog box displaying the last modified tagname's definition. Otherwise, the default **$AccessLevel** system tagname is displayed.
	Right-click	Displays an **Open** button. Click **Open** to open the **Tagname Dictionary** dialog box displaying the last modified tagname's definition. Otherwise, the default **$AccessLevel** system tagname is displayed.
Cross Referencing	Double-click	Opens the **Cross Reference** utility.
	Right-click	Displays an **Open** button. Click **Open** to open the **Cross Reference** utility.
TemplateMaker	Double-click	Opens the SuperTag **TemplateMaker** utility.
	Right-click	Displays an **Open** button. Click **Open** to open the SuperTag **TemplateMaker** utility.
Add-on Programs	Double-click, or click ⊞	Expands the view to display the add-on program's configuration commands. Double-click a command to open it's respective dialog box. • Right-click a command, an **Open** button appears. Click **Open** to open the command's respective dialog box. **Note** The add-on programs must be installed to appear in the Application Explorer.
	Double-click, or click ⊟	Collapses the group to hide its members.

Applications	Double-click, or click ⊞	Expands the view to display all other applications that you can launch from WindowMaker. • Double-click an application to launch it without exiting WindowMaker. • Right-click an application name to display a menu of commands that you can apply to the application.
	Right-click	Displays a **New** button. Click **New** to add an application to the Application Explorer (see below).
	Double-click, or click ⊟	Collapses the group to hide its members.

Adding Applications to the Application Explorer

One of the most powerful features of the WindowMaker Application Explorer, on the Windows NT operating system, is its ability to launch other FactorySuite and third-party Windows applications from within WindowMaker.

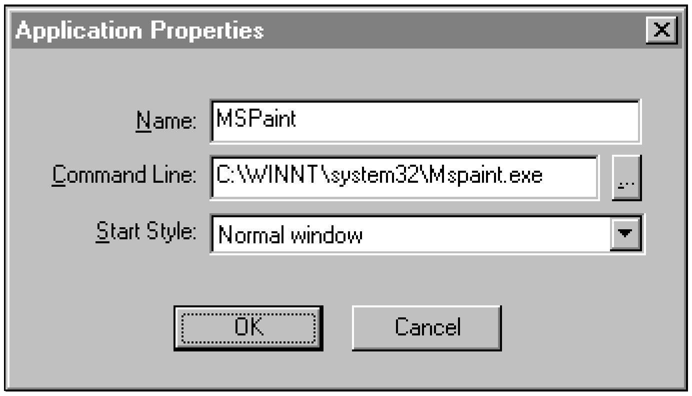
For example, you can run your I/O Server program and configure it at the same time that you are developing your application. You can launch third-party Windows programs that you frequently use such as Windows Notepad, Wordpad, Microsoft Excel, Microsoft Word, Microsoft Paint, and so on.

The InTouch add-on programs, SQL Access, SPC Pro and Recipe Manager are automatically added to the Application Explorer when you install them.

You can also configure the Application Explorer to open a specific document or spreadsheet in a program. For example, if you select a specific Microsoft Word document or Microsoft Excel spreadsheet, when you double-click that application's icon in the Application Explorer, the application will start up and automatically display the document or spreadsheet that you selected. These documents display the icon of the application in which they were originally created, or the .EXE configured as the associated application.

➢ **To add an application to the Application Explorer:**

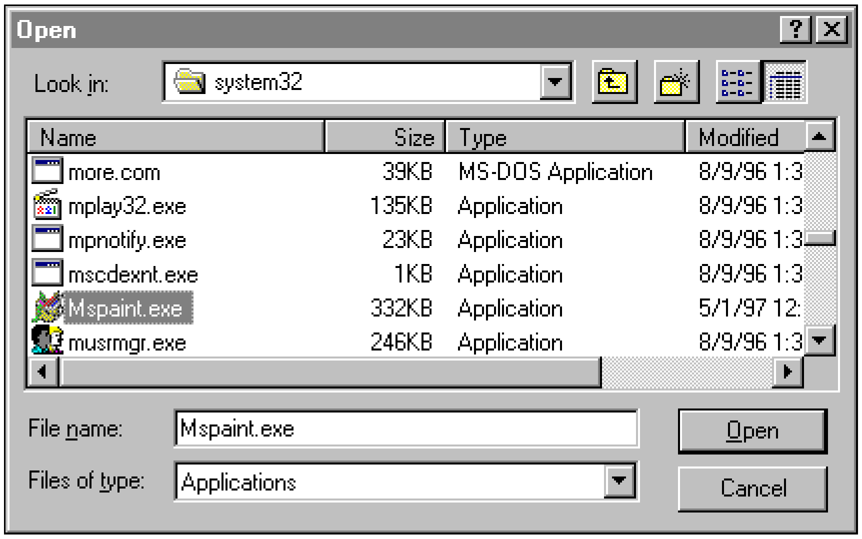
1. Display the Application Explorer.
2. Right-click **Applications**. A **New** button appears.
3. Click **New**. The **Application Properties** dialog box appears:



Name: MSPaint
Command Line: C:\WINNT\system32\Mspaint.exe
Start Style: Normal window

4. In the **Name** box, type the name that you want to display in the Application Explorer for the application.

5. In the **Command Line** box, type the full path for the application or, click the Ellipsis button. The **Open** dialog box appears:


Open

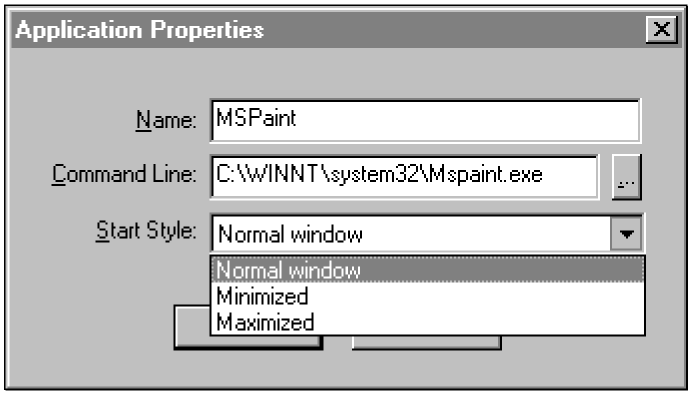
Look in: system32

Name	Size	Type	Modified
more.com	39KB	MS-DOS Application	8/9/96 1:3
mplay32.exe	135KB	Application	8/9/96 1:3
mpnotify.exe	23KB	Application	8/9/96 1:3
mscdexnt.exe	1KB	Application	8/9/96 1:3
Mspaint.exe	332KB	Application	5/1/97 12:
musrmgr.exe	246KB	Application	8/9/96 1:3

File name: Mspaint.exe

Files of type: Applications

6. Locate the application then, click **Open**. The **Application Properties** dialog box reappears:


Application Properties

Name: MSPaint

Command Line: C:\WINNT\system32\Mspaint.exe

Start Style: Normal window

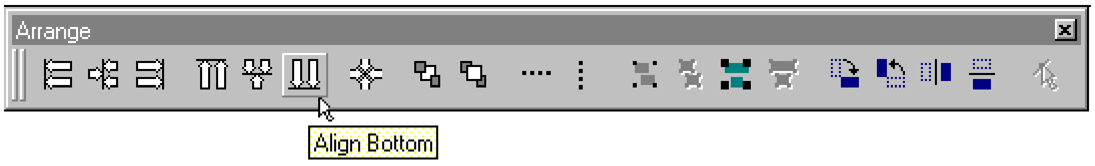
7. Click the **Start Style** arrow and select the style that you want for the application when you run it from WindowMaker.

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

The 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. For example:



Working with the Floating/Docking Toolbars

The WindowMaker toolbars have "floating and docking" capability. Meaning you can move any toolbar from its default "docked" position and dock it again on any edge of the WindowMaker window or, in the toolbar area at the top of WindowMaker's window. Docked toolbars can also be moved from their docked position at the edge of the window and floated within the window. When a toolbar is floating, it will have a title bar and you can change its size.

The Application Explorer can also be docked or floated anywhere in the window and its size can also be changed when it is floating just like any other toolbar.

For more information on the Application Explorer, see "The Application Explorer."

➢ **To change a docked toolbar's location in the window:**

1. Click the toolbar's "cool bars" or, on a blank area of the docked toolbar.
2. Hold down the left mouse button as you move the toolbar away from the edge of the window or, out of the toolbar area, or any edge of the WindowMaker window.
3. Move the toolbar to another edge of the window or, to a new position in the toolbar area.

   If you move a horizontally docked toolbar to the left or right edge of the WindowMaker window, it will automatically change to its default vertical shape when in position for docking to that edge. Likewise, if you move a vertical toolbar to the toolbar area at the top of the window or, to the bottom edge of the window, it will change to its default horizontal shape when in position for docking.
4. Release the mouse. The toolbar will be docked in the new location.

   When a toolbar is docked, you cannot change its size nor can you access its right-click menu.

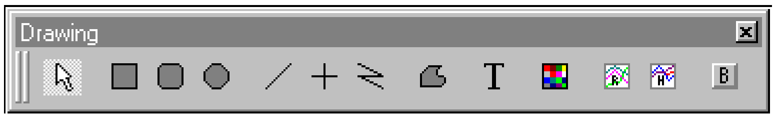
➢ **To show/hide a docked toolbar:**

1. On the **View** menu select the toolbar's name. (When you initially start WindowMaker, by default, all toolbars are showing.)
2. Repeat step 1 to reverse your selection.

   When you show a docked toolbar that has been hidden again, it will reappear in its last docked location in the window.

➢ **To float a docked toolbar:**

1. Click the toolbar's "cool bars" or, a blank area of the docked toolbar.
2. Hold down the mouse button as you move the toolbar from its docked position to a new location within the WindowMaker window.
3. Release the mouse button. The toolbar will appear as follows:



   When a toolbar is floating, you can change its size.

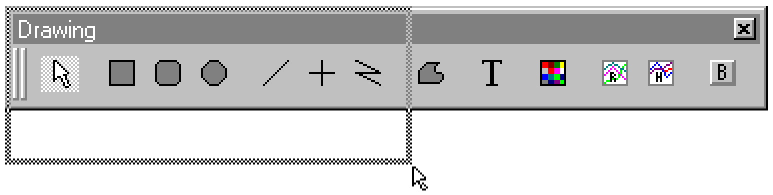
   When you dock a floating toolbar on the left or right edge of the WindowMaker window, it will automatically change to its default vertical shape when it is in position for docking to that edge. Likewise, if you move it to the toolbar area at the top of the WindowMaker window or, to the bottom edge of the WindowMaker window, it will change to its default horizontal shape when in position for docking.

   This will also occur when you dock a floating toolbar whose size you have changed. However, when you float the toolbar in the window again, it will return to its previous floated size.

➢ **To change the size of a floating toolbar:**

1. Move the mouse over any edge of the toolbar. The cursor will change to a double-ended arrow.
2. Click on the edge and hold down the mouse button as you move the mouse to size the toolbar.

   As you move the mouse, a box will appear to indicate the size the toolbar will be if you release the mouse button. For example:



3. Release the mouse button when the toolbar is the desired size.

   When you right-click the title bar of a floating toolbar, a menu appears displaying the commands you can apply to the toolbar.

   For more information on this menu, see "Right-Click Menus."

➢ **To hide/show a floating toolbar:**

1. To hide a floating toolbar, on the **View** menu, select the toolbar's name or, right-click the toolbar's title bar then, click **Hide** on the menu.

   To quickly hide the toolbar, click the ☒ button on the toolbar's title bar.

2. To show a hidden floating toolbar, on the **View** menu select the toolbar's name.

   The toolbar will reappear at its previous location and in its previous size.

➢ **To hide all toolbars at once:**

1. On the **View** menu, click **Hide All** or, click the Hide/Restore All tool on the **View** toolbar. All toolbars and the Application Explorer will be hidden.

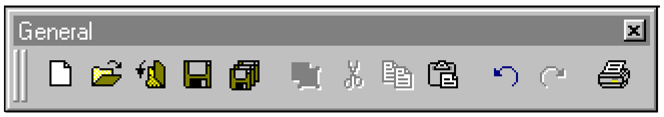
2. Repeat step 1 to show them again.

   All displayed toolbars will have a check mark preceding their names on the **View** menu.

The following section briefly describes each WindowMaker toolbar and each tool it contains.

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:



When you right-click a blank area of an open window, or right-click a window name under **Windows** in the Application Explorer, a menu appears that also contains most of these same windows commands.

For more information on right-click menus, see "Right-Click Menus."

Button	Description
🗋	Executes the **New Window** command on the **File** menu to open 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 **Duplicate** command on the **Edit** menu to duplicate the currently selected object(s) in the window.

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 **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.

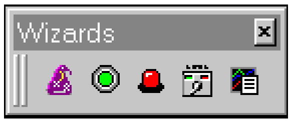
Executes the **Redo** command on the **Edit** menu to reverse (redo) last undo action or command applied to an object.

Note By default the number of undo/redo levels is set to 10. You can increase the **Levels of Undo** to 25 in the **WindowMaker Properties** dialog box. To access this dialog box, in the Application explorer, under **Configure**, double-click **WindowMaker** or, on the **Special** menu, point to **Configure** then, click **WindowMaker** on the submenu.

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

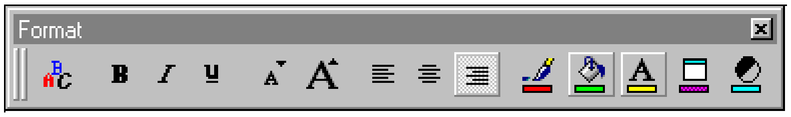
The **Wizards/ActiveX** toolbar, by default, only contains the wizard tool that you use to access the wizard **Selection Dialog** box. However, you can add any installed wizard or ActiveX control to the toolbar. For example:



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

Format Toolbar

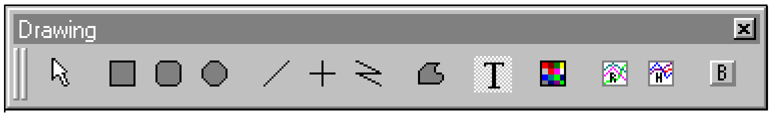
The **Format** toolbar is grouped with tools that execute most of the text object formatting commands found on the **Text** menu. It also contains the tools that you will use to access the color palette to select line, text, fill, window background and transparent object color.

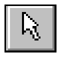
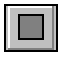
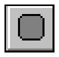
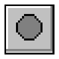
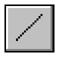
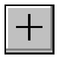
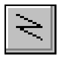
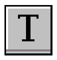
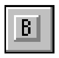


Button	Description
ABC	Executes the **Font** command on the **Text** menu to open the **Font** dialog box used to select the font, its style and size.
B	Executes the **Bold** command on the **Text** menu to apply **bold** styling to single or multiple text string selections and numeric value fields.
I	Executes the **Italic** command on the **Text** menu to apply *italic* styling to single or multiple text string selections and numeric value fields.
U	Executes the **Underline** command on the **Text** menu to apply underline styling to single or multiple text string selections and numeric value fields.
A	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 this tool on the **Text** toolbar.
A	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 this tool on the **Text** toolbar.
≡	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.
A	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

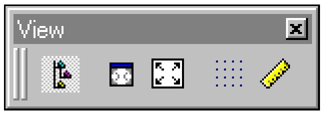
The **Draw Object** toolbar is grouped with all the tools that you will use to draw both simple graphic objects such as rectangles, ellipses, lines or text objects and, complex objects such as real-time trends, historical trends, bitmaps and 3-dimensional buttons with labels in your windows:



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.
	Polygon tool used to draw polygon objects.
	Text tool used to type 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.

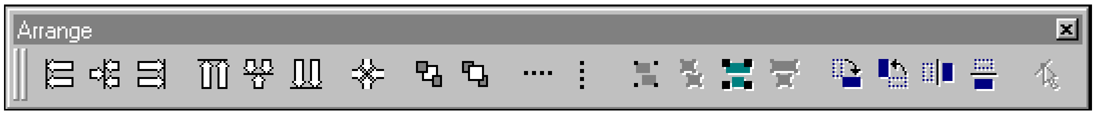


Button	Description
	Turns on and off the **Application Explorer** command on the **View** menu to show/hide the Application Explorer.
	Toggles the **Hide All** command on the **View** menu on and off to hide/show all docked toolbars.
	When the hide all mode is active, the overall size of WindowMaker remains the same. To return to normal mode, click the Hide/Restore All tool on the floating **View Toolbar**, or click **Hide All** on the **View** menu.
	In the hide all mode, all floating toolbars remain visible and the **View Toolbar** automatically floats on top of WindowMaker. If you dock any of the floating toolbars while 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 **Full Screen** on the **View** menu.
	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. Works with the **Snap to Grid** command on the **Arrange** menu.
	If the **Snap to Grid** option in the **WindowMaker Properties** dialog box is not selected, this tool will have no effect.
	Turns on and off the **Ruler** command on the **View** menu to show/hide the ruler.

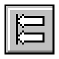
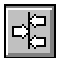
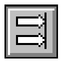
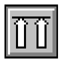
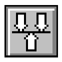
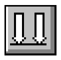
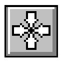
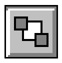
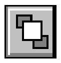
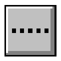
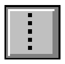
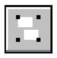
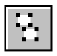
For more information on the ruler, see "The WindowMaker Ruler."

Arrange Toolbar

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



The following briefly describes each tool:

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, so 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, so 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 degree.

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

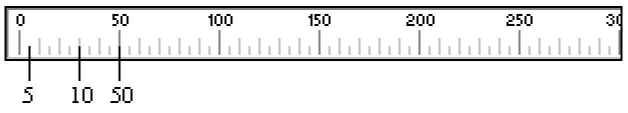
Executes the **Flip Vertical** command on the **Arrange** menu to flip selected objects vertically.

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

The WindowMaker Ruler

You can use the WindowMaker ruler to do precision alignment of the objects in your windows.

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. For example:



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, if you want to move an object 10 pixels at a time, hold down the SHIFT key while you press an arrow key. To move an object 50 pixels at a time, hold down CTRL key while you press an arrow key.

When you select an object and only press an arrow key, the object is moved 1 pixel at a time.

These features can be useful when you need to make fine alignment and location adjustments.

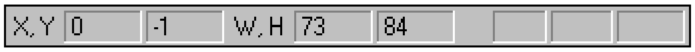
For more information on the using the arrow keys, see "Moving Objects with the Arrow Keys."

➢ **To show/hide the ruler:**

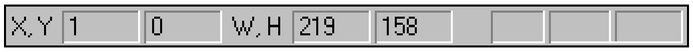
1. On the **View** menu, click **Ruler** or, click the ruler tool in the **View** menu.
2. Repeat step one to hide the ruler.

The WindowMaker Status Bar

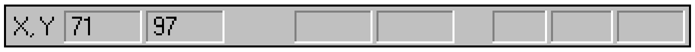
When you select an object in a window, the WindowMaker status bar displays the object's upper left X and Y pixel coordinates and the object's pixel height and width. For example:



When you select multiple objects, 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 show for the entire group. For example:



When you click a blank area of a window the status bar displays the X and Y coordinates for the current location of the cursor in the window. For example:



➢ **To show/hide the status bar:**

1. On the **View** menu, click **Status Bar**.
2. Repeat step one to hide the **Status Bar**.

The WindowMaker Color Palette

The WindowMaker palette that provides 16.7 million color support. (The color support is limited only by your video card capability.) By default, the palette offers you 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.

Using the Standard Color Palette

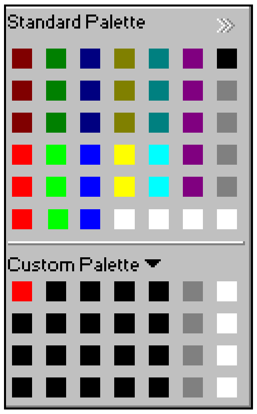
The WindowMaker color palette is used to apply color to static and dynamic properties of lines, rectangles, round rectangles, ellipses, polylines, polygons and text. It is also used to select the background color for your windows and the transparent color for bitmaps that allows you to view objects behind bitmaps.

For more information on transparent bitmaps, see Chapter 2 - Using WindowMaker.

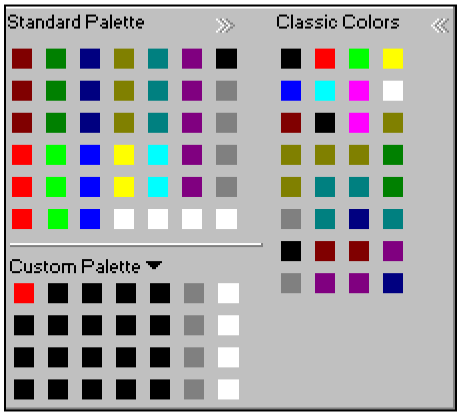
The color palette appears whenever you click a colored square in a dialog box or, you click one of the color tools to apply line, fill or text color to a selected object.

➢ **To use the standard color palette:**

1. To select a standard color, click the color that you want to use 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:

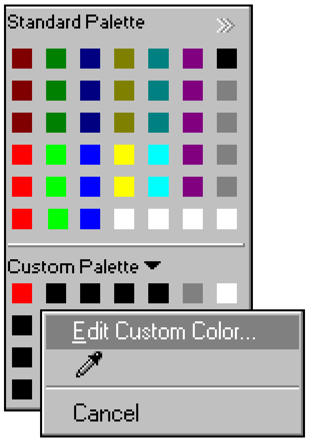


Creating a Custom Color Palette

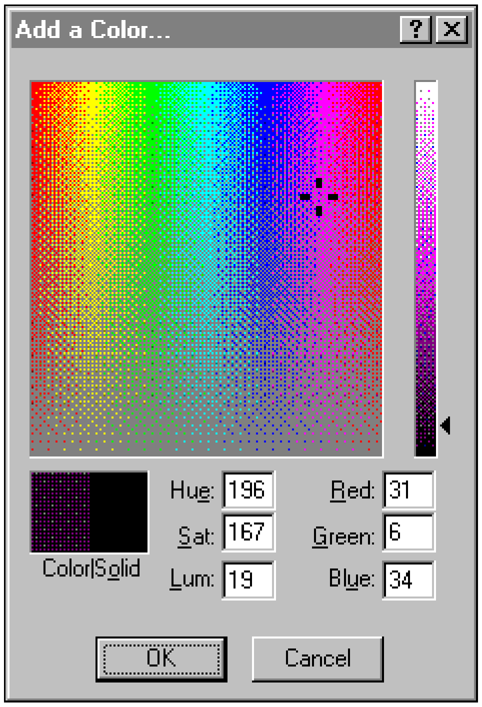
The WindowMaker color palette allows you to define custom colors and add them to your palette. It also allows you to import palettes created in other windows applications and add them to the standard palette. You can also export your custom palettes to other windows applications.

➢ **To create 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:



Hue: 196
Sat: 167
Lum: 19
Red: 31
Green: 6
Blue: 34

4. Click anywhere in the matrix then, use the slider at the right of the dialog box to adjust the color's attributes.

Hue, Sat, Lum
A combination of hue, saturation and luminosity can be used to define any color.

Hue is the value of a color wheel, where 0 is red, 60 is yellow, 120 is green, 180 is cyan, 200 is magenta and 240 is blue. Saturation is the amount of color in a specified hue, up to a maximum of 240. Luminosity is the brightness of a color. If you change any of these values, the red, green and blue scales will change to match.

The easiest way to experiment with different colors is to press and hold the mouse then, move the cursor around in the color matrix.

Red, Green, Blue
A combination of red, green and blue levels can be used to define any color. You can see the effect of changing these values in the color matrix. If you change these values, the values for hue, saturation and luminosity will change to match.

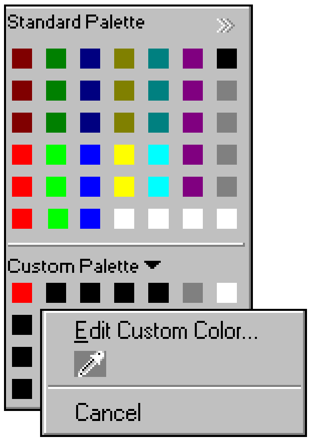
If you define a color using the **Hue**, **Sat**, **Lum** or **Red**, **Green**, **Blue** scales, you can view the color in the **Color|Solid** boxes to make sure you defined the color as you intended.

The **Color** (left) box shows the amount of white and black in the color you specified. The **Solid** (right) box, shows how the color will look if you choose 100% of the color with no white and black. To adjust the color, use the slider at the right of the dialog box. To specify that you want 100% of the color with no white or black, type ALT + O.

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

➢ **To select 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:

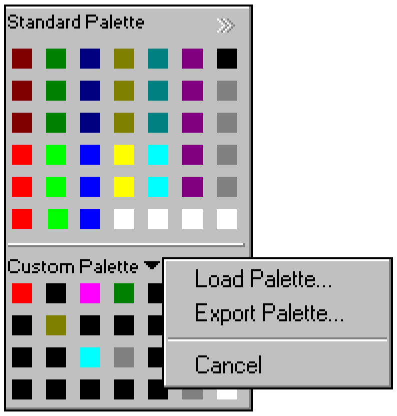


3. Click the blotter tool then, click the color that you want to add to the **Custom Palette** section of the color palette. You can select any color anywhere within the WindowMaker window or outside of WindowMaker completely. This feature is primarily intended to be used when you are creating transparent bitmaps.

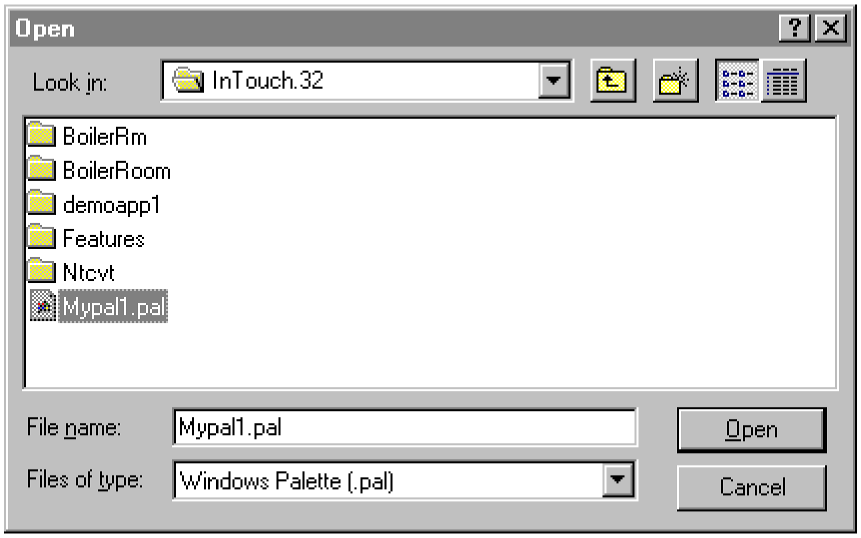
For more information on creating transparent bitmaps, see Chapter 2 - Using WindowMaker.

➢ **To import a custom palette:**

1. Open the color palette.
2. Click the **Custom Palette** arrow. The following menu appears:



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



4. Locate and select the palette (.PAL) file then, click **Open** or double-click the file name. The colors contained in your palette will be loaded into the **Custom Palette** section of the color palette.

➢ **To export a custom palette:**

1. Open the color palette.
2. Click the **Custom Palette** arrow then, click **Export Palette** on the menu (shown above).
3. The standard Windows **Save As** dialog box will appear. Specify the name that you want to save the palette as then, click **Save**.

   The palette must be saved with the .PAL extension.

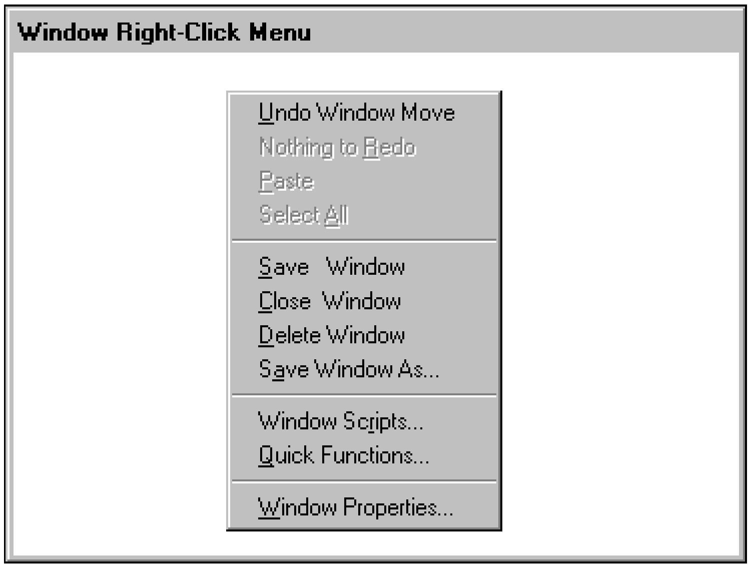
Right-Click Menus

InTouch supports right-click mouse functionality to display menus for frequently used commands for windows and graphic objects. Right-click menus containing the commands that can be applied to selected text in dialog box, text box, an animation link tagname or expression box or, in a QuickScript window is also supported. Instead of using the standard menus to find the command you want to use, you can simply right-click the window, object, dialog box text box, or the groups and their members in the Application Explorer.

Note To turn off the right-click functionality, add the line **oldrightmousebehavior=**1 to your INTOUCH.INI file.

➢ **To access the window right-click menu:**

1. Right-click a blank area of a window. The following menu appears:

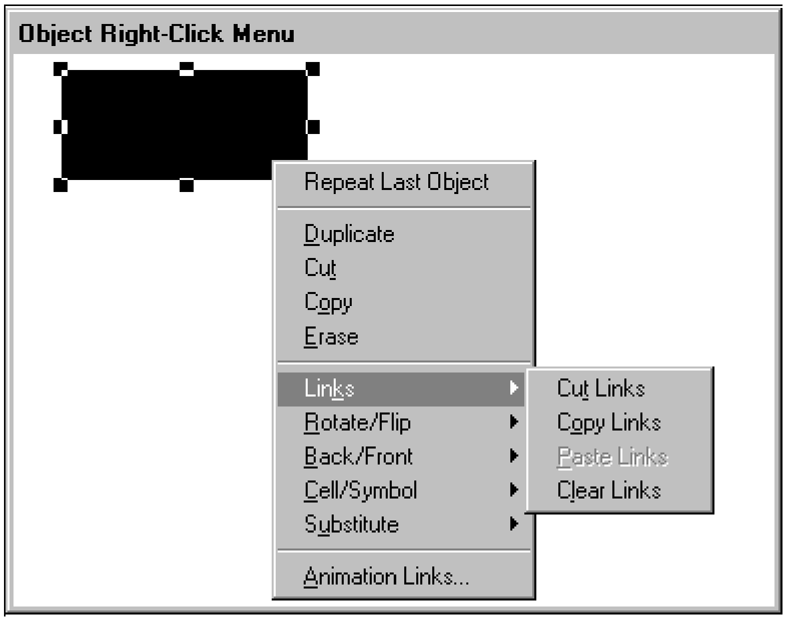


2. Click the command that you want to use on the menu.

Note Commands that are not applicable for the current state of the window will not be active.

➢ **To access the graphic object right-click menu:**

1. Right-click an object in the window. The following menu appears:



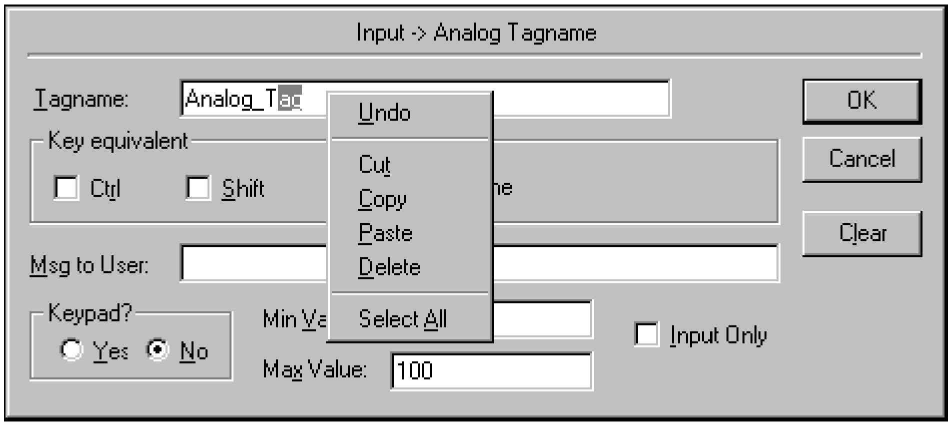
2. Click the command that you want to use on the menu.

   If a "submenu" exists for the command, an arrow will be displayed. To select a command in a submenu, point at the command in the initial right-click menu then, click the command that you want to apply to the object in the submenu.

Note Commands that are not applicable for the current state of the window will not be active.

➢ **To access the dialog box text right-click menu:**

1. In any WindowMaker dialog box, right-click a text box. The following menu appears:

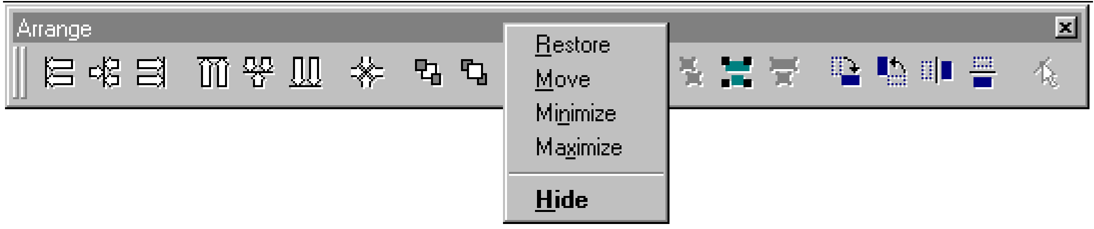


2. Click the command that you want to apply to the selected text.

Note Commands that are not applicable for the current state of the text will not be active. The **Select All** command will become active when partial or no text is selected.

➢ **To access a floating toolbar right-click menu:**

1. Right-click the floating toolbar's title bar. The following menu appears:

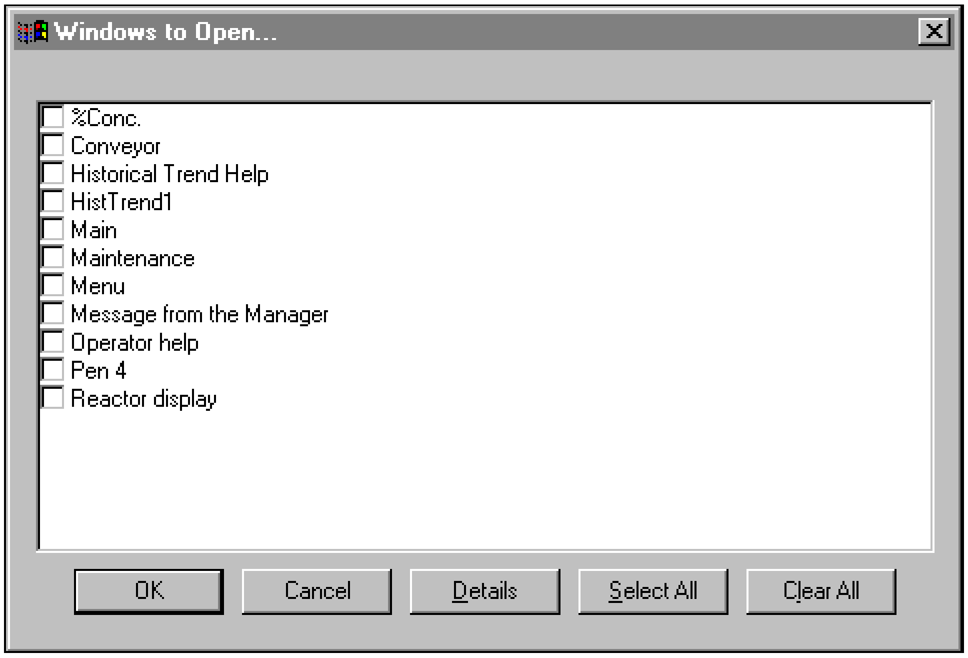


2. Click the command on the menu that you want to apply to the toolbar.

Common Window Dialog Box Features

When you are opening, saving, closing, deleting or duplicating a window(s) using the commands on the **File** menu in WindowMaker, the dialog boxes that you will use are very similar and have many common features. To avoid redundancy in the procedures describing how you perform these actions, the common features of those dialog boxes are described in this section.

When you right-click on a blank area of an open window, or you click the open, save, close, delete or save as window commands on the **File** menu, by default, the respective dialog box for the command you selected will appear in the "list view." Meaning that the names of all the windows that are applicable for the selected command will appear in a continuous list. For example:

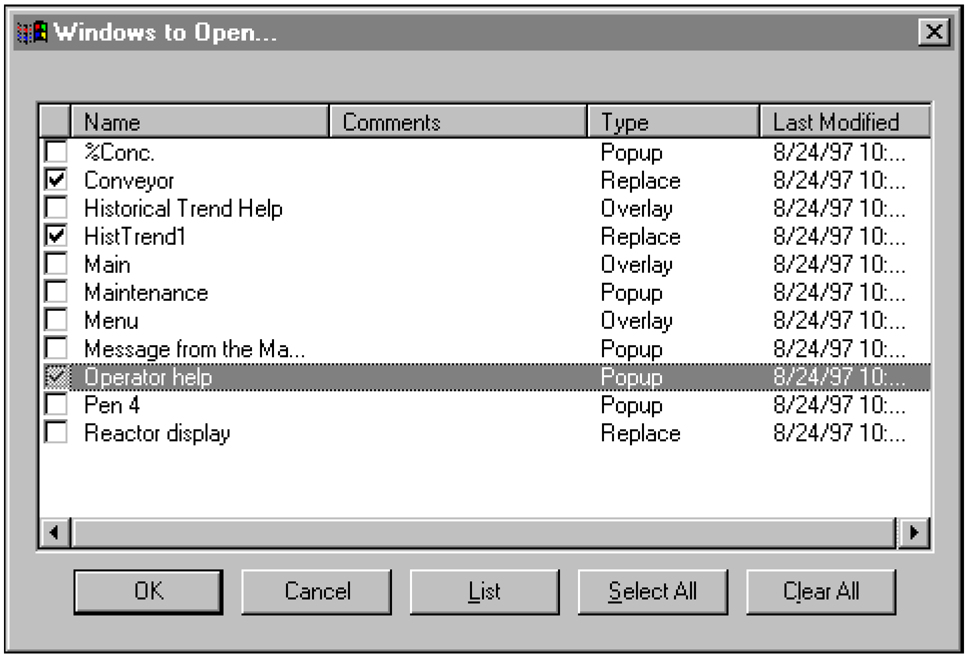

Windows to Open...

- %Conc.
- Conveyor
- Historical Trend Help
- HistTrend1
- Main
- Maintenance
- Menu
- Message from the Manager
- Operator help
- Pen 4
- Reactor display

Note A horizontal scroll bar will appear when the number of window names exceeds the default list space.

Click **Details** to change from the "list view" to the details view.

When you select the details view, the windows and their details are displayed in a multi-column format. The details displayed include any comments you entered for the window in the **Window Properties** dialog box, the window's type, the date and time it was last modified. For example:



	Name	Comments	Type	Last Modified
☐	%Conc.		Popup	8/24/97 10:...
☑	Conveyor		Replace	8/24/97 10:...
☐	Historical Trend Help		Overlay	8/24/97 10:...
☑	HistTrend1		Replace	8/24/97 10:...
☐	Main		Overlay	8/24/97 10:...
☐	Maintenance		Popup	8/24/97 10:...
☐	Menu		Overlay	8/24/97 10:...
☐	Message from the Ma...		Popup	8/24/97 10:...
☑	Operator help		Popup	8/24/97 10:...
☐	Pen 4		Popup	8/24/97 10:...
☐	Reactor display		Replace	8/24/97 10:...

Note In the details view, you can select any unopened window by clicking on any portion of its row, not just the check box. (The entire row will be highlighted.) You can click on a selected window a second time, to deselect it.)

A vertical scroll bar will also appear when the number of window names exceeds the default list space.

To sort the list by a detail type, click the column header for that detail. The details view sort sequences:

- **Name** - Alphabetically
- **Comments** - Alphabetically
- **Type** - Overlay, Replace then Popup
- **Last Modified** - From oldest date/time (top) to most recent (bottom)

Each time you click a column header, the list sort order will toggle from ascending to descending. For example, if the list is currently sorting in ascending order and you click a column header, the list will be resorted in descending order for the column selected.

To return the list to the default display, click the small box on the far left side of the column header.

To size the columns, place the cursor over the vertical lines that separate each detail header. When the cursor changes to an "I" bar, click and drag the header to the width you want for the column.

To quickly auto-size a column, double-click on the column's right vertical line separator.

To open selected window(s) click **OK**.
To cancel your selections and close the dialog box, click **Cancel**.
To return the dialog box to "list view," click **List**.
To select all listed windows, click **Select All**.
To clear all selected windows, click **Clear All**.

Miscellaneous Mouse Short Cuts

Double-clicking on any object or symbol automatically executes the **Animation Links** command (on the **Special** menu) with the object or symbol selected.

For more information on animation links, see Chapter 5 - Animation Links.

Double-clicking in a blank expression input field within a link definition dialog box launches the Tag Browser listing all tagnames defined in the application's Tagname Dictionary.

Double-clicking after a period (**.**) in an expression input field on a link definition dialog box will display the **Choose field name** dialog box containing a global listing of all the tagname **.fields**.

Double-clicking on a tagname in an animation link tagname or expression opens that tagname's definition in the Tagname Dictionary.

Double-clicking a SuperTag parent template name in an animation tagname or expression opens the Tagname Dictionary details dialog box for that SuperTags member tagnames.

For more information on the Tag Browser and tagname **.fields**, see Chapter 4 - Tagname Dictionary.

Right-clicking on a blank area of an open window, a text box in any WindowMaker dialog box or, on a graphic object will display a menu with the commands that you can apply to the window, text or object. Right-clicking the title bar of a floating toolbar will also display a menu of commands that you can apply to the toolbar.

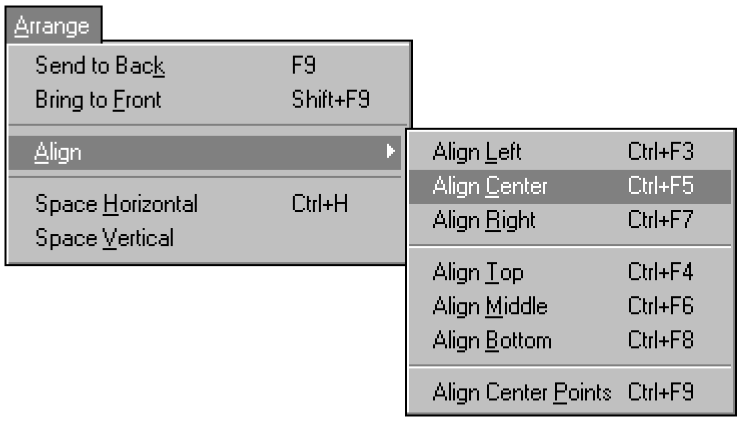
Short Cuts and Accelerators

InTouch provides many mouse and keyboard shortcuts for frequently used functions. Whenever a menu command has a keyboard shortcut, it is displayed on the menu to the right of the command. In addition, all commands may be executed with a three-key sequence beginning with the ALT key. The second key is the underlined character in the menu name, and the third key is the underlined character in the command.

For example, you can execute the **New Window** command on the **File** menu by using the sequence ALT + FN.

To execute a command on a right-click menu, type the underlined character in the command.

To execute a command on a submenu, you must press a three key sequence. For example, to execute the **Align Center** command on the **Align** submenu you would press ALT + AAC, or CTRL+F5



Note When you select a menu command that is followed by an ellipsis (**...**), a dialog box appears, and you must select or enter more information in order to complete the command.

Moving Objects with the Arrow Keys

In WindowMaker, you can use the up, down, left and right arrow keys to move a selected object or group of objects one pixel at a time in the direction of the arrow key being pressed. This feature can be very convenient when you need to make fine alignment and location adjustments. To move the object 10 pixels at a time, hold down the SHIFT key while pressing the arrow key. To move the object 50 pixels at a time, hold down the CTRL key while pressing the arrow key.

In WindowViewer, when you use the arrow keys, an algorithm is used to move from one touch-sensitive object to another. For example, the left cursor arrow key cuts a path to the left as wide as the height of the selected object. If it intersects any part of another touch-sensitive object within its path, that object takes the focus as the currently selected object. If no other touch sensitive object is encountered, the path wraps around to the left edge of the screen and continues to try to intersect another object. If no other object is encountered, the original object remains selected.

All arrow keys function in the same manner. The up and down arrow keys use the width of the selected object for the width of their paths.

If you know that the user of your application will only use cursor keys to navigate in the windows in your application, you should carefully plan the placement of the touch-sensitive objects in the window to ensure that they have intersecting paths.

In WindowViewer, the TAB key can also be used to transition between the touch-sensitive objects in a window. (However, the tabbing order cannot be guaranteed.)

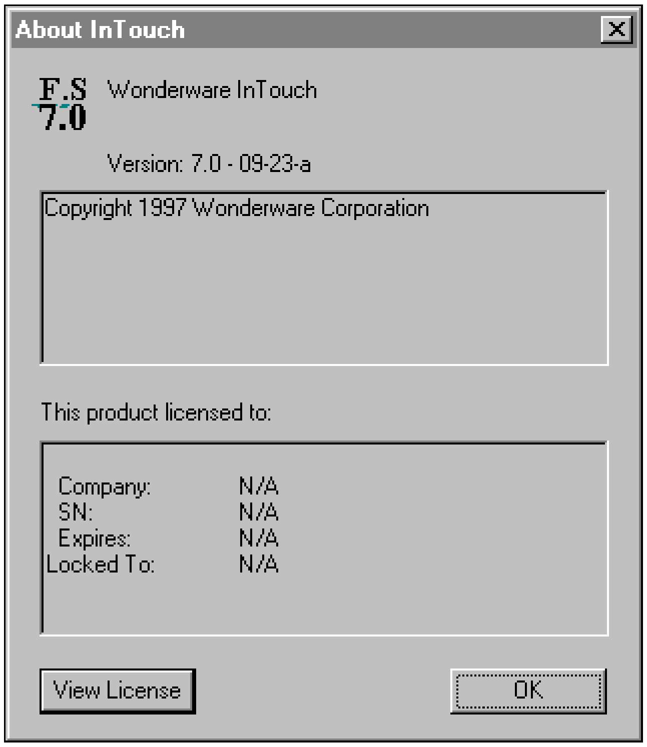
By careful application design, and understanding how the arrow keys react, you can create applications that can be used without a mouse.

Using WindowMaker Help

WindowMaker supports context-sensitive help as follows:

- For help on an open dialog box, press the F1 key. The respective Windows online help topic will appear.
- For help on the various QuickScript functions, in the QuickScript editor, click **Help** or, on the **Insert** menu, point to **Functions** then, click **Help**. A dialog box will appear listing all available QuickScript functions. Click the function that you want help for. The function's respective Windows online help topic will appear.

- To obtain information about your InTouch software, for example, the Version you are using, your Serial Number and your License expiration date and so on, on the WindowMaker **Help** menu, click **About**. The **About InTouch** dialog box appears:



Wonderware InTouch

Version: 7.0 - 09-23-a

Copyright 1997 Wonderware Corporation

This product licensed to:

Company:	N/A
SN:	N/A
Expires:	N/A
Locked To:	N/A

Click **View License** to launch the license viewing utility to obtain information regarding your FactorySuite license.

🕮 For more information on licensing, see your *FactorySuite System Administrator's Guide*.

CHAPTER 2

Using WindowMaker

By setting various properties for WindowMaker and WindowViewer, you can customize the functionality and final appearance of your application. For example, you can specify what menus you want available in WindowViewer, you can include company names in the title bar of your application, and so on.

This chapter describes how to configure WindowMaker and WindowViewer, work with WindowMaker's windows, edit and arrange graphic objects, and how to install and use wizards and ActiveX controls.

Contents

Simple Objects

WindowMaker has four basic types of simple objects; lines, filled shapes, text and buttons. Each of these simple object types have attributes that affect their appearance. These attributes include line color, fill color, height, width, orientation, etc. and can be either static or dynamic. A static attribute remains unchanged during the operation of the application. A dynamic attribute is linked to the value of an expression such that a change in the value of the expression results in a change in the attribute. For example, the fill color of an object can be linked to the value of a discrete expression. Based on the state of the expression, the fill would be one color when the expression is true and another color when it is false. Most of the attributes of simple objects can be made dynamic. An object may have more than one dynamic attribute. Dynamic attributes may be combined freely to achieve the desired result. The following briefly describes WindowMaker's simple object types:

Object	Description
Line	A line object is made up of one or more line segments depending on the type of line. Color is the only attribute of a line that you can animate. Width and style cannot be animated. They are simply assigned default attributes.
Filled Shapes	Filled shapes are two dimensional objects made up of a closed interior area surrounded by a line. Examples of filled shapes are rectangles, rounded rectangles, circles, ellipses and polygons. The attributes of a filled shape are: line color, line width, line style, fill color, percent color fill, height, width, location, visibility, orientation and size.
Text	Text is an object made up of a string of characters on a single line. The attributes of a text object are: font, size, color, **bold**, <u>underline</u>, *italic,* justification, visibility and location.
Buttons	The 3-dimensional buttons can be created for any desired size by using the Button tool on the WindowMaker **Draw Object Toolbar**. The default "text" string on the button face is edited by using the **Substitute Strings** command on the **Special** menu. Many types of links can be attached to buttons such as action scripts, key scripts, analog or discrete value input or output links. If an input or output link is attached to a button the value is displayed on the button as the text string.

Complex Objects

In addition to simple objects, InTouch also supports complex objects which are considerably different. The following briefly describes WindowMaker's complex object types:

Object	Description
Bitmap	The Bitmap tool is used to copy and paste bitmaps into your application. Once pasted in an application window, a bitmap can be rotated and, it can be defined with a transparent background so that it can float over other objects.
Trends	There are two trend tools: one is for creating trends that display real-time data and the second is for creating trends that display historical data (retrieved from historical log files). Both the real-time and the historical trends can be configured to display graphical representations of multiple tagnames over time.
Symbols	A symbol is a combination of simple objects (lines, filled shapes, and text) which is treated as a single object. Any attribute change applied to a symbol, whether it is a change in a static attribute in WindowMaker, or a change in a dynamic attribute in WindowViewer, will affect all the component objects of a symbol. For example, if you create a pump symbol from two circles and two rectangles, and then you define a fill Color Link on the symbol, the fill color will apply to all four objects. Similarly, in WindowMaker, a Fill Color selection would also change the fill color of all the component objects. The component objects of a symbol can have different values for the same attribute if these attributes were different before the objects were combined into the symbol and were not changed after they became a symbol. Symbols cannot contain bitmaps, buttons, cells, alarms or trends.
Cells	A cell is a collection of two or more objects, symbols, or other cells that are joined together to form a single unit. Cells maintain a fixed spatial relationship between their individual graphic elements. Each component of a cell can have its own links. Cells are used to create virtual devices such as a slide controller. A cell is created by selecting two or more objects, symbols, and/or cells, and then executing the **Make Cell** command on the **Arrange** menu. Once a set of objects is made into a cell, its interior details such as colors, sizes, animation links, and so forth, cannot be changed. The only way to change a cell's appearance or operation is by "breaking" it apart using the **Break Cell** command on the **Arrange** menu. The attributes of the components of a cell are changed in WindowViewer by the operation of links. Cells can be duplicated, copied, pasted, aligned, spaced, and so on. Cells cannot be sized. They must be broken apart, sized and made back into a cell. Cells are very useful in creating multiple similar devices that are associated with different tagnames.

Wizards

Wizards save you a considerable amount of time during application development. They are easy to use and easy to configure. When you select a wizard and paste it into a window, and then double-click it, its configuration dialog box appears and you can configure it.

For example, in the case of a "slider" wizard, the configuration would include items such as the tagname to effect, the minimum and maximum range labels for the slider, the fill color, and so on. Once the required configuration information is entered, the Wizard is ready to use. By using Wizards, you do not spend time drawing the individual components for the object, or entering the value ranges for the object, or animating the object.

"Complex" Wizards can be developed to provide "behind the scenes" types of operations. These operations may include creating complete display windows, creating or converting a database, importing an AutoCad drawing, and configuring other applications such as, the InTouch Recipe Manager and SPC add-on programs.

Note A proficient "C" programmer can develop custom Wizards by using Wonderware's Extensibility Toolkit. This toolkit is available through your distributor.

☞ For more information on Wizards, see "Working with Wizards."

ActiveX Controls

WindowMaker supports ActiveX controls which, in their simplest form, are mini-applications that talk to or run within your application. WindowMaker supports all ActiveX controls that are included in Wonderware FactorySuite components, for example, all InTrack ActiveX controls. WindowMaker also supports third-party ActiveX controls such as those installed with Office97.

You install the InTouch ActiveX controls just like any other wizard. If desired, you can add frequently used ActiveX controls to your WindowMaker **Wizards/ActiveX Toolbar**.

When you select an ActiveX control and paste it into a window, and then double-click it, its configuration dialog box appears. When you configure an ActiveX control, you specify a unique control name for it in order to reference it in a script (a default name is created when you initially add the control).

All ActiveX controls have properties, methods and events associated with them. You can associate an ActiveX property with a tagname of a corresponding data type. You can execute ActiveX methods through InTouch QuickScript functions. You can associate an ActiveX event with an ActiveX Event Script that executes when the event occurs. In other words, you can use InTouch QuickScript functions to handle control events, call control methods, and control properties.

In runtime, the tagnames and QuickScripts you defined in WindowMaker control the behavior of your ActiveX controls.

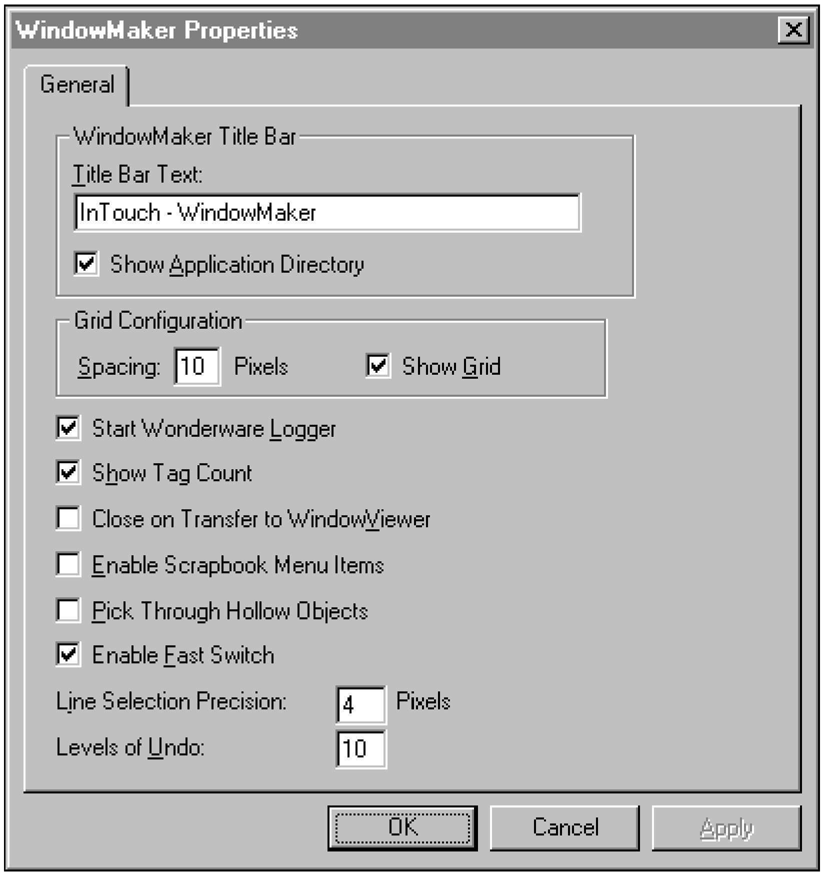
Customizing Your Development Environment

There are many properties that you can set to customize WindowMaker. For example, you can customize your application's title bar text to include the company name. You can set the pixel spacing for the grid and so on.

➢ **To set the properties for WindowMaker:**

1. On the **Special** menu, point to **Configure**, and then click **WindowMaker**, or in the Application Explorer under **Configure**, double-click **WindowMaker**. The **WindowMaker Properties** dialog box appears with the **General** properties sheet active:

   In the Application Explorer under **Configure**, you can also right-click **WindowMaker**, and then click **Open**.



WindowMaker Properties

General

WindowMaker Title Bar

Title Bar Text: InTouch - WindowMaker

[✓] Show Application Directory

Grid Configuration

Spacing: 10 Pixels [✓] Show Grid

[✓] Start Wonderware Logger

[✓] Show Tag Count

[ ] Close on Transfer to WindowViewer

[ ] Enable Scrapbook Menu Items

[ ] Pick Through Hollow Objects

[✓] Enable Fast Switch

Line Selection Precision: 4 Pixels

Levels of Undo: 10

   If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Title Bar Text** box, type the title that you want to appear in your application's title bar in runtime. For example:

   ABC Company, Paint APP1

Note You cannot change the title bar if you are using a "Promotional License."

   For more information on FactorySuite licensing, see your *FactorySuite System Administrator's Guide*.

3. **Show Application Directory** if you want to include the path to the application's directory in the title bar. For example:

   ABC Company, Paint APP1 - C:\DEMOAPP1

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

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

   If you do not select **Show Grid**, no grid will be visible in your windows when you turn snap to grid on.

   Click the Snap to Grid tool on the **View** toolbar or, on the **Arrange** menu, click **Snap to Grid** to turn the snap to grid functionality on and off.

   Objects are snapped to the grid by their upper left corner. If you select multiple objects, the snapping will be applied to the upper left corner of the first object selected in the group.

6. Select **Start Wonderware Logger**, if you want the FactorySuite Wonderware Logger application to automatically start when you start WindowMaker.

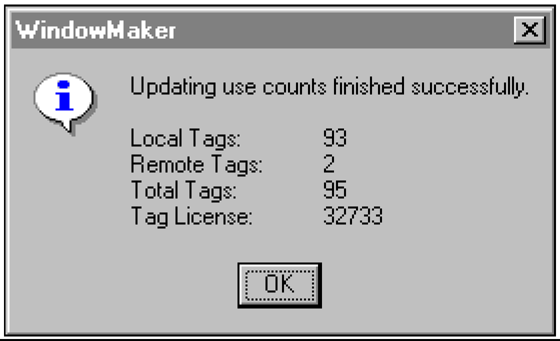
   The Wonderware Logger's behavior is a little different when you are on the Windows NT Operating System. For more information on the Wonderware Logger, see the "Welcome to InTouch" section.

7. Select **Show Tag Count**, if you want the number of tagnames defined in your Tagname Dictionary to be displayed in the WindowMaker menu bar.

   This can be very useful information if you are creating an application with a limited Tagname Dictionary size.

   Note Showing the tagname count will seriously impact performance of the Tagname Dictionary in WindowMaker.

   The displayed tagname count does not include remote tagname references. To determine your remote tagname reference usage, execute the **Update Use Counts** command on the **Special** menu. Once the system has completed updating the use counts, the following dialog box appears:



Updating use counts finished successfully.

```
Local Tags:      93
Remote Tags:     2
Total Tags:      95
Tag License:     32733
```

8. Select **Close on Transfer to WindowViewer**, if you want WindowMaker to automatically close when you start WindowViewer.

   If memory is not an issue and you are moving often between WindowViewer and WindowMaker, this option should not be selected.

Note When you select this option, the **Close WindowViewer** option located on the **WindowViewer Properties General Properties** sheet is automatically selected too.

9. Select **Enable Scrapbook Menu Items** if you are copying and pasting objects between WindowMaker and Scrapbook+.

Note The Windows Scrapbook+ program must be installed on your computer in order to use these menu commands. This option is used by legacy InTouch systems. It is not supported for FactorySuite nor is Scrapbook+ sold through Wonderware.

10. Select **Pick Through Hollow Objects**, if you want to be able to select objects that are behind "hollow" objects.

    🖰 If you select this option, and then draw four lines and join them to make a frame around another object, you can select the object within the frame without having to send the frame to the back.

11. Select **Enable Fast Switch**, if you want the fast switch between WindowMaker and WindowViewer to be displayed in your menu bar for both programs.

    🖰 If you select this option, in WindowMaker, the fast switch is the word **Runtime** displayed in the upper right hand of your screen. In WindowViewer, it is the word **Development**. To quickly switch between the two programs, click the fast switch.

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

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

    🖰 In most cases, the default setting of 4 should be sufficient.

13. In the **Levels of Undo** box, type the number of undo/redo levels that you want saved. You can have up to 25 levels. If you type zero, the undo/redo functionality is turned off.

    One level represents one action. The undo and redo stacks are empty when you create a new window or open an existing window. Both stacks are emptied when you save the window.

    🖰 For more information on undo and redo, see "Undoing Object Edits."

14. Click **OK**.

Note After you modify any of these settings, you must restart WindowMaker to apply your changes.

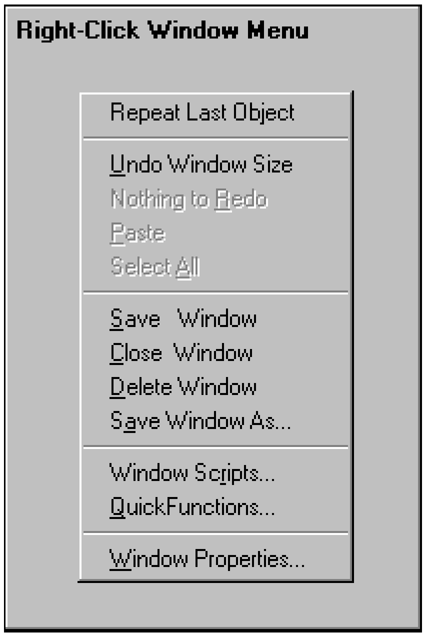
Working with WindowMaker Windows

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.

This section contains the procedures that you will follow to create, open, save, close, delete, and duplicate windows.

The **General Toolbar** contains tools that you can use to quickly apply most of the windows commands on the **File** menu.

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. For example:

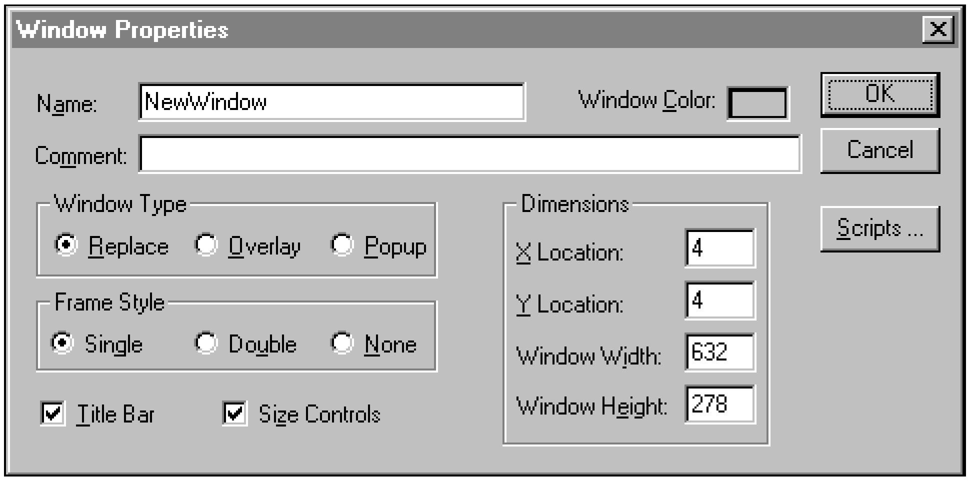


Creating a New Window

➢ **To create a new window:**

1. On the **File** menu, click **New Window** or, click the New Window tool on the **General Toolbar**. The **Window Properties** dialog box appears:

   To quickly create a new window, in the Application Explorer, right-click **Windows**, and then click **New**.


Window Properties

Name: NewWindow
Window Color:
Comment:
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

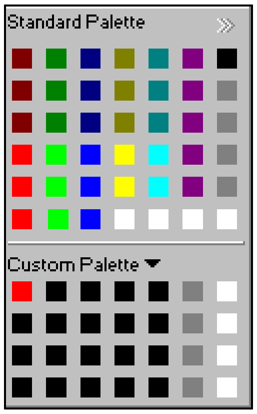
Note By default, the settings in this dialog box will reflect those of the previously created window or, if you select this command while a window is open in WindowMaker, the settings will reflect those of the active window. If a Window script(s) is attached to the active window, a message box will appear asking you if you want the window script(s) copied to the new window.

   If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Name** box, type the name that you want 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 type 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.

   For more information on using the color palette, see Chapter 1 - WindowMaker Program Elements.

5. Click the color that you want to use for the window's background.

6. Select the **Window Type** that you want to use. There are three window types:

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 except, it 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.

Note You can change a window's type by opening its **Window Properties** dialog box again. There are three ways to do this:

a) Open the window, and then on the **Windows** menu, click **Window Properties**. The **Window Properties** dialog box appears.

b) In the Application Explorer under **Windows**, right-click the window name, and then click **Properties**. The **Window Properties** dialog box appears. If the window is not open when you execute the command, it is automatically opened behind the dialog box.

c) Open the window, right-click a blank area of the window, and then click **Window Properties**. The **Window Properties** dialog box appears.

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** if you want the window to have a title bar. The title bar is also used to move the window by clicking the mouse on it, and then dragging the mouse.

   🖰 If your window has a title bar, you cannot select **Double** or **None** for the **Frame Style**.

9. Select **Size Controls** if you want the user to be able to resize the window in WindowViewer.

10. In the **Dimensions** group, type the pixel location that you want 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.
	🖱 Windows limits the minimum width of a window according to your display monitor. For example, for the standard VGA, the minimum is 102 pixels.
Window Height	The window's height in pixels.
	🖱 Windows limits the minimum height of a window according to your display monitor. For example, for standard VGA, the minimum is 26 pixels.

Note By default, the values in these boxes will be set to the dimensions of the previously created window. They are also automatically modified if you manually change the windows size in WindowMaker by using the window's border.

11. Click **Scripts** to access the **Window Script** editor. There are three types of scripts that you can apply 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.

Note If you attach a Window Script to the active window and then you create a new window, the script(s) from the active window can be copied to the new window. A message dialog box will appear asking if you want to copy the window script(s).

🖱 If you later decide that you need to attach a script to an open window, right-click a blank area of the open window, and then click **Window Scripts**. If the window is not open, in the Application Explorer, double-click **Windows** to show all window names. Right-click the window name, and then click **Window Scripts**.

👓 For more information on creating window scripts, see Chapter 6 - Creating QuickScripts in InTouch.

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 19 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 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 an 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 the them in WindowViewer.

You can use this technique with a popup window to hide cover the title bar and menu bar in WindowViewer. You can also create a touch link pushbutton or QuickScript to hide this popup window whenever you need to reveal the title bar and menu bar 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 which includes the ability to exit WindowViewer.

Note A Promotional InTouch license does not allow the title bar to be hidden. Therefore, if an application is developed under a Promotional InTouch license and WindowViewer is configured with the **Hide Title Bar** option selected when that application is viewed on a standard runtime license, the title bar will be hidden as configured and, all the windows in the application to move up.

Opening Windows

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

➢ **To open a window(s):**

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

   To quickly open a single window, in the Application Explorer, double-click **Windows** to open the list of all the window names in your application, and then double-click the window name. You can also right-click the window name, and then click **Open**.

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

   By default, all currently opened windows will already be checked.

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 you can close it or exit your application. All graphics, QuickScripts, properties, and so on associated with the window are also saved.

➢ **To save a window(s):**

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

   To quickly save a single window, in the Application Explorer, right-click the window name, and then click **Save**. You can also right-click any blank area of the window, and then click **Save Window**.

   To quickly save all currently open windows, click the Save All Windows tool on the **General Toolbar**, or the **Save All** command on the **File** menu.

2. Click the check box next to the name of window(s) that you want 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(s) that has been modified since it was last saved, you will be prompted to save your changes before WindowMaker will close the window.

➢ **To close a window(s):**

1. On the **File** menu, click **Close Window** or, click the Close Window tool on the **General Toolbar**. The **Windows to Close** dialog box appears listing the names of all currently open windows.

   To quickly close a single window, in the Application Explorer, right-click the window name, and then click **Close**. You can also right-click any blank area of the window, and then click **Close Window**.

2. Click the check box next to the name of the window(s) that you want 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.

➢ **To delete a window(s):**

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

   To quickly delete a single window, in the Application Explorer, right-click the window name, and then click **Delete**. You can also right-click any blank area of the window, and then click **Delete Window**.

2. Click the check box next to the name of the window(s) that you want to delete.
3. Click **OK** to close the dialog box and delete the selected window(s).

Duplicating a Window

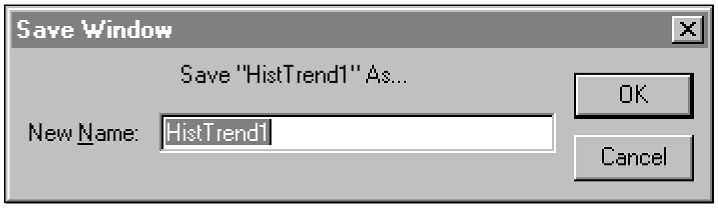
When you want to create a duplicate copy of an existing window, the window that you want to duplicate must be open.

➢ **To duplicate a window:**

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

   To quickly duplicate a window, in the Application Explorer, right-click the window name, and then click **Save As**. You can also right-click in any blank area of the window, and then click **Save Window As**.

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, type a valid name for the new window.
4. Click **OK** to close the dialog box and create the duplicate window.

Exporting Windows

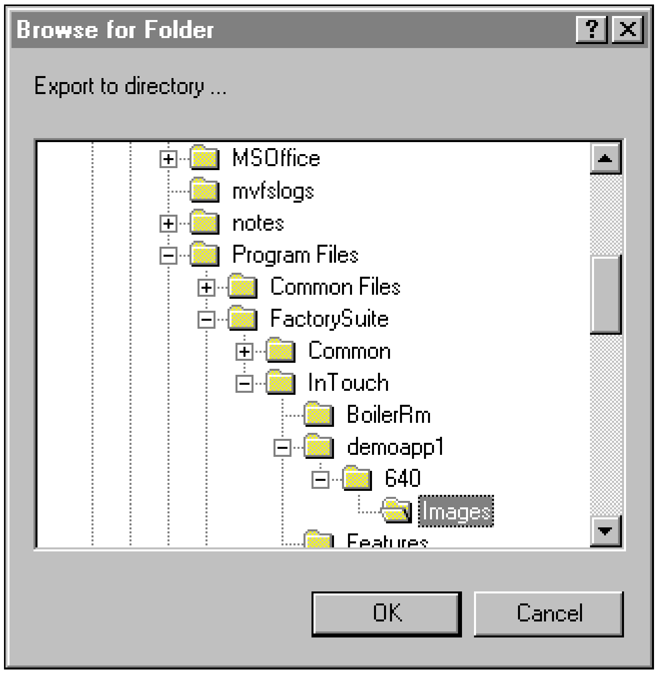
Exporting windows is very useful when you need to create or maintain a library application or you need to quickly create remote tagname references in another application. To move windows from one InTouch application to another, you must use the **Export Window** command on the **File** menu.

For more information on remote tagname references, see Chapter 4 - Tagname Dictionary.

Note If you attempt to copy InTouch window files by using any other copy methods, such as the File Manager or Windows Explorer copy commands, you may corrupt your application's Tagname Dictionary!

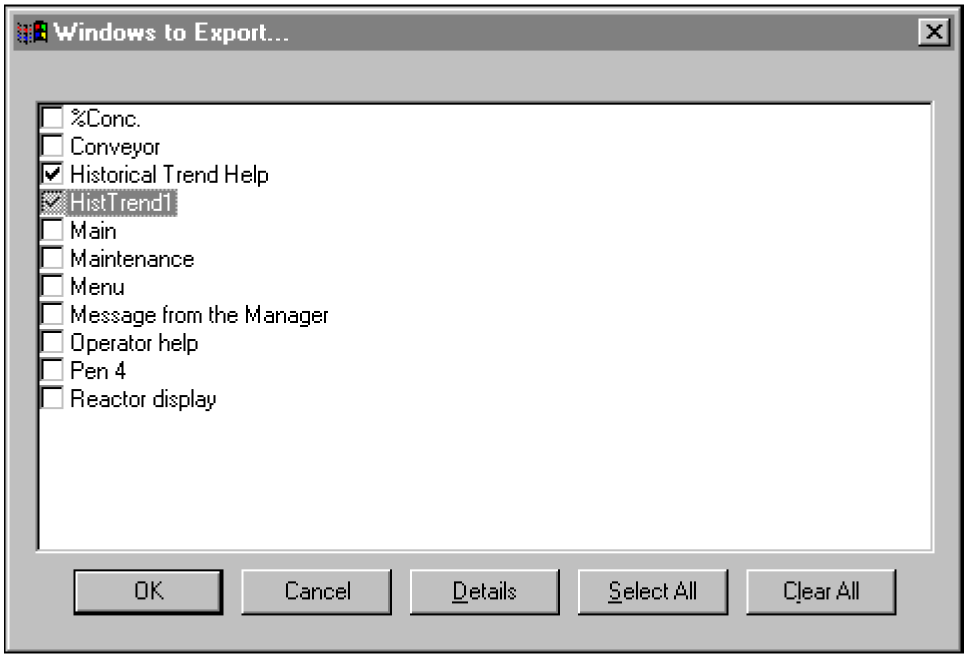
➢ **To export a window:**

1. Close all windows in your current application.
2. On the **File** menu, click **Export Window**. The **Browse for Folder** dialog box appears:



3. Locate and select the application directory (folder) that you want to export the window(s).
   - Click **Yes** if you want to export the window.
   - Click **No** if you want to export the window(s) to another InTouch application.

4. The **Windows to Export** dialog box appears:


Windows to Export...

- [ ] %Conc.
- [ ] Conveyor
- [x] Historical Trend Help
- [x] HistTrend1
- [ ] Main
- [ ] Maintenance
- [ ] Menu
- [ ] Message from the Manager
- [ ] Operator help
- [ ] Pen 4
- [ ] Reactor display

5. Select the window(s) that you want to export.

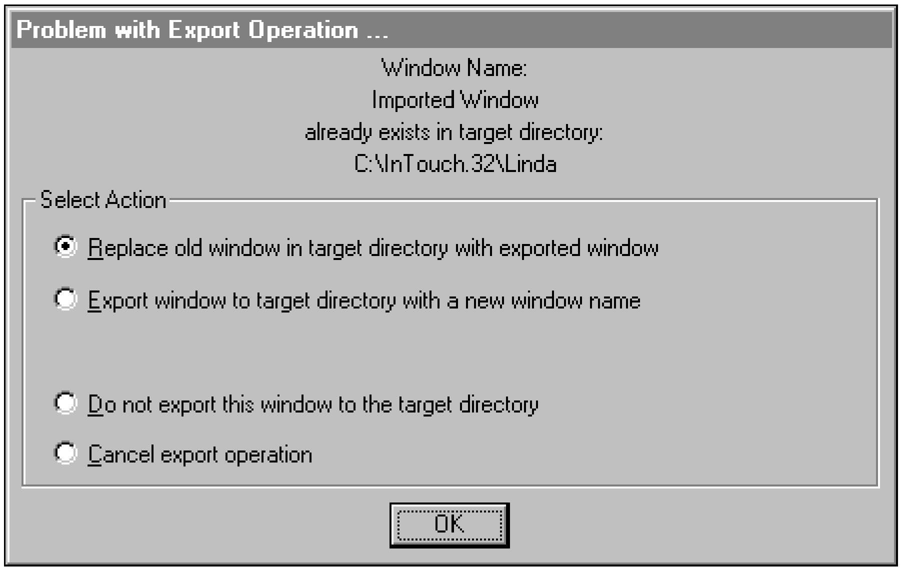
6. Click **OK**. The export operation will take place.

Note When you export a window, all of the objects and animation links associated with that window are exported with the window. However, the tagnames associated with the objects in the window are converted into "placeholder" tagnames. Placeholder tagnames are used to avoid any problems that might occur when the destination application's Tagname Dictionary does not contain the same tagnames.

For more information on converting placeholder tagnames, see Chapter 4 - Tagname Dictionary.

Problem with Export Operation

If a problem is encountered by the system when exporting the window, the **Problem with Export Operation** dialog box appears:


Problem with Export Operation ...

Window Name:
Imported Window
already exists in target directory:
C:\InTouch.32\Linda

Select Action

[✓ Replace old window in target directory with exported window]
Export window to target directory with a new window name
Do not export this window to the target directory
Cancel export operation

In the **Select Action** group, select the action that you want to take, and then click **OK**.

Importing Windows

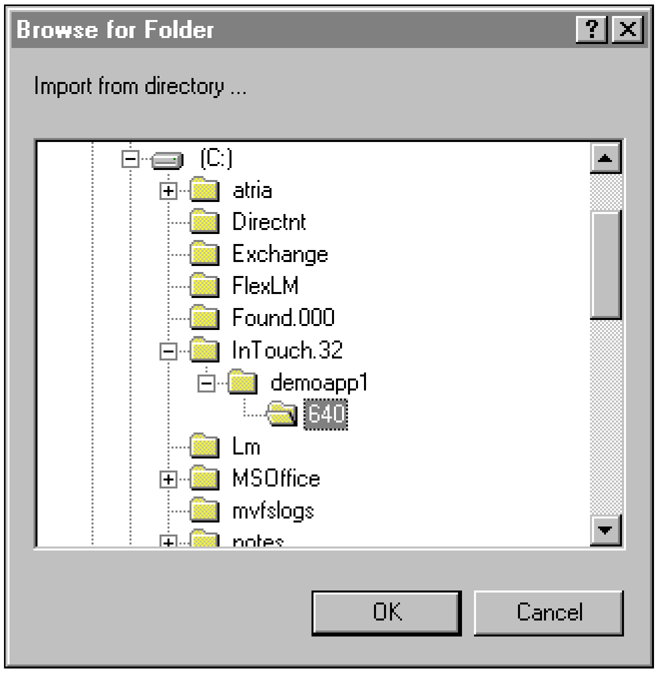
Importing windows from one InTouch application to your current application, can save you a considerable amount of development time. It also provides you with a quick and easy method for creating remote tagname references. It allows you to reuse your previously created windows, objects and window scripts.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

For more information on remote tagname references, see Chapter 4 - Tagname Dictionary.

Note If you attempt to move InTouch window files by using any other move methods, such as the File Manager or Windows Explorer move commands, you may corrupt your application's Tagname Dictionary!

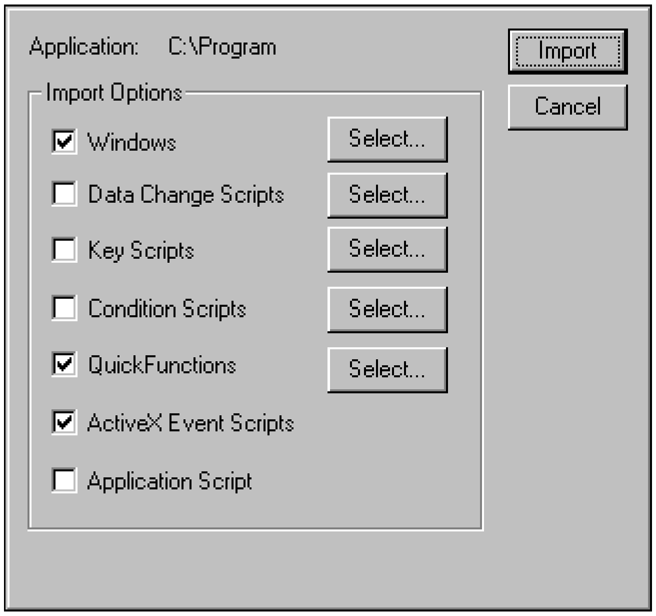
➢ **To import a window or QuickScript:**

1. Close all windows in your current application
2. On the **File** menu, click **Import**. The **Browse for Folder** dialog box appears:



3. Locate and select the application directory (folder) containing the windows that you want to import.

4. Click **OK**. The following dialog box appears:


Application: C:\Program

Import Options

- [✓] Windows
- [ ] Data Change Scripts
- [ ] Key Scripts
- [ ] Condition Scripts
- [✓] QuickFunctions
- [✓] ActiveX Event Scripts
- [ ] Application Script

5. Select the item(s) that you want to import, and then click **Select**. A dialog box will appear for you to select the window or QuickScript(s) that you want to import.

6. Once you have selected the window(s) or QuickScript(s) that you want to import, click **Import**. The system will automatically begin to import the selected items into your current application.

Note To import a window script, you must import the entire window. When you import a window, all of the objects and links associated with that window are imported with the window. However, the tagnames associated with the objects in the window (and the tagnames used in an imported script) are converted into "placeholder" tagnames.

When the tagnames in an imported script or window 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 not 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 only for placeholder tagnames. This increase in tagname length is not supported for standard tagnames.

For more information on converting placeholder tagnames, see Chapter 4 - Tagname Dictionary.

Note When you import a window from an application that contains SuperTags, only the SuperTag instances actually used in the imported window are imported into the new application. The entire SuperTag template structure is not imported. For example, if the application has hundreds of SuperTag member tagnames defined in it, and only 50 of those are used in the imported window, only those 50 are imported.

Working with Graphic Objects

Once you have created a new window in your application, you are ready to populate it with graphic objects. WindowMaker provides you with numerous tools for editing and arranging the various graphic objects that you will draw and paste into your windows. This section describes the procedures you will use to perform various editing functions on your graphic objects.

The **Draw Object Toolbar** contains the graphic drawing tools that you will use to create graphics in your windows.

The **View Toolbar** contains the Ruler tool that you can use to display a ruler for assisting you in alignment of your graphic objects in your windows. This toolbar also contains the tools that you will use to hide or show the Application Explorer, the toolbars, or a visible grid in your windows. It also contains the tool for switching to and from full screen mode.

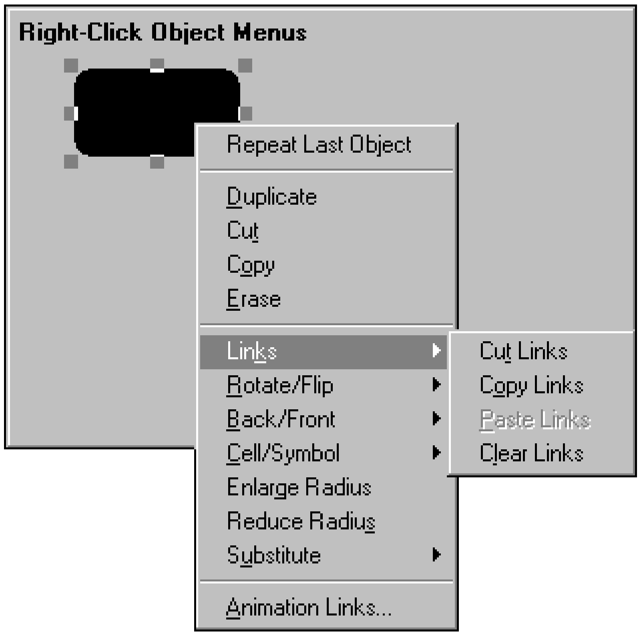
The **General Toolbar** and the **Format Toolbar** contain the tools that you can click to quickly apply many of the commands on the **Edit** menu and the **Text** menu to your graphic objects.

The **Arrange Toolbar** contains the tools that you can click to quickly apply the alignment commands on the **Arrange** menu.

You can also customize the **Wizards/ActiveX Toolbar** by adding any wizards or ActiveX controls that you repeatedly use.

For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

If you right-click a object, a menu will appear displaying the valid commands or actions that you can apply to the selected object. For example:



Note The commands on the right-click menus will vary. They are based upon the type of object that is selected.

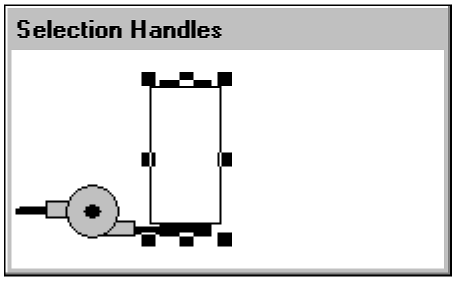
Selecting and Sizing Objects

After you draw an object, and you click it, several little boxes will surround it. These boxes are called "handles." You use these handles to resize and/or reshape the object.

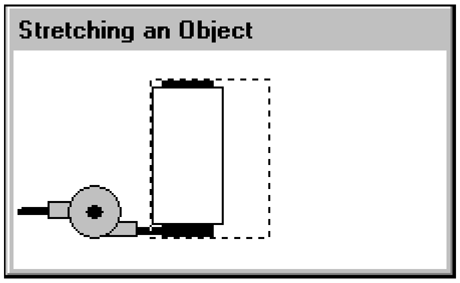
The notion of "selected" is a key concept of WindowMaker graphics editing. The presence of "handles" around an object indicates that it is "selected." Clicking directly on an object selects it. Clicking on a blank area of the window deselects any currently selected object(s) in that window. In general, any command that you execute is applied to all selected objects, if the command is valid for the object.

➢ **To change the size of an object:**

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

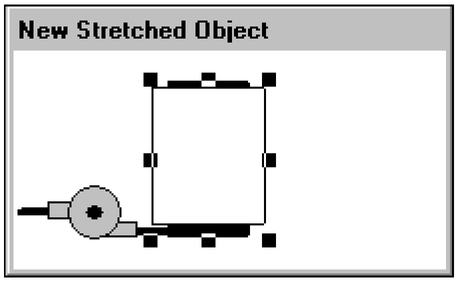


2. Press and hold down the left mouse button while you drag the handle to either stretch or shrink the object:



You can stretch/shrink the object in either direction, depending upon which handle is selected. If you use one of the four corner handles to size the object, it will be sized vertically and horizontally simultaneously.

3. Release the mouse button. The object will be redrawn at its new size:



If the size is not correct, on the **Edit** menu, click **Undo** or, click the Undo tool on the **General Toolbar** and try again.

For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

For more information on undoing and redoing edits, see "Undoing Object Edits."

Selecting all Objects in a Window

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

Selecting Multiple Objects

To select multiple objects, select your first object, then hold down the SHIFT key, while you click the other objects that you want to select. 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 that you want to deselect.

Selecting a Group of Objects

Move the cursor to a blank area of your window. Depress the 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 the objects that were completely within the rectangle will be selected.

Deselecting a Group of Selected Objects

If you hold the SHIFT key down while you draw 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 the SHIFT key down while you click 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.

Undoing Object Edits

WindowMaker keeps track of the editing and formatting changes that you make. You can configure WindowMaker to support up to 25 undo/redo levels. You can also disable the undo/redo functionality by setting the level to zero (0). By default, WindowMaker is set to support 10 levels where, one level represents one action.

Note The undo and redo stacks are empty when you create a new window or open an existing window. Both stacks are also cleared when you save the window.

For more information on configuring the undo/redo levels, see "Customizing Your Development Environment."

The **Undo** and **Redo** commands are located on the **Edit** menu. These commands can also be accessed by right-clicking the object. The commands are dynamic and will change to reflect the last action to which they can be applied. For example, if you move an object then decide that you want to return it to its original location in the window, right-click a blank area of the window and click **Undo Move** or, on the **Edit** menu click **Undo Move**.

The Undo and Redo tools are located on the **General Toolbar**.

You will use undo and redo to reverse actions or commands that you have applied to an object. You can undo or redo the following actions or commands:

Action/Command	Supported
Basic	Create, Select, De-select, Move and Re-size Object Line, Fill, Text and Window Color Window Move and Window Size
Editing	Duplicate, Cut, Copy, Paste, Delete Paste Bitmap and Adjust Bitmap - Original size Select All, Link Cut, Link Copy, Link Paste, Link Clear Enlarge Radius, Reduce Radius Reshape Object, Add Point, Delete Point
Arranging	Send to Back, Bring to Front, Align (all commands) Space Horizontal, Space Vertical, Rotate CW, Rotate CCW Flip Horizontal, Flip Vertical Make Symbol, Break Symbol Make Cell, Break Cell
Text	All operations (size, style, pitch, justification)
Line	All operations (width and styles)
Special	Animation Links (double-click on object), Substitute Strings

Duplicating Objects

➢ **To redraw the previously drawn object:**

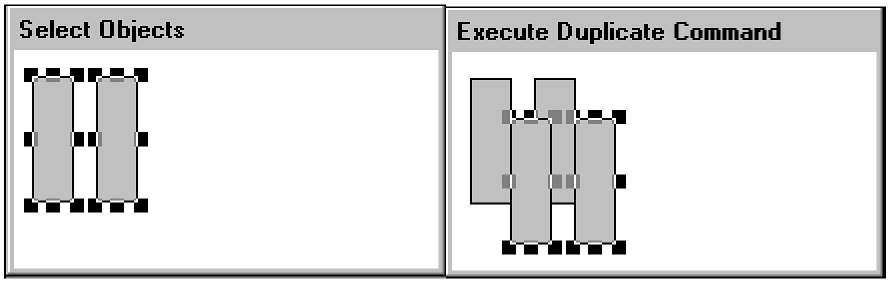
1. Draw the object.
2. Right-click the object, and then click **Repeat Last Object**.
3. Click the left mouse button, and then draw the same object again.

➢ **To duplicate an object:**

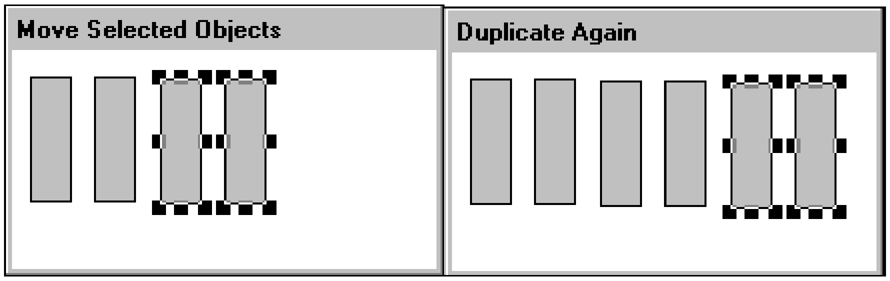
1. Select the object(s) you want to duplicate.
2. On the **Edit** menu, click **Duplicate** or, click the Duplicate Selection(s) tool on the **General Toolbar**.

   For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

   To quickly duplicate an object(s), right-click the object, and then click **Duplicate**.



   If you move the duplicated object(s) without deselecting it, and you duplicate it again, the second (and all subsequent duplications) will automatically be offset the same distance that the first duplicate was moved. For example:



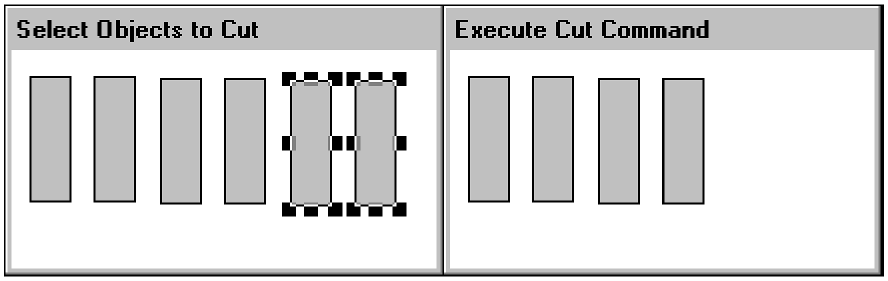
   You can repeat this procedure as many times as necessary.

Cutting Objects to the Windows Clipboard

➢ **To cut an object:**

1. Select the object(s) you want to cut.
2. On the **Edit** menu, click **Cut** or, click the Cut to Clipboard tool on the **General Toolbar**.

   To quickly cut an object(s), right-click the object, and then click **Cut**.



Note When you cut an object, it is erased from your window and copied to the Windows Clipboard. The object's attributes and animation links are also copied with it.

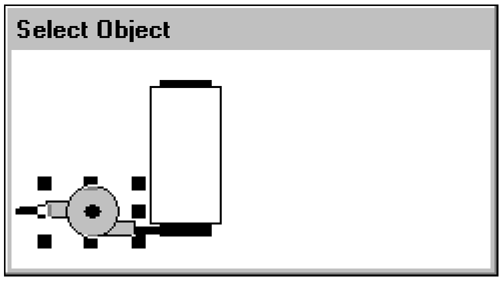
Copying Objects to the Windows Clipboard

➢ **To copy an object:**

1. Select the object(s) you want to copy.
2. On the **Edit** menu click **Copy** or, click the Copy to Clipboard tool on the **General Toolbar**.

   For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

   To quickly copy an object(s), right-click the object, and then click **Copy**.

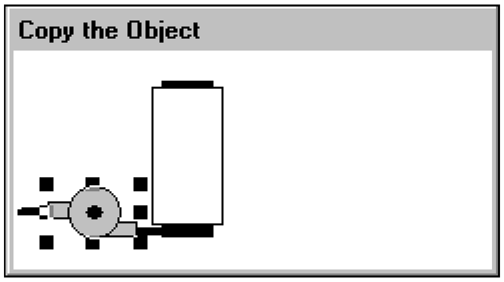


Note When you copy an object, it is not erased from your window. It is copied to the Windows Clipboard. The object's attributes and animation links are also copied with it.

Pasting Objects from the Windows Clipboard

➢ **To paste an object from the Windows Clipboard:**

1. Copy or cut the object:

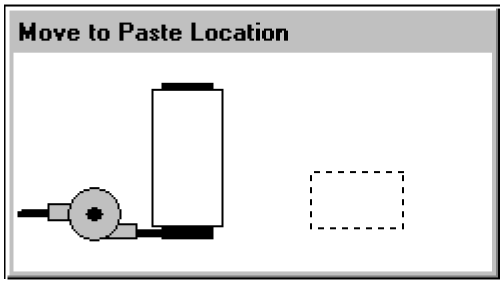


2. On the **Edit** menu, click **Paste** or, click the Paste from Clipboard tool on the **General Toolbar**.

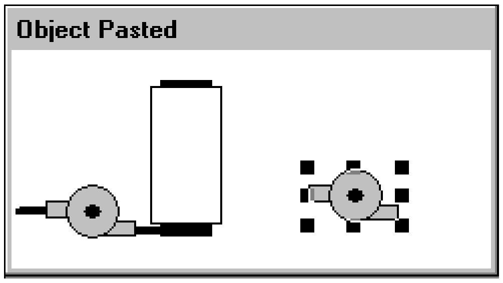
   To quickly paste the object, right-click the object, and then click **Paste**.

3. The cursor will change to a corner symbol.

4. 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 that you want to paste the object:



5. Release the mouse button to paste the object:



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

Note When you copy or cut an object to the Windows Clipboard, all the object's attributes and animation links are copied with it. If you subsequently paste that object into a window, all of its attributes will still be intact.

Cutting and Pasting Object Links

WindowMaker's link paste buffer is a temporary storage area that stores links that you cut or copy from an object. (The buffer only stores the links for your most recent cut or copy action.) You can paste the links stored in the link paste buffer to any object or symbol. If you select multiple objects, the links are pasted to each individual object.

If a pasted link has no apparent value to the object, for example, a line color link on a text object, the link is not pasted.

For more information on animation links, see Chapter 5 - Creating Animation Links.

➢ **To cut, copy, paste and clear links:**

1. Select the object that you want to apply the links command to.
2. On the **Edit** menu, point to **Links**, and then click the appropriate command.

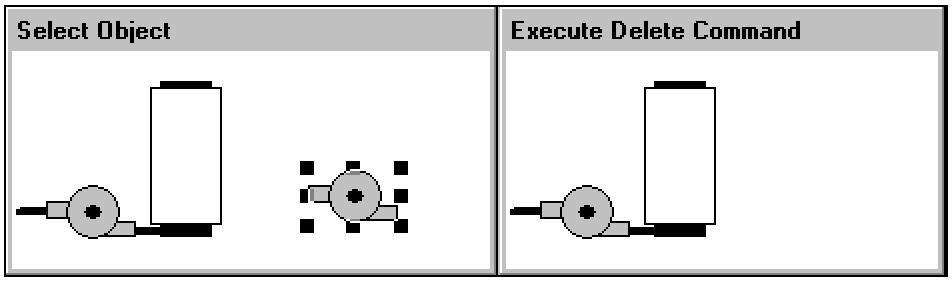
   To quickly access the link commands, right-click the object, then point to **Links**, and then click the appropriate links command.

Deleting Objects

➢ **To delete an object:**

1. Select the object(s) you want to delete.
2. On the **Edit** menu, click **Erase**.

   To quickly delete an object(s), right-click the object, and then click **Erase**, or select the object and press the DEL key.



Note Deleted objects are not copied to the Windows Clipboard.

Increasing or Decreasing a Rounded Object's Radius

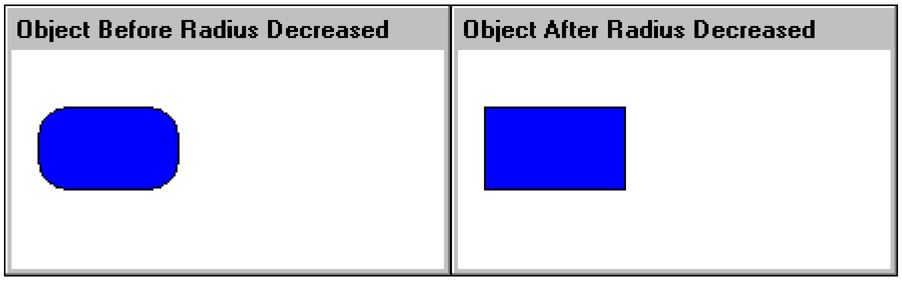
You can increase and/or decrease the corner radius of any object that you have drawn with the Rounded Rectangle tool.

➢ **To increase (or decrease) a rounded object's radius:**

1. Select the object.
2. On the **Edit** menu, click **Enlarge Radius** (or **Reduce Radius**).

   To quickly increase or decrease the radius, right-click the rounded object, and then click the appropriate command.

3. Repeat the command until the radius has increased to the desired degree. For example:

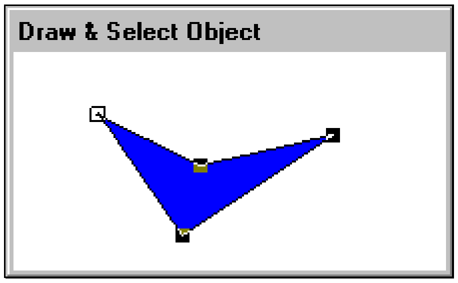


You can also use the keyboard short cut keys Shift+Plus, (+ symbol on the numeric keypad), to increase the radius or, Shift+Minus, (- symbol on the numeric keypad) to decrease the radius. Each time you press these key combinations, the command will be performed. If you hold the keys down, the command will execute continuously until the maximum increased or decreased radius for the object is reached.

Reshaping a Polyline or Polygon Object

➢ **To adjust the shape of polylines or polygons:**

1. Select the polyline or polygon object.

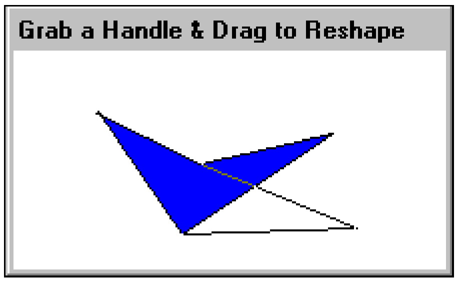


Each "point" where you clicked the mouse as you were drawing the object will reappear as a "handle."

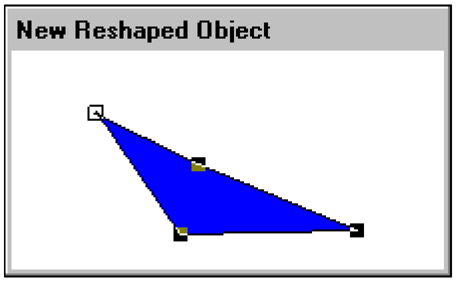
2. On the **Edit** menu, click **Reshape Object** or, click the Reshape Object tool on the **Arrange Toolbar**.

To quickly reshape a polyline or polygon, right-click the object, and then click the appropriate command.

3. To reshape the object, grab a handle and drag it to the desired location:



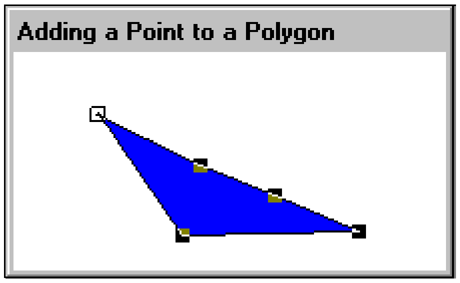
4. When you release the mouse, the object will be redrawn to its new shape:



➢ **To add or delete "points" on a polygon or polyline:**

1. Select the polyline or polygon object.
2. On the **Edit** menu, click **Add Point** or **Del Point** then click the spot of the object where you want the new point added or, click the point that you want to delete.

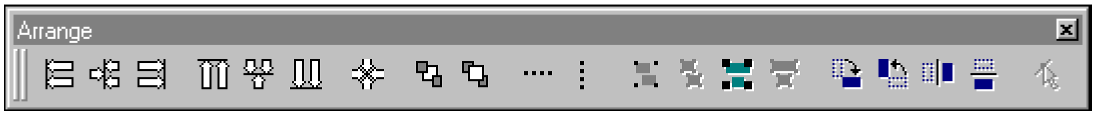
   🖰 To quickly add (or delete) points on a polyline or polygon, right-click the object, and then click the appropriate command.



Arranging Objects in your Window

WindowMaker provides you with numerous tools that you will use to arrange your objects in your windows. This section describes the various arranging tools that are available in WindowMaker.

The **Arrange Toolbar** contains tools that you can use to quickly apply most of the commands found on the **Arrange** menu to selected objects. For example:



For more information on the **Arrange Toolbar**, see Chapter 1 - WindowMaker Program Elements.

Aligning Objects

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

➤ **To align all selected objects:**

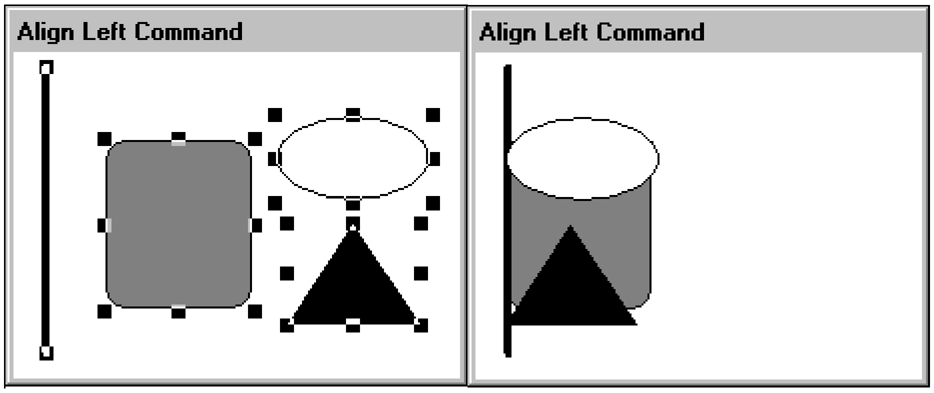
1. Select the objects(s).
2. On the **Arrange** menu, point to **Align**, and then click the appropriate align command. The selected object(s) will be aligned according to your selection.

   To quickly align objects, select the objects, and then click the appropriate tool on the **Arrange Toolbar**.

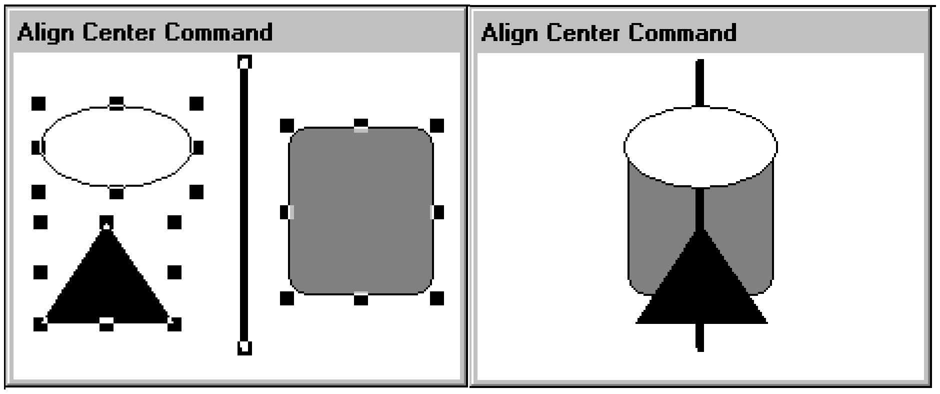
   For more information on the toolbars, see Chapter 1 - WindowMaker Program Elements.

The following examples illustrate the behavior for each alignment command:

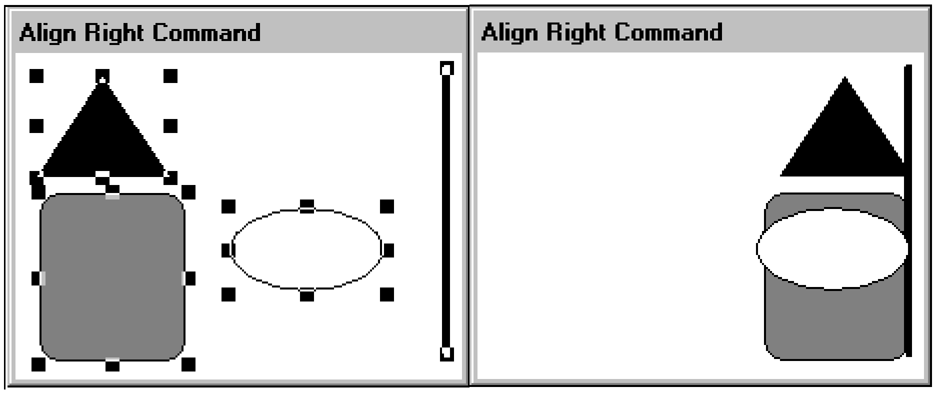
Align Left aligns the left edge of all selected objects with the left edge of the object that is farthest left in the group:



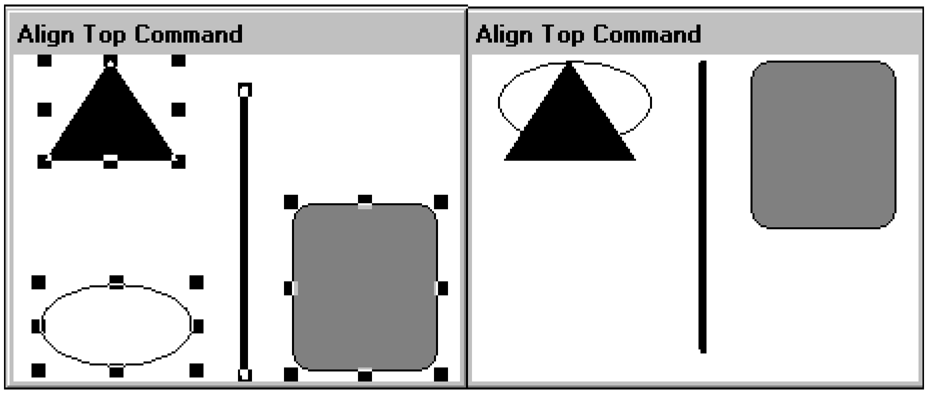
Align Center aligns all the selected objects with the vertical centerline of the group:



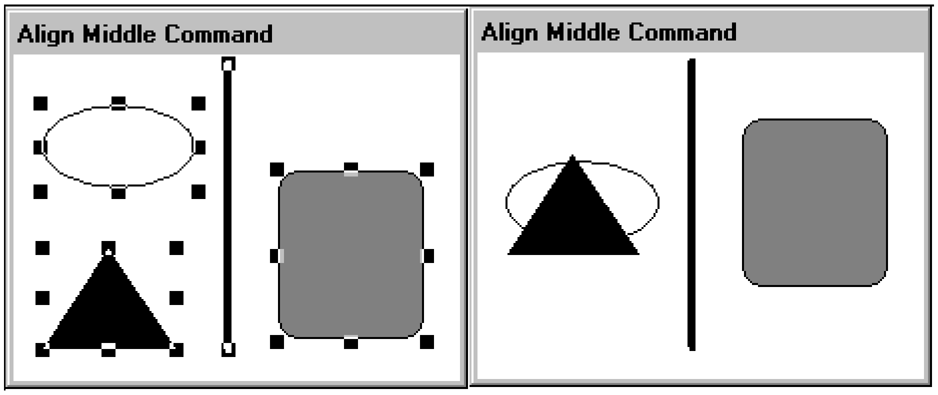
Align Right aligns the right edge of all selected objects with the right edge of the object that is farthest right in the group:



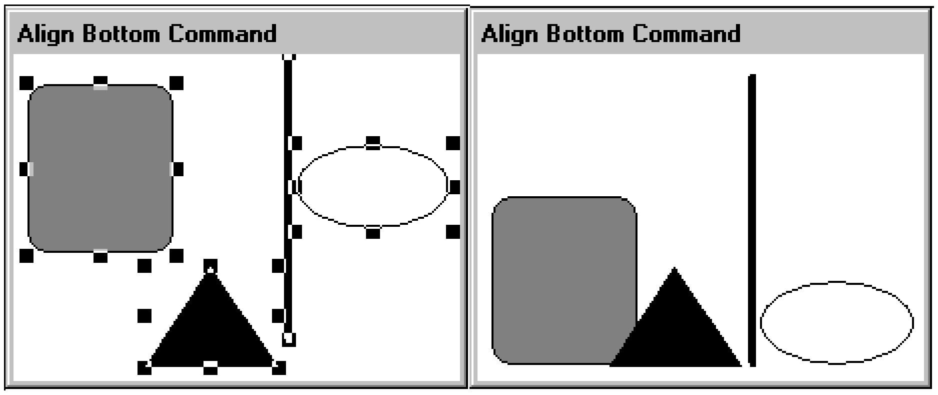
Align Top the top edge of all selected objects with the top edge of the object that is highest in the group:



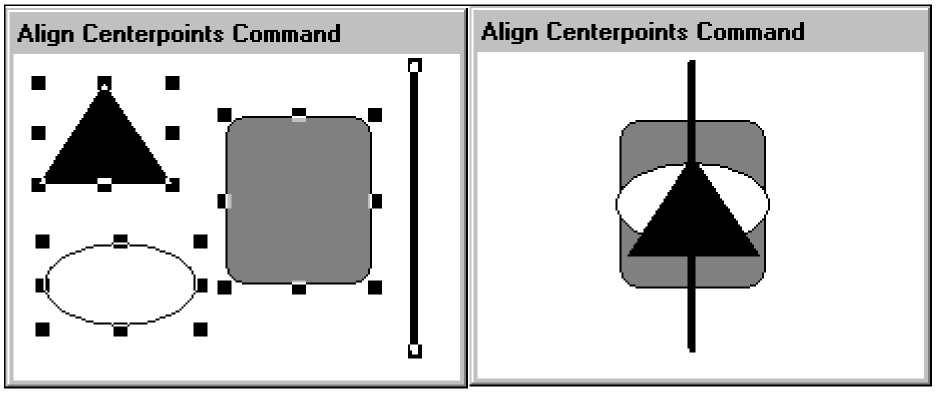
Align Middle aligns the middle of all the selected objects with the middle of the group:



Align Bottom the bottom edge of all selected objects with the bottom edge of the object that is lowest in the group:



Align Centerpoints aligns the centerpoints of all the selected objects with the centerpoint of the group:



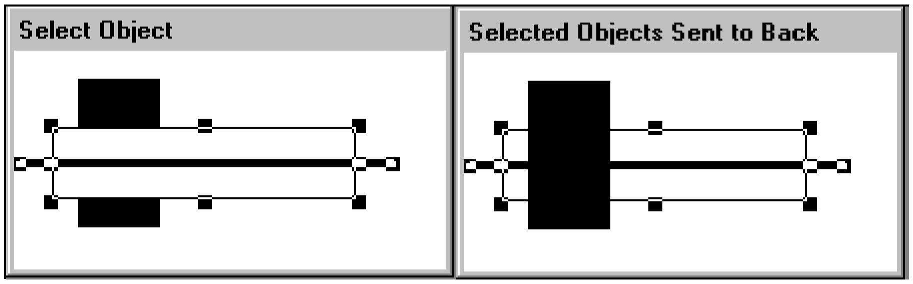
Layering Objects

You can layer the objects in your window by positioning objects in front or behind each other.

➢ **To position an object behind another object:**

1. Select the objects(s).
2. On the **Arrange** menu, click **Send to Back** or, click the Send to Back tool on the **Arrange Toolbar**. The selected object(s) will be redrawn behind the object(s) not selected in your window:

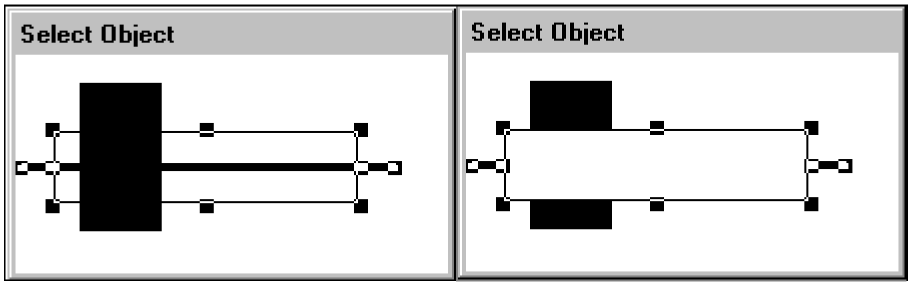
   To quickly send an object(s) to the back, right-click the object, then point to **Front/Back**, and then click **Send to Back**.



➢ **To position an object in front of another object:**

1. Select the objects(s).
2. On the **Arrange** menu, click **Bring to Front**, or click the Bring to Front tool on the **Arrange Toolbar**. The selected object(s) will be redrawn in front of the object(s) not selected in your window:

   To quickly bring an object(s) to the front, right-click the object, then point to **Front/Back**, and then click **Bring to Front**.



Notice that the farthest back object came to the very front.

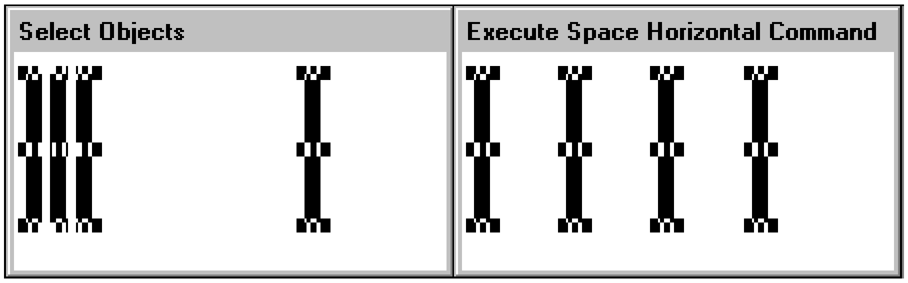
Controlling Horizontal and Vertical Spacing

You can evenly space objects horizontally between the left most selected object and the right most selected object. You can also control the vertical spacing between the top most selected object and the bottom most selected object.

➢ **To evenly space objects horizontally or vertically:**

1. Select the objects.
2. On the **Arrange** menu, click **Space Horizontally** (or **Space Vertically**) or, click the appropriate tool on the **Arrange Toolbar**. The selected object(s) will be redrawn spaced evenly between the two farthest placed objects. For example:

   For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.



Rotating Objects

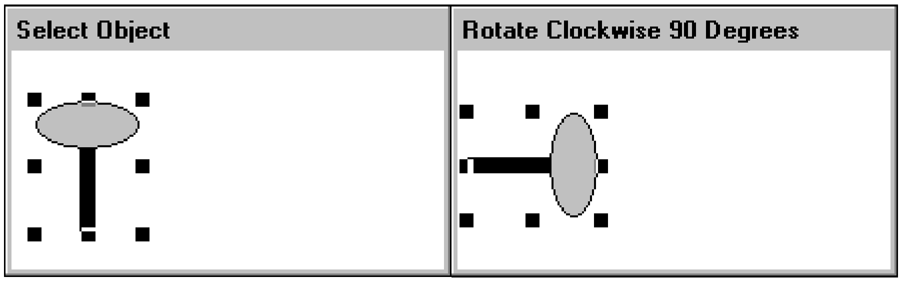
In WindowMaker, you can rotate most objects including bitmaps, JPEG, PCX, and TGA images and text objects. Objects can be rotated clockwise or counter clockwise 360 degrees in 90 degree increments (90 degrees, 180 degrees, 270 degrees and 360 degrees). Any links attached to the object are rotated with the object. You cannot rotate cells. However, you can rotate symbols.

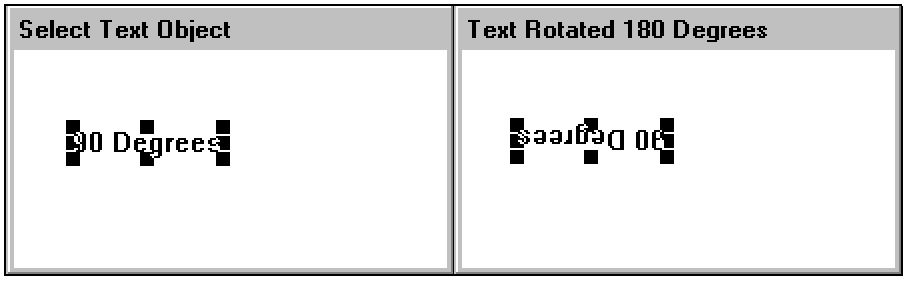
Note Rotating objects in WindowMaker has nothing to do with dynamically rotating objects in runtime. Objects are rotated in WindowViewer by linking them to an **Orientation** animation link. Text objects cannot be rotated in WindowViewer. However, bitmaps or images can be rotated by assigning an **Orientation** animation link to them.

➢ **To rotate a selected object(s) 90 degrees:**

1. Select the object(s).
2. On the **Arrange** menu, click **Rotate Clockwise** (or **Rotate CounterClockwise**). The selected object(s) will be rotated 90 degrees in the direction you chose:

   To quickly rotate an object, right-click the object, then point to **Rotate/Flip**, and then click the appropriate command.





To rotate an object 180 degrees, repeat this procedure. To rotate an object 270 degrees, perform this procedure twice, and so on.

Mirroring Objects

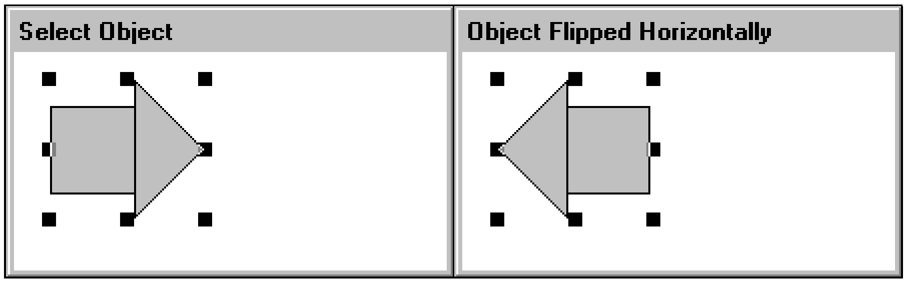
You can flip most WindowMaker objects horizontally or vertically including, bitmaps , JPEG PCX, and TGA images (text cannot be flipped, it can only be rotated). When you flip an object, you transform it into its horizontal or vertical mirror image. Any links attached to the object are flipped with the object.

➢ **To flip a selected object:**

1. Select the object(s).
2. On the **Arrange** menu, click **Flip Horizontal** (or **Flip Vertical**) or, select the appropriate tool on the **Arrange Toolbar**. The selected object(s) will flip. For example:

For more information on toolbars, see Chapter 1 - WindowMaker Program Elements.

To quickly flip an object(s), right-click the object, then point to **Rotate/Flip**, and then click the appropriate command.



Creating Cells and Symbols

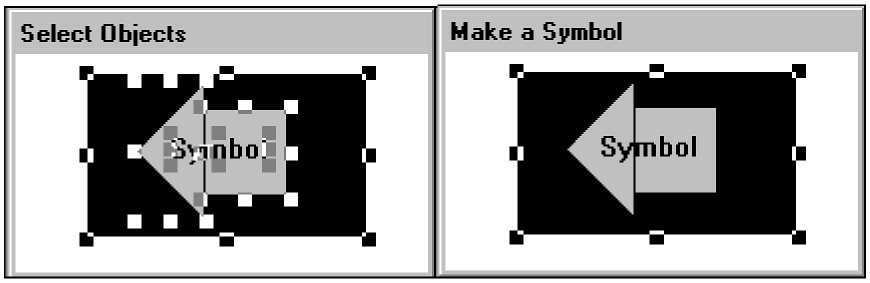
You can combine multiple objects into two different types of single units; cells and symbols. Multiple cells can be combined into a single cell. Cells are objects that maintain a fixed spatial relationship between individual graphic elements. The individual components within a cell, except another cell, can be animated. 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.

Double-clicking a cell will cause the **Substitute Tagnames** dialog box to appear (not the animation links selection dialog box as with objects and symbols).

For more information on substituting tagnames, see Chapter 4 - Tagname Dictionary.

Note When combining cells each cell will be retained, so when the combined cell is broken the original cells are restored.

A symbol can be made up of multiple symbols and/or multiple simple objects as illustrated below:



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.)

Note You cannot make a symbol if more than one of the selected objects has links. If you combine two symbols into a new symbol, the original symbol structure is lost. Therefore, if you break the new symbol, it will be broken into the individual components of each original symbol. The two original symbols are lost.

For more information on animation links, see Chapter 5 - Creating Animation Links.

➢ **To create a symbol or cell:**

1. Select the objects that you want to include in the cell or symbol...
2. On the **Arrange** menu, click **Make Cell** (or **Make Symbol**), or click the appropriate tool on the **Arrange Toolbar**.

   For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

   To quickly create a cell or symbol, select all the objects. Right-click one of the selected objects, point to **Cell/Symbol**, and then click the appropriate command.

➢ **To break a symbol or cell:**

1. Select the symbol or cell...
2. On the **Arrange** menu, click **Break Cell** (or **Break Symbol**), or click the appropriate tool on the **Arrange** toolbar.

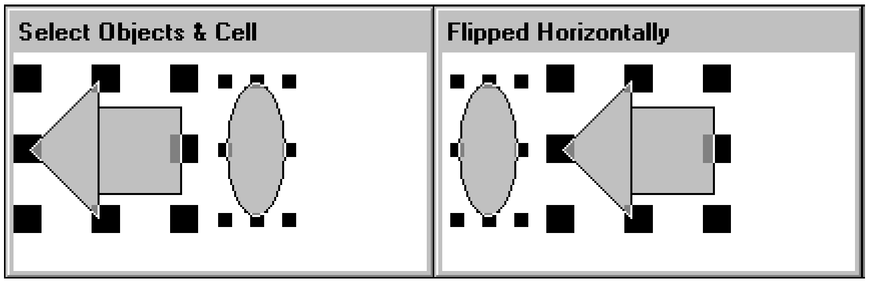
   For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

   If the symbol has links defined for it, the links are automatically saved to the link paste buffer.

To quickly break a cell or symbol, right-click the cell or symbol, then point to **Cell/Symbol**, and then click the appropriate command.

Flipping Cells

When you flip cells, they are not mirrored. Only the position of the cell in the group of objects is mirrored. For example:



Compare the location of the cell (on the left) and direction it is facing before it is flipped to the direction it is facing after it is flipped. Its position was flipped, but not its contents. The ellipse object was mirrored. The same applies to cells flipped vertically.

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.

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

Click the Snap to Grid tool on the **View** toolbar to turn snap to grid on and off, or on the **Arrange** menu, click **Snap to Grid**.

For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

➢ **To configure the grid:**

1. On the **Special** menu, point to **Configure** then click **WindowMaker** or, in the Application Explorer under **Configure**, double-click **WindowMaker**. The **WindowMaker Properties** dialog box will appear.
2. In the **Spacing** box, type 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.

   If you do not select **Show Grid**, no grid will be visible in your windows when you turn snap to grid on.

   For more information on configuring the grid, see "Customizing Your Development Environment."

Working with Images and Bitmaps

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

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

In WindowMaker, you can rotate bitmaps, JPEG, PCX, and TGA images. They 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 bitmap are rotated with it.

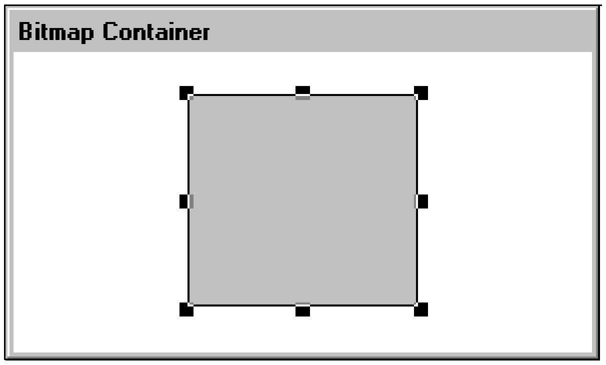
Note Rotating bitmaps in WindowMaker has nothing to do with dynamically rotating them in runtime. Bitmaps or images are rotated in WindowViewer by linking them to an **Orientation** animation link.

You can also define a bitmap with a transparent color, so that you can float it over other objects. When you define a bitmap with a transparent color, the window background color or any objects behind the bitmap will show through it everywhere the transparent color is used. (Only one transparent color may be used per bitmap.)

👓 For more information on transparent bitmaps and images, see "Creating a Transparent Bitmap."

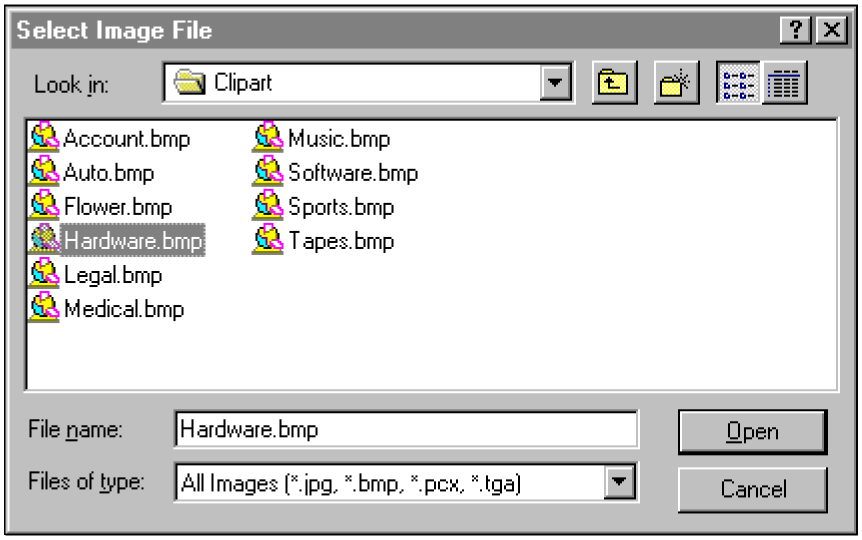
➢ **To import a bitmap or JPEG, PCX or TGA file type:**

1. Click the Bitmap tool (your cursor turns into a cross-hair) then draw a bitmap container in your window (the size is irrelevant).
2. Select the bitmap container:

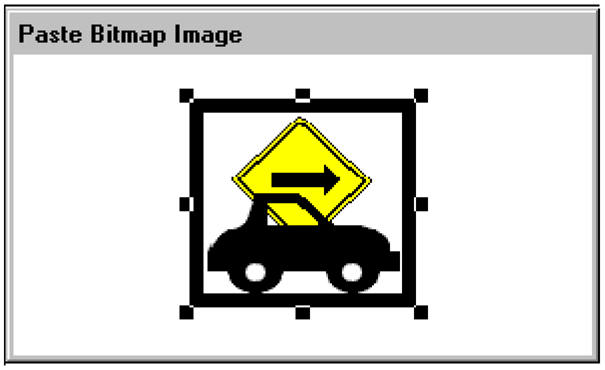


3. On the **Edit** menu, click **Import Image**. The Windows **Select Image File** dialog box appears:

   To quickly paste the image, right-click the bitmap container, and then click **Import Image**.

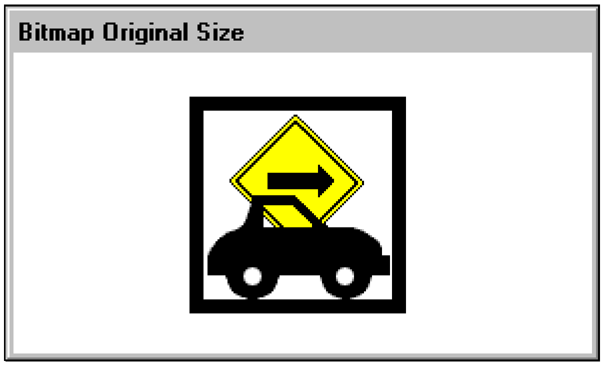


4. Locate and select the .BMP, .PCX, .TGA or .JPG file that you want to import as a bitmap, then click **Open** or double-click the image filename. The image will be pasted into your bitmap container:



5. To make the bitmap it's original size, select it, then on the **Edit** menu, click **Bitmap - Original Size**. The bitmap will be redrawn at its original size.

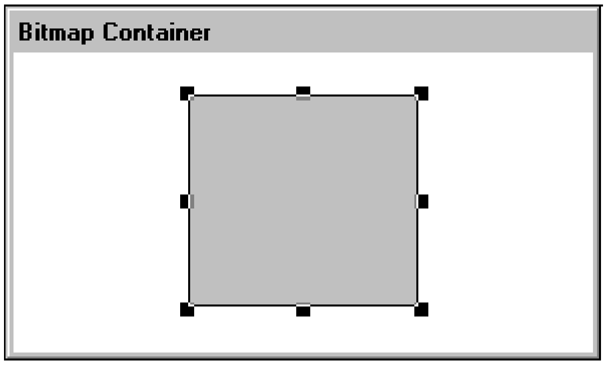
   To quickly size the bitmap, right-click the bitmap, and then click **Bitmap - Original Size**.



Pasting a Bitmap from the Windows Clipboard

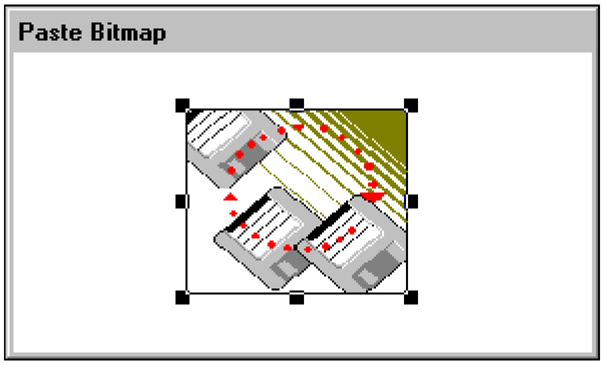
➢ **To paste a bitmap from the Windows Clipboard into a window:**

1. Copy the graphic to the Windows Clipboard. For example, display the graphic, then hold down the ALT key while you press the PRINT SCRN key to copy it to the Windows Clipboard.
2. Click the Bitmap tool (your cursor turns into a cross-hair), then draw a bitmap container in your window (the size is irrelevant).
3. Select the bitmap container:



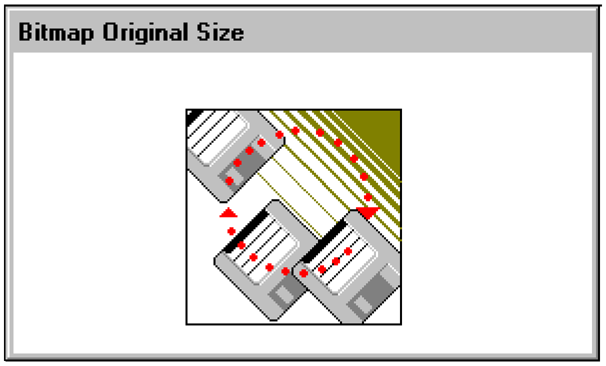
4. On the **Edit** menu, click **Paste Bitmap**. The bitmap from the Windows Clipboard will be pasted into the bitmap container:

   To quickly paste the bitmap, right-click the bitmap container, and then click **Paste Bitmap**.



5. To make the bitmap it's original size, select it, and then on the **Edit** menu, click **Bitmap - Original Size**. The bitmap will be redrawn at its original size:

   To quickly size the bitmap, right-click the bitmap, and then click **Bitmap - Original Size**.

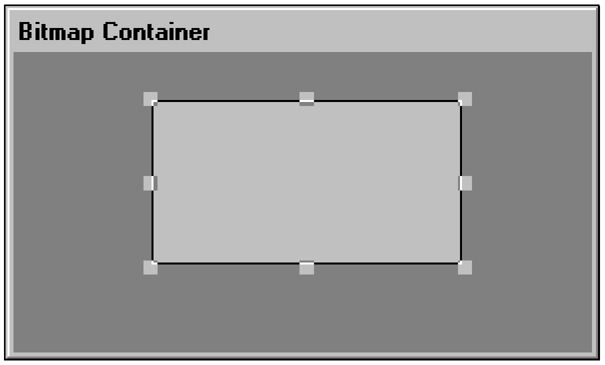


Creating a Transparent Bitmap

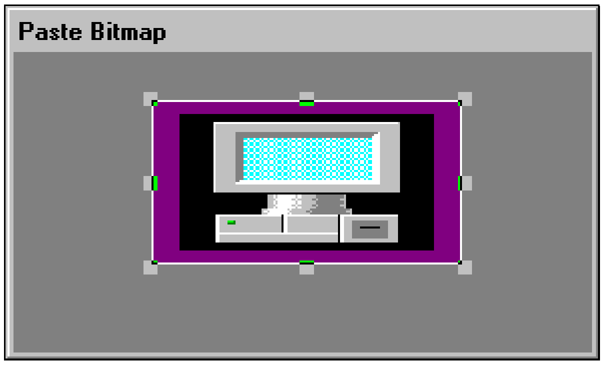
You can define a bitmap or image with a transparent color, so that you can float it over other objects. When you define a bitmap or image with a transparent color, the window's background or any objects behind the bitmap will show through it everywhere the transparent color is used.

➢ **To create a transparent bitmap:**

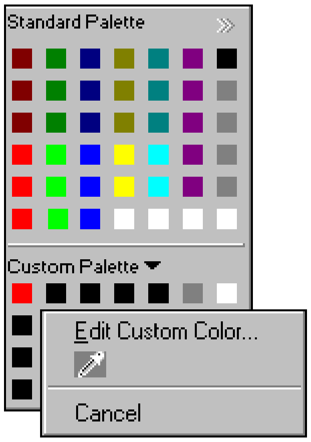
1. Click the Bitmap tool (your cursor turns into a cross-hair) then draw a bitmap container in your window (the size is irrelevant).
2. Select the bitmap container:



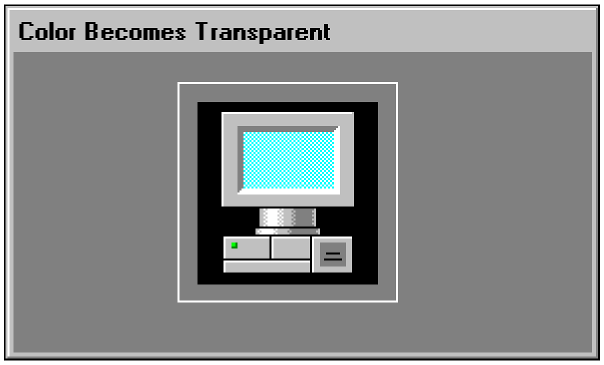
3. Right-click the bitmap container, and then click **Paste Bitmap** (if you have copied the graphic to the Windows Clipboard) otherwise, click **Import Image** (to locate and select the .BMP, .PCX, .TGA or .JPG file to open). The bitmap image will be pasted into the bitmap container:



4. Right-click the bitmap, and then click **Bitmap - Original Size** to return the bitmap to its original size.
5. With the bitmap selected, click the Transparent Color tool 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. The **Edit Custom Color** dialog box appears:



7. Click the Blotter tool (the **Edit Custom Color** dialog box will close).
8. Click the color in the bitmap that you want to make transparent. The color will be copied to the color square that you selected in the transparent color palette.
9. Click the color square to apply the transparent color to the bitmap:

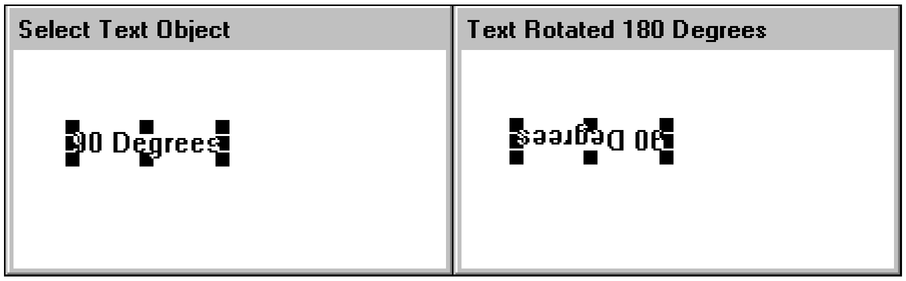


In this example, we made the wide border area of the bitmap transparent. Therefore the window background color now shows through the bitmap. If objects were behind the now transparent area, they would also shown through the bitmap.

Note Only one transparent color can be applied per bitmap.

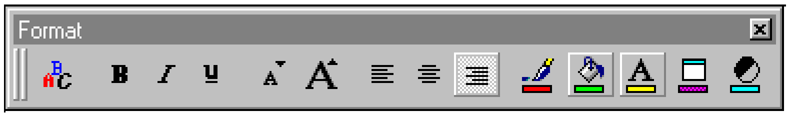
Working with Text Objects

In WindowMaker you can change the font, font style, font size, justification and rotation of any selected text object. You can also rotate it 360 degrees by 90 degree increments (90 degrees, 180 degrees, 270 degrees and 360 degrees). For example:



Note Rotating text objects in WindowMaker has nothing to do with dynamically rotating objects in runtime. **Orientation** animation links cannot be applied to text objects. Therefore, text objects cannot be rotated in WindowViewer.

The **Format Toolbar** contains tools that you can use to quickly apply most of the commands found on the **Text** menu to selected objects. For example:



For more information on the **Format Toolbar**, see Chapter 1 - WindowMaker Program Elements.

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.

For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

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** (Only 1 # sign is necessary)
0.0	Forces one leading zero and one decimal place; e.g., **.1** would display as **0.1** e.g., **77.1** would display as **77.1**
00000	Forces leading zeros as required; e.g., **123** would display as **00123** e.g., **1234** would display as **01234** e.g., **12345** would display as **12345**
#,##0.0	Inserts comma and leading zero if required, and one decimal place; e.g., **1234.56** would be displayed as **1,234.6** e.g., **123.4** would display as **123.4**
0,000.0	Forces comma, leading zeros and one decimal place. e.g., **12.3** would display as **0,012.3**

Note If you use a zero in the format, it must be followed by zeros. All places to the right of the decimal point must always be zeros. For example, **000.00** is correct, while **#0#0.0#** is incorrect.

All normal text formatting applies to numeric values. These include font, size, color justification and bolding.

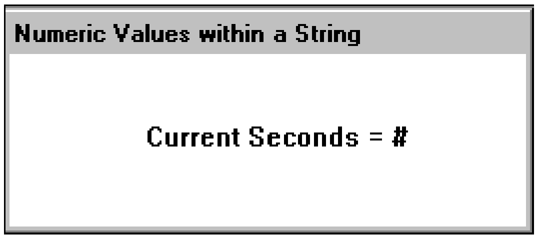
➢ T **To create a text object:**

1. Click the Text tool in the **Draw Object Toolbar**.
2. Click in the window and type the text string.

   To quickly access the various commands that can be applied to a text object, right-click the text object, and then click the appropriate command.

➢ **To display a numeric value within a text string:**

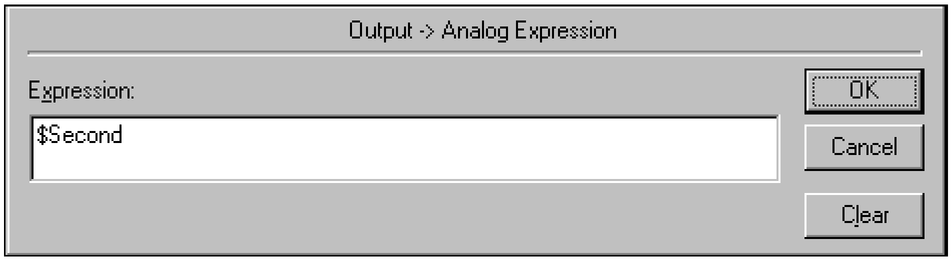
3. Click the Text tool, and then type a text object in the window using one of the previously described valid numeric formats. For example:



4. Select the object, and then on the **Special** menu, click **Animation Links** or double-click the text object. The animation links selection dialog box will appear.

   To quickly access the dialog box, right-click the text object, and then click **Animation Links**.

5. In the **Value Display** section, click **Analog**. The **Output -> Analog Expression** dialog box appears:



$Second

6. In the **Expression** box, type an analog tagname or expression. (In this example, the system tagname **$Second** is being used.)

7. Click **OK**.

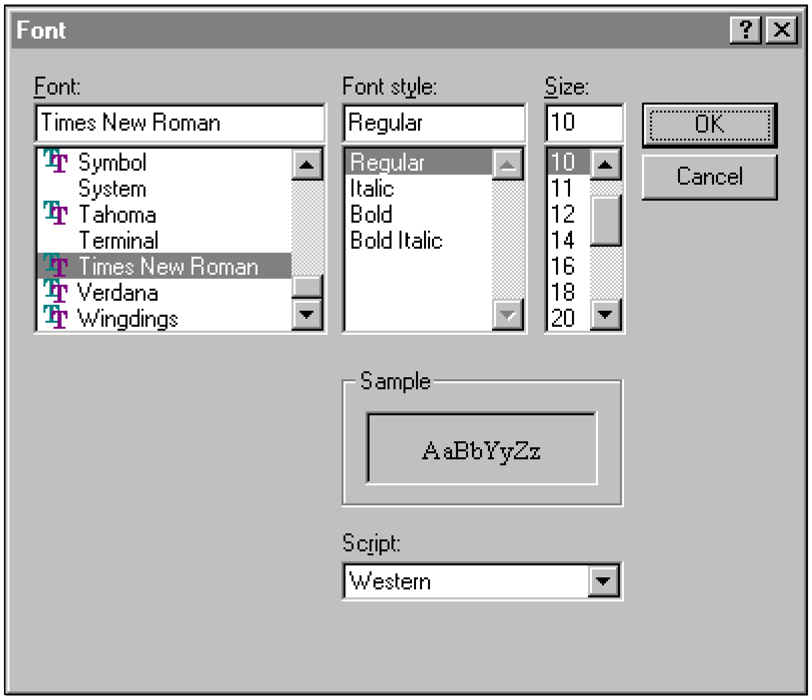
8. Click the **Runtime** fast switch in the upper right hand corner of the menu bar (or use the short cut keys ALT + !) to switch to WindowViewer or, on the **File** menu, click **WindowViewer**.

9. 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.

10. 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 or, on the **File** menu, click **WindowMaker**.

➢ **To change the font, font style and font size of a string:**

1. Select the text string, and then on the **Text** menu, click **Fonts**, click the Fonts tool on the **Format Toolbar**. The standard Windows **Font** dialog box appears:



Font: Times New Roman
Font style: Regular
Size: 10
Script: Western

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**.

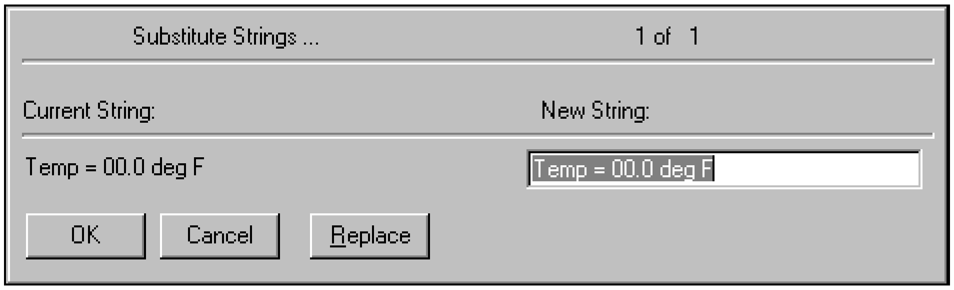
Note A font's point size will be enlarged or reduced in accordance with the range of point sizes available for the selected font. The default font for WindowMaker is System and cannot be sized. Choose a Windows True-Type font before changing the size.

Editing Text Objects

➢ **To change the text in an object:**

1. Select the object or button with the text.
2. On the **Special** menu, click **Substitute Strings**. The **Substitute Strings** dialog box appears:

   To quickly access the dialog box, right-click the text object, point to **Substitute**, and then click **Substitute Strings**.



Temp = 00.0 deg F

Temp = 00.0 deg F

   For more information on Analog Input/Display links, see Chapter 5 - Animation Links.

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

   You can also use this command on strings that are included in a symbol or cell and to change the label on buttons drawn with the Button tool.

   When you change a text string, it retains all of it's original attributes, that is font, style, color, and so on. All normal text formatting also applies to numeric values.

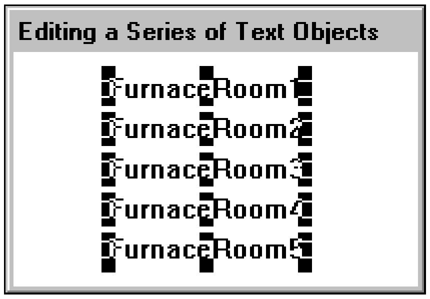
   You can also select and edit multiple string objects at the same time.

Replacing a Portion of a Text Object

You can change a portion of a text object's text and InTouch will automatically make the change to all selected text objects using the same text.

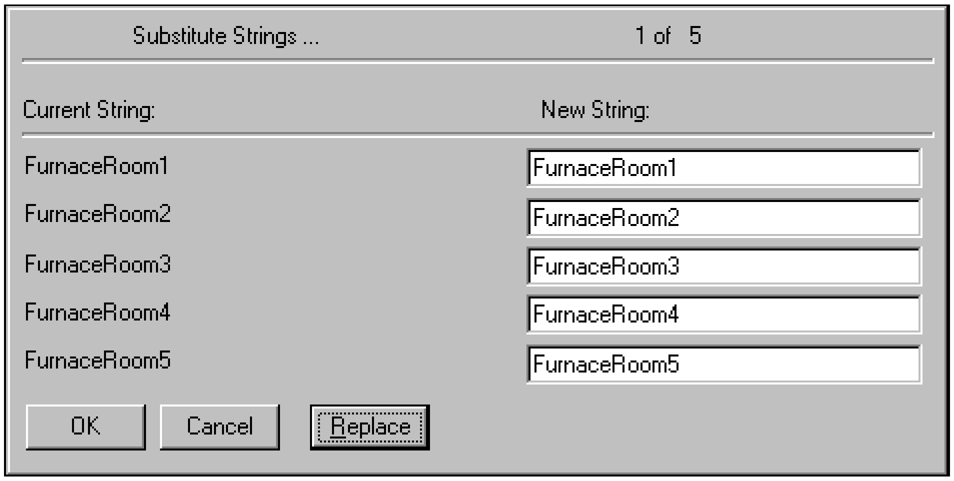
➢ **To change a portion of text in a series of text objects:**

1. Select all of the text objects.



2. On the **Special** menu, click **Substitute Strings**. The **Substitute Strings** dialog box appears:

   To quickly access the dialog box, right-click a text object, point to **Substitute**, and then click **Substitute Strings**..

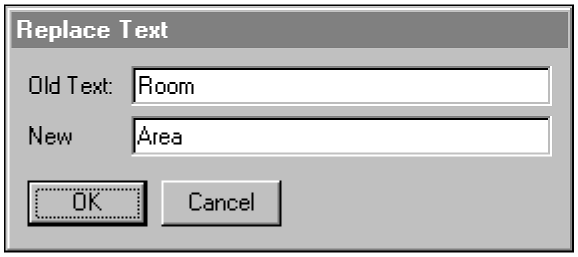


Substitute Strings ... 1 of 5

Current String:	New String:
FurnaceRoom1	FurnaceRoom1
FurnaceRoom2	FurnaceRoom2
FurnaceRoom3	FurnaceRoom3
FurnaceRoom4	FurnaceRoom4
FurnaceRoom5	FurnaceRoom5

If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

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

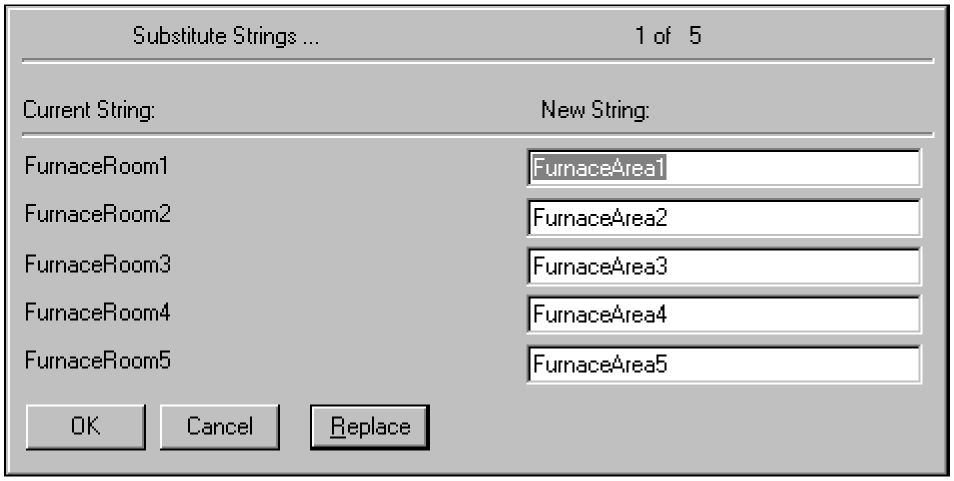


Replace Text

Old Text: Room

New Area

4. In the **Old Text** box, type the portion of the string that you want to replace.
5. In the **New** box, type the replacement text.
6. Click **OK**. The **Substitute Strings** dialog box reappears showing the change made to the selected text strings:



Substitute Strings ... 1 of 5

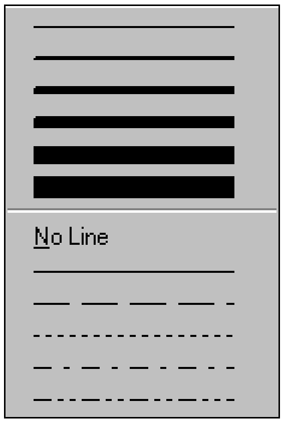
Current String:	New String:
FurnaceRoom1	FurnaceArea1
FurnaceRoom2	FurnaceArea2
FurnaceRoom3	FurnaceArea3
FurnaceRoom4	FurnaceArea4
FurnaceRoom5	FurnaceArea5

7. Click **OK**. All of the selected text objects will automatically be modified.

Working with Lines and Outlines

You can change the style and width of a line object including the outlines around ellipses, rectangles, polygons and bitmaps or images. You can apply a line style or width change to a single selected object or multiple selected objects.

The **Line** menu is divided into two sections. The top section contains the line widths and the bottom section contains the line styles. For example:



➢ **To apply a line command:**

Select the object, and then on the **Line** menu, click the desired line style or width.

If you do not select an object when you select a line style or width, the change will be applied to the default settings for all line tools in the **Wizard Toolbar**.

Note You can only change the width of solid lines. Broken lines can only be single pixel wide. Wider lines take longer to draw in runtime.

➢ **To remove an object's outline:**

Select the object, and then on the **Line** menu, click **No Line**. The object's outline will be removed.

Working with Wizards

Wizards save you a considerable amount of time during application development. They are easy to use and easy to configure. To configure a wizard, you install it, select it in the **Wizard Selection** dialog box, paste it into your window, and then double-click it. It's respective configuration dialog box will appear (assuming that it is a wizard that can be configured).

For example, if you wanted 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, and so on. You can save a considerable amount of development time by using Wizards because you don't have to draw the individual components for the object, or set the value ranges for the object, or animate the object.

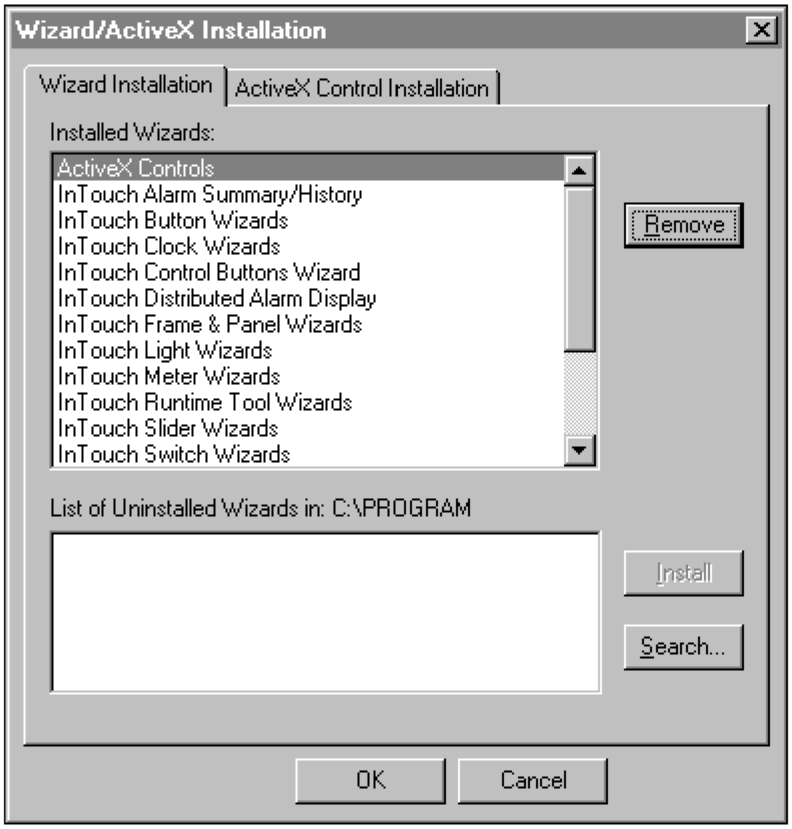
The FactorySuite InControl program includes the following five wizards that you can place in an InTouch window. These wizards allow easy and effective interaction between InControl and InTouch.

InControl Project	Launches an InControl project. This starts InControl for the specified project and allows the operator to use all the InControl functions to edit, compile, download, and run the programs in that project.
Configure Runtime Engine	Starts the runtime engine and select the node on which it runs.
InControl Mode	Used to set programs currently downloaded to the runtime engine to the specified mode (Run, Pause, Single Step).
InControl Edit	Launches an individual program in a project. This starts InControl within the development environment, at a specified line within the specified program. You can use all the tools available in the development environment to edit, compile, download, and run the program.
InControl Runtime Add Tag	Associates InTouch tagnames with InControl symbols (variables).

➢ **To install or remove wizards:**

1. On the **Special** menu, point to **Configure**, and then click **Wizard/ActiveX Installation**, or in the Application Explorer, double-click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears with the **Wizard Installation** property sheet active:

   In the Application Explorer, you can also right-click **Wizard/ActiveX Installation**, and then click **Open**.


Wizard/ActiveX Installation

Wizard Installation | ActiveX Control Installation

Installed Wizards:

ActiveX Controls
InTouch Alarm Summary/History
InTouch Button Wizards
InTouch Clock Wizards
InTouch Control Buttons Wizard
InTouch Distributed Alarm Display
InTouch Frame & Panel Wizards
InTouch Light Wizards
InTouch Meter Wizards
InTouch Runtime Tool Wizards
InTouch Slider Wizards
InTouch Switch Wizards

List of Uninstalled Wizards in: C:\PROGRAM

2. In the **Installed Wizards** list, select the wizard(s) that you want to remove from your application, and then click **Remove**. A message box will appear asking you to confirm the deletion.

Note The **Remove** button is active only when wizards are displayed in the **Installed Wizards** list.

   To select a group of wizards, click the first wizard in the list, hold down the SHIFT key and click the last wizard that you want to select. All wizards in the list between your first and last selection will be selected. To select multiple wizards that are not consecutively listed, click the first wizard, hold down the CTRL key, and then click the next wizard. Repeat this for all wizards that you want to select.

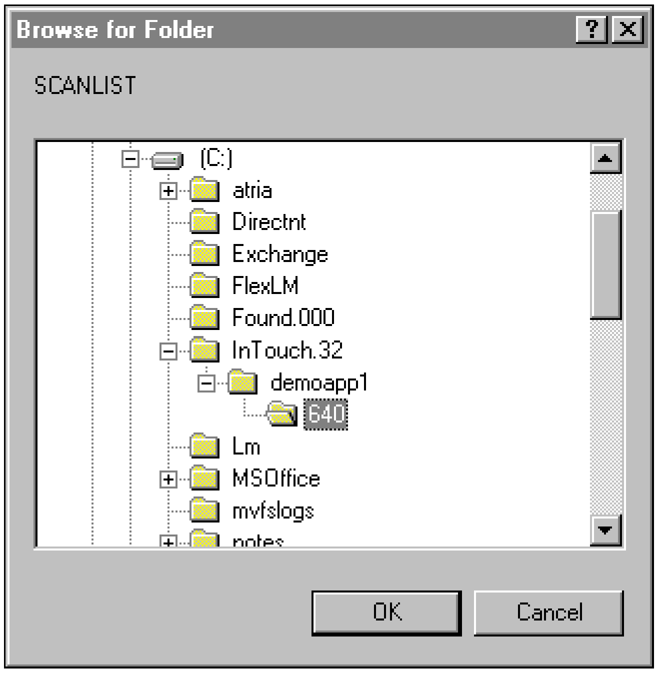
3. Click **Yes** to remove the wizard. The removed wizard(s) is moved to the **List of Uninstalled Wizards** list.

   When you remove a wizard, it is not deleted. However it is no longer loaded into memory.

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

Note The **Install** button is active only when wizards are displayed in the **List of Uninstalled Wizards** list.

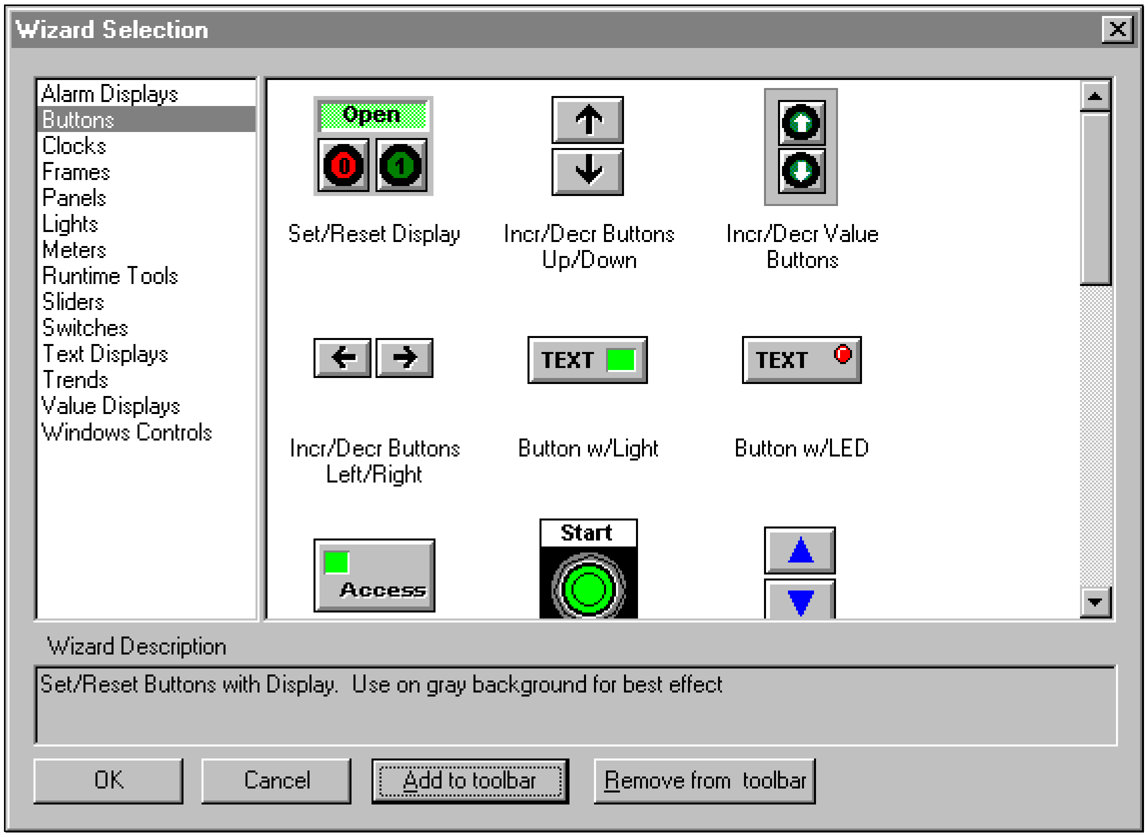
5. Click **Search** if you want to install wizards from another directory. The **Search for Wizard files** dialog box appears:



6. Locate the directory containing the wizards that you want to install, and then click **OK**. The wizard installation dialog box will reappear.
7. Any Wizards that were found will appear in the **List of Uninstalled Wizards** list and you can now install them as previously described.

➢ **To place a wizard in a window:**

1. Click the Wizard Dialog tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears:



Set/Reset Buttons with Display. Use on gray background for best effect

2. In the list of wizards, click the category of wizards that you want to use.

   All available wizards in that category will be shown the display area. For example, if you select **Buttons**, all available button wizards will immediately be shown in the display area.

3. Select the wizard that you want to use, and then click **OK** or double-click the wizard. The dialog box will close and your window will reappear.

   To add the wizard to the **Wizards/ActiveX Toolbar**, click **Add to toolbar**. Once you add a wizard to the **Wizards/ActiveX Toolbar**, you can select it and paste it into your open window at any time.

   Note The number of wizards that you can add to the toolbar is limited to your system resources.

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

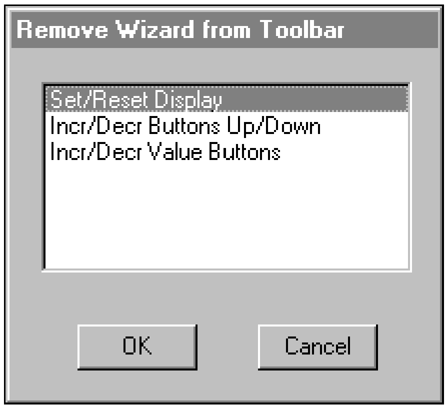
5. Double-click the wizard to configure it (if applicable).

Note Some toolbar functions may be used to modify applicable wizards directly. For example, the Reduce Font tool, Line Color tool, Fill Color tool, and so on.

For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

➢ **To remove wizards from the toolbar:**

1. Click the Wizard Dialog tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box will appear.
2. Click **Remove from toolbar**. The **Remove Wizard from Toolbar** dialog box appears:



3. Select the wizard(s) that you want to remove from the toolbar.
4. Click **OK**.

InTouch Windows Control Wizards

The windows control wizards are complex objects. Unlike normal wizards, they provide enhanced functionality through InTouch QuickScripts. They can be used for editing data objects and operator inputs. Windows control wizards also have InTouch tagname **.fields** and some of their properties are accessible both during development and runtime when their QuickScript functions are used.

You can use windows control wizards in your InTouch application to display text/data, gather user input or offer choices for the user at runtime. Choices may be in the form of list boxes, check boxes, combo boxes and radio (option) buttons. You can use the text boxes to display or input text/data.

When you configure a windows control wizard, you must specify a **Control Name** to identify the control. InTouch uses the **Control Name** to identify the control when you execute a windows control QuickScript function. Therefore, you must also specify the **Control Name** parameter in the QuickScript function. For example:

```
SetPropertyD ( "ControlName.Property", Discrete );
```

🕮 For more information on using the windows control QuickScript functions, see your *InTouch Reference Guide*.

Control Names do not add to the application's tagname count and must be unique for each control. Tagnames, although not required, are essential for productive use of the control. For example, selecting an item in a list box is not useful if the selected item is not automatically assigned to a tagname, thus making it accessible to InTouch.

Windows controls have properties (similar to tagname **.fields**) that can be modified at development (WindowMaker) and runtime (WindowViewer). They also support specific QuickScript functions that can be processed at runtime to modify lists, load files, disable controls, and so on.

Note To function properly, windows control objects cannot overlap each other. For verification, select the object in WindowMaker and insure that the object's handles do not touch another object.

The initial value of tagnames assigned to either a list box or combo box cannot be used to initialize the value of the list box or combo box.

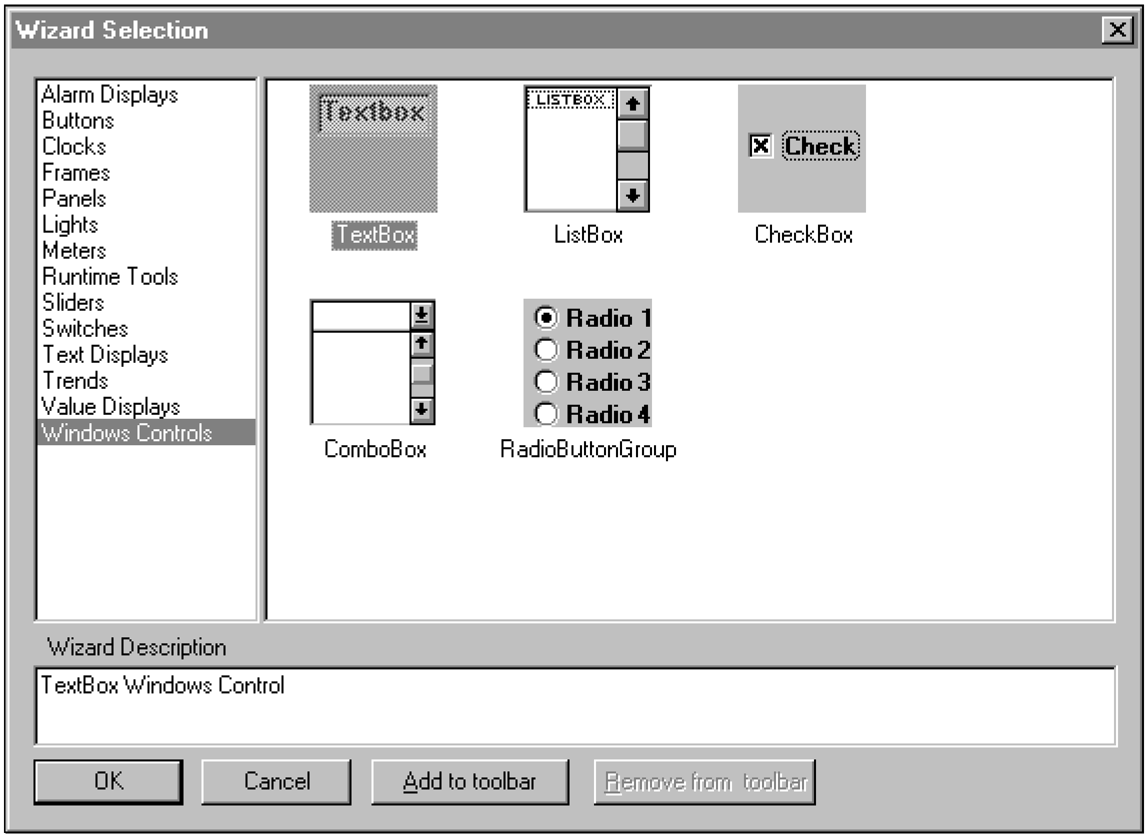
🕮 For more information on using the windows control**.fields**, see your *InTouch Reference Guide*.

Using InTouch Windows Control Wizards

The Windows control wizards available include: Text Boxes, Check boxes, Combo Boxes, List boxes and Radio (Option) Buttons. The windows control wizards also have tagname **.fields** and QuickScript functions that you can use to dynamically control them in runtime.

Windows control wizards are pasted into your windows just like any other wizard.

For more information on pasting wizards, see "To place a wizard in a window."



TextBox Windows Control

To achieve the best windows control 3-D effect, select a gray background for your window. If your window color cannot be gray, place a gray "Panel Wizard" behind the windows control wizard.

Windows Control Usability Guidelines

It is very important that you follow the guidelines below when you are using windows control wizards:

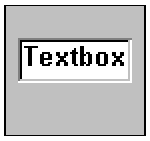
1. Windows control wizards work properly when they do not overlap other windows control wizards or other normal graphic objects.

   🖰 To verify that the windows control wizard is not overlapping any other object, select it in WindowMaker. Verify that none of its selection handles are touching another graphic object on the screen.

2. Windows control wizards should be used sparingly and intelligently.

   🖰 Placing 10 to 20 windows control wizards in one window results in non-intuitive, hard to navigate displays. When it is necessary for you to use numerous windows control wizards, we recommend that you call other dialog boxes with additional windows control wizards.

Text Box Control Wizard



Text boxes control wizards are versatile controls that can be used to get input from the user or to display text such as a notepad file (ASCII flat files only). You can configure text boxes to allow user input or as read-only for display purposes only You can only assign **Message** type tagnames to a text box control wizard.

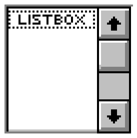
Text box control wizard QuickScript examples:

```
wcLoadText("TextBox_1",FileName);
```

```
wcSaveText("TextBox_1",FileName);
```

Note If the tagname has been defined with a maximum length, only that number of characters can be assigned from the text box contents to the tagname. If no tagname is assigned to the text box, its contents can be up to 65,535 characters.

List Box Control Wizard



List box control wizards display a list of choices to the user. The choices are displayed vertically in a single column. If the number of items exceeds what can be displayed in the list box, scroll bars will automatically appear on the control. List boxes require the user to choose from a list and do not allow user input. You can only assign **Message** type tagnames to a list box control wizard.

List box control wizard QuickScript example:

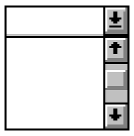
```
IF (ItemToAdd == "") THEN
   Show "Cannot Add Blank";
ELSE
   wcAddItem("ListBox_1",ItemToAdd);
   {Get the index of the item we just added.}
   {Since the list is sorted, we cannot assume anything about the new items location.}

   GetPropertyI("ListBox_1.NewIndex",ListBox_NewIndex);
   {Now, set the Item Data specified on the screen by the user.}
   {From this point on, this item will have this data associated with it.}
   {It allows you to associate a number with a string; the string being displayed in the
   list.}

   wcSetItemData("ListBox_1",ListBox_NewIndex,ListBox_ItemData);
   {Since we just added an item, update the "NumItems" variable.}

   GetPropertyI("ListBox_1.ListCount",ListBox_NumItems);
ENDIF;
```

List box and combo box control wizards use an internal, one-based numbering system (item index) that automatically assigns a number to each item in the list. For example, the first item in the list is assigned the number 1, the second is number 2, and so on. The item index is a 32-bit integer that is used as a parameter for windows control "item"

Note When using list boxes and combo boxes with the **wcLoadList()** and **wcSaveList()**, specific formatting and information must be provided.

🕮 For more information, on the windows control QuickScript functions, see your *InTouch Reference Guide*.

Combo Box Control Wizard



Combo box control wizards combine the features of a text box and a list box. The choices are displayed vertically in a single column. If the number of items exceeds what can be displayed in the list box, scroll bars will automatically appear on the control. Combo box control wizards allow the user to make a selection, either by typing text or selecting an item from the list. You can only assign **Message** type tagnames to a combo box control wizard.

There are three styles of combo boxes:

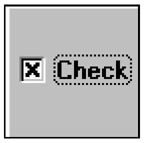
Type	Description
Simple	Combo boxes display their list at all times. To display the entries in the list box, the list box must be drawn large enough to display all entries. A vertical scroll bar is automatically inserted if there are more entries than can be displayed. Simple combo boxes allow the user to type in choices that are not in the list or will display the first item in the list matching the typed letters. If no match is found, the top of the list is displayed.
Drop Down	Combo boxes allow the user to either type text directly or click an arrow to open a list of choices. As with the simple combo box, this control allows the user to type choices not on the list or will display the first item in the list matching the typed letters. If no match is found, the top of the list is displayed.
Drop Down List	Combo boxes are similar to simple list boxes. They display a list of choices to select. Unlike list boxes, however, the list is not displayed until the arrow is clicked. This type of control is used to conserve screen space.

Combo box control wizard QuickScript example:

```
wcAddItem("ComboBox_1", UserMessage );
```

Where: *UserMessage* is a tagname assigned to a sting input link. When the operator types a new message, and then clicks this action pushbutton, linked to the **"On Down"** QuickScript, the message is displayed in the combo box wizard with the control name "ComboBox_1."

Check Box Control Wizard



A check box indicates whether a particular condition is on/off, true/false or yes/no. Check boxes work independently of each other, allowing the user to select or deselect any number of check boxes at the same time. Check boxes return a discrete value. They return 0 if not selected and 1 if selected. You can only assign **Discrete** type tagnames to a check box control wizard.

Check box control wizard QuickScript example:

```
{ Clear any previous machine }
Machine = "";

IF (Cutter_Selected) THEN
   Machine = Machine + "Cutter";

ENDIF;

IF (Mixer_Selected) THEN
   Machine = Machine + "Mixer";

ENDIF;
```

Where: *Cutter_Selected* is the tagname assigned to the control name "Checkbox_1" in the first check box wizard. *Mixer_Selected* is the tagname assigned to the control name "Checkbox_2" in the second check box wizard.

Machine is a tagname assigned to a string output link that displays the name for the check box selected.

Radio (Option) Buttons Control Wizard

Radio, or option buttons present a set of choices for the user. Unlike check boxes, radio buttons operate as part of a group. Selecting one radio button immediately clears all of the other buttons in the group. Radio buttons return an integer value. The value of a radio button control corresponds to the selected radio button.

For example, if Radio 1 option is selected, the current value is 1. If the Radio 4 option is selected, the value is 4. You can only assign **Integer** type tagnames to a radio button control wizard.

Radio (option) button control wizard QuickScript example:

```
SelectedMachine=1;
```

Where: *SelectedMachine* is an integer tagname assigned to the radio button wizard with the control name "RadioButtonGroup1. This is a Window **"On Show"** QuickScript that sets the value of the *SelectedMachine* tagname to 1. (This sets the default to select the first radio button in the group when the window is initially shown.) When the operator selects another radio button, the value of *SelectedMachine* will change according to the button selected. For example, if the radio button group had 4 choices and the operator selected the third button, *SelectedMachine*'s value would be set to 3.

Configuring a Windows Control Wizard

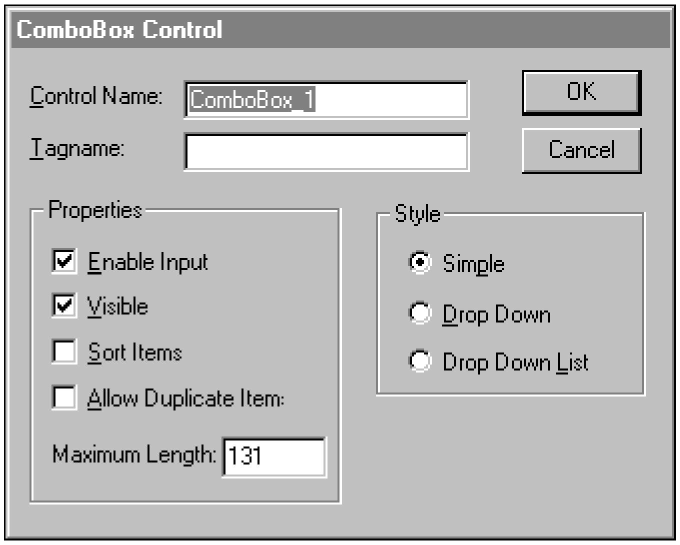
Each windows control wizard has its own unique configuration dialog box based on its intended functionality. The options shown in the dialog box are configurable properties not available through other WindowMaker tools. However, properties such as color, font style and size are modified by using the respective WindowMaker tools.

Most of the windows control wizards support QuickScript functions. For example, you could create a Data Change QuickScript to load and clear the lists, add and delete items in the lists, and so on.

📖 For more information on the windows control QuickScript functions and **.fields** see, your *InTouch Reference Guide*.

➢ **To configure a windows control wizard:**

1. Paste the wizard in your window.
2. Double-click it. Its respective configuration dialog box will appear. For example:



ComboBox Control

Control Name: ComboBox_1

Tagname:

Properties
- [✓] Enable Input
- [✓] Visible
- [ ] Sort Items
- [ ] Allow Duplicate Item:

Maximum Length: 131

Style
- [● Simple]
- ○ Drop Down
- ○ Drop Down List

   🖱 If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

3. In the **Control Name** box, type a unique name to identify the windows control.

   🖱 You must specify a unique **Control Name** for each windows control wizard you use for correct operation at runtime. WindowViewer uses the **Control Name** to identify the control when you execute a windows control QuickScript function. **Control Names** must start with an alpha character (underscores and numbers can be used after the first alpha character) and cannot include any special characters.

4. Specifying a **Tagname** when you configure the windows control is optional. However, if you do specify a tagname, its value is automatically set to the **.Value** property of the control (that is, the item index for the item selected in a list box).
5. Type in all other required entries and set all parameters for the particular windows control that you are configuring, as applicable.
6. Click **OK**.

Windows Control Wizard Properties

Windows control wizards have properties like InTouch tagname **.fields**. They can be read-write or read-only. Some properties are accessible at development and some at runtime. They are identified as *ControlName.x,* where *x* is the property.

For example, if the **.Visible** property of a windows control is equal to 0, the control will not be visible in the window. Similar to InTouch tagnames, **.Value** is the default property for the windows control wizard.

In WindowMaker, windows control wizard properties such as text font, size and color are modified using the respective WindowMaker toolbars or menu commands. The properties that are not supported by the toolbar or menu commands, are configured from within the wizard's configuration dialog box. Other properties of windows control wizards are dynamic and are read-write or read-only in runtime. This is similar to runtime properties of InTouch tagnames (**.fields**) such as **.Value** and **.Name**. Unlike InTouch tagnames, runtime properties for windows control wizards are accessed through QuickScript functions not animation link expressions.

The runtime properties may be either read-write or read-only depending on the property. The **GetProperty()** and **SetProperty()** QuickScript functions must be used to control or retrieve these properties. The following briefly describes each windows control property:

Property	Description
.Caption	Determines the "message" to be displayed with the check box.
.Enabled	Determines whether the control object can respond to operator-generated events.
.ListCount	Determines the number of items in a list box or combo box.
.ListIndex	Determines the corresponding index (*tagname* or *number*) of the currently selected item in the list. **Note** Index is a number that defines a specific item in a list. When using a list box, an index of -1 indicates that no item is currently selected. When using a combo box, an index of -1 indicates that the user has entered new text into the text entry field of the control.
.NewIndex	Returns the corresponding integer index (*tagname*) of the last item added to the list box or combo box through the **wcAddItem()** or **wcInsertItem()** functions.
.ReadOnly	Determines whether the contents of the text box are read-only or read-write.
.TopIndex	Determines the corresponding integer index of the top most item in the list box.
.Value	The default property for all InTouch windows control wizards. Changes made to this property are synchronized in the InTouch tagname and the windows control wizards.
.Visible	Determines whether the windows control is visible in the window.

Note The windows control wizard properties do not appear in the **Choose field name** dialog box.

For example:

```
[ErrorNumber=]GetPropertyM("ControlName.Property",Tagname);
```

Where:

Parameter	Description
ControlName	The **Control Name** configured for the windows control wizard, for example, CheckBox_1 or the name of an alarm object, for example, AlmObj_1.
.Property	Windows control or alarm object property.
Tagname	A valid InTouch tagname (of the same type to be returned) that will hold the property value when the function is processed.

For more information on windows control wizards, see "InTouch Windows Control Wizards. "

Windows Control Wizard Functions

The following briefly describes the InTouch QuickScript functions that are available for use with the InTouch Windows Control wizards:

Function	Description
wcAddItem	Adds the supplied string to the list box or combo box.
wcClear	Removes all items from the list box or combo box.
wcDeleteItem	Deletes the item associated with the item index argument in both list or combo boxes.
wcDeleteSelection	Deletes the currently selected item from the list. Applies to list boxes and combo boxes
wcErrorMessage	Given an error number, **wcErrorMessage()**, returns a string message describing the error. Applies to list boxes, text boxes, combo boxes, radio buttons and check boxes.
wcFindItem	Determines the corresponding index of the first item in the list box or combo box that matches the supplied string.
wcGetItem	Returns the value property of an item string associated with a corresponding index in a list box or combo box.
wcGetItemData	Retrieves the integer value associated with a list item in a list box or combo boxes.
wcInsertItem	Inserts a string into a list box or combo box.
wcLoadlist	Replaces the contents of the list box or combo box with new items.
wcLoadText	Replaces the contents of the text box with a new string.
wcSavelist	Replaces the contents of a filename with the items in a list object.
wcSaveText	Saves the text contained in a text box to a filename.
wcSetItemData	Assigns an integer value to an item in a list box.

Working with ActiveX Controls

ActiveX controls, originally known as OLE controls or OCXs, are standalone software components that perform specific functions in a standard way. ActiveX controls define standard interfaces for reusable components. ActiveX controls are not separate applications. Instead, they are servers that are placed into a control container. To use ActiveX controls, they must be placed in an ActiveX container. InTouch is an ActiveX container. Microsoft VisualBasic and internet browsers are also ActiveX containers.

ActiveX controls behave exactly like InTouch Wizards, except they bring compelling new functionality to InTouch applications. You can create ActiveX controls by using Visual Basic, Microsoft VC++ or other 3rd party development tools. You can also buy ActiveX controls from third-parties for specific functionality. These controls are packaged in the OCX form. Wonderware's FactorySuite InTrack component also provides you with several ActiveX controls. Additionally, IndustrialSQL's ActiveTrend allows you to run the IndustrialSQL Trend program (or a functional subset) from within InTouch and ActiveEvent allows you to notify the IndustrialSQL Event sub-system when an event has occurred in another application.

There are three main components of ActiveX controls: *properties*, *methods* and *events*. Properties are very similar to variables that you can modify, for example, Calendar.day, Control.height, and so on. Methods are similar to script function calls that you can call from the container.

For example, **Browser.Navigate("http://www.wonderware.com")**, **Engine.start()**. Events occur through the ActiveX container. For example, **Control.click (shift)**. **FileViewer.DoubleClick (name)**, and so on.

InTouch allows you to access ActiveX control properties, methods and events. You can associate these properties with InTouch tagnames or you can access them through InTouch QuickScripting.

Note In order for an ActiveX Event script to function properly, the ActiveX control for which the script was created, must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

You can use one or more ActiveX controls in your InTouch application. InTouch allows you to easily select and paste an ActiveX control into any application window and to add them to your **Wizards/ActiveX Toolbar**. You can also import ActiveX Event scripts from one application to another.

➢ **To use an ActiveX control in InTouch:**

1. Install the ActiveX control(s) that you want to use.
2. Select and paste the ActiveX control into a WindowMaker window.
3. Configure the ActiveX control's properties and assign them to tagnames.
4. Associate ActiveX events to ActiveX Event scripts.
5. Call ActiveX methods and set ActiveX control properties in ActiveX Event scripts, or other InTouch QuickScripts.

The following WindowMaker edits can be made to an ActiveX control:

- An ActiveX control's size can be changed, if sizing is supported by the control.
- ActiveX controls can be duplicated, cut, copied, pasted and deleted.
- All aligning commands (left, right, top, bottom, centerpoint) can be applied to an ActiveX control.
- ActiveX controls can be added to the **Wizards/ActiveX Toolbar**.
- ActiveX controls can be included with other objects when creating a cell.
- The WindowMaker menu commands and their equivalent toolbar tools can be used to directly modify many ActiveX properties. For example, Reduce Font, Line Color, Fill Color, and so on.

InTouch does not support the following types of ActiveX controls:

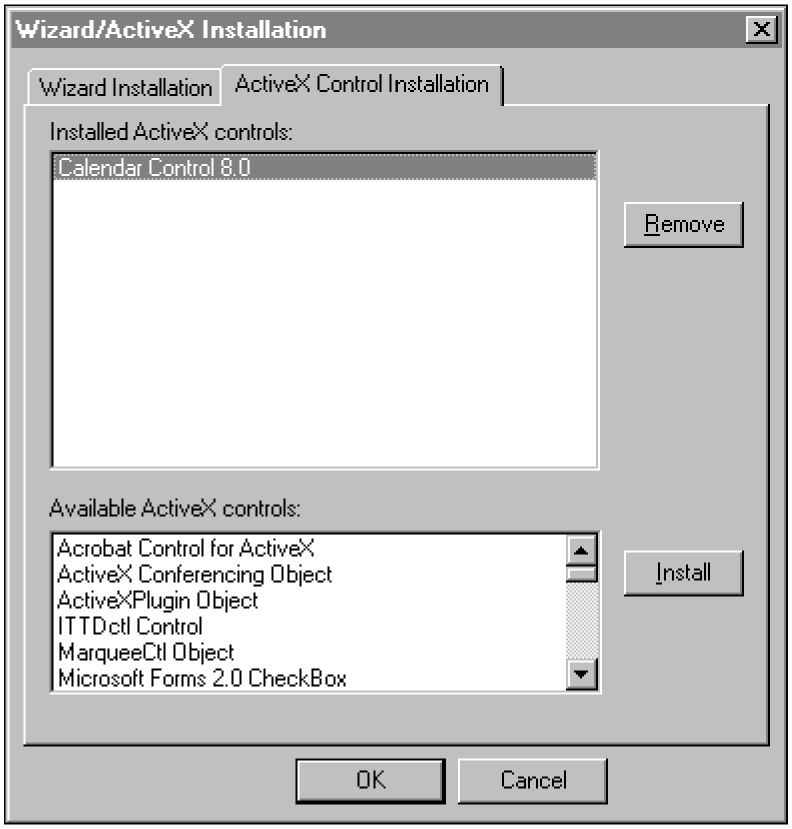
- Windowless Controls
- Simple Frame Site (Group Box)
- Containers
- Data Controls
- Dispatch Objects
- Arrays, Blobs, Objects, Variant Types

➢ **To install or remove an ActiveX control:**

1. On the **Special** menu, point to **Configure**, and then click **Wizard/ActiveX Installation**, or in the Application Explorer, double-click **Wizard/ActiveX Installation**. The **Wizard/ActiveX Installation** dialog box appears.

   In the Application Explorer, you can also right-click **Wizard/ActiveX Installation**, and then click **Open**.

2. Click the **ActiveX Control Installation** tab to activate the **ActiveX Installation** property sheet:


Wizard/ActiveX Installation
Wizard Installation
ActiveX Control Installation
Installed ActiveX controls:
Calendar Control 8.0
Remove
Available ActiveX controls:
Acrobat Control for ActiveX
ActiveX Conferencing Object
ActiveXPlugin Object
ITTDctl Control
MarqueeCtl Object
Microsoft Forms 2.0 CheckBox
Install
OK
Cancel

3. In the **Installed ActiveX controls** list, select the control(s) that you want to remove from your application, and then click **Remove**. An interactive message box will appear asking you to confirm the deletion.

   To select a group of controls, click your first selection, hold down the SHIFT key and select your last selection. All controls in between will be selected as well. To select multiple controls that are not consecutively listed, click the first control, and then hold down the CTRL key as you click another.

4. Click **Yes** to remove the control(s). The removed control(s) is moved to the **Available ActiveX controls** list.

   When you remove a control, it is not deleted. However it is no longer loaded into memory. Therefore, it will not function properly.

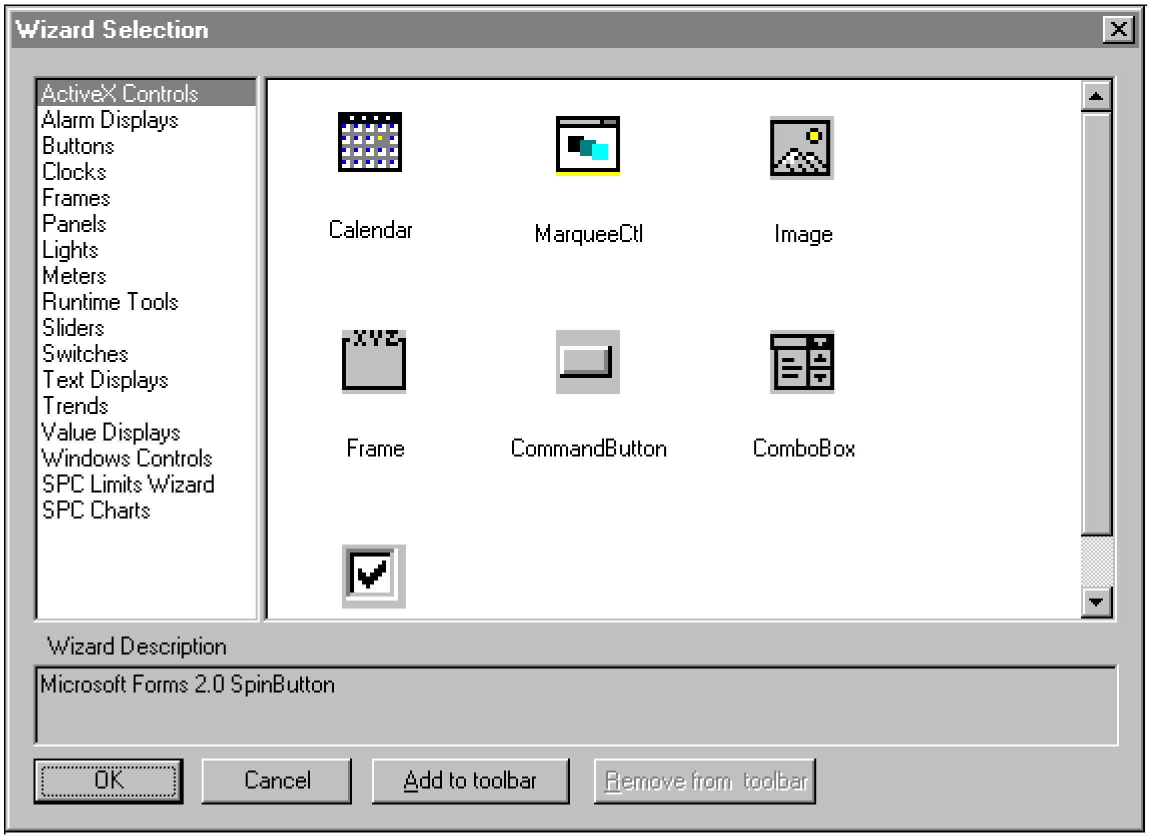
5. To install ActiveX controls, select them in the **Available ActiveX controls** list, and then click **Install**.

   The **Install** button is active only when controls are displayed in **Available ActiveX controls** list.

6. Click **Close**.

➢ **To place an ActiveX control in a window:**

1. Click the Wizard Dialog tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box appears:



Microsoft Forms 2.0 SpinButton

2. In the list of wizards, click the **ActiveX Controls** category. All available ActiveX controls will be shown the display area.

3. Select the ActiveX control that you want to use, and then click **OK**, or double-click the control. The dialog box will close and your window will reappear.

   To add the ActiveX control to the **Wizards/ActiveX Toolbar**, click **Add to toolbar**. Once you add a control to the **Wizards/ActiveX Toolbar**, you can select it and paste it into your open window at any time.

   Note The number of ActiveX controls that you can add to the toolbar is limited to your system resources.

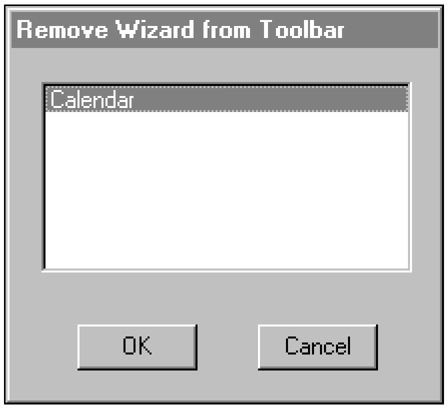
4. The cursor will change to the corner symbol, ⌜W , when you return to the window. Click the location in the window where you want to paste the ActiveX control.

5. Double-click the control to configure it properties.

   For more information on the WindowMaker toolbars, see Chapter 1 - WindowMaker Program Elements.

➢ **To remove ActiveX controls from the toolbar:**

1. Click the Wizard Dialog tool in the **Wizards/ActiveX Toolbar**. The **Wizard Selection** dialog box will appear.
2. Click **Remove from toolbar**. The **Remove Wizard from Toolbar** dialog box appears:



3. Select the ActiveX control(s) that you want to remove from the toolbar.
4. Click **OK**.

Configuring an ActiveX Control

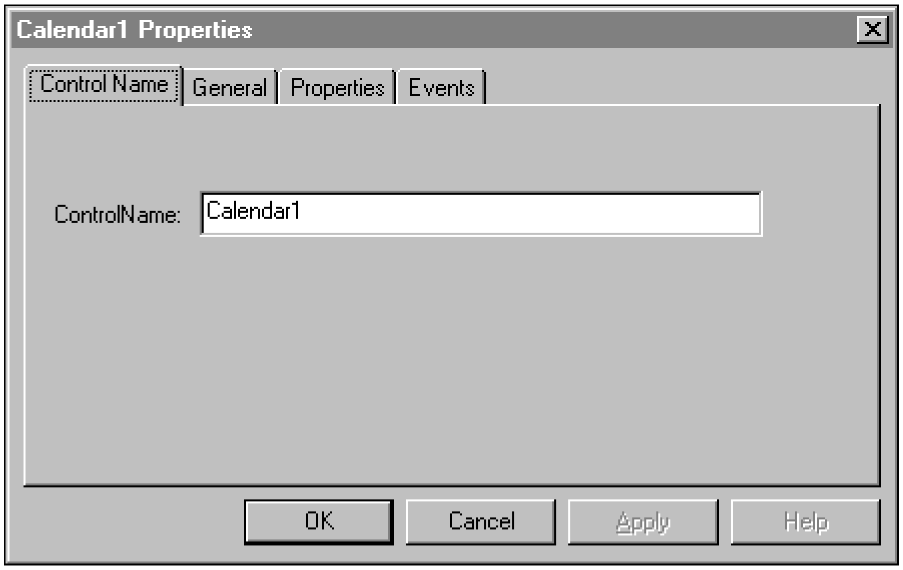
When you paste an ActiveX control into an InTouch window, you must configure its properties to interact with InTouch. Each control must be named for reference from InTouch QuickScripts. A default control name such as, Calendar1, will be generated when you paste the ActiveX control. (This control name will be global within your InTouch application.)

The ActiveX control's properties must be assigned to InTouch tagnames. Each property type must be assigned to an equivalent InTouch tagname type.

➢ **To name an ActiveX control:**

1. Paste the ActiveX control into your WindowMaker window.
2. Double-click the control, or right-click the control, and then click **Properties**. The control's respective **Properties** dialog box will appear.

Note Each ActiveX control's **Properties** dialog box is unique to the control. The number of tabs displayed is based upon the properties of the particular control. Some ActiveX controls may require you to configure more properties than others do. For example some controls may require you to configure their **Colors** and **Fonts**, while others may not have these properties. However, for all ActiveX controls, InTouch adds three tabs; **Control Name, Properties** and **Events**. For example:


Calendar1 Properties

ControlName: Calendar1

3. Click the **Control Name** tab, and then type a unique name for the ActiveX control in the **ControlName** box.

   You must define a unique name for each ActiveX control used in your InTouch application. The Control Name is used in script functions to identify the control. For example:

   ```
   #Calendar1.day = Tag1;
   #Calendar1.year = 1997;
   ```

Note If you use the default Control Name, for example, Calendar1, and you subsequently duplicate the ActiveX control, InTouch will automatically increment the Control Name. In this case, the duplicate ActiveX control's name would be Calendar2.

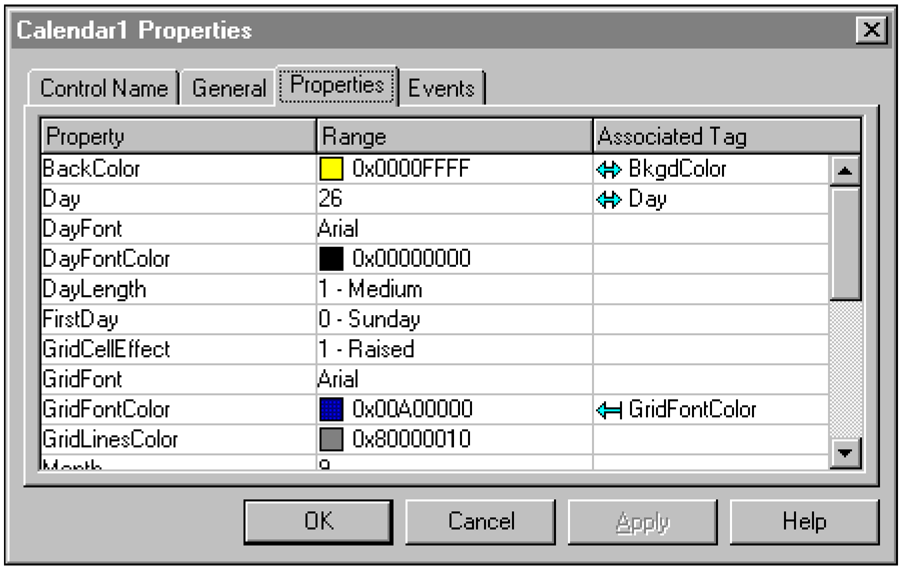
Configuring ActiveX Control Properties

The properties that you can configure for a particular ActiveX control are determined by the ActiveX control designer. Each ActiveX control's **Properties** property sheet displays three columns: **Property**, **Range** and **Associated Tag**. The **Property** and **Range** columns are read-only. The **Associated Tag** column is used to associate InTouch tagnames with the respective property in the **Property** column.

Note When you click certain items in the **Range** column an arrow will appear that you can click to view the list of possible values for the item. The items in the list are for viewing purposes only and cannot be changed.

➢ **To configure an ActiveX control's properties:**

1. Click the **Properties** tab in the ActiveX control's **Properties** dialog box to activate the **Properties** property sheet:



Property	Range	Associated Tag
BackColor	🟨 0x0000FFFF	⇔ BkgdColor
Day	26	⇔ Day
DayFont	Arial	
DayFontColor	⬛ 0x00000000	
DayLength	1 - Medium	
FirstDay	0 - Sunday	
GridCellEffect	1 - Raised	
GridFont	Arial	
GridFontColor	🟦 0x00A00000	⇐ GridFontColor
GridLinesColor	▨ 0x80000010	

2. Click in the middle of each cell in the **Associated Tag** column, and then type a tagname for the respective property.

   If you type in a tag name that is not defined in the Tagname Dictionary, you will be prompted to define it now.

   If you double-click a blank cell, or click the ... button, the Tag Browser will appear displaying the tagnames for the selected tag source. Double-click the tagname that you want to use, or select it, and then click **OK**. The tagname is automatically inserted into the cell.

   For more information on the Tag Browser, see Chapter 4 - Tagname Dictionary.

3. Once you specify the tagname, double-click in the cell to the left of the tagname to select the association direction for the tagname to its respective property. (Continuously double-clicking will cycle you through the various association direction choices. The association direction choices are described below.)

   There are actually two fields in each cell in the **Associated Tag** column. The association direction selection and the tagname entry. The ActiveX control determines the association direction and the property type determines the tagname type that must be used.

You can select one directional or bi-directional association. However, if the association direction you select is not valid for the property or tagname, the control will automatically change it accordingly. For example, if you select ⇐, when the tagname's value changes, its associated property is changed accordingly. A certain subset of the symbols below will appear based upon the potential relationship between the property and the tagname.

For example, if you associate a property to a writeable tagname, you will see a different set of associations than if you associate the same property to a read-only tagname. Select the appropriate association symbol as follows:

⇐	The tagname sets the value of the associated property.
⊢	This symbol indicates that the property is read-only and the tagname cannot change the property's value.
⇒	The property sets the value of the associated tagname.
⊣	This symbol indicates that the tagname is read-only and the property cannot change the tagname's value.
⇔	Value can be set from both the tagname or the property. (Tagname takes precedence.)
⊢⊣	The tagname and the property are both read-only.
⇐⊣	The tagname can change the property's value, but the property cannot change the tagname's value. The property cannot change the tagname's value because the property is non-bindable, or the tagname is read-only.
⊢⇒	The property can change the tagname's value, but the tagname cannot change the property's value. The tagname cannot change the property's value because the property is read-only.

4. Click **OK**.

Note You can also access or change properties through ActiveX Event scripts and/or other InTouch QuickScripts. All ActiveX script functions are qualified by the # (pound) sign. The valid syntax to access ActiveX properties is:

```
#ControlName.PropertyName
```

Examples:

```
#Calendar1.Day = 29;
Tag1 = #Calendar1.year;
```

For more information, see, "Configuring an ActiveX Control."

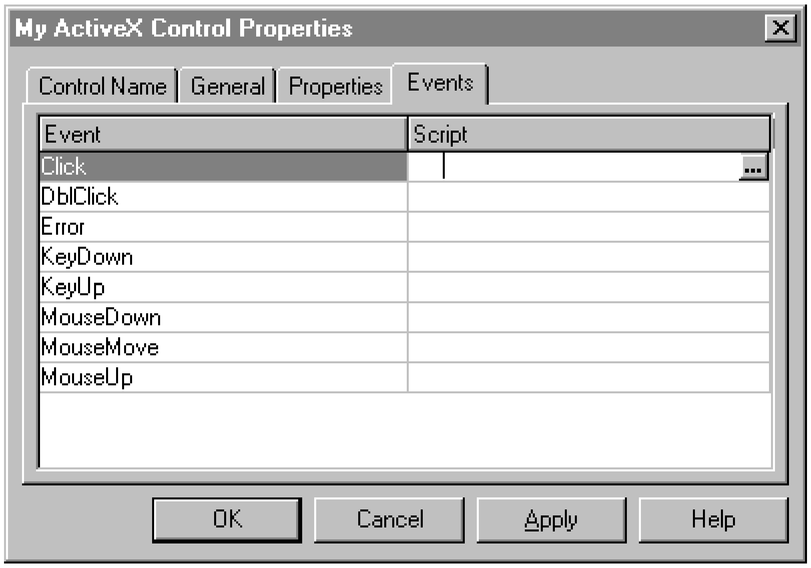
Using ActiveX Control Methods

ActiveX control methods are similar to ActiveX control properties. You activate methods in runtime (WindowViewer). ActiveX control methods are accessed through ActiveX Event scripts and/or other InTouch QuickScripts.

Note In order for an ActiveX Event script to function properly the ActiveX control for which the script was created, must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

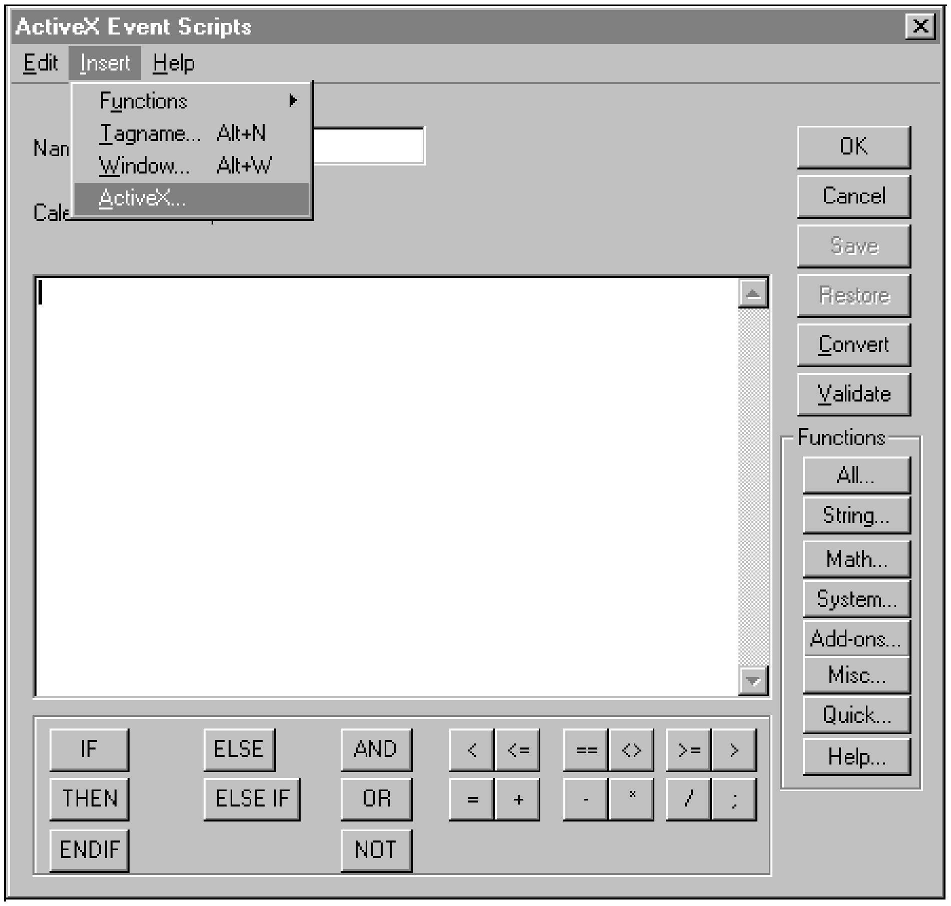
➢ **To use ActiveX methods and/or properties:**

1. In the ActiveX control's **Properties** dialog box, click the **Events** tab to activate the **Events** property sheet:

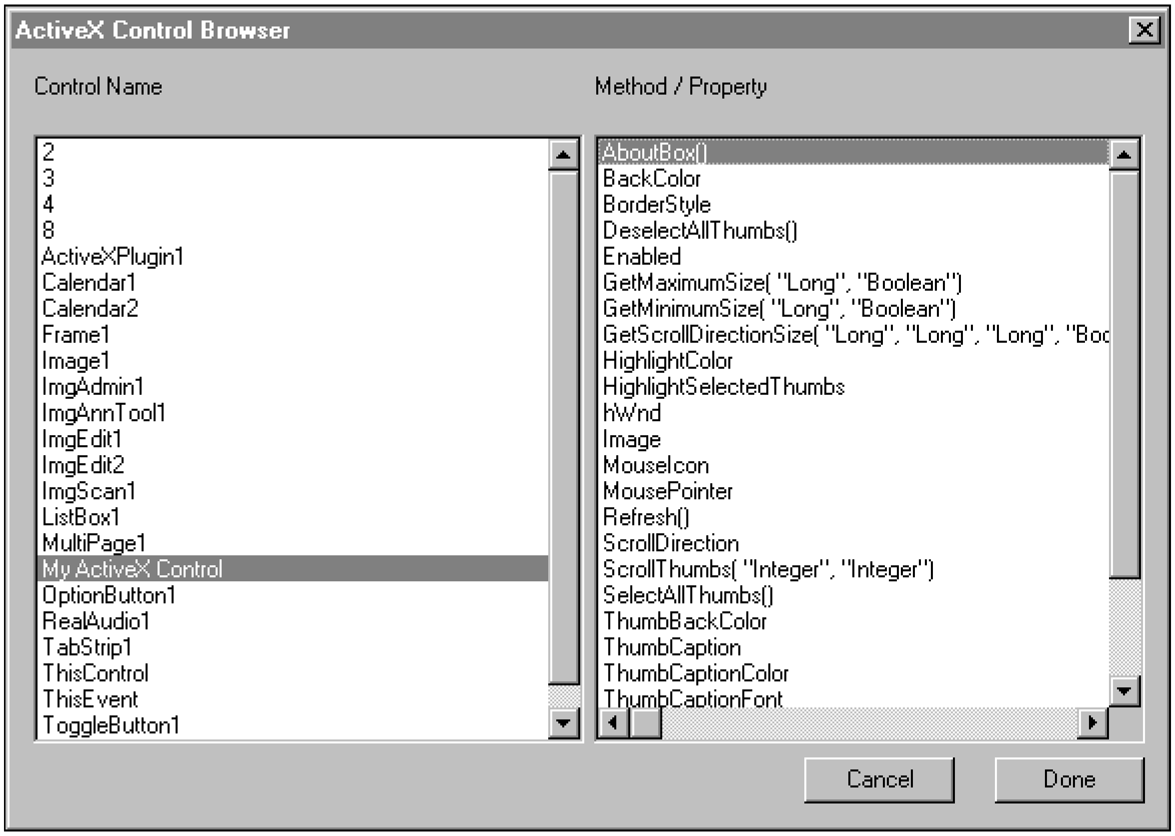


Event	Script
Click	
DblClick	
Error	
KeyDown	
KeyUp	
MouseDown	
MouseMove	
MouseUp	

2. Double-click a blank cell in the **Script** column. The **ActiveX Event Scripts** editor appears:



3. On the **Insert** menu, click **ActiveX**. The **ActiveX Control Browser** appears:



Control Name	Method / Property
2	AboutBox()
3	BackColor
4	BorderStyle
8	DeselectAllThumbs()
ActiveXPlugin1	Enabled
Calendar1	GetMaximumSize("Long", "Boolean")
Calendar2	GetMinimumSize("Long", "Boolean")
Frame1	GetScrollDirectionSize("Long", "Long", "Long", "Boo
Image1	HighlightColor
ImgAdmin1	HighlightSelectedThumbs
ImgAnnTool1	hWnd
ImgEdit1	Image
ImgEdit2	MouseIcon
ImgScan1	MousePointer
ListBox1	Refresh()
MultiPage1	ScrollDirection
My ActiveX Control	ScrollThumbs("Integer", "Integer")
OptionButton1	SelectAllThumbs()
RealAudio1	ThumbBackColor
TabStrip1	ThumbCaption
ThisControl	ThumbCaptionColor
ThisEvent	ThumbCaptionFont
ToggleButton1	

4. In the **Control Name** list, select the ActiveX control whose methods or properties you want to access.

   🖰 The names of all ActiveX controls currently being used in your application will be listed.

Note If you select **This Control** instead of the actual **Control Name**, the methods and properties displayed will be those for the ActiveX control currently selected in your application. By selecting **This Control** instead of the actual **Control Name**, you can create generic ActiveX Event script functions. You can then copy and paste these functions into any other ActiveX Event script, or any other InTouch QuickScript without having to change the **Control Name** in the new script. For example:

```
#ThisControl.Navigate  (“http:\\www.wonderware.com”);
#ThisControl.Navigate(URL);   { where URL is a tagname}
```

This Control is accessible <u>only</u> through ActiveX Event scripts. It is <u>not</u> accessible through any other type of InTouch QuickScript.

5. In the **Method / Property** list, select the method or property that you want to use in your script.

   🖰 Properties are the items in the list that include parenthesis. For example, **Display()**.

6. Click **Done**. The selected control name and method or property are automatically inserted into your script.

   🖰 ActiveX control's methods and properties are also accessed through the **Insert** menu in all other InTouch QuickScripts types.

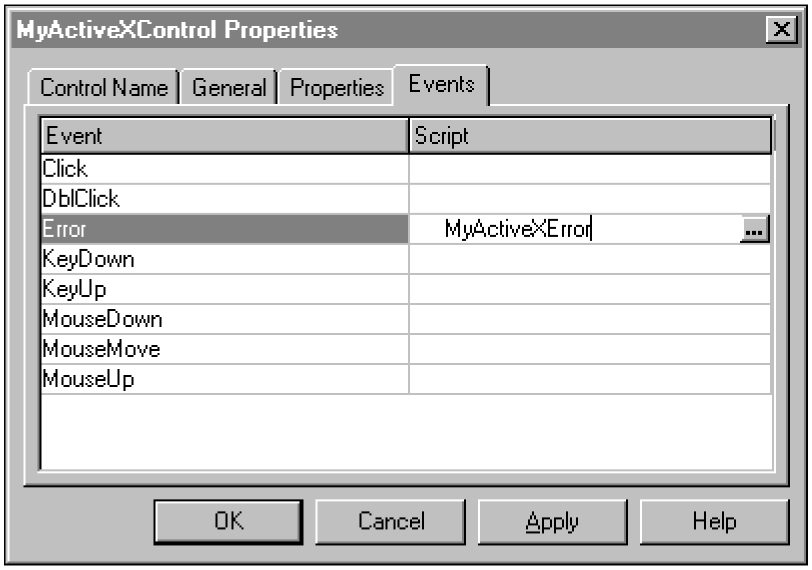
Using ActiveX Control Event Parameters

You can execute ActiveX control events in runtime (WindowViewer) by designing a particular action and associating it to the event. For example, if your ActiveX control has an error event handler, you could create a ActiveX Event script that displays a window with an error message when an error occurs or any other InTouch QuickScript. ActiveX Event scripts are provided to support event actions. You can associate a named event script to each event.

Note In order for an ActiveX Event script to function properly the ActiveX control for which the script was created, must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

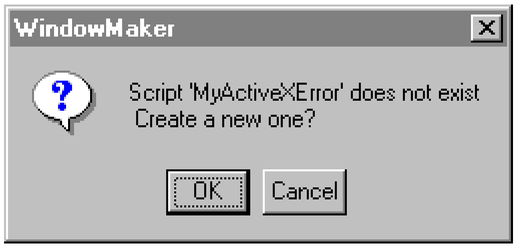
➢ **To use ActiveX Event parameters:**

1. Double-click the ActiveX control for which you want to create an ActiveX Event script. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Events** tab to activate the **Events** property sheet:



Event	Script
Click	
DblClick	
Error	MyActiveXError
KeyDown	
KeyUp	
MouseDown	
MouseMove	
MouseUp	

3. In the **Event** column select the event to which you want to associate an ActiveX Event Script.
4. In the respective cell in the **Script** column, type a unique name for the ActiveX Event Script that you want to create and then double-click the name, or click **OK**. The following message box appears:

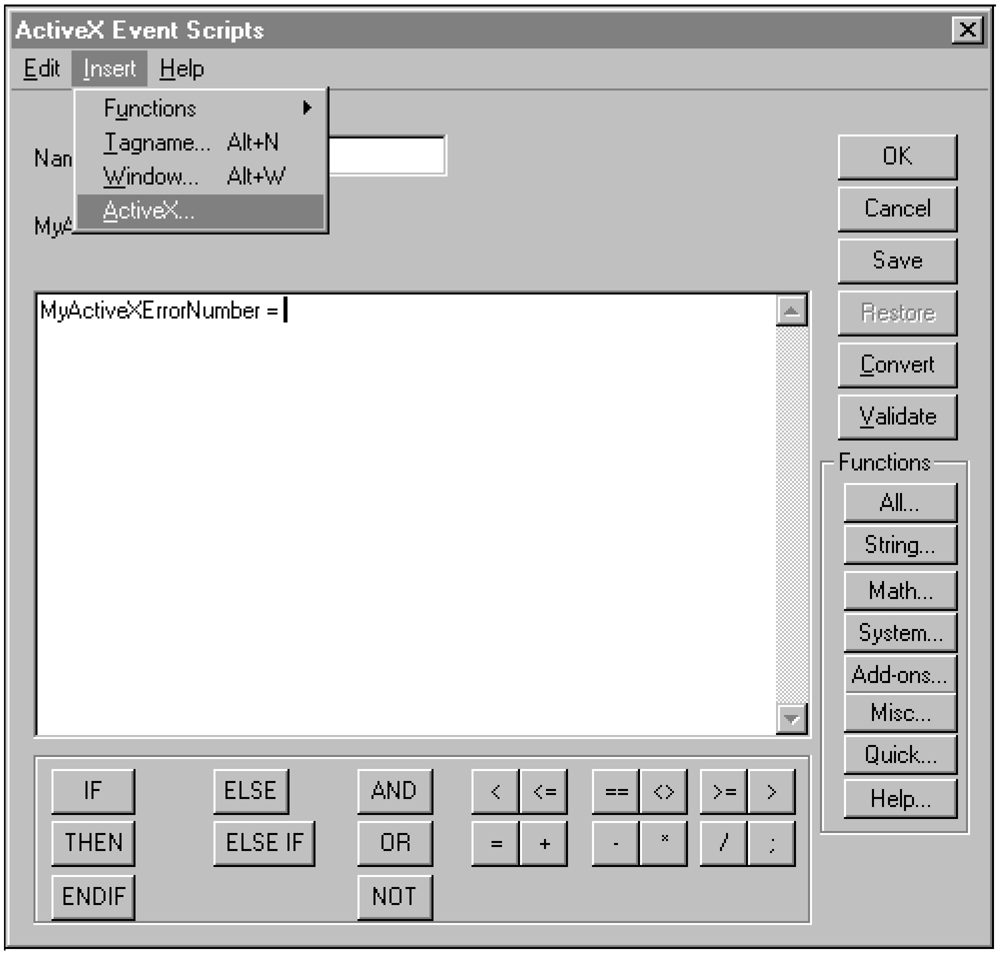


Script 'MyActiveXError' does not exist
Create a new one?

Click **OK**. The ActiveX Event script editor will appear displaying the name that you typed in the **Name** input box (see example below). If you double-click a blank **Script** cell, when the ActiveX Event script editor appears, you must then type a name for the ActiveX Event script.

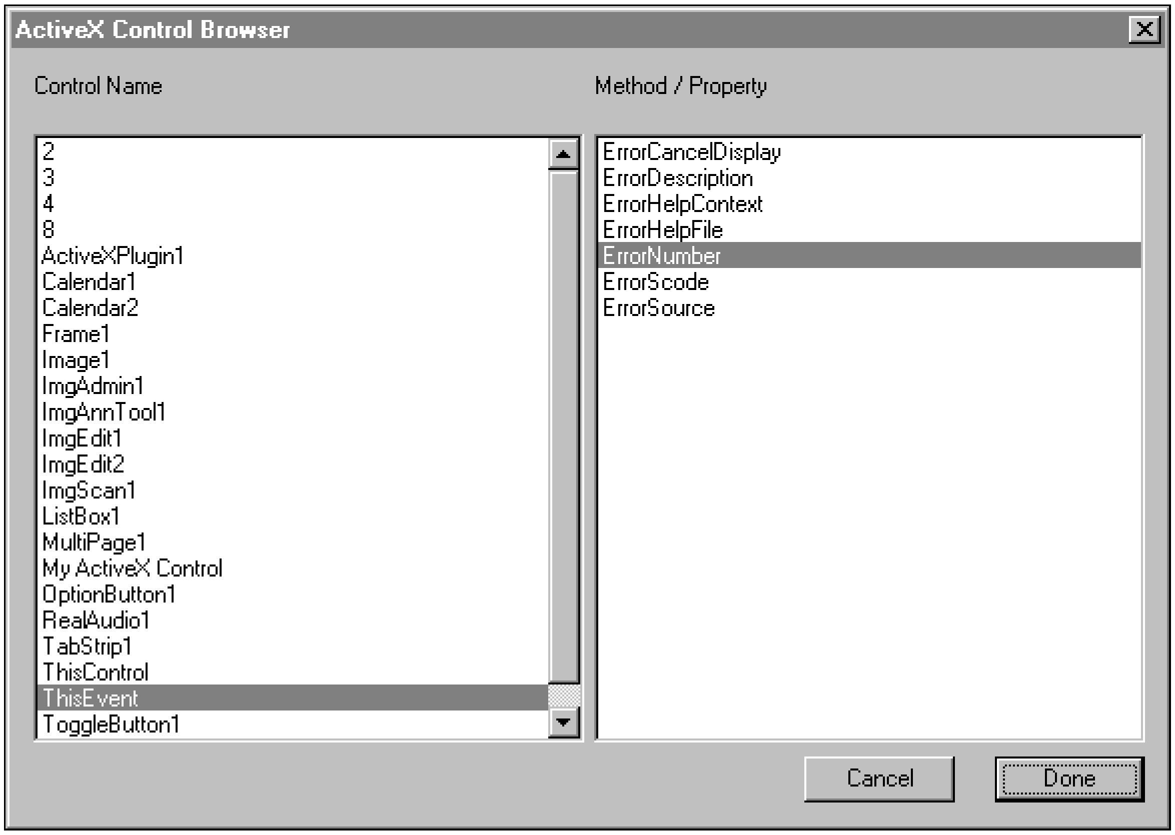
If the ActiveX Event script that you want to use already exists, click the ... button. The **Choose ActiveX Script** dialog box will appear listing all existing ActiveX Event scripts in your application.

For more information, see "Reusing ActiveX Event Scripts."



```
MyActiveXErrorNumber =
```

5. On the **Insert** menu, click **ActiveX**. The **ActiveX Control Browser** appears:



Control Name list:

2
3
4
8
ActiveXPlugin1
Calendar1
Calendar2
Frame1
Image1
ImgAdmin1
ImgAnnTool1
ImgEdit1
ImgEdit2
ImgScan1
ListBox1
MultiPage1
My ActiveX Control
OptionButton1
RealAudio1
TabStrip1
ThisControl
ThisEvent
ToggleButton1

Method / Property list:

ErrorCancelDisplay
ErrorDescription
ErrorHelpContext
ErrorHelpFile
ErrorNumber
ErrorScode
ErrorSource

6. In the **Control Name** list, select **This Event** to access the parameters for the selected event. In this case, the selected event is, **Error**.

Note This Event is accessible only through ActiveX Event scripts. It is not accessible through any other type of InTouch QuickScript. You must select **This Event** to access the event parameters for an ActiveX control.

Events may or may not pass parameters in runtime. Event parameters can be accessed by using the **ThisEvent** keyword. For example:

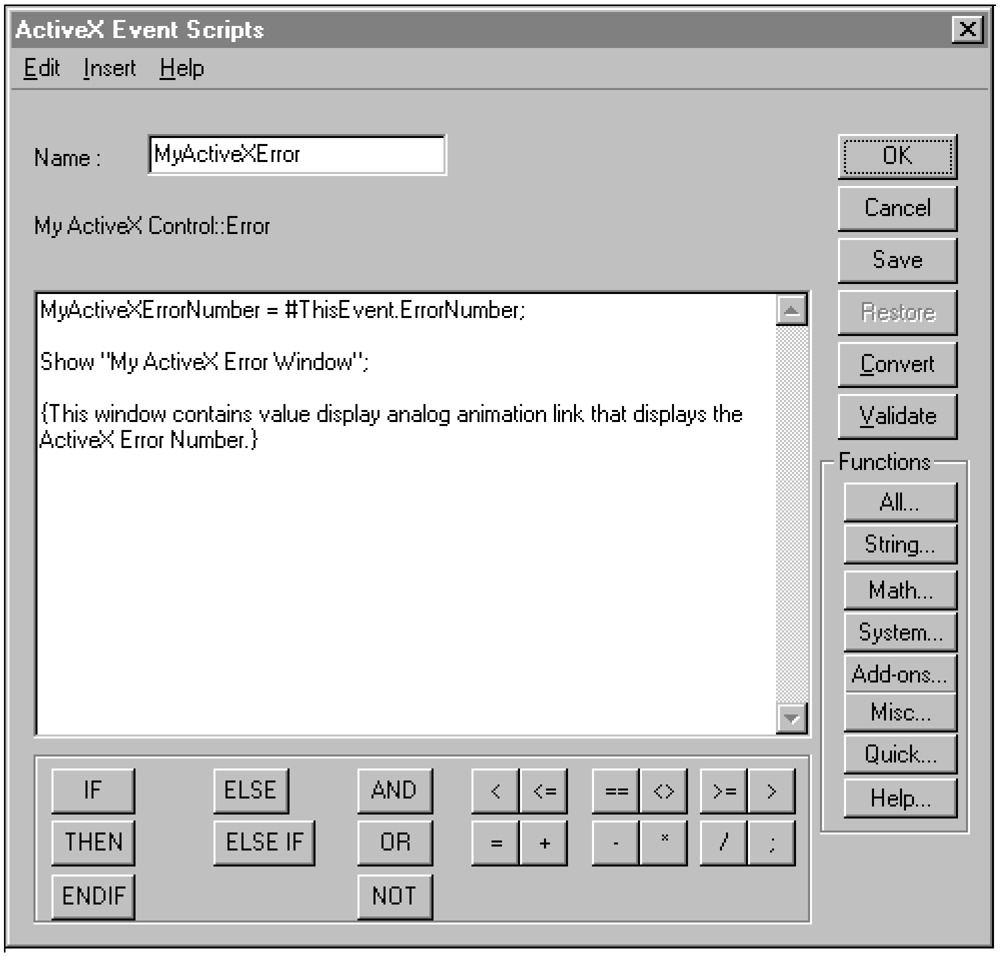
```
MyActiveXErrorNumber = #ThisEvent.ErrorNumber;
```

Where: **#** indicates that this is an ActiveX script function. **ThisEvent** relates to the event selected in the ActiveX control's **Event** property sheet, and **ErrorNumber** is the parameter passed by the selected event.

7. In the **Method / Property** list, select the event that you want to use in your ActiveX Event script.

8. Click **Done**.

The selected control name, in this case, **This Event**, and selected event parameter are both automatically inserted into your script at the cursor location. For example:



Name : MyActiveXError

My ActiveX Control::Error

```
MyActiveXErrorNumber = #ThisEvent.ErrorNumber;

Show "My ActiveX Error Window";

{This window contains value display analog animation link that displays the
ActiveX Error Number.}
```

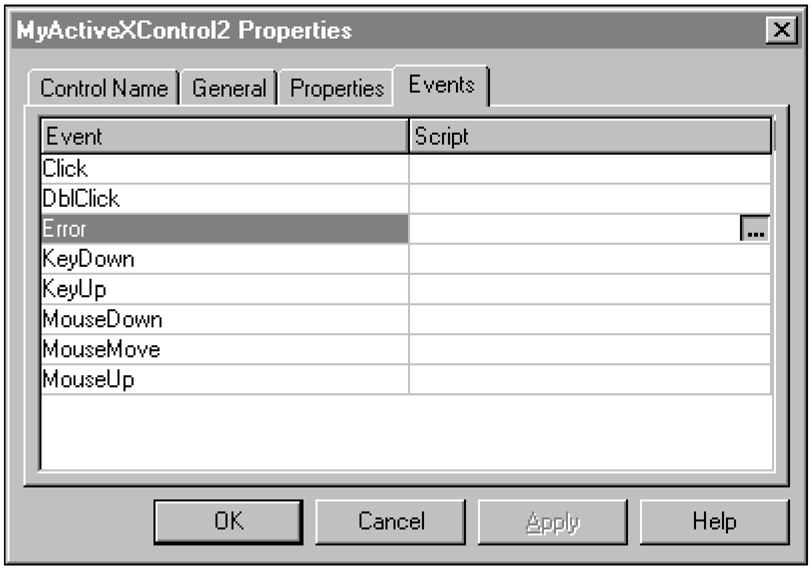
9. Click **OK** to save your ActiveX Event script and close the script editor. The ActiveX control's **Properties** dialog box reappears.
10. Click **OK** to close the **Properties** dialog box, or continue to create ActiveX Events scripts.

Reusing ActiveX Event Scripts

ActiveX Event scripts can only be reused for the same event for the same kind of ActiveX control. For example, the mouse down event may be a stock event on hundreds of ActiveX controls. However, an ActiveX Event script written for mouse down on ActiveX ControlA cannot be reused for mouse down on ActiveX ControlB unless the two controls are the same type.

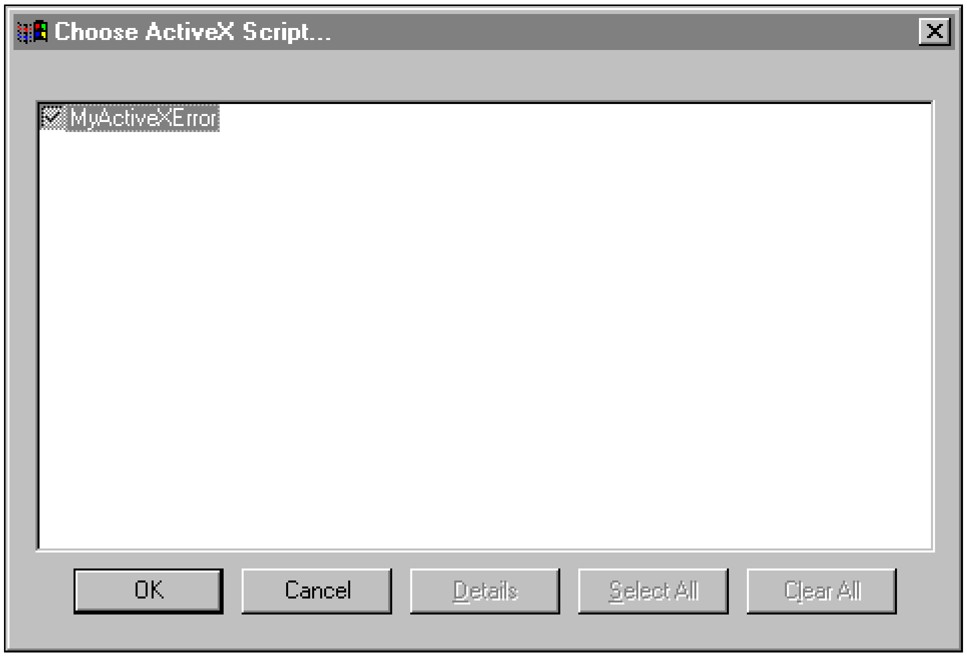
➢ **To reuse an ActiveX Event script:**

1. Double-click the ActiveX control for which you want to reuse an existing ActiveX Event script. The selected ActiveX control's **Properties** dialog box will appear.
2. Click the **Events** tab to activate the **Events** property sheet:



Event	Script
Click	
DblClick	
Error	...
KeyDown	
KeyUp	
MouseDown	
MouseMove	
MouseUp	

3. In the **Script** column for the respective event, click the ... button. The **Choose ActiveX Script** dialog box appears:



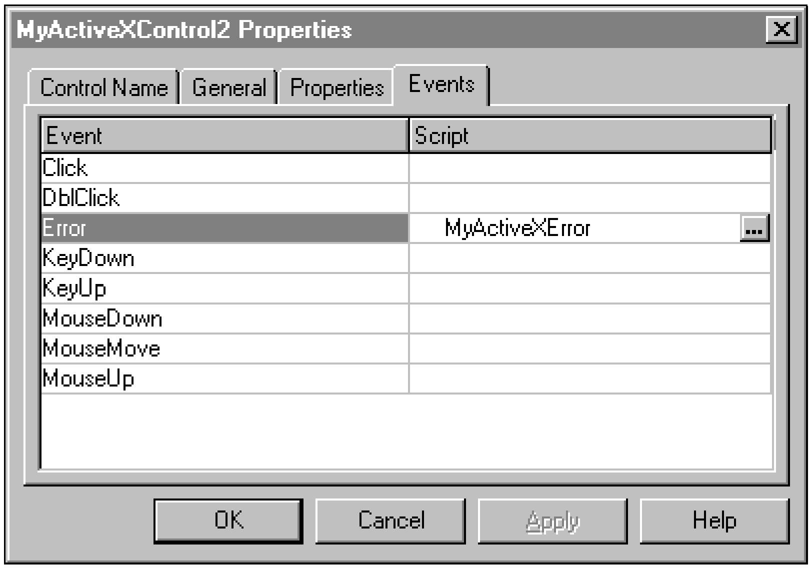
☑ MyActiveXError

Note This dialog box will only display the ActiveX Event scripts that were written for the same type of ActiveX control and the same selected event.

For example, let's assume that you are creating an ActiveX Event script for a second ActiveX Calendar control's "Click" event. You have already created two other ActiveX Event scripts named Click1 and Click2 in your application. Click1 was created for a different ActiveX Calendar control's "Click" event, and Click2 was created for an ActiveX InSQLTrend control's "Click" event. When you click the ... button and the **Choose ActiveX Script** dialog box appears, it will only display the Click1 script since was created for the same type of ActiveX control and the same event.

4. Select the ActiveX Event script that you want to use, and then click **OK**.

   The name of the selected script is automatically inserted into the **Script** cell where you previously clicked the ... button. For example:



Event	Script
Click	
DblClick	
Error	MyActiveXError
KeyDown	
KeyUp	
MouseDown	
MouseMove	
MouseUp	

5. Click **OK** to close the **Properties** dialog box, or continue to create ActiveX Events scripts.

Importing ActiveX Event Scripts

Importing ActiveX Event scripts from one InTouch application to your current application, can save you a considerable amount of development time. When you move ActiveX Event scripts from one InTouch application to another, you must use the **Import** command on the WindowMaker **File** menu.

Note When you import ActiveX Event scripts, from one application to another, all ActiveX Events scripts are imported. Additionally, in order for an imported ActiveX Event script to function properly in the new application, the same ActiveX control and the same event for which the script was originally created, must also be used in the new application, and it must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

For more information on importing scripts, see Chapter 6 - Creating InTouch QuickScripts.

Customizing Your Runtime Environment

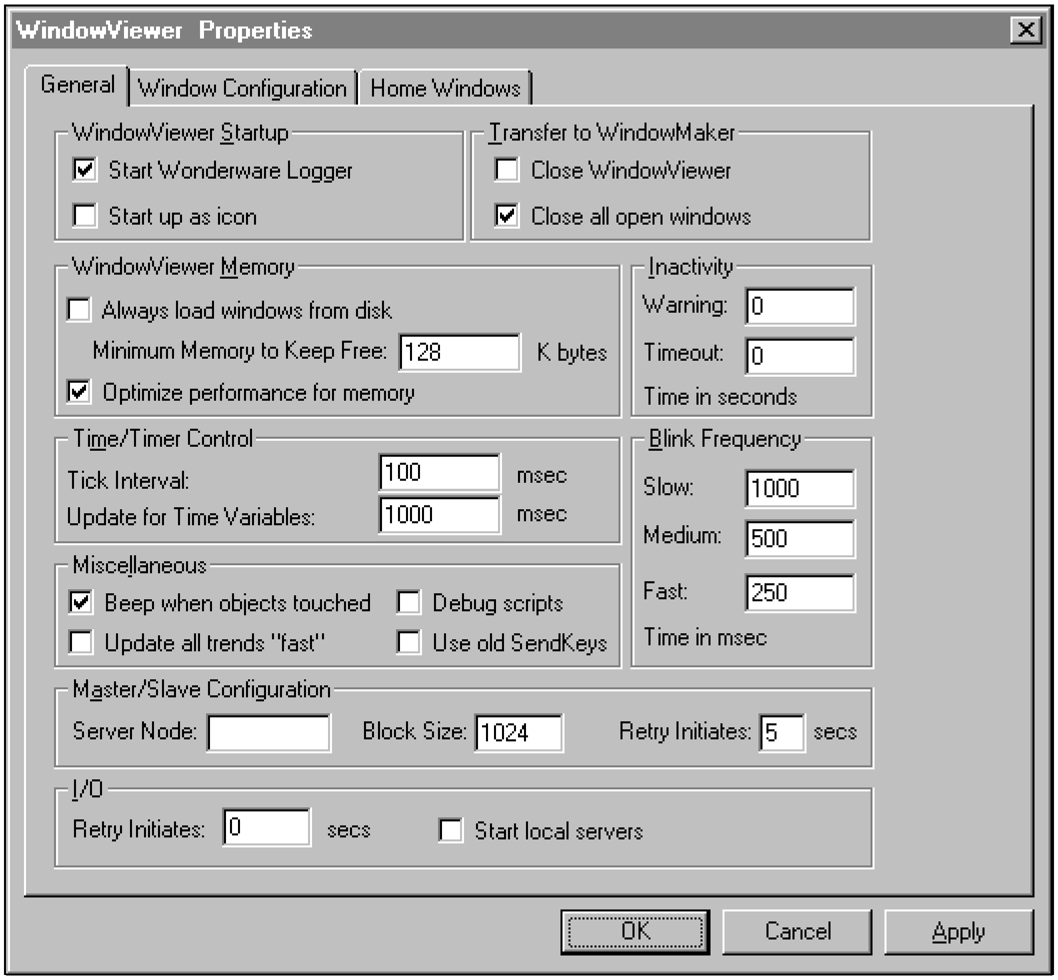
Like WindowMaker, there are many properties that you can set to customize your runtime environment (WindowViewer). For example, you can set the blinking speed for blinking objects, the system inactivity timeout and warning values, the windows that are automatically opened when WindowViewer is started from its icon or its menu command.

Setting WindowViewer's General Properties

➢ **To set the properties for WindowViewer:**

1. On the **Special** menu, point to **Configure**, and then click **WindowViewer**, or in the Application Explorer under **Configure**, double-click **WindowViewer**. The **WindowViewer Properties** dialog box appears with the **General** properties sheet active:

   In the Application Explorer, you can also right-click **WindowViewer**, and then click **Open**.



WindowViewer Startup
- [✓] Start Wonderware Logger
- [ ] Start up as icon

Transfer to WindowMaker
- [ ] Close WindowViewer
- [✓] Close all open windows

WindowViewer Memory
- [ ] Always load windows from disk
- Minimum Memory to Keep Free: 128 K bytes
- [✓] Optimize performance for memory

Inactivity
- Warning: 0
- Timeout: 0
- Time in seconds

Time/Timer Control
- Tick Interval: 100 msec
- Update for Time Variables: 1000 msec

Blink Frequency
- Slow: 1000
- Medium: 500
- Fast: 250
- Time in msec

Miscellaneous
- [✓] Beep when objects touched
- [ ] Debug scripts
- [ ] Update all trends "fast"
- [ ] Use old SendKeys

Master/Slave Configuration
- Server Node:
- Block Size: 1024
- Retry Initiates: 5 secs

I/O
- Retry Initiates: 0 secs
- [ ] Start local servers

   If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Select **Start Wonderware Logger**, if you want the FactorySuite Wonderware Logger program to automatically start when you start WindowViewer.

   The Wonderware Logger's behavior is a little different when you are on the Windows NT Operating System. For more information on the Wonderware Logger, see "Welcome to InTouch."

3. Select **Start up as icon** if you want WindowViewer to start up as an icon instead of a window.

   Select this option only when you are using WindowViewer to gather data for other I/O-interconnected applications.

4. Select **Close WindowViewer** if you want WindowViewer to automatically close when you start WindowMaker.

   If memory is not an issue, and you are using the fast switch to move between WindowViewer and WindowMaker, this option should not be selected.

   The fast switch option is selected in the **WindowMaker Properties - General** dialog box.

   When you select this option, the **Close on Transfer to WindowViewer** option located on the **WindowMaker Properties/General** property sheet is automatically selected too.

5. Select **Close all open windows** if you want all open windows to automatically be closed when you transfer from WindowViewer to WindowMaker.

   Selecting this option will free up memory on your system.

6. Select **Always Load Windows from Disk** if a low memory situation exists.

   Selecting this option causes your application windows to be loaded from disk and not saved in RAM memory when you close them.

7. In the **Minimum Memory to keep free** box, type the number of K bytes of memory that you want to keep free for other Windows applications.

8. Select **Optimize performance for memory** to significantly increase the drawing update speed. Selecting this option significantly increases the update rate for text fields.

   If your system is low on memory do not enable this option.

9. In the **Warning** box, type the number of seconds that can elapse with no operator activity (mouse clicks or keystrokes) before the system discrete tagname **$InactivityWarning** is set to 1 (True).

   You can use **$InactivityWarning** in a Condition QuickScript to show a window warning the operator that he/she is about to be logged off the system. If the operator clicks the mouse, presses a key, or performs an action using any other pointing device before the specified timeout elapses, they are not logged off. **$InactivityWarning** and the timer are reset.

10. In the **Timeout** box, type the number of seconds that can elapse with no operator activity (mouse clicks, keystrokes, and so on) before the system discrete tagname **$InactivityTimeout** is set to 1 (True). When **$InactivityTimeout** is true, the system equates the logged on operator name to the reserved name "None" and sets the security tagname, **$AccessLevel**, to 0.

    You can use **$InactivityTimeout** in a Condition QuickScript to show a window telling the operator that he/she has been logged off the application.

    You can use the **Timeout** feature independently of the **Warning** feature. However, the **Timeout** value must be greater than the **Warning** value for proper use of both system tagnames.

    For example, **Warning** becomes true after 30 seconds of inactivity and **Timeout** becomes true after an additional 15 seconds (for a total of 45 seconds) of inactivity.

11. In the **Tick Interval**box, type the speed interval that InTouch will use to check its internal timers.

This setting controls how fast an Application **While Running**, Window **While Showing**, Condition **While On True/On False**, Key and Touch Pushbutton Action **While Down** QuickScripts will be executed.

Note Scripts cannot execute faster than every 10 milliseconds on the Windows NT operating system or every 50 milliseconds on Windows 95.

For more information, see Chapter 6 - Creating QuickScripts in InTouch.

12. In the **Update Time Variables every** box, type the frequency (in milliseconds) that you want WindowViewer to update the time-based system tagnames such as $Msec, $Second, $Minute, and so on.

    We recommend that you use the default setting of 1000 milliseconds. However, you can type a zero to prevent updating of all time variables.

13. Select **Beep when objects touched** if you want all touch-sensitive objects to beep when selected in WindowViewer.

14. Select **Update all trends "Fast"** if you want your trend objects to be updated faster.

    Select this option only when you are absolutely certain that no objects are overlapping your runtime trend objects. If you select this option and any object is overlapping the trend object, it will not be drawn properly.

15. Select **Debug Scripts**if you want a message to be written to the Wonderware Logger program each time a QuickScript is executed.

    If you select the **Debug** menu option in the **WindowViewer Properties/Window Configuration** property sheet, you will be able to turn this command on and off in runtime from WindowViewer's **Special** menu.

16. Select **Use old SendKeys** only if you are using an international application that was developed using InTouch Version 3.26 or earlier. (This is a legacy option and it is not used for FactorySuite.)

17. In the **Slow, Medium, Fast** boxes type the speeds (in milliseconds) that you want to use for your blink animation links.

18. In the **Server Node** box, type the name of the alarm server node that you want remote alarm nodes to retrieve their alarm information. You cannot enter the local node name here. (This is a legacy option and it is not used for FactorySuite.)

    The node name used here is not used as part of the distributed alarm system.

    For additional information, refer to Chapter 3 - Building a Distributed Application.

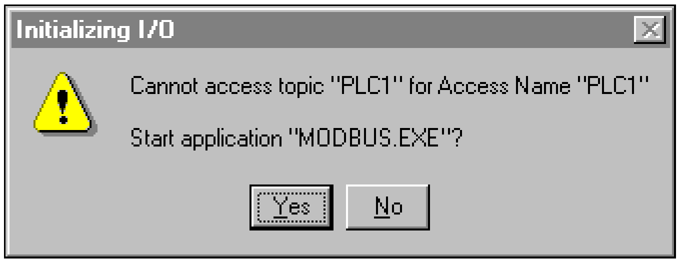
19. In the **Block Size** box, type the size that you want the blocks of data transferred between the alarm server node and its clients to be.

    When the InTouch network alarm server is communicating with other InTouch nodes, I/O Advise and Data Messages are packed into blocks to improve throughput.

20. In the **Retry Initiates** box, type the number of seconds that you want to elapse before the InTouch network alarm server attempts to establish and update I/O conversations to the client alarm nodes. (This is a legacy option and it is not used for FactorySuite.)

    This value is only used for data transfer between the alarm server node and its clients.

21. Select the **Start Local Servers** option if you want a dialog box to display whenever you start WindowViewer and the server you are trying to communicate with is not running. For example:

    Cannot access topic "PLC1" for Access Name "PLC1"

    Start application "MODBUS.EXE"?

    Click **Yes** to start the server or, click **No** to ignore the message and close the dialog box.

22. Click **OK** to save your property settings and close the dialog box.

Note After you modify any of these parameters, you must restart WindowViewer to apply your changes.

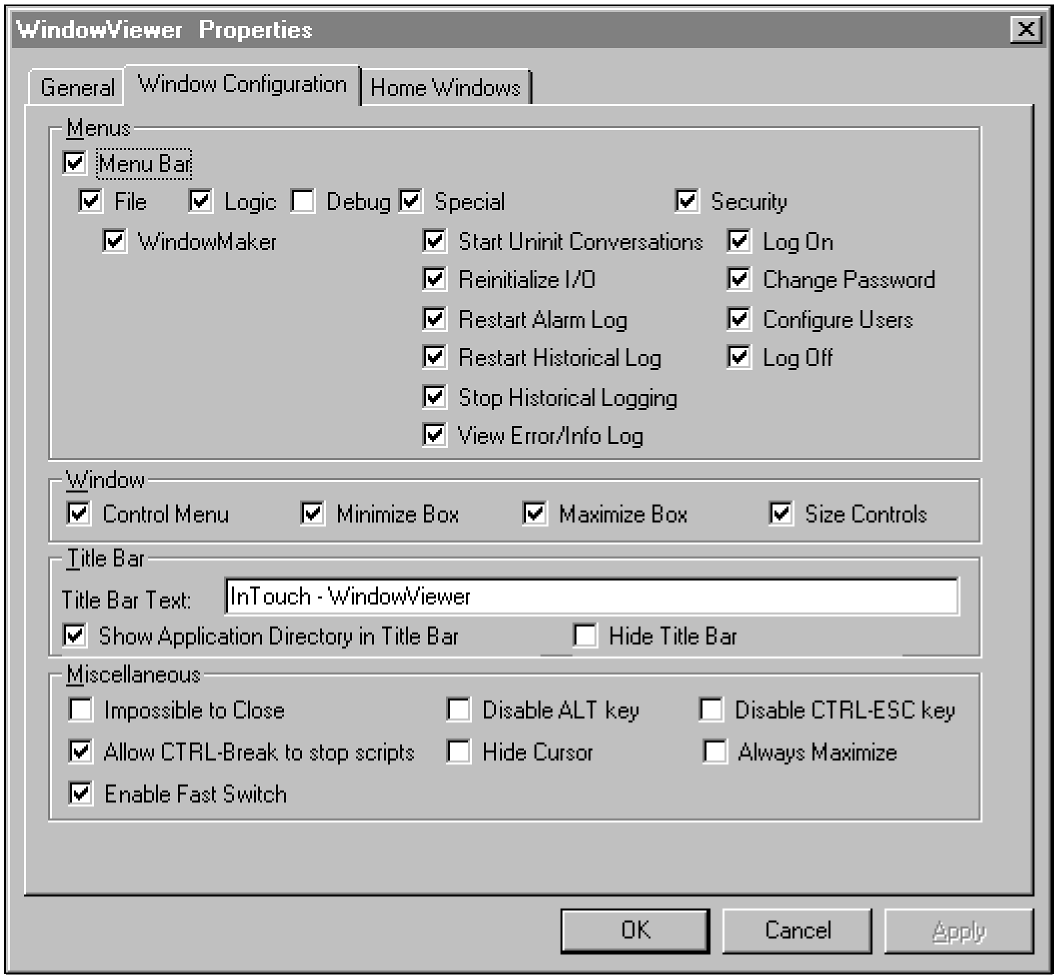
Setting WindowViewer's Window Configuration Properties

➢ **To configure the WindowViewer program window:**

1. On the **Special** menu, point to **Configure**, and then click **WindowViewer**, or in the Application Explorer under **Configure**, double-click **WindowViewer**. The **WindowViewer Properties** dialog box appears:

   ☝ In the Application Explorer, you can also right-click **WindowViewer**, and then click **Open**.

2. Click the **Window Configuration** tab:



WindowViewer Properties

General | Window Configuration | Home Windows

Menus
- ☑ Menu Bar
- ☑ File ☑ Logic ☐ Debug ☑ Special ☑ Security
- ☑ WindowMaker
- ☑ Start Uninit Conversations
- ☑ Reinitialize I/O
- ☑ Restart Alarm Log
- ☑ Restart Historical Log
- ☑ Stop Historical Logging
- ☑ View Error/Info Log
- ☑ Log On
- ☑ Change Password
- ☑ Configure Users
- ☑ Log Off

Window
- ☑ Control Menu ☑ Minimize Box ☑ Maximize Box ☑ Size Controls

Title Bar
- Title Bar Text: InTouch - WindowViewer
- ☑ Show Application Directory in Title Bar ☐ Hide Title Bar

Miscellaneous
- ☐ Impossible to Close ☐ Disable ALT key ☐ Disable CTRL-ESC key
- ☑ Allow CTRL-Break to stop scripts ☐ Hide Cursor ☐ Always Maximize
- ☑ Enable Fast Switch

OK Cancel Apply

   ☝ If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

3. In the menus group, select the menus and commands that you want displayed in runtime.

   ☝ By default, the menu bar is displayed when WindowViewer is running. Clear the **Menu Bar** option to prevent the menu bar from showing.

   Clear the **WindowMaker** command if you want to prevent the operator from being able to switch to the WindowMaker program. (Selecting this option does not affect the fast switch to WindowMaker.)

   Clear the **Logic** menu if you want to prevent the operator from starting and stopping all QuickScripts from executing during runtime.

You can use the system tagname, **$LogicRunning** to allow the operator to start and stop all QuickScripts.

Note If you select the **Allow CTRL-Break to stop scripts** option, the operator will be able to stop all QuickScripts from executing regardless of whether the **Logic** menu is displayed or not.

Asynchronous QuickFunctions that are currently executing cannot be stopped. However, you can prevent new asynchronous QuickFunctions from executing.

Select the **Debug** menu <u>only</u> when you need to "debug" your application.

Select the **Window** controls that you want available in runtime.

Note You must clear the **Control Menu** option (also called the System menu) in order to hide the close (**X** button) in the upper right hand corner of the application.)

---

4. In the **Title Bar Text** box, type the title that you want to appear in your application's title bar in runtime. For example:

   ABC Company, Paint APP1

   Note You cannot change the title bar if you are using a "Promotional License."

   📖 For more information on FactorySuite licensing, see your *FactorySuite System Administrator's Guide*.

5. Select **Show Application Directory** if you want to include the path to the application's directory in the title bar. For example:

   ABC Company, Paint APP1 - C:\DEMOAPP1

6. Select **Hide Title Bar** if you want to hide the application's title bar in runtime.
7. Select **Impossible to Close** if you want to prevent the operator from closing WindowViewer.

   Note You must clear the **Control Menu** option (also called the System menu) in order to hide the close (X) box in the upper right hand corner of the application.)

   ---

8. Select **Allow CTRL-Break to stop scripts** if you want to allow the operator to press the CTRL + BREAK key sequence to stop the execution of <u>all</u> QuickScripts whenever necessary during runtime.

   Note Asynchronous QuickFunctions that are currently executing cannot be stopped. However, you can prevent new asynchronous QuickFunctions from executing.

   ---

9. Select **Disable ALT key** if you want to disable the ALT key and prevent the operator from executing menu commands by using the ALT + accelerator key. For example, ALT + FX to exit the application.

   Note You must clear the **Control Menu** option (also called the System menu) in order to hide the close (X) box in the upper right hand corner of the application.)

   ---

10. Select **Hide Cursor** if you want to prevent the cursor from being displayed during runtime because a touch-screen will be used.
11. Select **Disable CTRL-ESC key** if you want to prevent the operator from accessing the Windows **Start** menu to close and/or switch applications.
12. Select **Always Maximize** if you want to keep the WindowViewer program maximized at all times.
13. Click **OK** to save your settings and close the dialog box.

Note After you modify any of these parameters, you must restart WindowViewer to apply your changes.

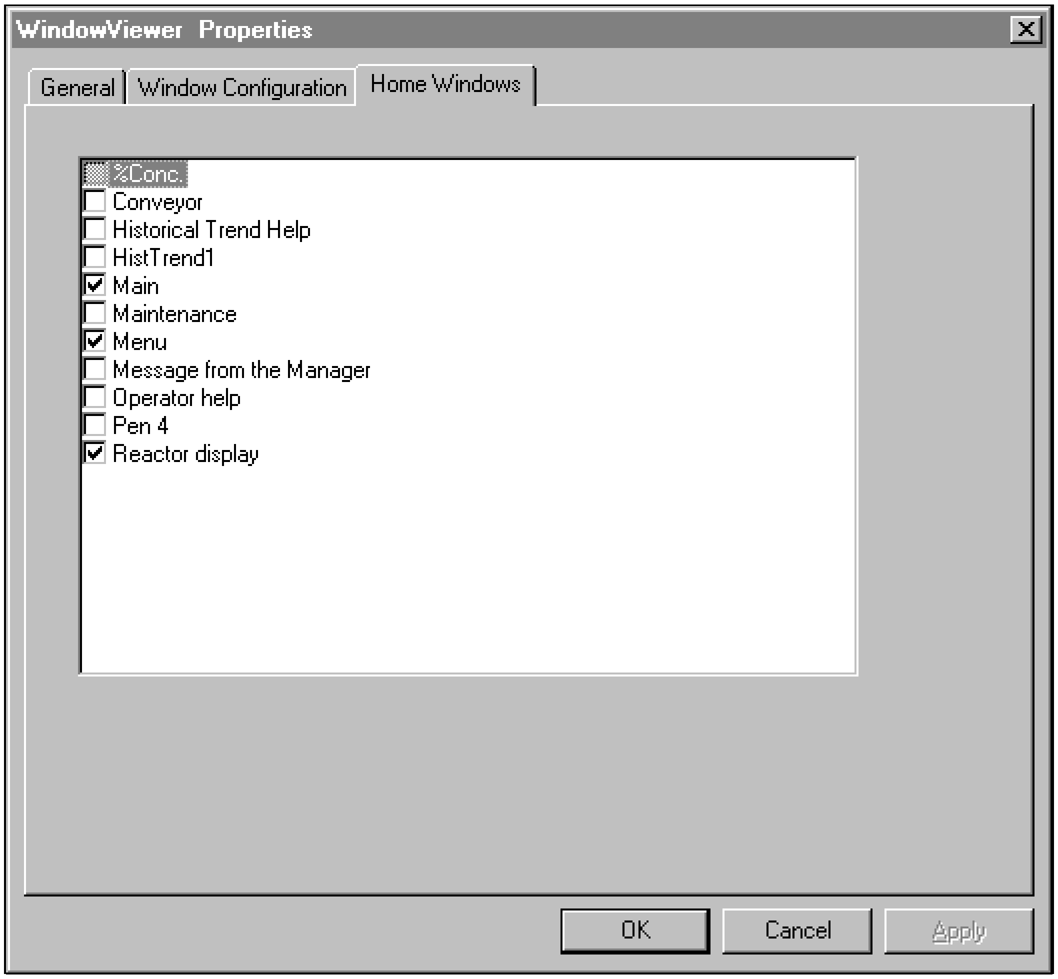
Selecting WindowViewer's Home Windows

➢ **To select the WindowViewer default start up windows:**

1. On the **Special** menu, point to **Configure,** and then click **WindowViewer**, or in the Application Explorer under **Configure**, double-click **WindowViewer**. The **WindowViewer Properties** dialog box appears:

   🖱 In the Application Explorer, you can also right-click **WindowViewer**, and then click **Open**.

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



- [ ] %Conc.
- [ ] Conveyor
- [ ] Historical Trend Help
- [ ] HistTrend1
- [✓] Main
- [ ] Maintenance
- [✓] Menu
- [ ] Message from the Manager
- [ ] Operator help
- [ ] Pen 4
- [✓] Reactor display

3. Select the window(s) that you want to automatically open when WindowViewer is started directly.

   Note The home windows selections have no effect when you use the fast switch to start WindowViewer. Home windows are automatically opened when you start WindowViewer directly from either its icon on its menu command.

4. Click **OK**.

Using InTouch Security

Applying security to your application is optional. However, by applying security to your application, you can control specific functions that an operator is allowed to perform by linking those functions to internal tagnames. In addition, when you establish security on your application, audit trails can be created that tie the operator to all alarms/events that occur during the time he/she is logged on to the system.

Security is based on the concept of the operator "logging on" to the application, typing his/her name and his/her password. Therefore, you must configure a user name, password and access level for each operator.

Note There is no association between Microsoft operating system security and InTouch security.

When you create a new application, by default, the user name is set to "Administrator" with an access level of 9999 (which allows access to all security commands). Once you add a new user name to the security list and restart WindowMaker or WindowViewer, the default user name is automatically reset to "None" with an access level of "0" (which prevents access to the Configure Users command in both WindowMaker and WindowViewer). Therefore you must configure a user name for the System Administrator with an access level equal to or greater than 9000 in order to be able to access the security user list later.

Once an operator logs on to the application, access to any protected function will be granted upon verification of the operator's password and access level against the value specified for the internal security tagname linked to the function.

For example, you can control access to a window, or the visibility of an object and so on, by specifying that the logged on operator's "Access Level" must be greater than 2000.

Note The operator can log on to the application by executing the **Log on** menu command under **Security** in the WindowViewer **Special** menu (if the **Special** menu is displayed) or, you can create a custom log on window with touch-sensitive input objects that are linked to internal security tagnames.

📖 For more information on the internal security tagnames, see the *InTouch Reference Guide*.

🖱 The commands used to establish security on an application located under **Security** on the **Special** menu in both WindowMaker and WindowViewer. The security commands are used to log on and off the application, change passwords and to configure the list of valid user names, passwords and access levels.

Using the Security Internal Tagnames

Once you implement security on your application, there are two internal security tagnames that you can use on buttons, in animation link expressions or QuickScripts, and so on, to control whether or not the logged on operator is allowed to perform specific functions:

Tagname	Type	Valid Values	Access
$AccessLevel	Integer	0-9999	Read Only
$Operator	Message	16-characters max	Read Only

For example, to make an object become visible based on the logged on user's access level, the following statement could be used in a visibility animation link's expression:

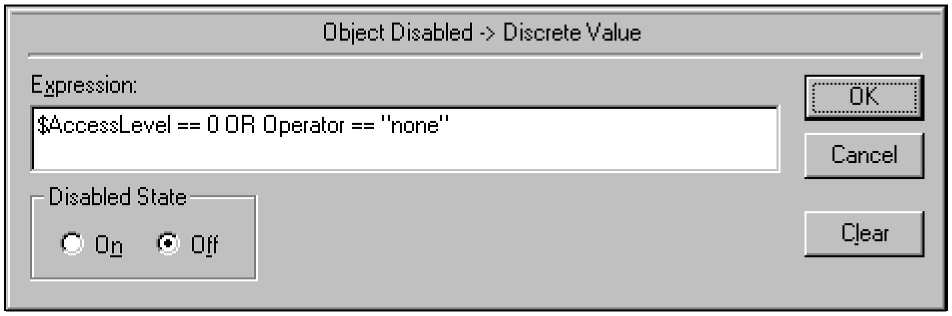
```
$AccessLevel >= 2000;
```

Or, a QuickScript can be bounded by an **IF** statement:

```
IF $Operator == "DayShift" THEN
```

```
    Show "Control Panel Window";
    {and other lines that only execute for the DayShift Operator}
ENDIF;
```

You can also control an object's touch functionality based upon the value of an internal security tagname by using the **Disable** animation link. For example:



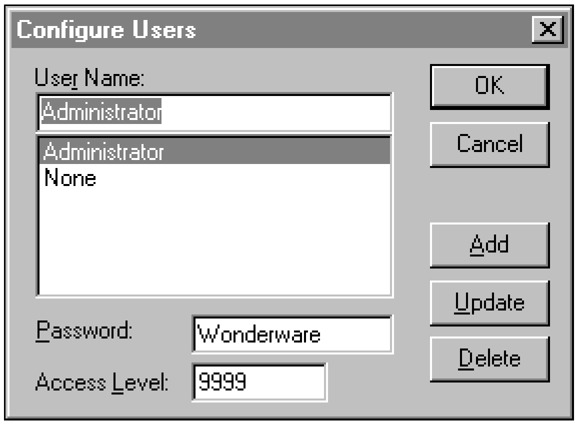
Expression:
$AccessLevel == 0 OR Operator == "none"
Disabled State: On [✓ Off]

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

Configuring the Operator's Security Level

➢ **To configure security for the operators of your application:**

1. On the **Special** menu, point to **Security**, and then click **Configure Users**. The **Configure Users** dialog box appears:`



User Name: Administrator
Administrator
None
Password: Wonderware
Access Level: 9999

If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **User Name** field, type the name that you want to assign to the operator.
3. In the **Password** field, type a password (up to 32 characters).
4. In the **Access Level** field, type a value (lowest = 0 to highest = 9999).
5. Click **Add** to add the user name to the security list.

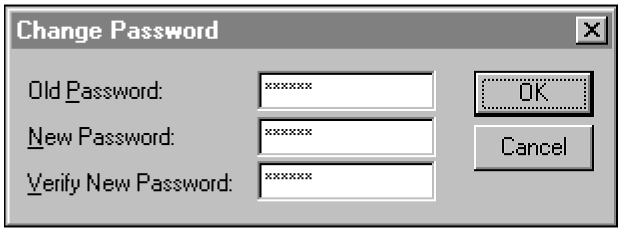
To modify an existing user name, select the desired name in the **User Name** list. Type your changes, and then click **Update** to accept the changes. To delete a user name, select it in the list, and then click **Delete**.

Note The **None** and **Administrator** names are reserved and only the password (**Wonderware**) or **Administrator** may be changed. Once you have configured user names for your application, you should change the **Administrator** name's password since it will more than likely become commonly known to most users of the system. The **Administrator** default access level (9999) is the highest and allows access to everything including, the **Configure Users** menu command.

Changing a Security Log On Password

➢ **To change the password for an operator:**

1. On the **Special** menu, point to **Security**, and then click **Change Password**. The **Change Password** dialog box appears:`



   ☝ If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Old Password** field, type the old password.
3. In the **New Password** field, type the new password (up to 32 characters).
4. In the **Verify Password** field, type the new password again.
5. Click **OK**.

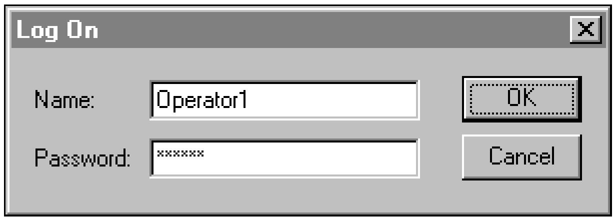
   ☝ To prevent anyone who may be watching the operator from seeing the password, the information entered is displayed on the screen as asterisks.

Note If you do not plan on displaying the **Special** menu in WindowViewer, you can create a discrete button and link it to the **$ChangePassword** internal tagname to set the **$ChangePassword** tagname equal to 1 to cause the **Change Password** dialog box to be displayed. Once displayed, the operator can change his/her password.

Logging on to an Application

➢ **To "log on" to an application:**

1. On the **Special** menu, point to **Security**, and then click **Log On**. The **Log On** dialog box appears:



   ☝ If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Name** box, type your user name.
3. In the **Password** box, type your password.
4. Click **OK**.

   ☝ If the information is entered incorrectly or is invalid, a message box indicating that log on failed will appear.

   If log on is successful, the **$AccessLevel** internal tagname will be set to its predefined value (configured in the security user list).

Creating a Custom Security Log on Window

If the **Special** menu will not be displayed in WindowViewer, you can create a custom log on window that the operator uses to log on to the application.

➢ **To create a custom log on window:**

Link the **$OperatorEntered** and **$PasswordEntered** system tagnames to user input objects or, use them in a QuickScript to set the "User Name" and "Password." (These are internal message type tagnames that are intended for write operation only.)

For example:

```
Set the User Name string into ->     $OperatorEntered
Set the User Password string into -> $PasswordEntered
```

If the entries are valid, the **$AccessLevel** and **$Operator** internal tagnames are set to their predefined values (configured in the security user list). i

Also, when you are not displaying the **Special** menu in WindowViewer, you can link a **User Input - Discrete** button to the **$ChangePassword** tagname to show the **Change Password** dialog box and allow the operator to change his/her password. When the operator clicks the button, the value of the **$ChangePassword** tagname is set to 1 and the **Change Password** dialog box appears. When the operator closes the dialog box, the system resets the value to 0. (This is a system discrete tagname intended for write operation only.)

You can also link a **User Input - Discrete** button to the **$ConfigureUsers** tagname to allow an authorized operator to access the **Configure Users** dialog box to edit the security user name list. When the operator clicks the button, the value of the **$ConfigureUsers** tagname is set to 1 and the **Configure Users** dialog box appears. When the operator closes the dialog box, the system resets the value to 0. (This is a system discrete tagname intended for write operation only.)

Logging Off an Application

➢ **To log off the application:**

On the **Special** menu, point to **Security**, and then click **Log Off**.

When this command is executed, the "User Name" is reset to "None" with an Access Level of "0".

You can configure the application to automatically log off the operator after a specified amount of time has elapsed with no activity by the operator.

For more information, see "Automatically Logging Off the System."

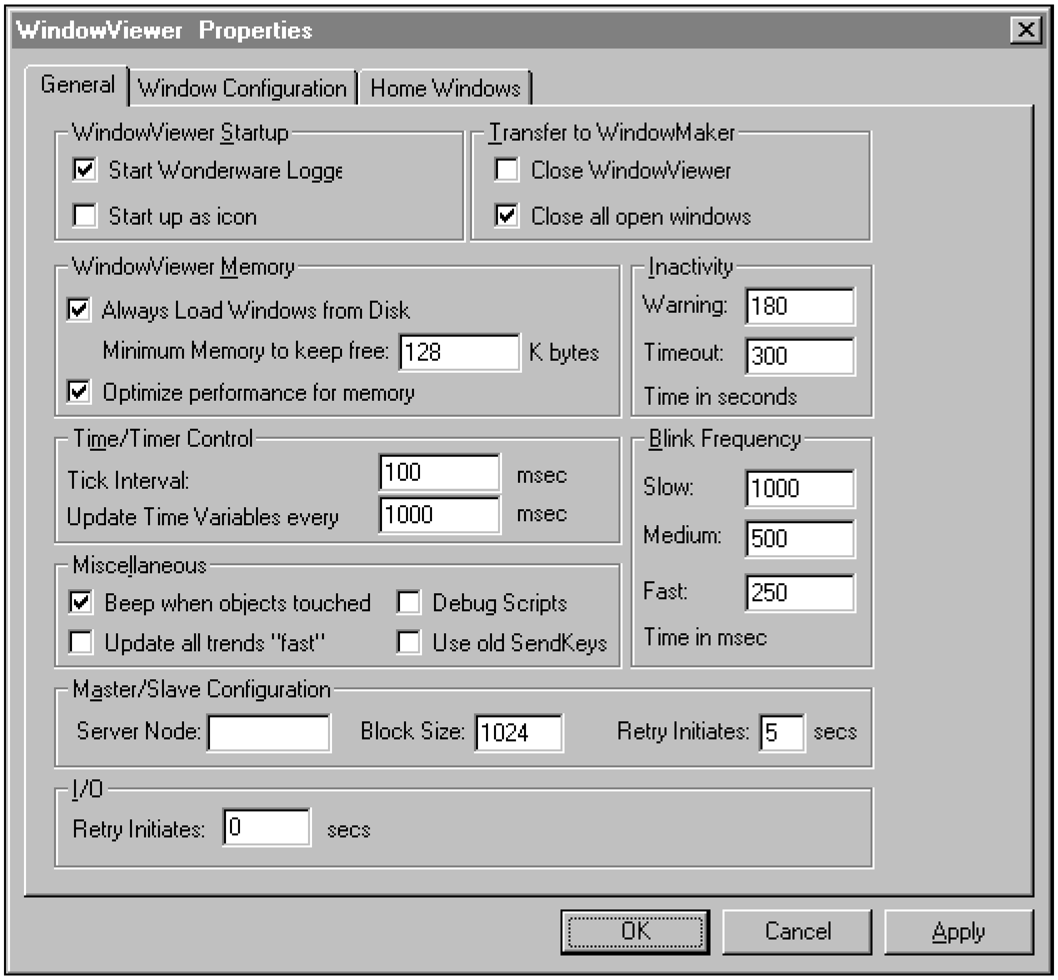
Automatically Logging Off the System

You can configure your application to automatically log off the operator when there has been no activity for a specified period of time by using the warning and timeout settings.

➢ **To configure inactivity:**

1. On the **Special** menu, point to **Configure**, and then click **WindowViewer**, or in the Application Explorer under **Configure**, double-click **WindowViewer**. The **WindowViewer Properties** dialog box appears with the **General** properties sheet active:

   ☝ In the Application Explorer, you can also right-click **WindowViewer**, and then click **Open**.



WindowViewer Properties

General | Window Configuration | Home Windows

WindowViewer Startup
- [✓] Start Wonderware Logge
- [ ] Start up as icon

Transfer to WindowMaker
- [ ] Close WindowViewer
- [✓] Close all open windows

WindowViewer Memory
- [✓] Always Load Windows from Disk
- Minimum Memory to keep free: 128 K bytes
- [✓] Optimize performance for memory

Inactivity
- Warning: 180
- Timeout: 300
- Time in seconds

Time/Timer Control
- Tick Interval: 100 msec
- Update Time Variables every 1000 msec

Blink Frequency
- Slow: 1000
- Medium: 500
- Fast: 250
- Time in msec

Miscellaneous
- [✓] Beep when objects touched
- [ ] Debug Scripts
- [ ] Update all trends "fast"
- [ ] Use old SendKeys

Master/Slave Configuration
- Server Node:
- Block Size: 1024
- Retry Initiates: 5 secs

I/O
- Retry Initiates: 0 secs

OK | Cancel | Apply

   ☝ If you right-click any of the text entry boxes in any dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Warning** box, type the number of seconds that can elapse with no operator activity (mouse clicks or keystrokes) before the system discrete tagname **$InactivityWarning** is set to 1 (True).

   ☝ You can use **$InactivityWarning** in a Condition QuickScript to show a window warning the operator that he/she is about to be logged off the system. If the operator clicks the mouse, presses a key, or performs an action using any other pointing device before the specified timeout elapses, they are not logged off. **$InactivityWarning** and the timer are reset.

3. In the **Timeout** box, type the number of seconds that can elapse with no operator activity (mouse clicks, keystrokes, and so on) before the system discrete tagname **$InactivityTimeout** is set to 1 (True). When **$InactivityTimeout** is true, the system equates the logged on operator name to the reserved name "None" and sets the security tagname, **$AccessLevel**, to 0.

   You can use **$InactivityTimeout** in a Condition QuickScript to show a window telling the operator that he/she has been logged off the application.

   You can use the **Timeout** feature independently of the **Warning** feature. However, the **Timeout** value must be greater than the **Warning** value for proper use of both system tagnames.

   Example: Warning becomes true after 30 seconds of inactivity and **Timeout** becomes true after an additional 15 seconds (for a total of 45 seconds) of inactivity.

4. Click **OK**.

   You can use **$InactivityWarning** in a Condition QuickScript to show a window warning the operator that he/she is about to be logged off the system. If the operator clicks the mouse, presses a key, or uses his/her touch screen before the specified timeout elapses, they are not logged off. (**$InactivityWarning** and the timer are reset.)

5. In the **Timeout** box, type the number of seconds that can elapse with no operator activity (mouse clicks, keystrokes, and so on) before the system discrete tagname **$InactivityTimeout** is set to 1 (True). When **$InactivityTimeout** is true, the system equates the logged on operator name to the reserved name "None" and sets the security tagname, **$AccessLevel**, to 0.

   You can use **$InactivityTimeout** in a Condition QuickScript to show a window telling the operator that he/she has been logged off the application.

   You can use the **Timeout** feature independently of the **Warning** feature. However, the **Timeout** value must be greater than the **Warning** value for proper use of both system tagnames.

   Example: Warning becomes true after **30** seconds of inactivity and **Timeout** becomes true after an additional **15** seconds (for a total of **45** seconds) of inactivity.

C H A P T E R 3

Building a Distributed Application

InTouch is designed to support both stand-alone and distributed applications. Stand-alone applications are those that use just one Operator Interface (OI) for each monitored system, such as in a boiler package control. Stand-alone applications are generally easier to configure, with minimal to no networking, and require only simple maintenance. Distributed applications, conversely, are much more complex, often with several layers of networks. Distributed applications, typically, have a central development station, central data storage, with many *client* stations which interact with the central station and each other.

InTouch provides many features that greatly ease the building and maintenance of distributed applications. One of the most powerful is "Network Application Development" (NAD). NAD allows many *client* stations to maintain a copy of a single application without restricting the development of that application. InTouch NAD also provides automatic notification to these *client* stations when the application changes.

This chapter describes how to use the distributed features of InTouch, the different architectures you can use, and the advantages and disadvantages of each.

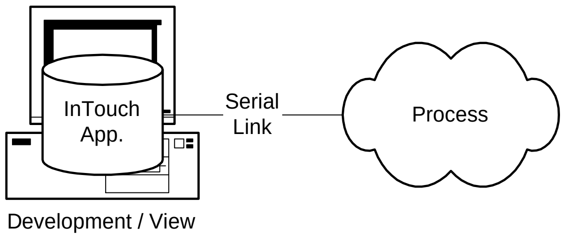
Contents

Network Architectures

InTouch is a highly configurable package that can be set up in many different ways, depending on your application's needs. This section provides an overview of the various architectures available with InTouch, and the relative advantages and disadvantages of each. While mention is made of various application components such as alarms and history, these systems are covered in greater depth later in their respective chapters.

Stand-alone Application

Stand-alone applications are defined as those with a single operator interface for each monitored process. These typically consist of one non-networked personal computer (PC) that functions as the primary operator interface (OI). This computer is connected to the industrial process via a direct connection, such as a serial cable.



In this architecture, a single InTouch application is installed on the computer. If development work is required, the application can be developed directly on this computer. It can also be copied to another computer, and modified, and then copied back to the original computer. While not a network architecture, the stand-alone architecture is covered for completeness.

Advantages

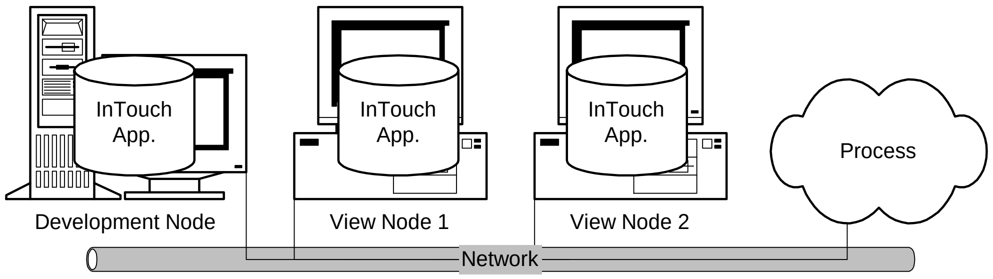
- Easy to maintain

Disadvantages

- Limited to single node

Client-Based Architecture

The client-based architecture is the first of the networked architectures, and is a direct outgrowth of the stand-alone. It provides a unique copy of one InTouch application for each computer running WindowViewer and NetDDE (View node). This application can be installed on each computer's hard drive, or in a unique location on a network server. In the example below, an application would be developed and tested on the development node, and then copied to each View node.



As each View node has an identical copy of the application, each must also have identical access to any data sources referred to by the application. These sources may be I/O Servers, SQL databases, DOS files, an so on. If a central data source is used, for example a network-shared I/O Server, each View node will maintain a separate conversation with the shared server, which can result in increased network loading. Therefore, it is a good idea to consider individual I/O Servers on each node if heavy network use is expected.

The client-based architecture has several tradeoffs when it comes to application maintenance. Since each node has its own copy of the application, the development node has unrestricted editing capability for that application. Modifications can be made and tested on that node without affecting the running process. The drawback of this approach is in the effort required to distribute the modified application to the View nodes. Each View node must be shut down locally, the new application copied to it, and then View must be restarted.

Advantages

- Unrestricted development of the application
- Inherent redundancy since each node can be self-sufficient
- No limit to the number of View nodes you can use

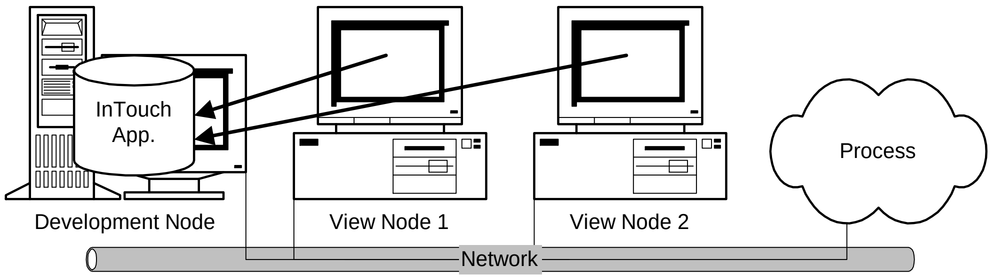
Disadvantages

- Distributing applications is difficult
- All nodes must have identical access to the same data sources

Note This architecture has been superseded by the NAD architecture, which is discussed later in this chapter. It is described here only for the purposes of presenting a complete overview of networking architectures.

Server-Based Architecture

The server-based architecture allows several View nodes to share a common InTouch application. In the example below, the two View nodes are accessing the same application from the development node. Each View node must create a logical drive in the networking software and map it to the shared network drive of the development node. Each View node must also have the shared application registered with the InTouch program.



As in the client-based architecture, each View node must have identical access to any data sources referred to by the application. There are also ways to tailor the data source locations by using a combination of scripts to get the node name and change each data source location based on that name.

For more information, see "Configuring InTouch for Common Data Sources."

While both WindowMaker and WindowViewer are running from the same application directory, WindowMaker is restricted from many editing functions, including: changing tagname types, deleting unused tagnames, and configuring the application. This architecture does allow the View nodes to be updated when the application changes.

Advantages

- Single application to maintain
- View nodes automatically updated when application changes

Disadvantages

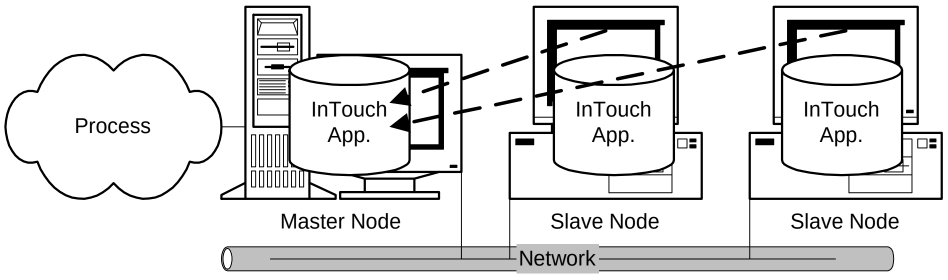
- Development of application is restricted
- No redundancy if the Development station goes down
- All nodes must have the same screen resolution

Note This architecture has been superseded by the NAD architecture, which is discussed later in this chapter. It is described here only for the purposes of presenting a complete overview of networking architectures.

Master/Slave Architecture

The master/slave architecture was developed to overcome some of the drawbacks of the client-based and server-based architectures. While it still allows View nodes to be set up in a client or server type architecture, it does not require that they all have the same data sources.

The architecture defines one node as a "Master" node (usually the computer connected to the industrial process). This node acts as a server to the remote or "Slave" View nodes running the same application. In the example below, each "Slave" node can either run their own unique copy of the application or a common application. Either way, once running, each "Slave" node references all their I/O data sources via the "Master" node that is connected to the monitored process.



Developing a master/slave application takes some planning, as all tagnames must be I/O type tagnames. Each of these tagnames must use a fully qualified NetDDE reference to the "Master" node. For example, **Node = Master7, App = View, Topic = tagname**. When this application runs on the "Master" node, the references point to local resources; when run from a "Slave" node, the references point to the remote "Master" resources.

To ease the maintenance of distributing the applications, a system tagname $**ApplicationVersion** can be used to notify clients of changes. $**ApplicationVersion** reflects the current version number of the application and is incremented on the "master" node each time changes are made. $**ApplicationVersion** can be monitored and used to flag the "slave" nodes of application changes.

To reduce network load, typically the InTouch I/O type tagnames on the master node reference an I/O Server connected to the process. However, the slave nodes reference WindowViewer on the master node. This means the Tagname Dictionarys are not identical between the master and slave nodes. As such, issues related to tagname scaling and maintenance of two separate InTouch applications should be considered.

Advantages

- Prevents nodes from flooding networks by funneling all communications through one source (Master node)
- Provides automatic notification of application changes via **$ApplicationVersion**

Disadvantages

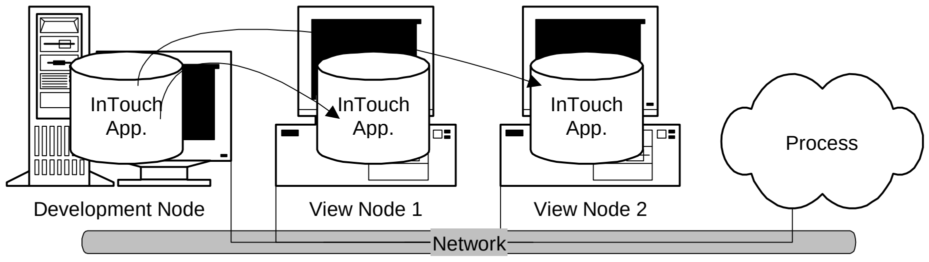
- Distributing applications is difficult
- Single source of application—no redundancy if Master node goes down
- Database is typically not the same between master and slave

Network Application Development (NAD)

Network Application Developmentor NADis an architecture that combines the best of the client-based and server-based architectures. NAD provides automatic notification of application changes and automatic distribution of the updated applications to View nodes. NAD can even be used to automatically distribute master/slave applications.

Note You cannot use the NAD features if you are using WindowViewer as an NT service.

In the NAD architecture, a master copy of an application is maintained on a central network location. Each View node loads that network application as they would in a server-based architecture but, instead of running the application from the server, the application is copied to and run from a user-defined location. This provides the client-based advantage of redundancy. In the example below, the two View nodes both have the master application registered from the development node, but actually run it from their own hard drives.



When a View node copies and runs a master application, it automatically monitors for changes in the master copy. These changes are indicated by a flag in the master application directory. This flag is set manually when the application developer uses the **Notify Clients** command on the WindowMaker **Special** menu while editing that application. When this flag changes, each View node has a user-definable action that specifies the response of that node. This can range from ignoring the flag, to automatically shutting down and restarting the View node, which reloads the master application.

Advantages

- Single application to maintain
- View nodes automatically notified when application changes
- Each View node has user-definable action for application updates
- Unrestricted development of the application

Disadvantages

- When distributing a large, complex application to numerous nodes, slowed system response time may be apparent on the initial down load, updates are optimized
- Limits flexibility of having different applications running on different nodes
- Changing the application and notifying clients will momentarily suspend all view nodes while the application is transferred. The larger the application, the longer the time required for copying
- Application transfer may be a problem for slow networks or over serial connections

Configuring Network Resources

InTouch provides many configurable options for distributed use. This section contains hands-on examples of how to setup and use these options to distribute your InTouch applications.

Configuring UNC Paths for Files

InTouch supports the use of Universal Naming Conventions (UNCs) for application directory entries, configuration items, remote history, and distributed alarms. UNC allows direct access to network-based files without needing to create a mapped drive letter. Each UNC address may consist of three parts: Node, Share, and Path in the form \\Node\Share\Path. Node refers to the computer node name that contains the file share. Share refers to the logical name assigned to the shared directory on that computer. Path refers to the normal DOS path to that file with respect to the share.

Before you can access a file through UNC, you must create a file share on the computer you want to access. You can share an entire drive, or just a directory of it; and even customize the security of that share. Either way, you must create a share name that will be used in the UNC address. For more information on creating a file share, refer to the manuals for your Windows operating system.

Once the share is created, you can use a UNC address to refer to that drive anywhere that you would normally enter a file path. For example, the InTouch program allows either a UNC path or a standard DOS path to be entered for the location of InTouch applications.

For example, let's assume that you have a computer with the network name of "EngineRm" that you have shared the root drive "C:\" with the share name of "Root". To set up a UNC path to the "C:\InTouch.32\Apps\Boiler" application you must use the following UNC:

\\EngineRm\Root\InTouch.32\Apps\Boiler

If the "Boiler" directory itself was shared as "Boiler," the UNC could be shortened to:

\\EnginerRm\Boiler

No path is required if the share is the path.

Note If you need to write to a file referred to by a UNC address, the share must be a read/write share, even on a local node. If you create a share that is password-protected, you will not be able to access the share with a UNC unless you first set up a network drive mapping. You can set up a drive mapping from the remote node by using Windows Explorer

Configuring DDE Shares

InTouch is shipped with the Wonderware NetDDE product, portions of which were licensed to Microsoft as *Network DDE* for use in Windows 95, and Windows NT. The NetDDE provided with InTouch conveys additional capabilities over the Microsoft Network DDE. NetDDE provided with InTouch is used only on the Windows 95 operating system while, Wonderware's *NetDDE Extensions* is used with the Windows NT operating system for configuring WinSock for use with Microsoft Network DDE.

If you are using the Windows NT operating system, you will need to set up a DDE share for any node containing I/O resources that other View nodes may need to access. For example, if you have a node running GE Genius I/O Server, you need to create a share for that I/O Server.

Windows NT Operating System

Microsoft includes *NT Network DDE* with its Windows NT operating system. In order to allow *NT Network DDE* to act as a resource for InTouch, DDE shares must be created for all nodes running on the Windows NT operating system with I/O resources that InTouch View nodes may need. (The "Server" referred to here is synonymous with Data Source it is not to be confused with the NT Server operating system.)

🕮 For more information on configuring shares, see your *FactorySuite System Administrator's Guide*.

Wonderware SuiteLink Communication Protocol

Wonderware FactorySuite is shipped with Wonderware's communications protocol 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:

- Value Time Quality (VTQ) places a timestamp 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.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft’s standard Winsock interface.

👓 For more information on using SuiteLink, see Chapter 9 - I/O Communications.

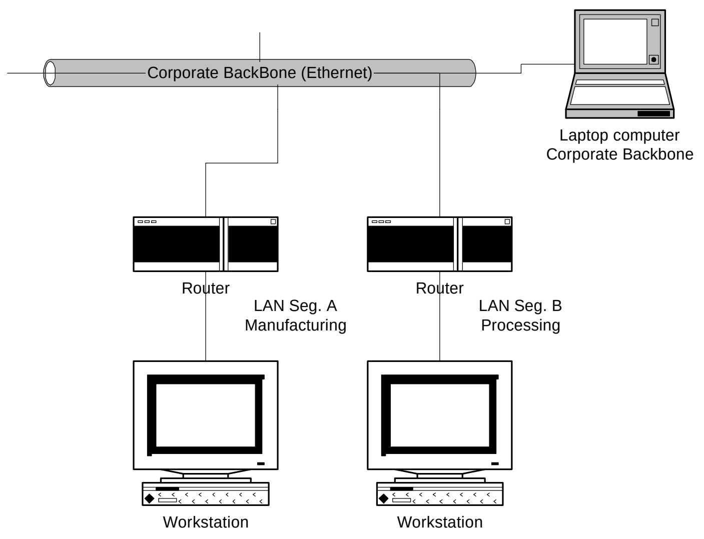
Troubleshooting Networks

There are some key networking architecture constraints that must be considered when using InTouch's distributed network features. One of the first things to recognize when implementing a large network of InTouch nodes is that all Ethernet connections are not the same. Many MIS departments segment their networks using devices called "routers." These routers act as traffic cops to regulate the flow of traffic from one Ethernet segment to another. Routers have the ability to "filter" out certain types of network traffic and addresses.

If you are using NetBEUI, it can be frustrating trying to connect to a remote I/O Server in another building or different city and find that it does not work, even though the networks are connected. If this happens it usually means the router has been set up to filter out NetBEUI traffic. One way around this is to reprogram the router to allow this traffic. However, reprogramming the router does have one drawback in that the NetBEUI broadcasts will now reach a larger audience and add traffic to the other segments on the network.

Note Wonderware's SuiteLink protocol cannot be used with NetBEUI.

Network Segmentation Example



To prevent this, switch to a TCP/IP protocol and set up the router to route from one router to the next automatically. The router must be configured to allow bi-directional data flow (A to B, and B to A) to minimize network traffic crossing the router onto other segments. By using TCP/IP, you will improve network performance while allowing conversations over very long distances using a large variety of network services (such as, Internet, Frame Relay, ISDN, and so on). Another key reason for you to use TCP/IP is the quick adoption of TCP/IP as the protocol of choice for the PLC suppliers.

Microsoft makes switching to TCP/IP a very easy task. The Windows NT Server comes with the Network Client for Windows 95 or Windows NT and has TCP/IP support built in. The Windows NT Server also has the ability to administer the allocation of TCP/IP addresses and names dynamically.

Microsoft's main networking product, Windows NT Server, also poses some challenges for networking InTouch nodes.

For example, whenever you try to retrieve InTouch files such as historical log files, from a different Domain, you must supply a password and have an account in that domain to make the connection. The creation of these accounts and constant interruptions for passwords make this almost unworkable. There are two solutions to avoid this:

1. You can make all the computers join the same domain; however, there can be administrative problems if there are too many computers in the domain.
2. You could setup a "Trust Relationship" between the domains to allow computers in one domain to share the resources of another domain without having to create additional accounts and supply passwords each time you connect.

The latter setup is definitely preferable, as well as being easier to setup and manage. Other, more advanced models of NT Domain Architecture are available that you may have to implement which have similar constraints.

It is recommended that you consult a Microsoft NT systems specialist on the design and topology of your whole network implementation in an instance such as this.

Configuring InTouch for Common Data Sources

InTouch allows you to build your application using several different architectures. Regardless of the architecture you choose, it is important to consider the data sources that application will access and how it will access them.

Each architecture shares a common trait in that the application is run by each View node as if it owned it. While this may not seem problematic at first, it's important when you consider the data source references that an InTouch application may contain. Typical data sources are Access Names, SQL connect strings, or Recipe files.

Each of these sources are retrieved through a reference address, such as D:\PROCESS\RECIPE.CSV in the case of a Recipe file, or DSN=PROCDB in the case of a SQL connect. While these addresses may make sense from the perspective of the computer they were developed on, they may be meaningless when that application is copied to, and run from, a View node that has no D: drive or registered ODBC data source name called PROCDB.

If you plan to distribute an application to more than one node, you need to consider the impact of your data source addresses. There are two basic ways to do this:

1. Create identical copies of the data sources on each View node, or
2. Use only global addresses for the data sources.

The following sections will cover these options as they pertain to the two major data sources: remote data and file access.

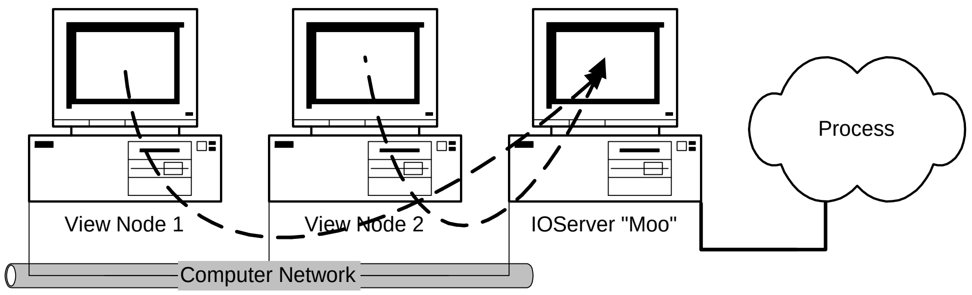
InTouch Access Names

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.

For more information on Access Names, see Chapter 9 - I/O Communications.

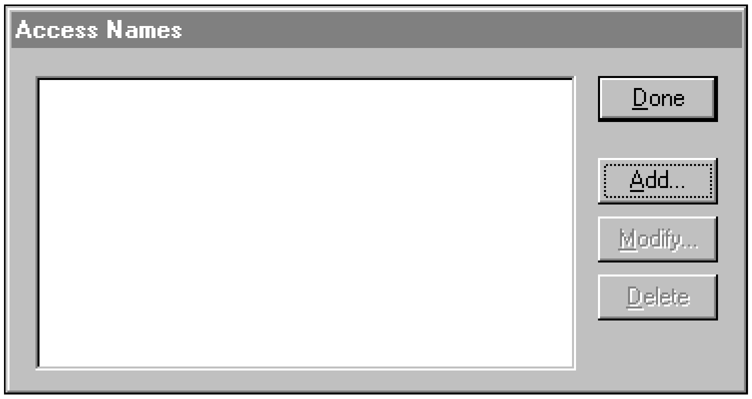
Global Addresses to I/O Data Sources

Global addresses to I/O data allow all of the View nodes to share a common network-based I/O Server. This prevents the cost of multiple I/O Servers, but is less fault-tolerant and can result in lower overall performance. In the example below, two View nodes, each running a copy of the same application, are referencing the same I/O data source. Since each application uses a fully qualified I/O address for the data source, all references point to the same I/O Server.

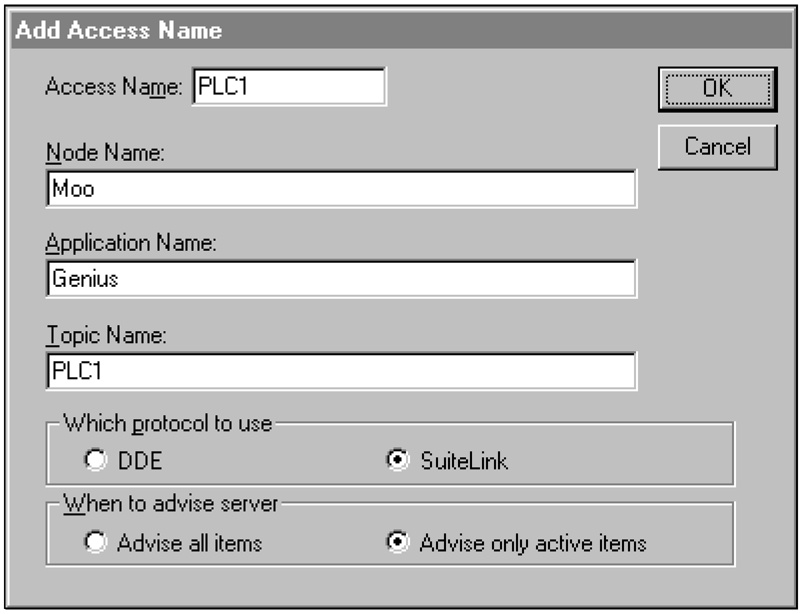


➢ **To set up this configuration:**

1. On the **Special** menu, click **Access Name** or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears:



2. Click **Add**. The **Add Access Name** dialog box appears:



Access Name: PLC1

Node Name: Moo

Application Name: Genius

Topic Name: PLC1

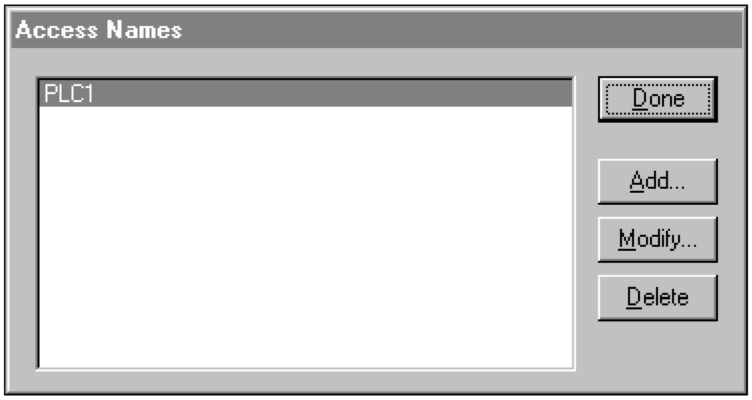
Which protocol to use: DDE [✓ SuiteLink]

When to advise server: Advise all items [✓ Advise only active items]

3. In the **Access Name** box, type **PLC1**.
4. In the **Node Name** box, type **Moo**. (Do not prefix the node name with \\.)
5. In the **Application Name** box, type **Genius**.
6. In the **Topic Name** box, type **PLC1**.

   You can define any name you want for the **Access Name**, but the **Application Name** and **Topic Name** must reference the computer with the I/O Server.

7. Select the protocol that you are using.
8. Click **OK**. The **Access Names** dialog box reappears showing the new Access Name in its list:

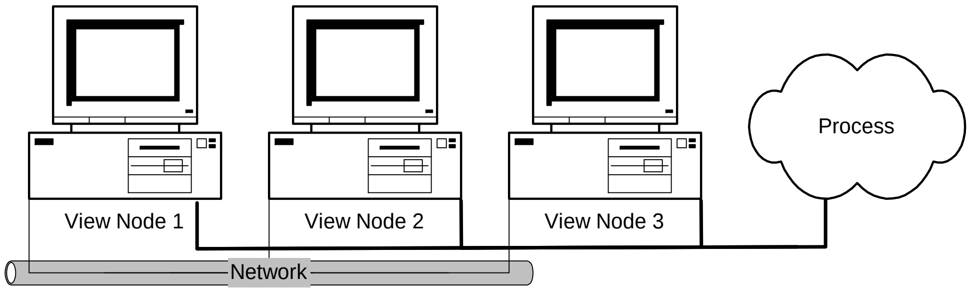


9. Click **Done**.

Local Addresses to I/O Data Sources

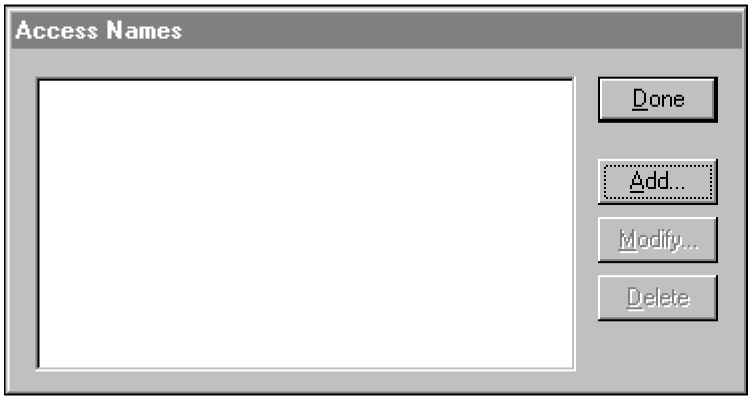
Local addresses to I/O data can be used when each View node has it's own I/O Server. This architecture provides fault-tolerant operation, as each View node can independently run if the network goes down. In the example below, two View nodes, each running a copy of the same application, are referencing their own I/O data source. Since each application uses a local I/O address for the data source, each reference points to the local I/O Server.

This method however, significantly increases the load on the process connection network. That is, three nodes triples the traffic created by one node, as each node's requests must be separately processed.

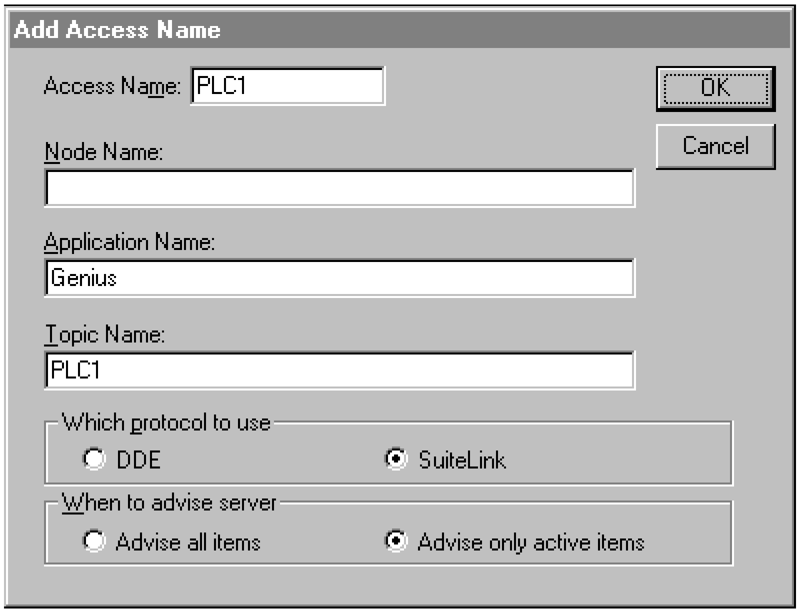


➢ **To set up this configuration:**

1. On the **Special** menu, click **Access Name** or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears:



2. Click **Add**. The **Add Access Name** dialog box appears:



Access Name: PLC1

Node Name:

Application Name: Genius

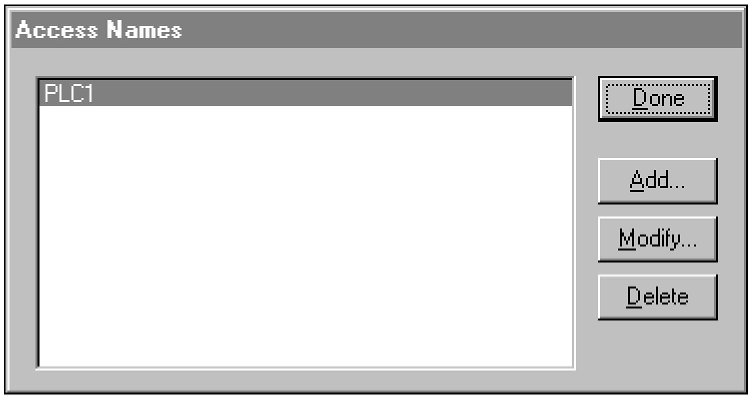
Topic Name: PLC1

Which protocol to use: DDE [✓ SuiteLink]

When to advise server: Advise all items [✓ Advise only active items]

3. In the **Access Name** box, type **PLC1**.
4. Leave the **Node Name** box blank.
5. In the **Application Name** box, type **Genius**.
6. In the **Topic Name** box, type **PLC1**.

   You can define any name you want for the **Access Name**, but the **Application Name** and **Topic Name** must reference the computer with the I/O Server.
7. Select the protocol that you are using.
8. Click **OK**. The **Access Names** dialog box reappears showing the new Access Name in its list:



PLC1

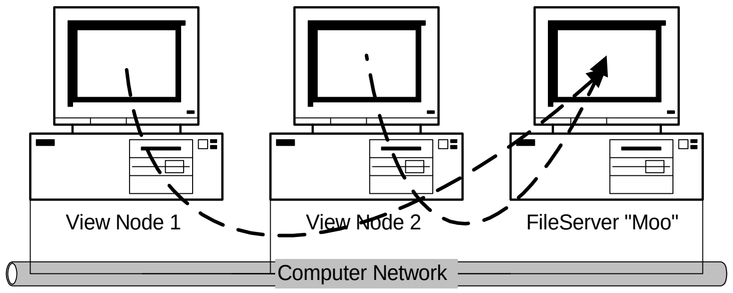
9. Click **Done**.

File Access

InTouch uses DOS files, FAT or NTFS to read and write reference data. Certain programs such as Recipe Manager, utilize files very heavily. In a distributed application, file references can be set up as global addresses to a network file server or as local addresses to a local file.

Global Addresses to File Data Sources

Global addresses to file data allow all of the View nodes to share a common network-based set of files. This provides single-source maintenance of the files, but is less fault-tolerant than local copies. In the example below, two View nodes, each running a copy of the same application, can reference the same Recipe file. Since each application uses a drive letter mapped to a fully-qualified network path for the file, all references point to the same file.



➢ **To set up this configuration:**

Map a network drive to the shared path containing the referenced files. In a script to retrieve a Recipe file, type the following:

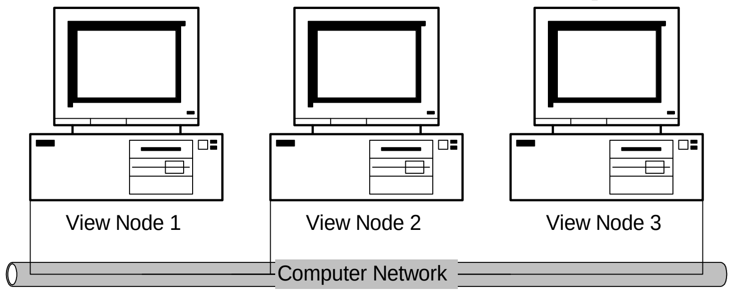
```
RecipeSelectRecipe("G:\Directory\Recipe.CSV", "review",
"RecipeName");
```

where "G:\" is the mapped drive letter that refers to \\Moo\Share. Every View node must be independently configured to have this same "G:\" drive mapped.

For more information, see "Configuring UNC Paths for Files."

Local Addresses to File Data Sources

Local addresses to file data can be used when each View node has it's own copy of the file. This architecture provides fault-tolerant operation, as each View node can independently run if the network goes down, but requires that any changes to the files be copied to all the View nodes. In the example below, three View nodes, each running a copy of the same application, can reference their own copies of the Recipe file. Since each application uses a local address for the file, each reference points to the local file.



➢ **To set up this configuration:**

Use a local address (for example, **C:\Directory**) to directly reference files. In a script to retrieve a Recipe file, type the following:

```
RecipeSelectRecipe("C:\Directory\Recipe.CSV", "review",
"RecipeName");
```

where "**C:**\" is a local drive.

A copy of the file "Recipe.csv" must be stored on each machine in its local directory "**C:**\directory." If the file is modified, it must be copied again to each machine. Because of the difficulty in maintaining this configuration any file access should be "Read Only" and modification to the local file should not be permitted.

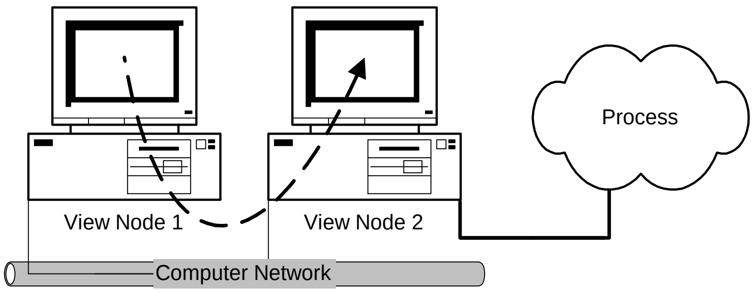
Configuring InTouch for Master/Slave

InTouch supports an architecture called Master/Slave that allows several networked InTouch applications to share a common InTouch Tagname Dictionary. The applications are configured so that a "Master" node (a computer that is connected to a PLC or a network of PLCs) can act as a server to the "Slave" or remote nodes that are running the same application.

The requirements for this application are as follows:

1. InTouch, I/O Server and NetDDE (for Windows 95) (MS Network DDE for Windows NT operating system) or the SuiteLink service must be running on the "Master" node.
2. InTouch and NetDDE (for Windows 95) (MS Network DDE for Windows NT operating system) or SuiteLink must be running on the "Slave" nodes.
3. A physical network connection must exist between the master node and the remote nodes.

Developing a Master/Slave application that can be copied to several nodes with no reconfiguration takes some advance planning. You should only use I/O type tagnames. No memory tagnames should be used. The example documented below shows how to configure an InTouch application for Master/Slave. This example has two View nodes; one of which is connected to a controlled process through a GE Genius I/O Server.

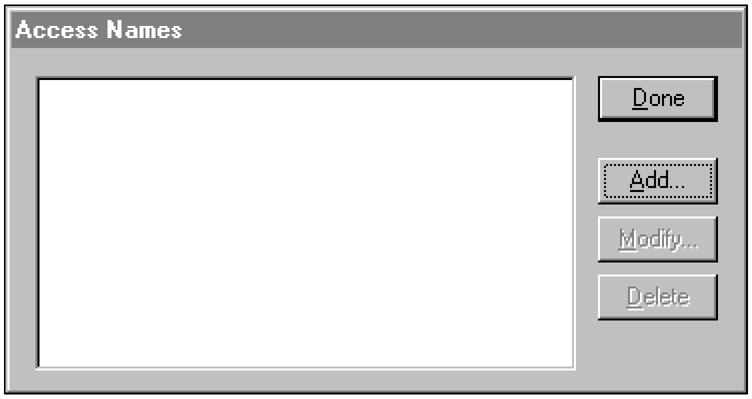


➢ **To setup this Master/Slave application:**

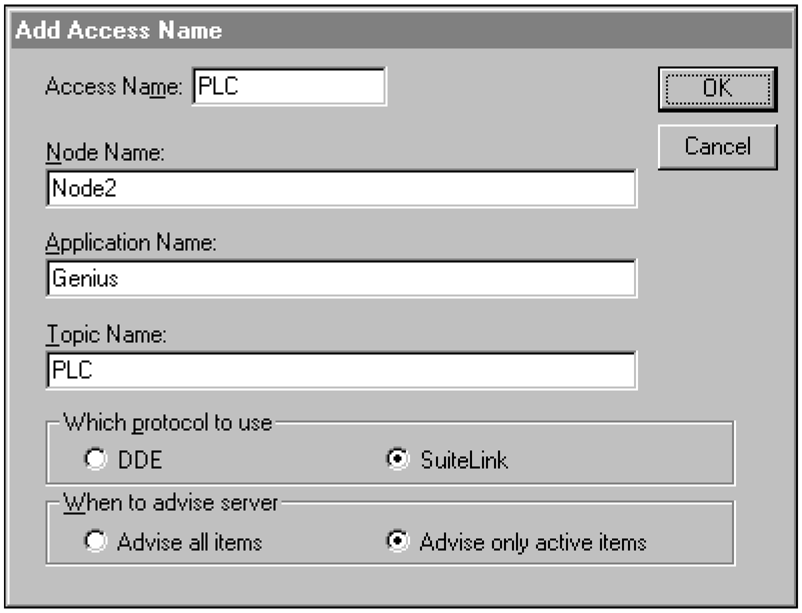
For Windows 95, set up the NetDDE node names for each node. For each node, configure the node name for that computer in the NetDDE program menu by using the **Local Node Name** command on the **Configure** menu.)

For purposes of this example, two nodes named "Node1" and "Node2" will be used. The two InTouch applications will be identical, but the GE Genius I/O Server will be running only on Node2. (The system could be configured so that Node1 was also running a GE Genius I/O Server and, thereby, act as a redundant backup for Node2.)

1. On the **Special** menu, click **Access Name** or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears:



2. Click **Add**. The **Add Access Name** dialog box appears:



Access Name: PLC

Node Name: Node2

Application Name: Genius

Topic Name: PLC

Which protocol to use: ( ) DDE [✓ SuiteLink]

When to advise server: ( ) Advise all items [✓ Advise only active items]

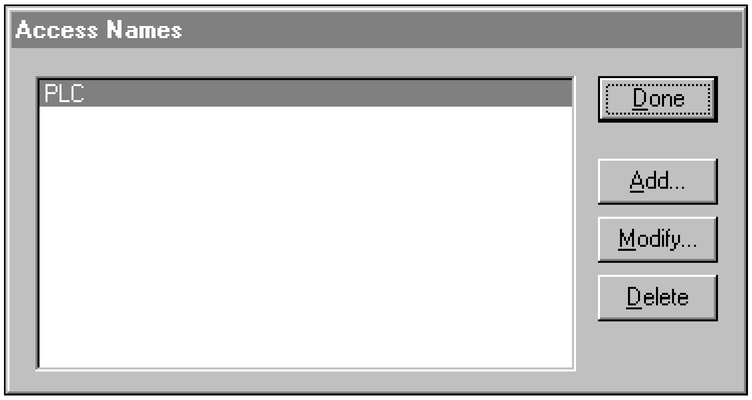
3. In the **Access Name** box, type **PLC**.
4. In the **Node Name** box, type **Node2**.
5. In the **Application Name** box, type **Genius**.
6. In the **Topic Name** box, type **PLC**.

   You can define any name you want for the **Access Name**, but the **Application Name** and **Topic Name** must reference the computer with the I/O Server.

7. Select the protocol that you are using.

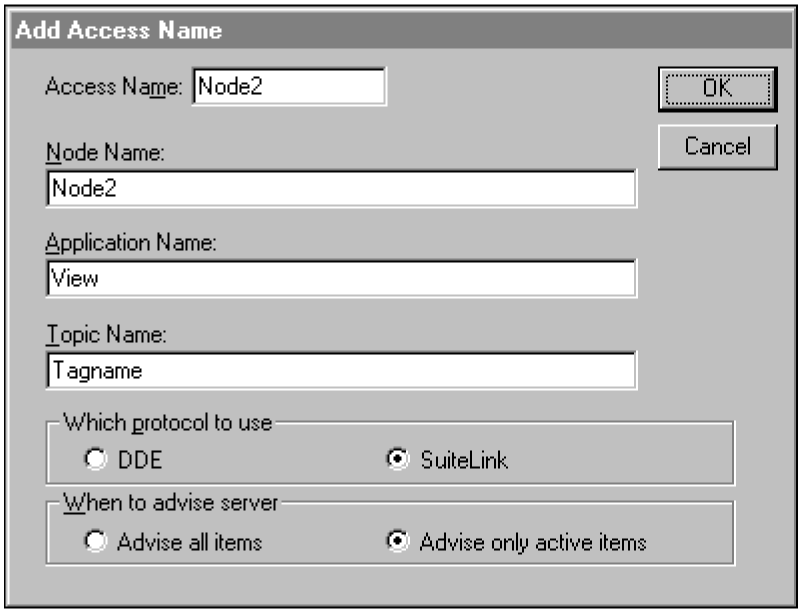
   The Access Name, PLC, will reference the I/O Server on Node2 so that all nodes, including Node2, will access PLC data from the GE Genius I/O Server on Node2. A Topic Name in the I/O Server must also be configured with the name "PLC"

8. Click **OK**. The **Access Names** dialog box reappears showing the new Access Name in its list:



9. Click **Done**.

10. Set up another Access Name for the global InTouch data as follows:


Add Access Name

Access Name: Node2

Node Name: Node2

Application Name: View

Topic Name: Tagname

Which protocol to use: ○ DDE ● SuiteLink

When to advise server: ○ Advise all items ● Advise only active items

The DDE Access Name "Node2" will reference InTouch data on the Master node so that Node1 and Node2 can read and write to Node2's tagnames.

11. Set up tagnames to be used in the application. For example:

Tagname	Type	Access Name	Item
Temp1	I/O Real	PLC	40001
Temp2	I/O Real	PLC	40002
Ack_System2	I/O Discrete	Node2	$System.Ack

Temp1 and Temp2 are tagnames that are read from the GE Genius I/O Server which in turn reads from the PLC. Ack_System2 is a tagname that will be used to write to **$System.Ack** on Node2 to acknowledge all alarms on that node. If the distributed alarm system is used, there is no need to create an Ack_System2Tag. This is only required for the standard alarm system.

For more information on acknowledging alarms see, Chapter 7 - Alarms/Events.

12. Copy the application to all necessary remote nodes, either manually or through NAD. Once you are finished, start NetDDE (for Windows 95) (MS Network DDE for NT, will start automatically), the I/O Server on Node2 and WindowViewer on all nodes.

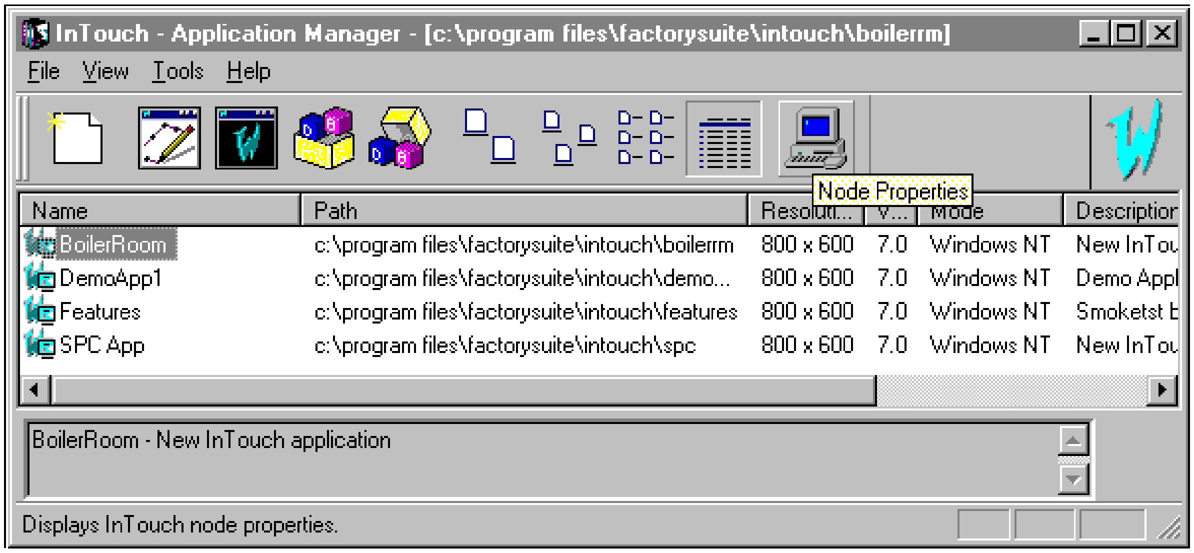
Configuring an InTouch Application for NAD

Network Application Development or NAD is an architecture that combines the best of the client-based and server-based architectures. NAD provides automatic notification of application changes and automatic distribution of the updated applications to View nodes. NAD can even be used to automatically distribute master/slave applications.

Note You cannot use the NAD features if you are using WindowViewer as an NT Service.

➢ **To configure an application for NAD:**

1. Start the InTouch program (INTOUCH.EXE). The **InTouch Application Manager** dialog box appears:

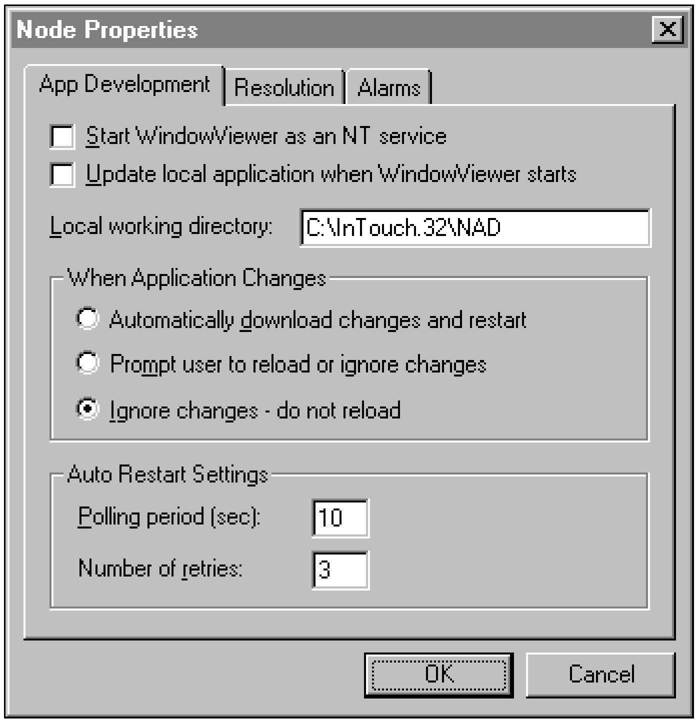

InTouch - Application Manager - [c:\program files\factorysuite\intouch\boilerrm]

Name	Path	Resoluti...	V...	Mode	Descriptior
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTou
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Appl
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst b
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTou

BoilerRoom - New InTouch application

Displays InTouch node properties.

2. Click the Node Properties tool or on the **File** menu, click **Properties**, or right-click a blank area of the application display window, and then click **Properties**. The **Node Configuration** dialog box appears with the **App Development** property page active:


Node Properties

App Development | Resolution | Alarms

[ ] Start WindowViewer as an NT service
[ ] Update local application when WindowViewer starts

Local working directory: C:\InTouch.32\NAD

When Application Changes
- ○ Automatically download changes and restart
- ○ Prompt user to reload or ignore changes
- [✓ ● Ignore changes - do not reload]

Auto Restart Settings
- Polling period (sec): 10
- Number of retries: 3

Note The **App Development** property sheet provides several options that allow you to specify how NAD will function. These settings are configured on each View node, **NOT ON THE DEVELOPMENT NODE**. This allows unique configurations for each View node.

When you run WindowViewer as an NT service, it allows continuous operation of WindowViewer through operating system log-ins, log-outs, for example, operator shift changes. By selecting this option you also allow automatic start up of InTouch following power failure or when the machine is turned off and on. By doing this, you provide unmanned station startup of WindowViewer without compromising NT operating system security. However, you cannot use NAD features if you are using WindowViewer as an NT service.

4. Select **Update local application when WindowViewer starts** if you want to copy the master application to the local working directory or View node on startup of WindowViewer.

   The initial copying of the master application may take longer than subsequent updates.

5. In the **Local working directory** box, type the directory that you want WindowViewer to copy the master application to.

   If this is the development node, you can type a local directory path, such as **c:\InTouch\NAD**. You can also type a networked remote UNC path, such as **\\node\share\path**. This is convenient for file server based networks where most file storage is kept in a central location. If this is a client node (runtime only), it will likely use a local directory path. If you do not specify a directory, WindowViewer automatically creates a local subdirectory named "NAD" in the directory from which WindowViewer is launched.

   It is recommended that you use a local directory whenever possible to prevent network delays and failures from affecting the operation of WindowViewer.

Warning! Do not use a "root" directory or a UNC pathname that points to a root directory. The View node will recursively delete all files and subdirectories in the specified destination application directory before copying the master application directory. Therefore, never use the path of the master application directory or a UNC to the master application directory.

This directory should be considered a temporary directory and no files should be saved to it except those copied by NAD itself.

For more information on UNC paths, see "Configuring UNC Paths for Files."

6. In the **When Application Changes** group, select the option for the action that you want WindowViewer to take when the master application changes.

Automatically download changes and restart	Automatically shuts down WindowViewer on the View node, copies the updated master application (if configured to do so), then restarts WindowViewer on the View node.
Prompt user to Reload or ignore changes	Causes an interactive message box to appear on the View nodes notifying the operator that the application has changed and asks if he wants to restart WindowViewer. If the operator selects **Yes** WindowViewer shuts down on the View node and the updated master application is copied from the development node (if configured to do so), then WindowViewer restarts. If the operator selects no, it will behave exactly as **Ignore changes - do not reload**.
Ignore changes - do not reload	Causes the View node to ignore any changes the development node made to the master application. Also used when customizing NAD update functions. For more information, see "Customizing NAD Update Function."

7. In the **Polling Period (sec)** box, type the number of seconds that WindowViewer will wait before checking the master application for changes.

Note Caution is advised when specifying this setting, a value too small will cause WindowViewer to spend too much time checking for master application changes. This can interfere with WindowViewer servicing the running application.

8. In the **Number of retries** box, type the number of attempts that will be made to shutdown and restart WindowViewer when the master application changes.

   This option is only valid if you have selected **Automatically download changes and restart**.

9. Click **OK**.

Customizing NAD Update Function

In addition to the three update options described above, NAD provides the following tools that you can also use to customize the update behavior of an application:

Tool	Description
$**ApplicationChanged**	Provides an indication when a master application has changed. This tagname could be used to cause a message to appear telling the operator that the master application has changed. You can also use the $**ApplicationChanged** system tagname in a data change script to build a node update notification script. This script could include launching your own dialog boxes or closing down certain processes. Then, you could use the **RestartWindowViewer()**to initiate the shutdown process.
RestartWindowViewer()	When this QuickScript function executes, it automatically shuts down WindowViewer, copies the updated master application (if configured to do so), and then restarts WindowViewer. For example, you could use this function on a button to allow an operator to choose an appropriate time for updating the application. The function can also be used in a QuickScript to automatically update at a specified time or between shifts.

Note To use these functions, in the **NAD Configuration** dialog box, the **When Application Changes** option must be set to: **Ignore changes - do not restart**. Setting this option prevents the system from interfering with customized functions defined.

📖 For additional information on these functions, see your *InTouch Reference Guide*.

Manually Notifying Clients of Application Changes

During application development, you can execute the **Notify Clients** command on the WindowMaker **Special** menu to automatically update InTouch client applications.

When you execute this command a flag is set to notify all remote View nodes that the master application has changed. These clients in turn, may automatically start an update process based on the parameters defined for each node.

The Application Copying Process

When the WindowViewer node copies an application, it makes every attempt to retain the attributes (read-only, system, hidden, and so on.) of the master application during the copy process. WindowViewer also copies all files and subdirectories of the master application. The copy process does not copy the following files: ***.WVW**, ***.LGH**, ***.LOG**, ***.IDX**, ***.LOK**, ***.FSM**, ***.WBK**, ***.CBK**, ***.DBK**, ***.GBK**, and ***.NBK**.

Note WindowViewer will recursively delete all files and subdirectories in the destination application directory. This directory should be considered a temporary directory (no files should be placed into it).

Application Editing Locks

InTouch applications can only be edited by one developer at a time. To prevent multiple developers from trying to edit an application, WindowMaker locks an application during the edit session. If you try to load an application into WindowMaker that has a lock created, you will receive a message telling you that the application cannot be edited because it is being edited by another computer. The name of the node editing the application will also be stated in the message.

Note If WindowMaker is abnormally shut down with an application loaded, the APPEDIT.LOK file is automatically deleted. However, you can manually remove the lock by deleting the APPEDIT.LOK file from the application directory.

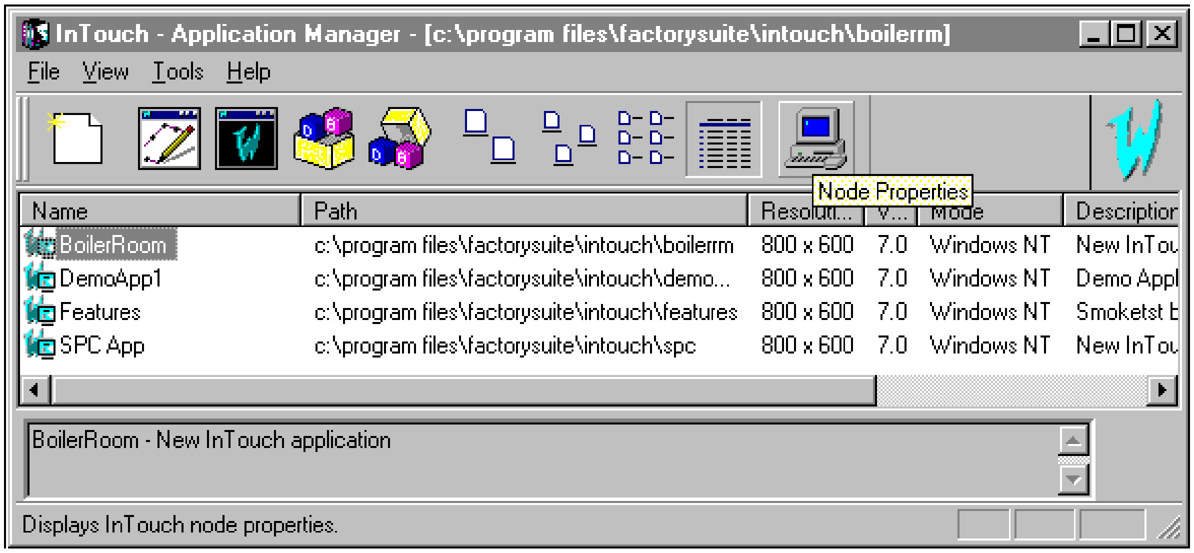
Dynamic Resolution Conversion (DRC)

Dynamic Resolution Conversion(DRC) works with other distributed features to provide independence from screen resolution restrictions. In a NAD architecture, an InTouch application is created and maintained on a development node, and then copied to several View nodes. DRC allows all of these nodes to view the application, even if they are running at different screen resolutions.

DRC enables each View node to scale the application to a number of user-defined options, including a custom resolution. This scaling takes place while WindowViewer compiles the application, and does not require WindowMaker. Since each View node can use a different DRC setting, each View node must have its own settings configured.

➤ **To configure an application for DRC:**

1. Start the InTouch program (INTOUCH.EXE). The **InTouch Application Manager** dialog box appears:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\boilerrm]
File View Tools Help
Node Properties

Name	Path	Resoluti...	V...	Mode	Descriptior
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTou
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Appl
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst t
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTou

BoilerRoom - New InTouch application

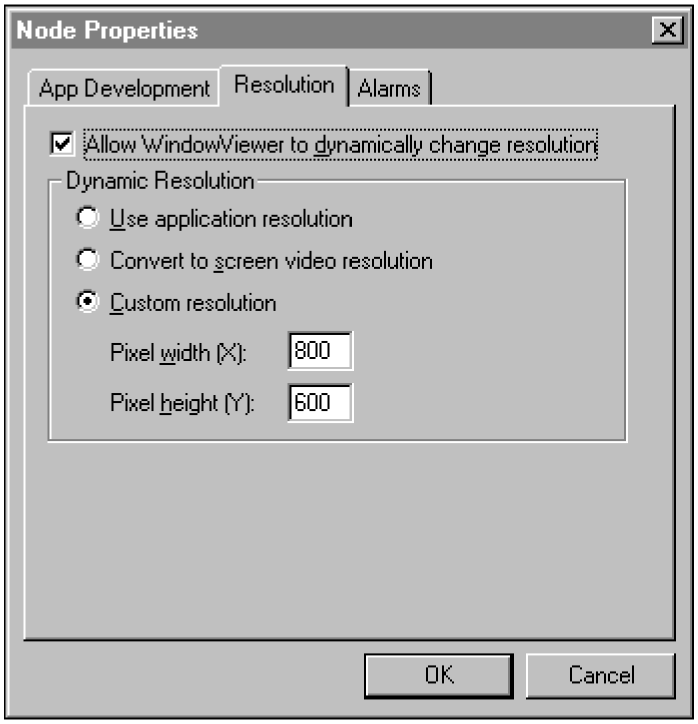
Displays InTouch node properties.

3. Click the Node Properties tool or on the **File** menu, click **Properties**. The **Node Configuration** dialog box will appear.

To quickly access the dialog box, right-click a blank area of the application display window, and then click **Properties**.

Note When an application is selected in the Application Manager window, selecting the **Properties** command on the **File** menu will display the **Properties** dialog box for that application.

4. Click the **Resolution** tab:



☑ Allow WindowViewer to dynamically change resolution

Dynamic Resolution

○ Use application resolution

○ Convert to screen video resolution

● Custom resolution

Pixel width (X): 800

Pixel height (Y): 600

5. Select **Allow WindowViewer to dynamically change resolution** if you want WindowViewer to locally scale the master application, based on the resolution option you select. (The three resolution options are described below.)

   If you do not select this option, WindowViewer will only run the application if the node's screen resolution is identical to the screen resolution of the application development node. If the resolutions are different, WindowViewer prompts the operator to run WindowMaker to convert the application to the node's resolution. Use caution when doing this if you have set up a UNC path to the master application directory as this will only modify the original application.

6. Select **Use Application resolution** if you want WindowViewer to run the application at the resolution it was developed for and ignore the node's resolution. For example, if the application was developed at 640x480 and the node's resolution is 1024x768, WindowViewer will not dynamically scale the application. Instead, the application will be displayed at 640x480.

7. Select **Convert to screen video resolution** if you want WindowViewer to run the application at the node's resolution and ignore the resolution the application was developed at. For example, if the node is running at 640x480 and the application was developed at 1280x1024, WindowViewer will dynamically scale the application (smaller) to fit the node's 640x480 display. (This will more than likely be the most commonly used setting.)

8. Select **Custom Resolution** if you want WindowViewer to run the application at the resolution you specified in the **Pixel width (X)** and **Pixel height (Y)** (must be integer values) boxes. The application's resolution and the node's resolution are both ignored. For example, if **Pixel width (X)** and **Pixel height (Y)** are set to 512 and 384, respectively, the application will dynamically be scaled to fit in a 512x384-pixel area on the node's display.

9. Click **OK**.

Working with Multiple Monitor Systems

There are several advanced graphics adapter cards on the market today that allow you to have more than one VGA monitor connected to your system at a time. These monitors act in tandem, creating a virtual screen which can be very large. As an example, a popular system connects four 17" monitors stacked as a cube: two on the bottom and two on the top. Since each screen has a resolution of 800x600, the virtual screen created is 1600x1200 pixels.

Dynamic Resolution Conversion (DRC) makes it easy to support these multi-monitor systems. Simply select from the DRC resolution conversion options, and you can take full advantage of all the virtual display or just a portion of it.

If an application is scaled to run on an even number of the monitors, a problem exists when certain dialogs are displayed over the span of the monitors. One of these dialogs, the **Keypad**, can cause particular problems as certain keys may not be accessible. To solve the problem, InTouch provides several multi-monitor configuration options.

➢ **To configure the multi-monitor settings on a node:**

1. Using a suitable text editor, for example, Windows Notepad open the **WIN.INI** file located in your Windows directory.
2. Locate the **[InTouch]** section and add the following parameters:
3. **[InTouch]**

MultiScreen=1	turns on multi-screen mode
MultiScreenWidth=640	width in pixels of a single screen
MultiScreenHeight=480	height in pixels of a single screen

For example, if your computer's resolution is 2560 x 1024 split on two horizontal screens, enter the following:

[InTouch]
MultiScreen=1
MultiScreenWidth=1280
MultiScreenHeight=1024

Note The above entries affect the numeric keypad and the QWERTY keyboard. Other InTouch dialog boxes and option boxes are not affected.

Running WindowViewer as an NT Service

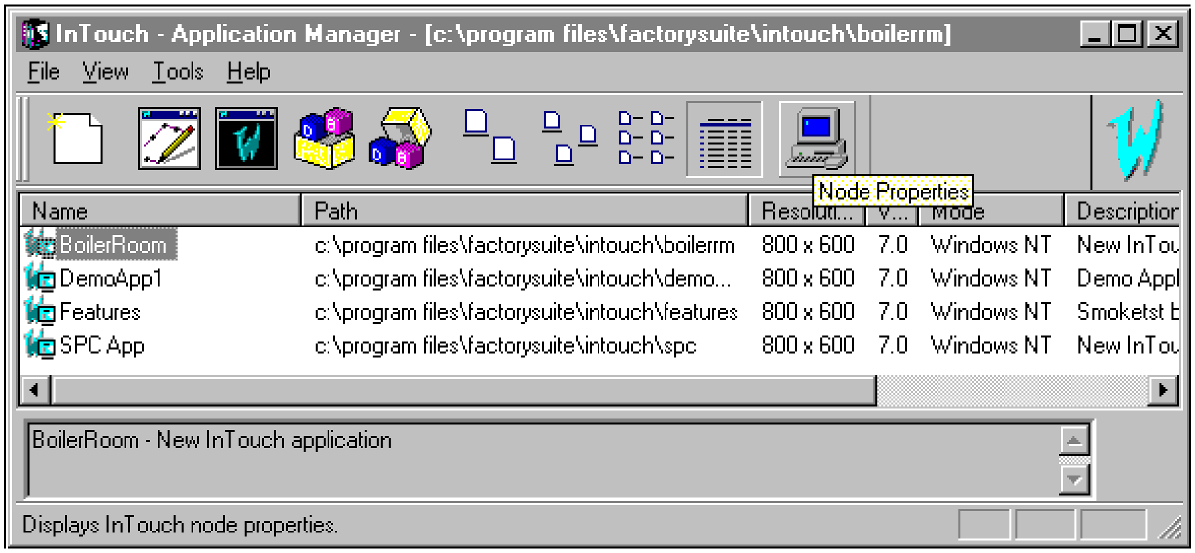
Beginning with InTouch 7.0, you can create client-server configurations very easily. You can configure a node that acts as a server node. This server node can then store the Tagname Dictionary and historical log data, execute InTouch QuickScripts, provide an alarming facility and I/O data. Any client node can then retrieve this information from the server node and display graphics.

Running WindowViewer as an NT Service allows you to take advantage of all the features that an NT Service provides. For example, continuous operation after the operator logs off and automatic startup at system boot time without operator intervention. This allows unmanned station startup of WindowViewer without compromising NT operating system security.

Note All NAD features are disabled when WindowViewer is installed as an NT Service.

➢ **To configure WindowViewer to start as an NT service:**

1. Start the InTouch program (INTOUCH.EXE). The **InTouch Application Manager** dialog box appears:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\boilerrm]
File View Tools Help
Node Properties

Name	Path	Resolu...	V...	Mode	Description
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTou
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Appl
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst b
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTou

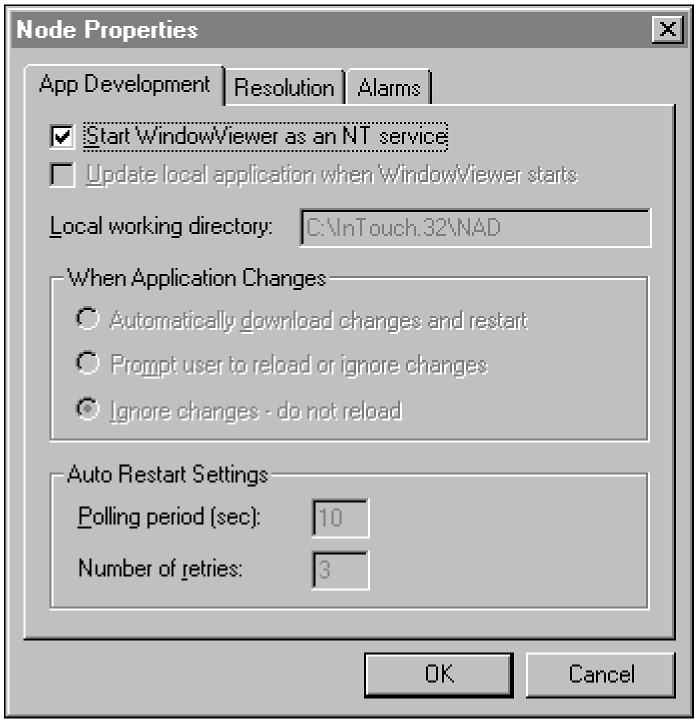
BoilerRoom - New InTouch application

Displays InTouch node properties.

2. Click the Node Properties tool or, on the **File** menu, click **Properties**. The **Node Configuration** dialog box appears with the **App Development** property page active:

   To quickly access the dialog box, right-click a blank area of the Application Manager's window, and then click **Properties**.

Note When an application is selected in the Application Manager window, selecting the **Properties** command on the **File** menu will display the **Properties** dialog box for that application.


Node Properties

- [✓] Start WindowViewer as an NT service
- [ ] Update local application when WindowViewer starts

Local working directory: C:\InTouch.32\NAD

When Application Changes
- ○ Automatically download changes and restart
- ○ Prompt user to reload or ignore changes
- ● Ignore changes - do not reload

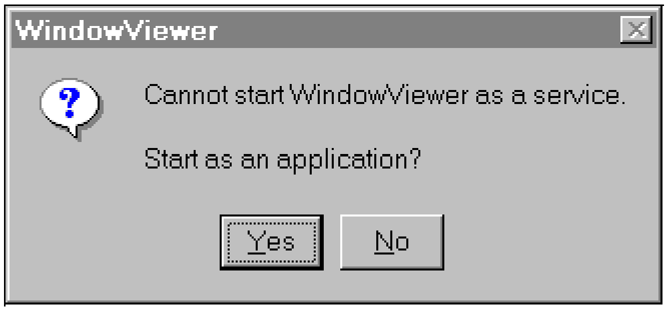
Auto Restart Settings
- Polling period (sec): 10
- Number of retries: 3

3. Select the **Start WindowViewer as an NT service** option to configure WindowViewer to automatically run as an NT service.

4. Click **OK**.

Notes

1) If WindowViewer is configured as an NT service and subsequently started directly (from its icon, the Windows startup menu and so on), there will be approximately a 15 second delay before WindowViewer will display a window. This delay is due to WindowViewer attempting to connect to the NT Service Control Manager. Upon failing to connect to the Service Control Manager, WindowViewer will display the following message box:


WindowViewer

Cannot start WindowViewer as a service.

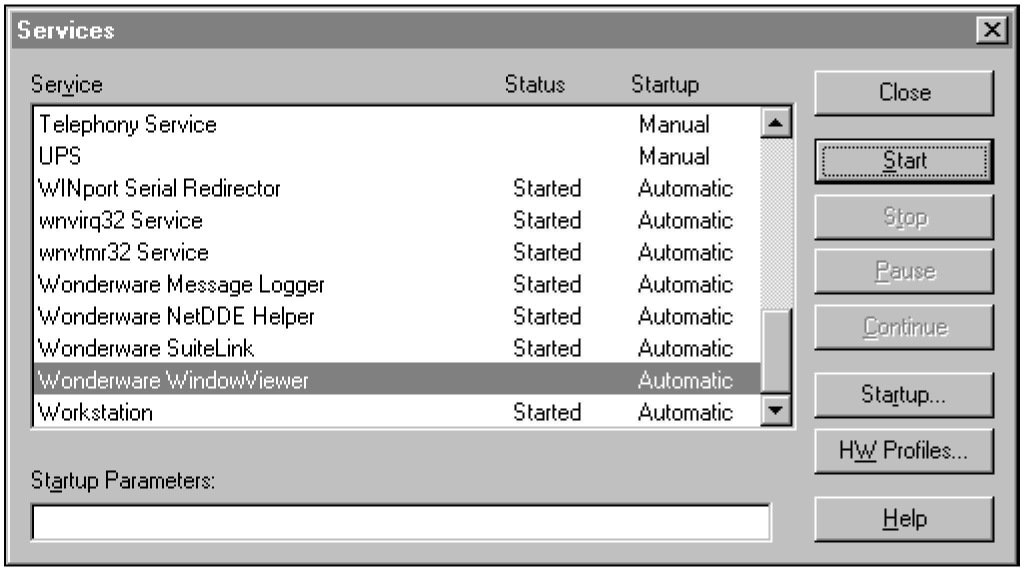
Start as an application?

If you click **Yes**, WindowViewer is started as an application not an NT service. If click **No** the command to start WindowViewer is canceled.

2) If you stop WindowViewer from running as an NT service by clearing the **Start WindowViewer as an NT service** option, WindowViewer service is automatically uninstalled. However, it can be run as an application.

➢ **To start WindowViewer service from Services dialog box:**

1. In the Windows Control Panel, double-click **Services**. The **Service** dialog box appears:



Service	Status	Startup
Telephony Service		Manual
UPS		Manual
WINport Serial Redirector	Started	Automatic
wnvirq32 Service	Started	Automatic
wnvtmr32 Service	Started	Automatic
Wonderware Message Logger	Started	Automatic
Wonderware NetDDE Helper	Started	Automatic
Wonderware SuiteLink	Started	Automatic
Wonderware WindowViewer		Automatic
Workstation	Started	Automatic

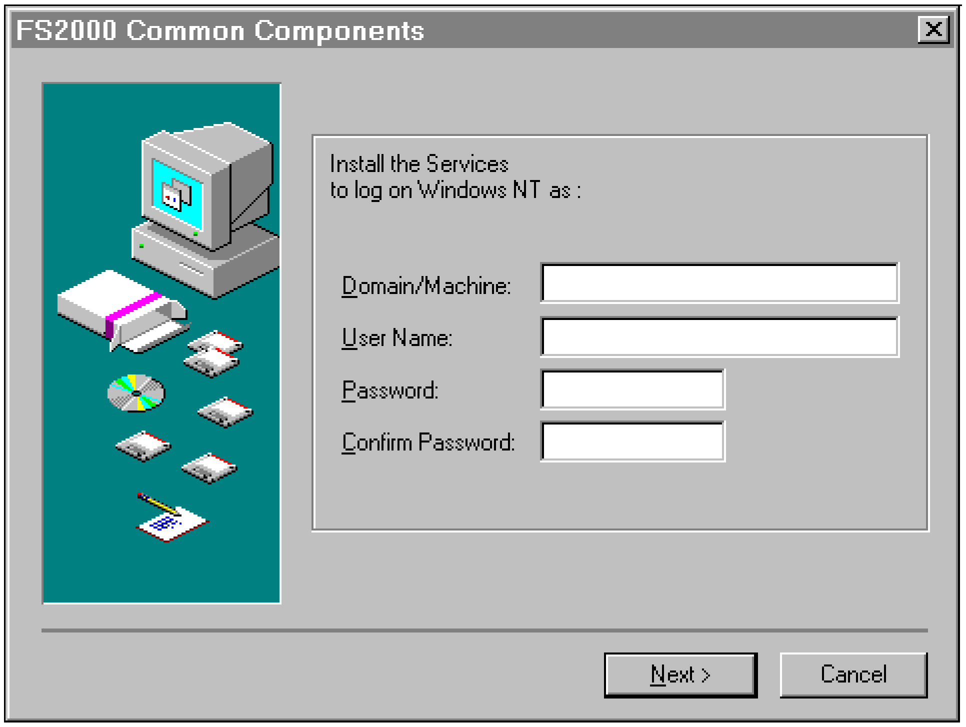
2. Select **Wonderware WindowViewer**, and then click **Start**.
3. Click **Close**.

   After you perform these steps, WindowViewer can be started as both an NT service and as an application.

Configuring System Privileges

During InTouch installation, you will be prompted to provide your user name and password for your administrative account. This information is used to set up your NT user impersonation account. The Wonderware services such as, Wonderware NetDDE Helper and Wonderware WindowViewer, will use this information for automatically logon, and for automatically starting the appropriate services during unattended start up.

The User Name and Password that you provide must be a valid Administrator level logon configured through the NT User Manager.

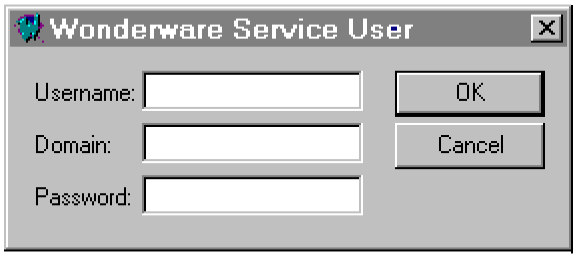

FS2000 Common Components

Install the Services
to log on Windows NT as :

Domain/Machine:
User Name:
Password:
Confirm Password:

1. In the **Domain/Machine** box, type the name of your system domain or the node name.
2. In the **User Name** box, type your user identification.
3. In the **Password** box, type your system password.
4. In the **Confirm Password** box, retype you system password to verify it.

   After installation, if you need to alter this information, run the Wonderware Service User application (WWUSER.EXE) located in your installed directory. For example, \Program Files\FactorySuite\Common. When you run this utility, the **Wonderware Service User** dialog box appears:

   
   Wonderware Service User

   Username:
   Domain:
   Password:

   Enter the information as described above.

Distributed Applications and Time Zones

InTouch provides services that ease the use of applications across multiple time-zones. These services are used by both the alarm and history systems to permit values to be viewed at the local time they occurred. For example, if an engineer in California is viewing an alarm that occurred in a manufacturing plant in Kansas at 10AM, that engineer would see the local California time that the alarm occurred; 8AM. The same is true if the engineer is viewing historical data from that plant.

The key to these services is the use of UCT (Universal Coordinated Time), also known as Greenwich Mean Time or GMT, as the time reference. Each computer is configured with both the local time and a UCT offset for the time zone where it resides. In the above example, the time zone for the computer in California is set at GMT eight hours, while the time zone for the computer in Kansas is set at GMT six hours.

InTouch uses these GMT offsets as the basis for retrieving all alarm and historical data. In the above example, when the InTouch application in California received the alarm from the application in Kansas, it also looked at the GMT offset of both computers to determine the local California time that the alarm occurred. Thus, a 10AM alarm in a UCT six time zone equals an 8AM alarm in a GMT eight time zone. To use this feature, the GMT offset must be configured for each computer.

➢ **To set your time zone when using the Windows 95 operating system:**

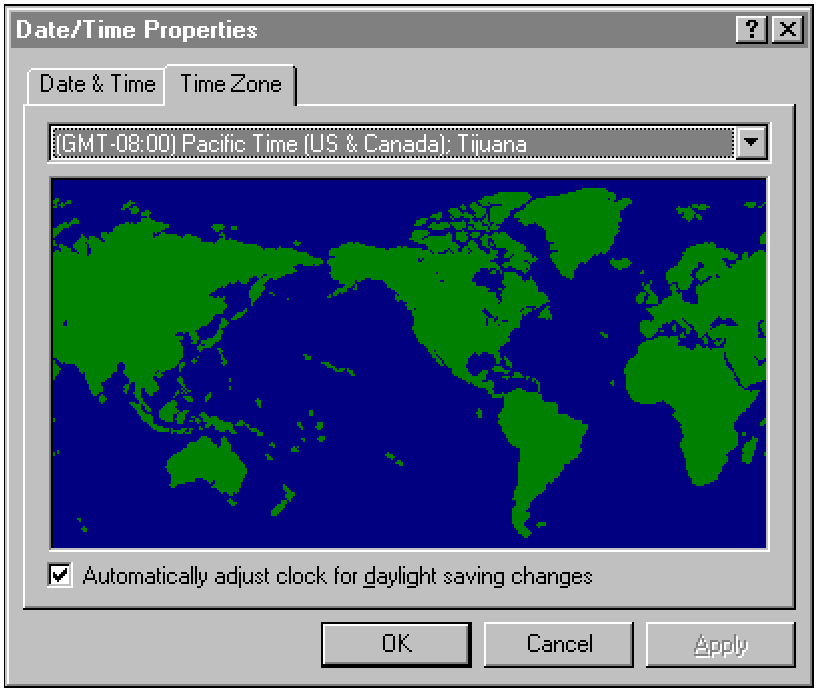
The setting, **set TZ=GMT[+ | -]*X***, must be included in the AUTOEXEC.BAT of the computer.

Where: *X* is the offset from GMT for the time zone the computer resides in.)

For example, to set the **TZ** environment variable to correspond to the current time zone in California, you could use either **set TZ=GMT8** or **set TZ=GMT+8**.

➢ **To set your time zone on the Windows NT operating system:**

1. Open the Windows Control Panel.
2. Double-click the **Date/Time** icon or double-click the clock in the Windows **Taskbar**. The **Date/Time Properties** dialog box appears:



Date/Time Properties

Date & Time | Time Zone

(GMT-08:00) Pacific Time (US & Canada); Tijuana

☑ Automatically adjust clock for daylight saving changes

3. Click the **Time Zone** tab, and then click the arrow to open the list of time zones.
4. Select your time zone in the list.
5. Click **OK**.

Note The Windows 95 and Windows NT operating systems may be set to automatically adjust the clock for daylight savings time. It is recommended that this feature be disabled. To disable this feature, use the Date/Time utility in the Windows Control Panel or double-click the clock on the Windows **Taskbar**.

Automatically Changing the System Time

Windows 95 and Windows NT will attempt to automatically adjust the clock for daylight savings time. You should turn this feature off by using the Date/Time utility in the Windows Control Panel and use InTouch QuickScripts to automatically set the clock at these times.

➢ **To set the clock forward in the spring:**

Create the following Condition QuickScript:

```
$Year == yyyy and $Month == 04 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 0 ;
```

where: yyyy = the year (i.e., 1993, 1994, or 1995 ...)
dd = the date of the time change
DaylightSavingsTime = a user-defined memory discrete tagname to indicate daylight savings time

ON TRUE:
```
DaylightSavingsTime = 1;
StartApp "c:\windows\control.exe" ;
SendKeys "%(st)" ;
SendKeys "%(t)" ;
SendKeys "03" ;
SendKeys "~" ;
SendKeys "%({F4})" ;
```

➢ **To set the clock back in the fall:**

Create the following Condition QuickScript:

```
$Year == yyyy and $Month == 10 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 1 ;
```

where: yyyy = the year (i.e., 1993, 1994, or 1995 ...)
dd = the day of the time change for that year
DaylightSavingsTime = a user-defined memory discrete tagname to indicate daylight savings time

Note Whenever a systems clock is set back, the historical logging engine could overwrite existing data in the historical log file. To prevent this loss of data, we recommend that you first back up the log files before changing the clock back.

ON TRUE:
```
DaylightSavingsTime = 0;
StartApp "c:\windows\control.exe" ;
SendKeys "%(st)" ;
SendKeys "%(t)" ;
SendKeys "01" ;
SendKeys "~" ;
SendKeys "%({F4})" ;
```

Distributed Alarms

InTouch provides two alarm systems: standard and distributed. Both provide services to display, log, print, and acknowledge process alarms and system events. The standard system is used to display and acknowledge events and alarms generated by the local InTouch application. The distributed system expands this scope to allow the display and acknowledgment of alarms generated by the local alarm systems of other InTouch applications.

Both the standard and distributed systems can be used in a distributed application. The major difference is that the standard system is limited to only those alarms generated by an identical InTouch application, while the distributed system has no such limitation.

For more information on setting up and configuring the distributed alarm system, see Chapter 7 - Alarms/Events.

Distributed History

InTouch provides a distributed history system that allows retrieval of historical data from any InTouch 5.6 (or later) application, even those across a network. This system extends the capabilities of the standard InTouch history by allowing remote retrieval of data from multiple historical databases simultaneously. These databases are referred to as history providers. Up to eight history providers can be displayed simultaneously, one for each historical trend chart pen.

Note History providers can be configured as native InTouch history or IndustrialSQL (InSQL) history providers.

For more information on setting up and configuring the distributed history system, see Chapter 8 - Real-time and Historical Trending.

C H A P T E R 4

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.

The two database utility programs, DBDump and DBLoad are also described in this chapter. DBDump allows you to export an InTouch application Tagname Dictionary as a text file which can be accessed from another package such as Microsoft Excel for modifying, storing, etc. DBLoad allows a database of tagnames created in another package such as Excel or a DBDump file from another InTouch application to be loaded into an existing InTouch application.

Contents

Tagname Dictionary Special Features

Beginning with InTouch Version 7.0, the Tagname Dictionary has been enhanced to include the following special features:

Feature	Description
Tag Browser	The Tag Browser is used for selecting tagnames and tagname **.field**s, remote tagname references and SuperTag member tagnames from FactorySuite applications, or any other tag source that supports the Tagname Dictionary interface.
Tagname Cross Referencing	Tagname Cross Referencing allows you to cross reference a tagname to the locations where it is used in your application including windows, scripts, SQL configuration, SPC triggers, and so on. You can print the cross reference information or save it to a file.
SuperTags	InTouch supports a SuperTag structure that allows you to define composite tagname types. You can define SuperTag templates with up to 64 member tagnames and 2 nesting levels. Member tagnames behave exactly like normal tagnames and they support trending and alarming and all tagname **.fields**.
Remote Tagname References	Remote tagname references allow InTouch to access data from an I/O server without creating a tagname in the local Tagname Dictionary. Remote references allow you to import or export a window or QuickScript without requiring conversion of the tagnames from placeholders.
Extended Tagname Support	InTouch can support up to 61,405 tagnames in its Tagname Dictionary. (The number of tagnames that your system supports is determined by your software license.)

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 a I/O Server, it must be a 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. You use them to create system constants and simulations. You can also use them to create calculated variables that are accessed by other Windows programs. For example, you can define a memory tagname with the initial value of 3.1416 or, you could store recipes in groups of memory tagnames. In simulations, you can use memory tagnames to control the actions of a background QuickScript. For example, you could define a memory tagname "COUNT" what 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 $\pm 3.4e^{38}$. 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.

For more information on using I/O tagnames, see Chapter 9 - I/O Communications.

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. 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. You use **Group Var** type tagnames 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 that the operator clicks 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.

For more information on Alarms, see Chapter 7 - Alarms/Events.

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 executed, Pen1 in the historical trend associated with the **Hist Trend** tagname "MyHistTrendTag," would begin trending the historically logged data for the "MyLoggedTag."

For more information on using **Tag ID** tagnames, see your *InTouch Reference Guide*.

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, and so on. 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";  or  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.

SuperTags

InTouch SuperTags allow you to define composite tagname types. You can define SuperTags with up to 64 member tagnames and 2 nesting levels. Member tagnames behave exactly like normal tagnames. They support trending, alarming and all tagname **.fields**.

For more information on SuperTags see, "Creating InTouch SuperTags."

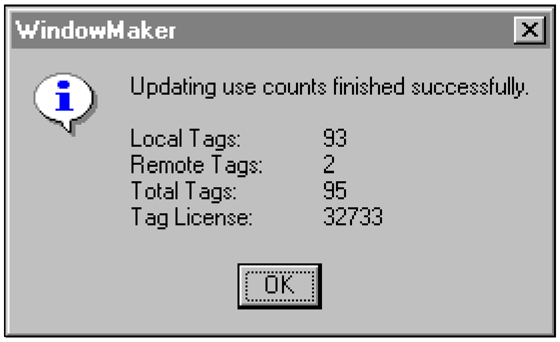
Extended Tagname Support

InTouch can support up to 61,405 tagnames in its Tagname Dictionary. The number of tagnames that your system supports is determined by your software license.

➢ **To determine the tagname support for your system:**

1. Close all your windows.
2. On the **Special** menu, click **Update Use Counts**.

   A message box will appear telling you that updating use counts can take quite a while. You may at that time cancel the command or continue.
3. Click **Yes** to continue updating the use counts.
4. Once the system has completed updating the use counts, the following dialog box appears:



WindowMaker

Updating use counts finished successfully.

```
Local Tags:      93
Remote Tags:     2
Total Tags:      95
Tag License:     32733
```

5. The **Tag License** line will display the number of tagnames supported by your license.
6. Click **OK**.

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 an 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 and so on.

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 mean that you simply 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.)

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

☞ For more information on the Tag Browser, see "The Tag Browser."

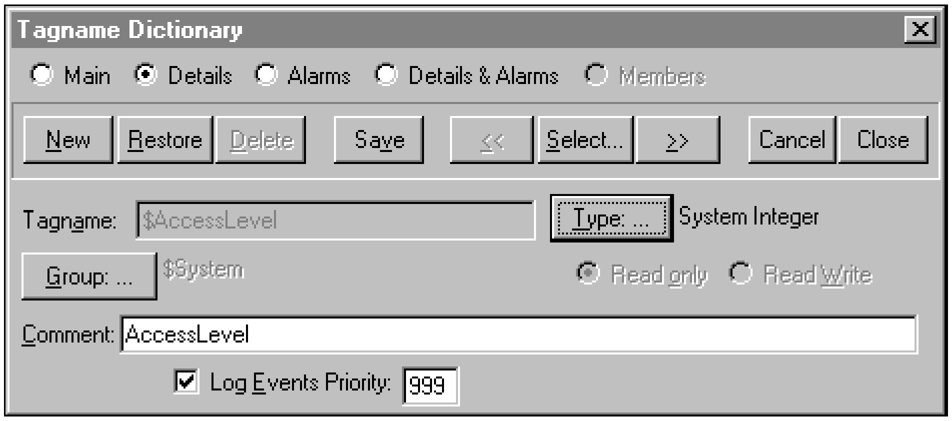
The options at the top of the **Tagname Dictionary** dialog box are used to display the various tagname detail level dialog boxes as follows:

Dialog Box	Description
Main	Displays the main tagname dictionary dialog box (show above). In the case of SuperTags, **Main** shows only the parent or root tagname. Any changes made to the parent or root tagname can overwrite Member tagname information. After making a change, click **Save**. A message box will appear asking if you want to overwrite member tagnames with the root tagname changes.
Details	Displays the respective details level dialog box for the tagname type selected.
Alarms	Displays the respective alarms configuration dialog box for the tagname type selected.
Details & Alarms	Displays the both respective details and alarm configuration dialog boxes for the tagname type selected.
Members	Displays the member details dialog box for a SuperTag type tagname.

If you right-click any of the text entry boxes in any of the Tagname Dictionary dialog boxes, a menu will appear displaying the commands that you can apply to the selected text.

➢ **To define a new tagname:**

1. On the **Special** menu, click **Tagname Dictionary** or, in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box appears:

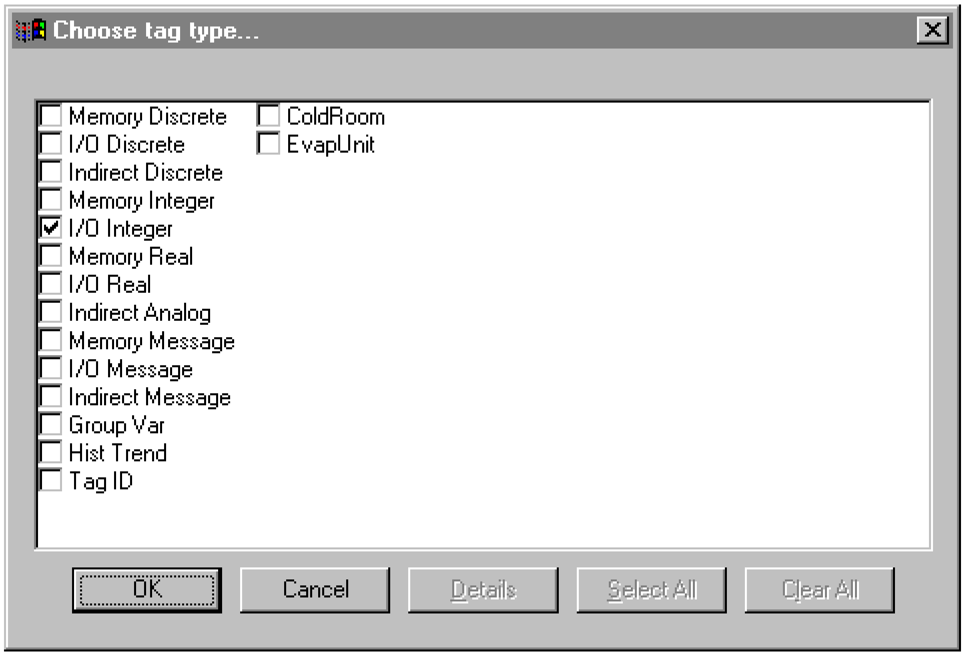


Tagname: $AccessLevel
System Integer
$System
[✓ Read only] Read Write
Comment: AccessLevel
[✓] Log Events Priority: 999

2. Click **New**. (The **Tagname** box clears.)
3. In the **Tagname** box, type the name you want to use for the 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 &.

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



Memory Discrete
I/O Discrete
Indirect Discrete
Memory Integer
[✓] I/O Integer
Memory Real
I/O Real
Indirect Analog
Memory Message
I/O Message
Indirect Message
Group Var
Hist Trend
Tag ID
ColdRoom
EvapUnit

5. Select the type for the tagname, then click **OK**. The respective details dialog box for the selected type will appear. (The detail dialog boxes are described later in this chapter.)

   The names of any SuperTag 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.

For more information on tagname types, see "Tagname Types."

For more information on creating InTouch SuperTags see, "Creating InTouch SuperTags."

Note If a tagname is currently linked to an object or used in a QuickScript, its type can only be changed when WindowViewer is not running.

---

6. Click **Group** to assign the tagname to a specific Alarm Group. The **Alarm Groups** dialog box will appear. Select the name of the Alarm Group that you want to assign to the tagname, then click **Done**.

Note If you do not assign the tagname to a specific Alarm Group, by default, InTouch will assign it to the root group, **$System**.

Once you create a tagname and assign it to an Alarm Group, if you do not close the dialog box, all subsequent tagnames that you define will be assigned to the same Alarm Group, unless you change it.

For more information on defining Alarm Groups, see Chapter 7 - Alarms/Events.

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

Note The distributed alarm system can be configured to use the tagname **Comment** box to store an operator's comments regarding an acknowledged alarm.

The first time you access the **Tagname Dictionary** dialog box, the default comment for the internal system tagname **$AccessLevel** will be displayed in the **Comment** box. You should delete this comment to prevent it from being associated with any tagnames that you define. To delete the comment, select it and press the DEL key.

For more information on using the distributed alarm system, see Chapter 7 - Alarms/Events.

10. Select **Log Data** if you want the tagname to be logged to the historical log file during runtime whenever its engineering unit value changes more than th e **Log Deadband**value you specify or, by default once an hour, regardless of change.

Notes

1) In order for your tagnames to actually be logged, you must enable logging through the **Configure Historical Logging** command on the **Special** menu.

2) If you decide to later clear this option so the tagname will not be logged, the data previously logged for the tagname will not be accessible. Also, if you make changes in WindowMaker to logging while WindowViewer is running, they will not take affect until WindowViewer is restarted.

---

11. Select **Log Events** if you want 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, 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.

    For more information on events and priorities, see Chapter 7 - Alarms/Events.

12. Select **Retentive Value** if you want 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. When you select this option, the initial value of the tagname will constantly be updated to reflect the current value of the tagname. When WindowViewer is exited, the initial value is set based on the last retained value. If this option is later cleared, the initial value of the tagname will be set to the last retained value.

13. Select **Retentive Parameters** if you want 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.

    Note Since changes are logged immediately, we strongly recommended that you only select the above two retentive options for values that do not change often.

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 exists that you use 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 completed defining the basic tagname, you will need to define the details for the tagname and, if required, the alarm conditions. The steps that you need to follow to define the details for each tagname type are described in the following sections.

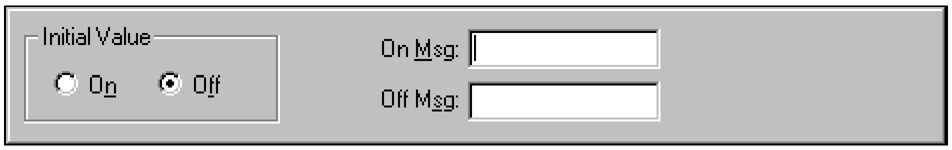
Defining Memory Discrete Tagname Details

Memory discrete type tagnames exist internally within your InTouch application. You define a **Memory Discrete** type tagname when you need an internal tagname with a value of either 0 (False, Off) or 1 (True, On).

➢ **To define the details for a memory discrete tagname:**

1. When you select **Memory Discrete** as the type for your tagname, the following details dialog box will appear.

   If it does not appear, click **Details** at the top of the **Tagname Dictionary** dialog box.


Initial Value
On [✓ Off]
On Msg:
Off Msg:

2. Click the **Initial Value** that you want stored in the tagname when the runtime database is first loaded.
3. If you define a discrete alarm state for this tagname that is "on" when the tagname's value is equal to 1 (On, True), type the message in the **On Msg** box that you want to be displayed in your alarm window's value/limit field.
4. If you define a discrete alarm state for this tagname that is "on" when the tagname's value is equal to 0 (Off, False), type the message in the **Off Msg** box that you want to be displayed in your alarm window's value/limit field.
5. If you want to define alarm conditions for the tagname, click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog to display the respective alarm conditions dialog box for the type of tagname you are defining.

   For more information on alarms, see Chapter 7 - Alarms/Events.
6. Once you have completed defining your tagname, click **Done** to save your tagname definition and close the tagname dialog boxes.

Defining Memory Analog Tagname Details

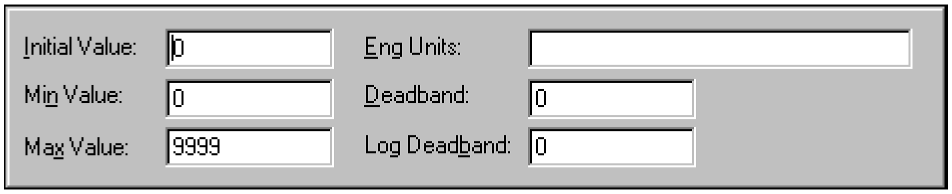
Memory analog type tagnames exist internally within your InTouch application. There are two memory analog types: **Memory Integer** and **Memory Real**. You define a **Memory Integer** type tagname when you need an internal tagname with a 32-bit signed integer value between -2,147,483,648 and 2,147,483,647.

You define a **Memory Real** type tagname when you need an internal tagname with a floating point value 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.)

➢ **To define the details for a memory analog tagname:**

1. When you select **Memory Integer** or **Memory Real** as the type for your tagname, the following details dialog box will appear.

   If it does not appear, click **Details** at the top of the **Tagname Dictionary** dialog box.



Initial Value:	0	Eng Units:	
Min Value:	0	Deadband:	0
Max Value:	9999	Log Deadband:	0

2. In the **Initial Value** box, type the value you want stored in the tagname when the runtime database is first loaded.
3. In the **Min Value** box, type the minimum value you want to use for Historical Trend charts, I/O and the **.Min EU** tagname .field.
4. In the **Max Value** box, type the maximum value you want to use for Historical Trend charts, I/O and the **.Max EU** tagname .field.
5. In the **Eng Units** box, enter the label you want to use for the tagname's engineering units.
6. In the **Deadband** box, type the amount the tagname's engineering units can change before the database is updated.
7. In the **Log Deadband** box, type the amount the tagname's engineering units must change before the tagname is logged to the historical log file.

Notes

1) You must select **Log Data** for the tagname if you want it to be logged to disk when its engineering units change more than the **Log Deadband** value.

2) If you change the **Log Deadband** value while WindowViewer is running, your changes will not take effect until historical logging has been stopped and restarted.

For more information on historical logging, see Chapter 8 - Real-time and Historical Trending.

8. If you want to define alarm conditions for the tagname, click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog to display the respective alarm conditions dialog box for the tagname type you are defining.

   For more information on alarm conditions, see "Defining Tagname Alarm Conditions."

9. Once you have completed defining your tagname, click **Close** to save your tagname definition and close the tagname dialog boxes.

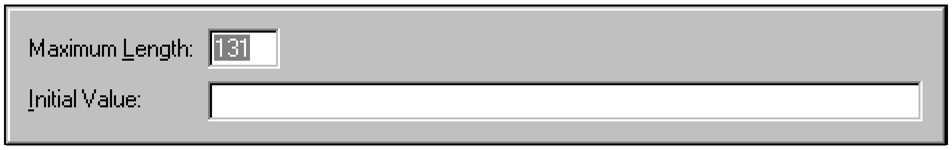
Defining Memory Message Tagname Details

Memory message type tagnames exist internally within your InTouch application. You define a **Memory Message** type tagname when you need an internal text string tagname that can be up to 131 characters long.

➢ **To define the details for a memory message tagname:**

1. When you select **Memory Message** as the type for your tagname, the following details dialog box will appear.

   If it does not appear, click **Details** at the top of the **Tagname Dictionary** dialog box.



Maximum Length: 131

Initial Value:

2. In the **Maximum Length** box, type the maximum number of characters to be allowed in the tagname's message. (InTouch allows a maximum of 131, which is displayed as the default.)
3. In the **Initial Value** box, type the text string that you want displayed for the tagname when WindowViewer is initially started.
4. Once you have completed defining your tagname, click **Close** to save your tagname definition and close the tagname dialog boxes.

Defining I/O Discrete Tagname Details

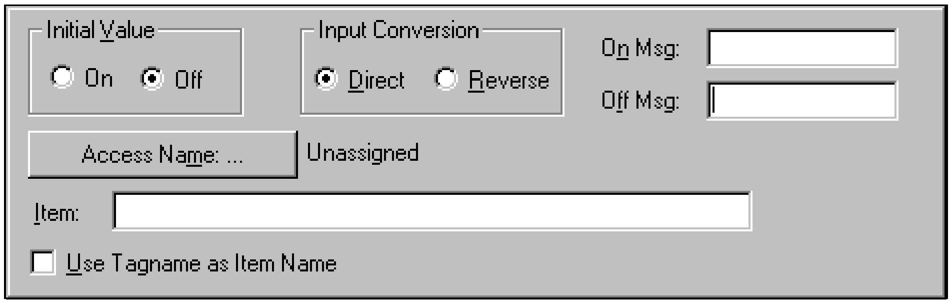
All tagnames that read or write their values to or from another Windows program are I/O type tagnames. This includes all inputs and outputs from programmable controllers, process computers, other Windows programs and data from network nodes.

You define an **I/O Discrete** type tagname when you need an I/O tagname with a value of either 0 (False, Off) or 1 (True, On).

➢ **To define the details for a I/O discrete tagname:**

1. When you select **I/O Discrete** as the type for your tagname, the following details dialog box will appear.

   If it does not appear, click **Details** at the top of the **Tagname Dictionary** dialog box.



Initial Value: On [✓ Off]

Input Conversion: [✓ Direct] Reverse

On Msg:

Off Msg:

Access Name: ... Unassigned

Item:

Use Tagname as Item Name

2. Click the **Initial Value** that you want stored in the tagname when the runtime database is first loaded. (**Off** equals **0**, **On** equals **1**.)

3. Click the **Input Conversion** that you want applied to the value when the runtime database is updated:

   Direct - The I/O input value is read unchanged directly from the server program.

   Reverse - The I/O input value is reversed when read from the server program. For example, if the I/O input value in the server program is 0, InTouch will automatically reverse it, save it and display a 1.

4. If you define a discrete alarm state for this tagname that is "on" when the tagname's value is equal to 1 (On, True), type the message in the **On Msg** box that you want to be displayed in your alarm window's value/limit field.

5. If you define a discrete alarm state for this tagname that is "on" when the tagname's value is equal to 0 (Off, False), type the message in the **Off Msg** box that you want to be displayed in your alarm window's value/limit field.

6. Click **Access Name** to define or select the Access Name that you want to assign to this tagname. (If an Access Name already appears to the right of this button, and you do not define or select a different one, it will be assigned to the tagname.

   For more information on Access Names, see Chapter 9 - I/O Communications.

7. In the **Item** box, type the valid item name for the data point in the server program that the tagname will read/write its value to/from. For example, if you want to read a value from a register in a PLC, enter the valid identification for that register as the item name.

   Item names are auto-indexed. For example, if you enter and save item name R4001, then click **New** (to define a new tagname), the item name will automatically be indexed to R4002. If an item name 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 and so on.

8. Select the **Use Tagname as Item Name** option if you want to use the tagname for the item name.

9. If you want to define alarm conditions for the tagname, click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog to display the respective alarm conditions dialog box for the tagname type you are defining.

   For more information on alarm conditions, see "Defining Tagname Alarm Conditions."

10. Once you have completed defining your tagname, click **Close** to save your tagname definition and close the tagname dialog boxes.

Defining I/O Analog Tagname Details

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, other Windows programs and data from network nodes. There are two memory analog types: **I/O Integer** and **I/O Real**.

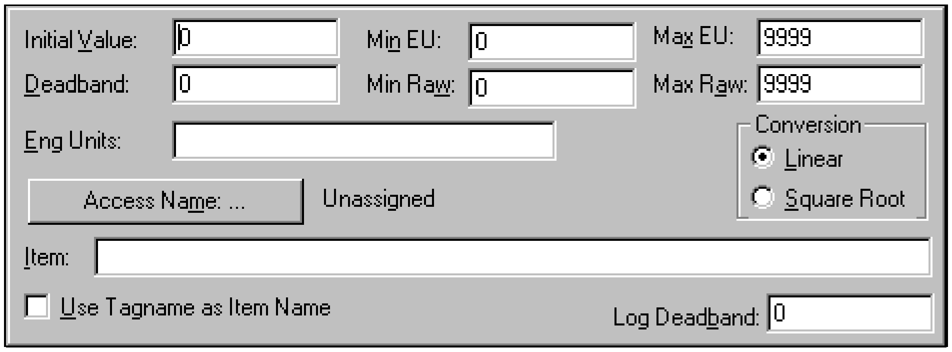
You define an **I/O Integer** type tagname when you need an I/O tagname with a 32-bit signed integer value between -2,147,483,648 and 2,147,483,647.

You define an **I/O Real** type tagname when you need an I/O tagname with a floating point value 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.)

➢ **To define the details for an I/O analog tagname:**

1. When you select **I/O Integer** or **I/O Real** as the type for your tagname, the following details dialog box will appear.

   If it does not appear, click **Details** at the top of the **Tagname Dictionary** dialog box.



Initial Value: 0
Min EU: 0
Max EU: 9999
Deadband: 0
Min Raw: 0
Max Raw: 9999
Eng Units:
Conversion: [✓ Linear] Square Root
Access Name: ... Unassigned
Item:
☐ Use Tagname as Item Name
Log Deadband: 0

2. In the **Initial Value** box, type the value you want stored in the tagname when the runtime database is first loaded.
3. In the **Deadband** box, type the amount the engineering units for the tagname can change before the database is updated.
4. In the **Min EU** box, type the engineering units value for the tagname when the minimum raw count value is received.
5. In the **Min Raw** box, type the minimum value of the low clamp on the raw I/O integer values.
6. In the **Max EU** box, type the engineering units value for the tagname when the maximum raw count value is received.
7. In the **Max Raw** box, type the maximum value of the high clamp on the raw I/O integer values.

   You can use the **Min EU**, **Min Raw**, **Max EU** and **Max Raw** values to scale your I/O tagnames.

   For more information on scaling tagnames, see "Scaling I/O Tagnames."
8. In the **Eng Units** box, enter the label you want to use for your tagname's engineering units.
9. Select the type of **Conversion** that you want the database to use to scale the raw counts when calculating the engineering units.

   Linear - The result is calculated using linear interpolation between the end points.

   The algorithm for linear scaling of input is:

```
EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) / (MaxRaw -
MinRaw)) + MinEU
```

The algorithm for linear scaling of output is:

```
RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) / (MaxEU -
MinEU)) + MinRaw
```

Square Root - The raw counts values are used for interpolation. This is useful for scaling inputs from nonlinear devices such as pressure transducers.

The algorithm for square root scaling of input is:

```
EUValue = sqrt(RawValue - MinRaw) * ((MaxEU - MinEU) /
sqrt(MaxRaw - MinRaw)) + MinEU
```

The algorithm for square root scaling of output is:

```
RawValue = square((EUValue - MinEU) * (sqrt(MaxRaw - MinRaw) /
(MaxEU -MinEU))) + MinRaw
```

10. Click **Access Name** to define or select the Access Name that you want to assign to this tagname. (If an Access Name already appears to the right of this button, and you do not define or select a different one, it will be assigned to the tagname.)

    For more information on Access Names, see Chapter 9 - I/O Communications.

11. In the **Item** box, type the valid item name for the data point in the server program that the tagname will read/write its value to/from. For example, if you want to read a value from a register in a PLC, enter the valid identification for that register as the item name.

    Item names are auto-indexed. For example, if you enter and save item name R4001, then click **New** (to define a new tagname), the item name will automatically be indexed to R4002. If an item name 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 and so on.

12. Select the **Use Tagname as Item Name** option if you want to use the tagname for the item name.

13. If you want to define alarm conditions for the tagname, click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog to display the respective alarm conditions dialog box for the tagname type you are defining.

    For more information on alarm conditions, see "Defining Tagname Alarm Conditions."

14. In the **Log Deadband** box, type the amount the tagname's engineering units must change before the tagname is logged to the historical log file.

    Notes

    1) You must select **Log Data** for the tagname if you want the tagname to be logged to disk when its Engineering units change more than the **Log Deadband** value.

    2) If you change the **Log Deadband** value while WindowViewer is running, your changes will not take effect until historical logging has been stopped and restarted.

    For more information on historical logging, see Chapter 8 - Real-time and Historical Trending.

15. Once you have completed defining your tagname, click **Close** to save your tagname definition and close the tagname dialog boxes.

Defining I/O Message Tagname Details

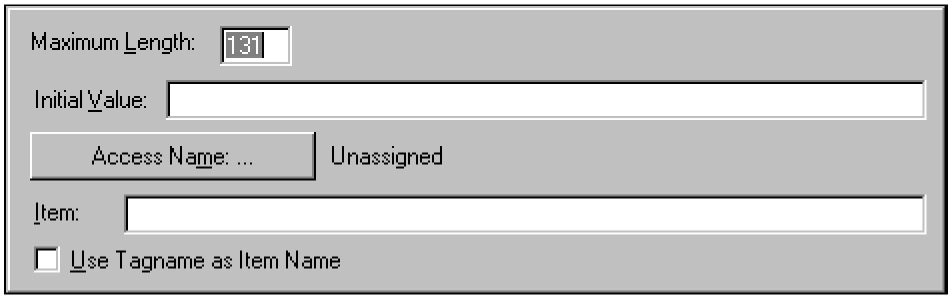
All tagnames that read or write their values to or from another Windows program are I/O type tagnames. This includes all inputs and outputs from programmable controllers, process computers, other Windows programs and data from network nodes.

You define an **I/O Message** type tagname when you need an internal text string tagname that can be up to 131 characters long.

➢ **To define the details for a I/O message tagname:**

1. When you select **I/O Message** as the type for your tagname, the following details dialog box will appear.

   If it does not appear, click **Details** at the top of the **Tagname Dictionary** dialog box.



Maximum Length: 131

Initial Value:

Access Name: ... Unassigned

Item:

☐ Use Tagname as Item Name

2. In the **Maximum Length** box, type the maximum number of characters to be allowed in the tagname's message. (InTouch allows a maximum of 131, which is displayed as the default.)
3. In the **Initial Value** box, type the text string that you want displayed for the tagname when WindowViewer is initially started.
4. Click **Access Name** to define or select the Access Name that you want to assign to this tagname. (If an Access Name already appears to the right of this button, and you do not define or select a different one, it will be assigned to the tagname.)

   For more information on Access Names, see Chapter 9 - I/O Communications.
5. In the **Item** box, type the valid item name for the data point in the server program that the tagname will read/write its value to/from. For example, if you want to read a value from a register in a PLC, enter the valid identification for that register as the item name.

   Item names are auto-indexed. For example, if you enter and save item name R4001, then click **New** (to define a new tagname), the item name will automatically be indexed to R4002. If an item name 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 and so on.
6. Select the **Use Tagname as Item Name** option if you want to use the tagname for the item name.
7. Once you have completed defining your tagname, click **Close** to save your tagname definition and close the tagname dialog boxes.

Defining SuperTag Member Tagname Details

Member tagnames are defined in SuperTag Templates. Member tagnames behave exactly like normal tagnames and can be of type Discrete, Integer, Real, Message or, another SuperTag. Like normal InTouch tagnames, member tagnames support trending, alarming and all tagname **.fields**.

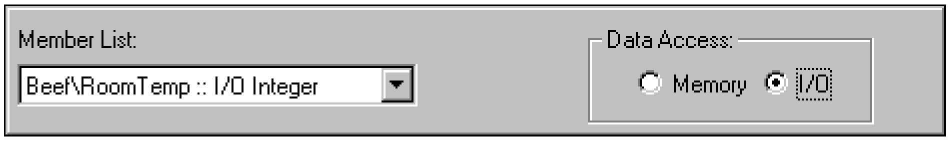
For more information on member tagnames see, "Creating InTouch SuperTags."

When you define a tagname and select a SuperTag template for its tagname type, by default, all member tagnames defined in SuperTag templates will be set to the data access type "Memory." If this is the type that you need for them to be, no special configuration is necessary. However, if you need any of the member tagnames in the SuperTag template to be defined as I/O types, you must do some additional configuring.

➢ **To define I/O SuperTag member tagnames:**

1. When you select a SuperTag template as the type for your tagname, the following details dialog box appears:

   If it does not appear, click **Members** at the top of the **Tagname Dictionary** dialog box.



Member List: Beef\RoomTemp :: I/O Integer

Data Access: Memory [✓ I/O]

   Notice that the new tagname that you typed in the **Tagname** box becomes the "parent" for all the member tagnames in the **Member List**.

2. Click the **Member List** arrow, and then select the member tagname in the list that you want to define as an I/O data access type
3. In the **Data Access** group, select **I/O**. The respective I/O details dialog box for the member tagname's type (Discrete, Analog (Real or Integer) or Message) will appear..
4. Enter the required I/O details just as you would for a normal InTouch I/O type tagname.
5. To save your changes, select another member tagname in the list and configure it, or click **Close**.

Defining Tagname Alarm Conditions

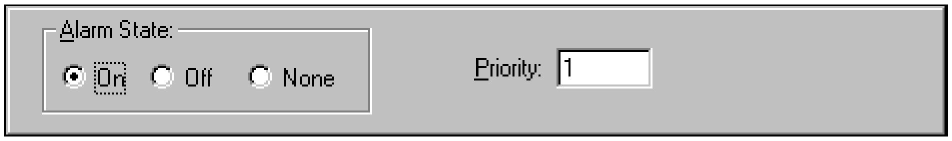
You can define alarm conditions for tagnames at the same time that you define the tagname. There are two types alarm detail dialog boxes. One for discrete type tagnames and one for analog (integer or real) type tagnames.

Defining Discrete Tagname Alarm Conditions

You can define an alarm condition for a discrete type tagname's **On** state or **Off** state.

➢ **To define alarm conditions for a discrete tagname:**

1. On the **Special** menu, click **Tagname Dictionary** or, in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box will appear.
2. Click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog box to display the discrete alarm details dialog box:



Alarm State: [✓ On] Off None

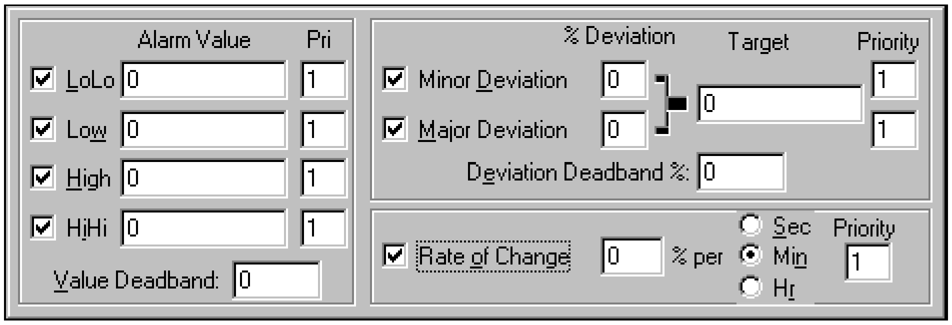
Priority: 1

3. Click the **Alarm State** that you want the tagname to be in when in alarm.
4. In the **Priority** box, type 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 that you want 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

➢ **To define alarm conditions for an analog tagname:**

1. On the **Special** menu, click **Tagname Dictionary** or, in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box will appear.
2. Click either **Alarms** or **Details & Alarms** at the top of the **Tagname Dictionary** dialog box to display the analog alarm details dialog box:



	Alarm Value	Pri
✓ LoLo	0	1
✓ Low	0	1
✓ High	0	1
✓ HiHi	0	1

Value Deadband: 0

	% Deviation	Target	Priority
✓ Minor Deviation	0	0	1
✓ Major Deviation	0		1

Deviation Deadband %: 0

✓ Rate of Change 0 % per Sec [✓ Min] Hr Priority 1

3. Select the alarm types (**LoLo**, **Low**, **High**, **HiHi**) that you want to use to detect when the value of an analog type tagname is beyond an absolute limit.
4. In the **Alarm Value** box, type 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 (i.e., 100.75).

5. In any of the **Pri** (priority) boxes type 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 you want to be displayed in a window, logged to disk or printed.
6. In the **Value Deadband** box, type 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 you want to use 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, type 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, type the desired or reference value of the tagname from which minor and/or major deviation percentages are based.
10. In the **Deviation Deadband %** box, type 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** if you want 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, type the maximum allowable percentage change.
13. Select **Sec**, **Min**, or **Hr** for the time interval units of the change.

For more information on alarms, see Chapter 7 - Alarms/Events.

Creating InTouch SuperTags

InTouch supports a template structure that allows you to define composite tagname types called SuperTags. SuperTag templates can contain up to 64 member tagnames and 2 nesting levels. Meaning, a SuperTag parent can contain up to 64 embedded child members and each child member can contain up to 64 sub-member tagnames for a total of 4095 member tagnames. (When one SuperTag template parent is embedded into another SuperTag template it becomes a "child member.") All SuperTag template sub-member tagnames behave exactly like normal tagnames. They support trending, alarming and all tagname **.fields**.

For convenience, InTouch provides you with a "TemplateMaker" that you can use to create your SuperTags. The TemplateMaker allows you to create, edit and delete SuperTag templates and member tagnames. InTouch saves all SuperTag templates in the file SUPERTAG.DAT in your InTouch installation directory (not the application directory). This allows the templates that you create to be used in any application.

InTouch also provides you with the ability to create SuperTags in several alternative ways. For example, you can create SuperTags directly from the Tagname Dictionary, in animation link tagname or expression input boxes, InTouch QuickScripts, or in an external file that you load into your application by using the InTouch DBLoad utility.

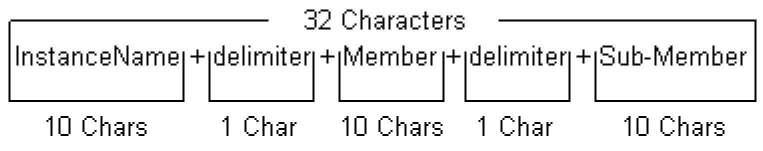
For more information, see "Alternative Methods for Creating SuperTags."

When you create a SuperTag parent template, its name is automatically added to the tagname **Tag Types** dialog box in the Tagname Dictionary and is immediately available for selection when you create a new tagname. You do not need to restart WindowMaker to define tagnames that use a newly created SuperTag type.

Important Note If you modify an existing SuperTag template, all existing instances of that SuperTag are not affected. (Instances are tagnames defined in the Tagname Dictionary that use a SuperTag for their type.) In other words, modifications that you make to a SuperTag are not retroactive. However, all new instances that you define using the modified SuperTag will use the new composition. Similarly, if you add a member tagname to a SuperTag instance through an alternative method, its template is not updated.

InTouch SuperTag Syntax

Since InTouch tagnames are limited to 32 characters, each SuperTag ParentInstance\ChildMember\Sub-member is restricted to a maximum of 32 characters. A SuperTag reference can only be a maximum of two templates (ParentInstance\ChildMember) and one member deep as illustrated below:



Each member in a SuperTag template is accessible in the standard format that you currently use to access the **.fields** of normal InTouch tagname types. The SuperTag reference syntax is supported throughout InTouch where normal tagnames can be used. For example, a valid SuperTag reference would be:

```
ColdRoom4\EvapUnit1\FanMotor2.MaxEU
```

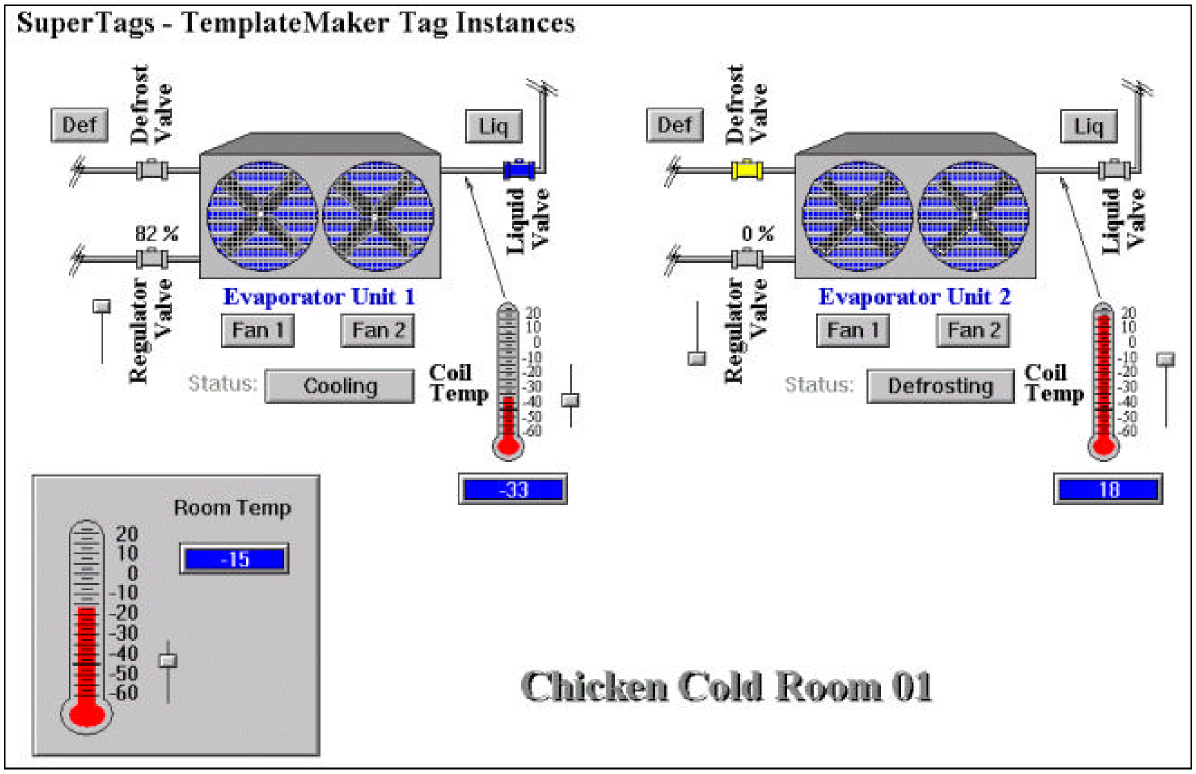
Remote tagname references also support SuperTags. For example:

PLC1:"Turkey\EvapUnit2\PrsRegVlv.EngUnits"

For more information on using remote tagname references, see "Remote Tagname Referencing."

Creating a SuperTag Template Structure

To realistically illustrate the SuperTag concept in a factory environment, let's assume that we have four identical refrigerated storage rooms in which we store beef, pork, chicken and turkey. Each of these cold rooms has a room temperature, and two evaporator units. Each evaporator unit has seven data values that we need to monitor or control in runtime. For example:



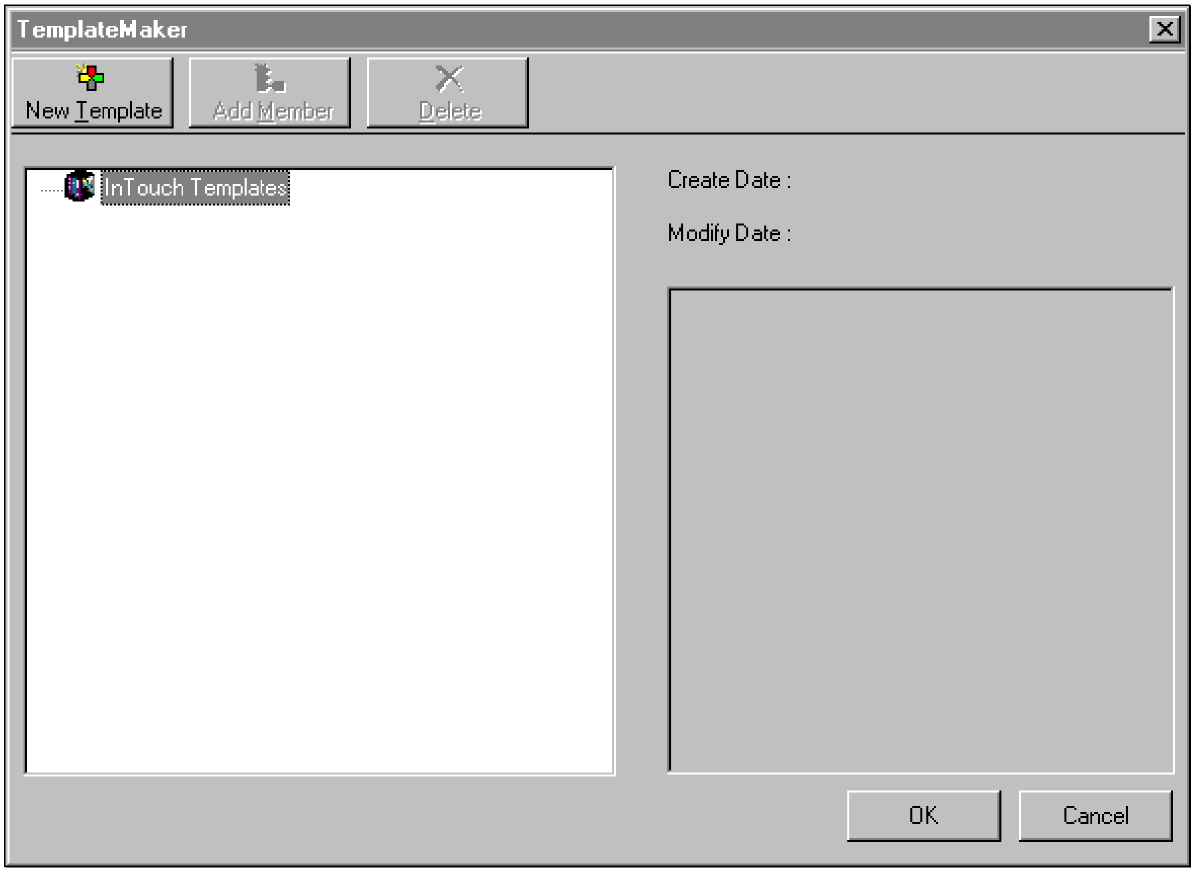
If we do not create SuperTag templates to accomplish this, we would need to manually define each individual tagname for every data value in each cold room multiplied by our total number of cold rooms. In other words, we would have to organize and define dozens of tagnames in the Tagname Dictionary!

By using SuperTags we can save hours of development time and minimize our chance for errors. Using our Cold Room scenario described above, we will create one SuperTag parent template called "EvapUnit." (This EvapUnit will later become a child member of the ColdRoom parent template. This a "detail-up" design concept.) EvapUnit will be defined with seven sub-member tagnames:

Member Tag	Type	Description
FanMotor1	Discrete	Motor Starter for Fan 1
FanMotor2	Discrete	Motor Starter for Fan 2
DefrostVlv	Discrete	Defrost Gas Valve State
LiquidVlv	Discrete	Liquid Refrigerant Valve State
CoilTemp	Real	Temperature of the refrigerant
PrsRegVlv	Integer	Pressure Regulator Valve (0-100%)
EvapStatus	Message	Evaporator Unit Status String

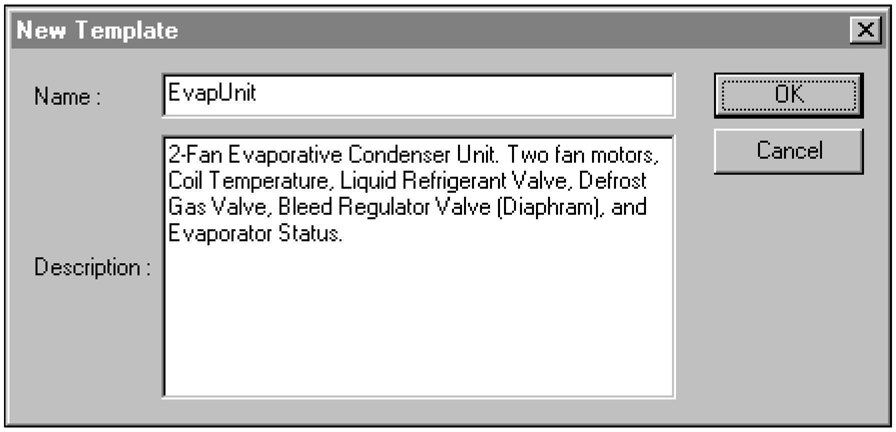
➢ **To create a SuperTag parent template:**

1. On the **Special** menu, click **TemplateMaker** or, in the Application Explorer, double-click **TemplateMaker**. The **TemplateMaker** dialog box appears:



2. In the TemplateMaker window, select **InTouch Templates**, and then click **New Template** or, right-click **InTouch Templates**, and then select **New Template**. The **New Template** dialog box appears:

   You can also select **InTouch Templates**, and then right-click a blank area of the window.



Name : EvapUnit

Description : 2-Fan Evaporative Condenser Unit. Two fan motors, Coil Temperature, Liquid Refrigerant Valve, Defrost Gas Valve, Bleed Regulator Valve (Diaphram), and Evaporator Status.

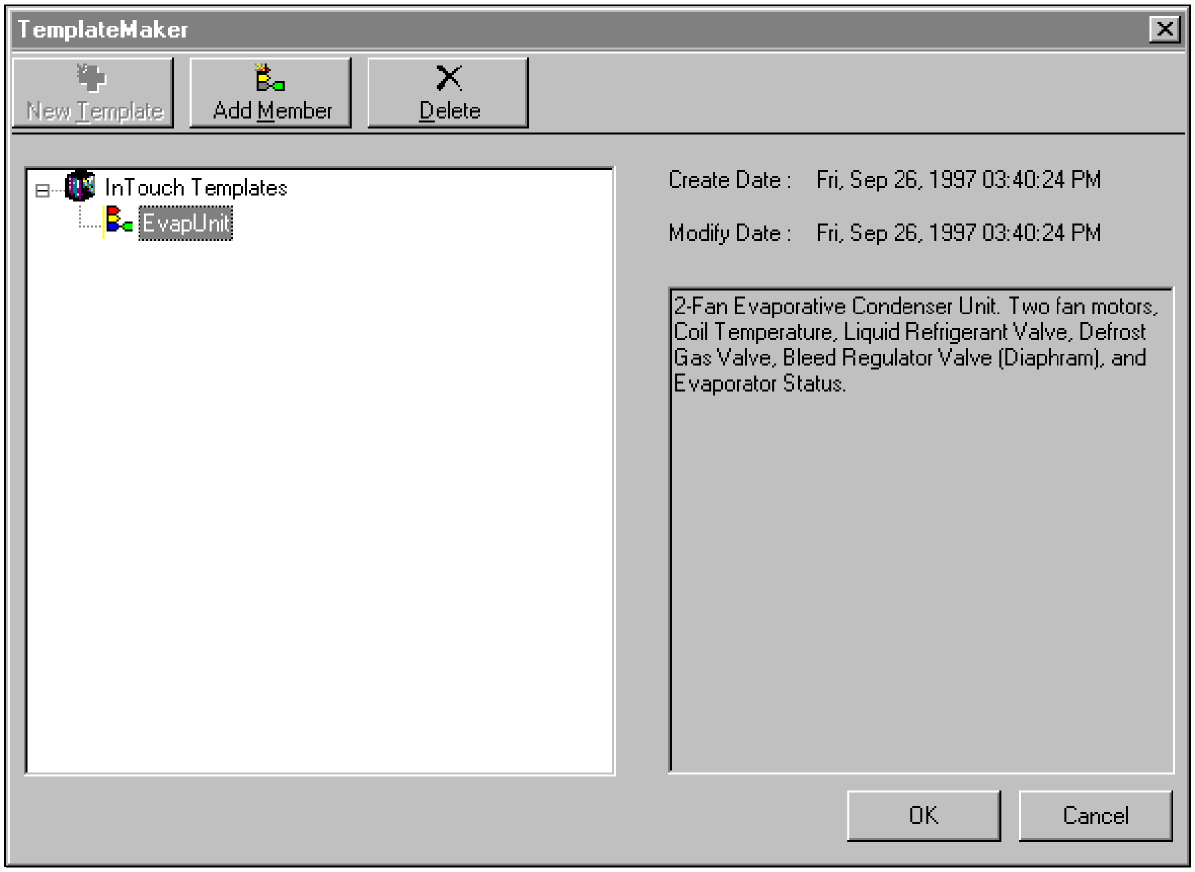
   If you right-click any of the text entry boxes in any of the TemplateMaker dialog boxes, a menu will appear displaying the commands that you can apply to the selected text.

3. In the **Name** field, type a unique name for the new template (maximum of 10 characters.)

- As you add new parent templates, their names immediately appear as a tagname type in the **Tag Types** dialog box in the Tagname Dictionary and are immediately available for selection. You do not need to restart WindowMaker to define new tagnames and assign them to the SuperTag type.

4. In the **Description** field, type any information that you want to describe the template.

5. Click **OK**. The **TemplateMaker** dialog reappears displaying the new template name in its window:



Create Date : Fri, Sep 26, 1997 03:40:24 PM

Modify Date : Fri, Sep 26, 1997 03:40:24 PM

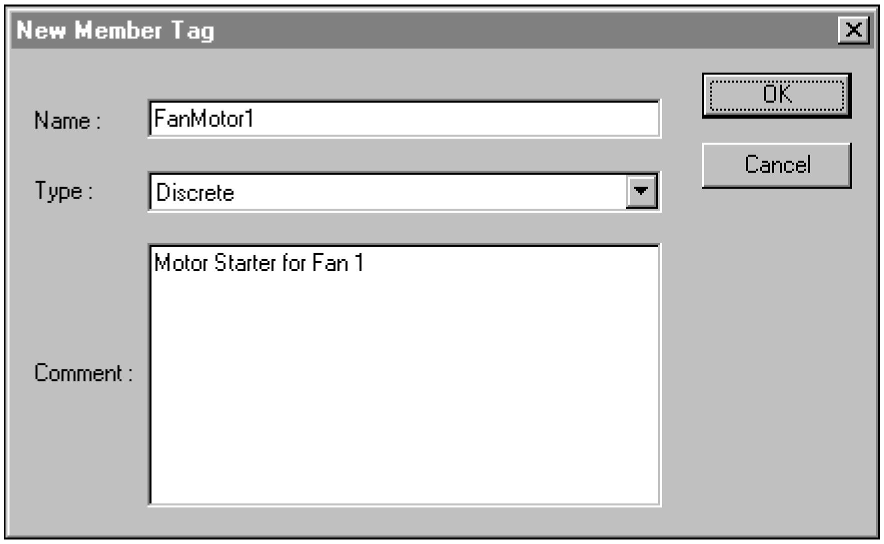
2-Fan Evaporative Condenser Unit. Two fan motors, Coil Temperature, Liquid Refrigerant Valve, Defrost Gas Valve, Bleed Regulator Valve (Diaphram), and Evaporator Status.

- Notice that once a template is created, the **New Member** and **Delete** buttons become active. The day, date and time the template was created and/or last modified, and the template's description are also now displayed when the template name is selected.

Note The TemplateMaker window displays all your currently defined SuperTag parent templates and their child members in a hierarchical list. To expand a template's view, click the left mouse button on the ⊞ next to the template name. All member tagnames defined for the parent template name will be displayed. To collapse a view, click the left mouse button on the ⊟.

➢ **To create SuperTag member tagnames:**

1. In the TemplateMaker window, select the SuperTag template (in this case, EvapUnit), and then click **Add Member**, or right-click the SuperTag template name, and then click **Add Member**. The **New Member Tag** dialog box appears:



Name : FanMotor1

Type : Discrete

Comment : Motor Starter for Fan 1

2. In the **Name** box, type the name that you want to use for the member tagname.
3. In the **Type** box, type the tagname type for the member or, click the **Type** arrow and select the type in the list. A type can be Discrete, Integer, Real, Message or another SuperTag template.

   If you type the first letter of a type, the first type in the list box beginning with that letter will automatically be displayed in the box. If multiple types exist that begin with the same letter, you can continuously type the letter to cycle through the names.

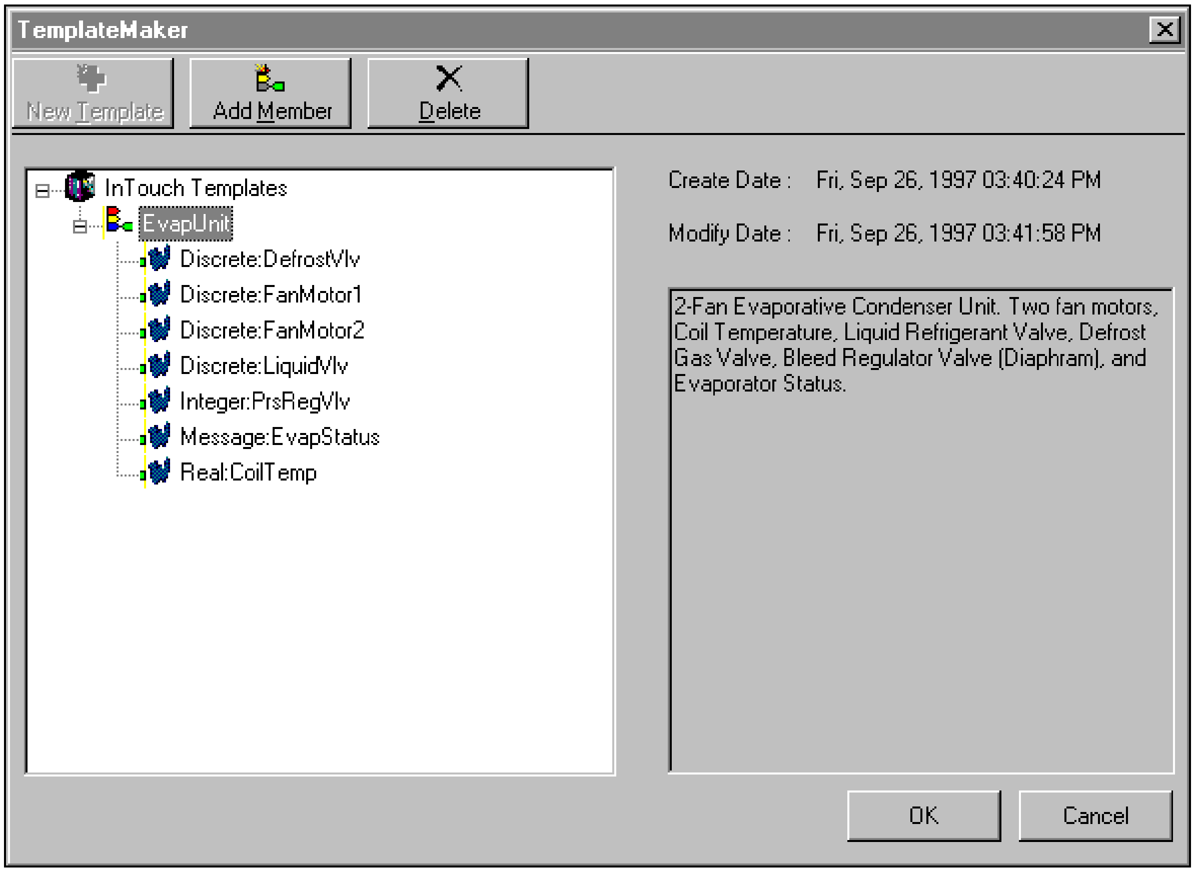
Note The type you specify here is only a placeholder for the SuperTag template. By default, all member tagnames are set to "Memory" types when you define them in the TemplateMaker. However, when you define a template instance in the Tagname Dictionary, you will need to specify whether they are truly "Memory" or "I/O" type tagnames.

   For more information on the **Members** dialog box, "Defining SuperTag Member Tagname Details."

4. In the **Comment** field, type any information that you want to describe the member tagname.
5. Click **OK**.

   Repeat this procedure to add additional member tagnames to the SuperTag template.

The new member tagnames are added beneath the SuperTag parent template in the TemplateMaker window: For example:



Create Date : Fri, Sep 26, 1997 03:40:24 PM

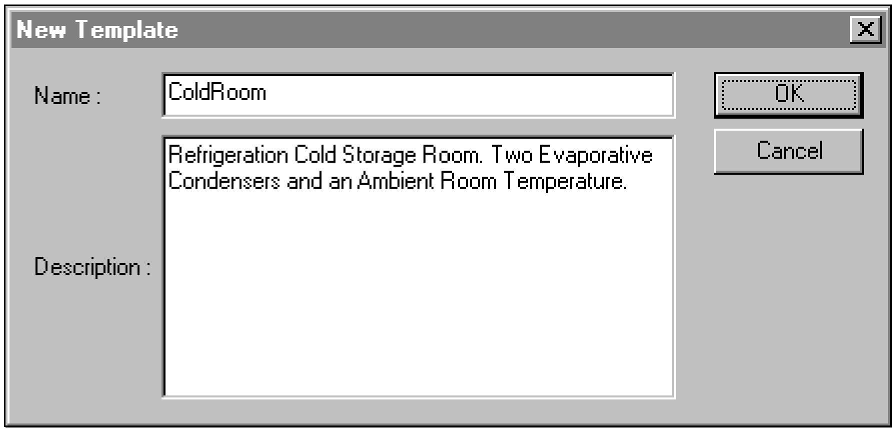
Modify Date : Fri, Sep 26, 1997 03:41:58 PM

2-Fan Evaporative Condenser Unit. Two fan motors, Coil Temperature, Liquid Refrigerant Valve, Defrost Gas Valve, Bleed Regulator Valve (Diaphram), and Evaporator Status.

Notice that if a member tagname is selected, the **New Member** button is no longer active since members can only be created for existing SuperTag parent templates. The day, date and time the member tagname was created and/or last modified, and the template's description are displayed when the member is selected.

We will now create another parent template called ColdRoom. ColdRoom will have one member tagname called RoomTemp and two EvapUnit child member templates (EvapUnit1 and EvapUnit2). The two child member templates will use the parent, EvapUnit, SuperTag template for their type.

1. In the TemplateMaker window, select **InTouch Templates**, and then click **New Template**, or right-click **InTouch Templates**, and then click **New Template**. The **New Template** dialog box appears:



Name : ColdRoom

Description : Refrigeration Cold Storage Room. Two Evaporative Condensers and an Ambient Room Temperature.

2. In the **Name** field, type a unique name for the new parent template (maximum of 10 characters.)

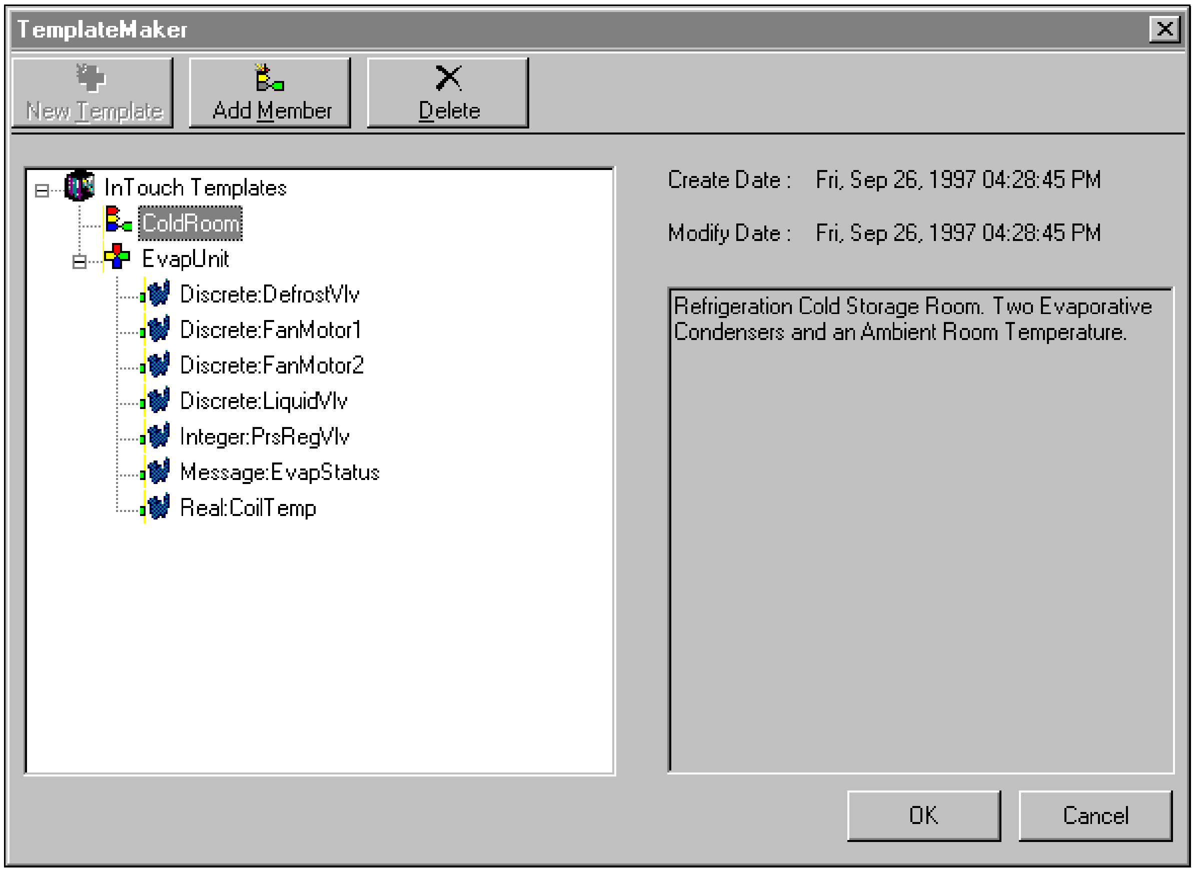
   As you add new parent templates, their names immediately appear as a tagname type in the **Tag Types** dialog box in the Tagname Dictionary and are immediately available for selection. You do not need to restart WindowMaker to define new tagnames and assign them to the SuperTag type.

3. In the **Description** field, type any information that you want to describe the template.

4. Click **OK**. The **TemplateMaker** dialog reappears displaying the new template name in its window:

5. Click **OK**.

The parent template is added to the list of **InTouch Templates** in the TemplateMaker window: For example:



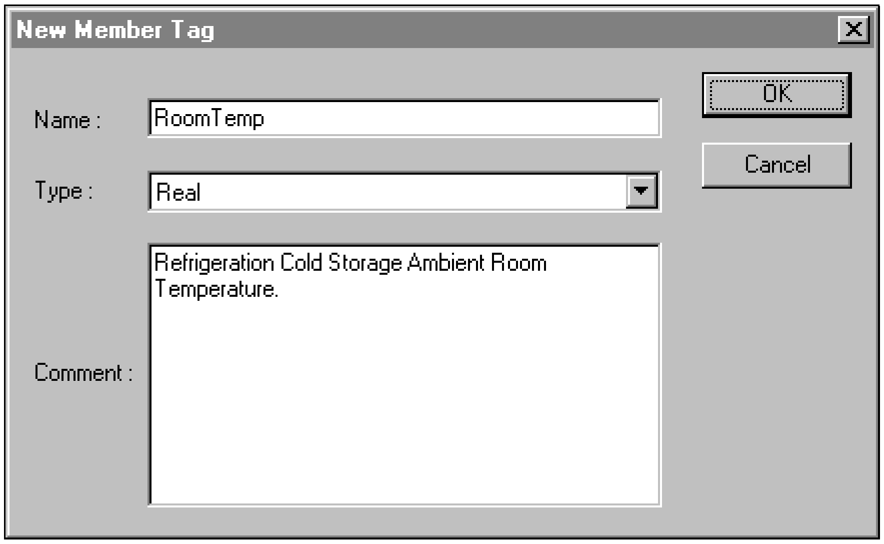
Create Date : Fri, Sep 26, 1997 04:28:45 PM

Modify Date : Fri, Sep 26, 1997 04:28:45 PM

Refrigeration Cold Storage Room. Two Evaporative Condensers and an Ambient Room Temperature.

We now need to create three members for our ColdRoom parent template; two EvapUnit child members and one member tagname called RoomTemp.

1. In the TemplateMaker window, select the SuperTag parent template (in this case, ColdRoom), and then click **Add Member**, or right-click the SuperTag parent template name, and then click **Add Member**. The **New Member Tag** dialog box appears:



Name : RoomTemp

Type : Real

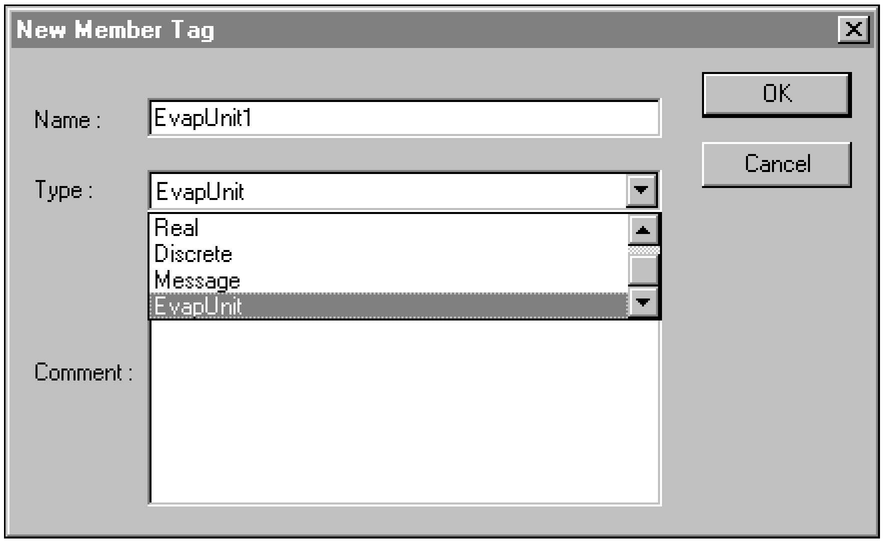
Comment : Refrigeration Cold Storage Ambient Room Temperature.

2. In the **Name** box, type the name that you want to use for the member tagname.

3. In the **Type** box, type the tagname type for the member or, click the **Type** arrow and select the type in the list. A type can be Discrete, Integer, Real, Message or another SuperTag template.
4. In the **Comment** field, type any information that you want to describe the member tagname.
5. Click **OK**.

Next we will create our two child member templates, EvapUnit1 and EvapUnit2 which use the EvapUnit template type.

1. In the TemplateMaker window, select the SuperTag parent template (in this case, ColdRoom), and then click **Add Member**, or right-click the SuperTag parent template name, and then click **Add Member**. The **New Member Tag** dialog box appears:

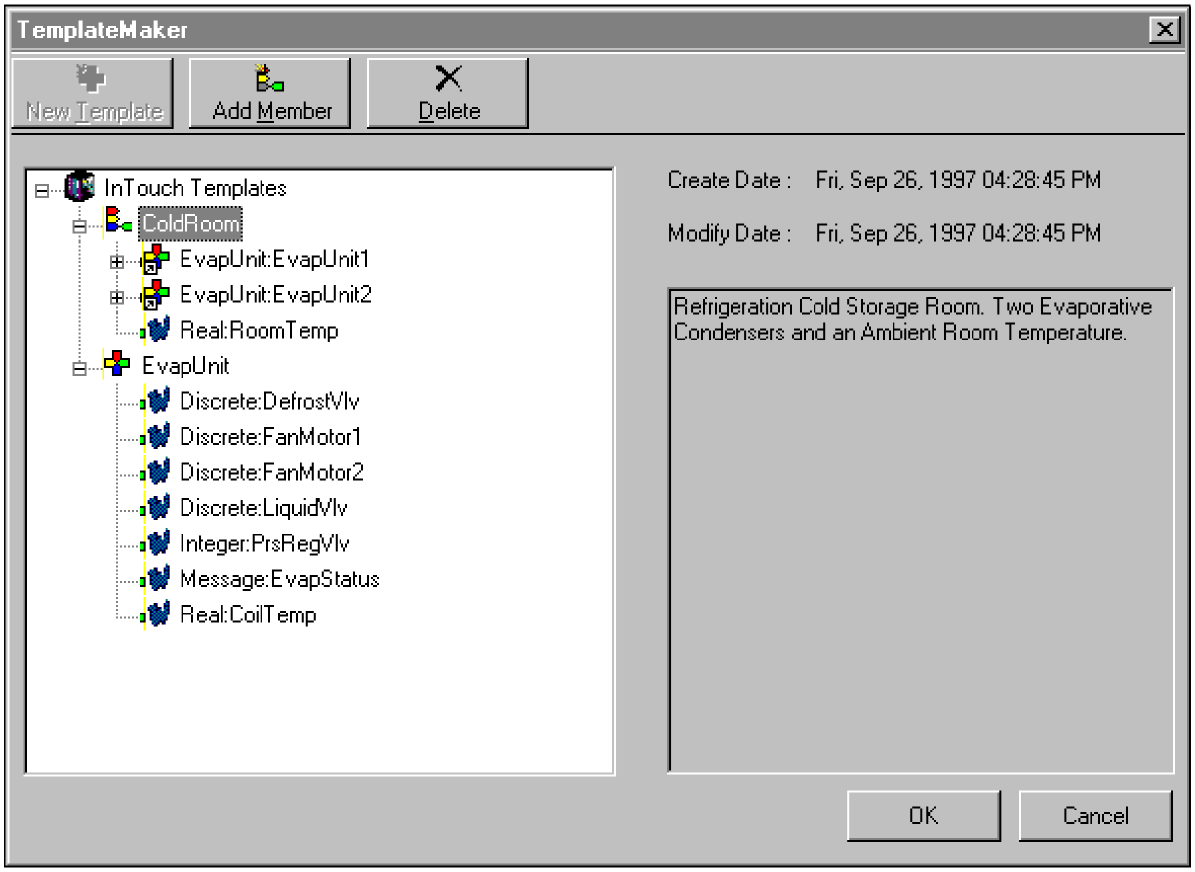


Name : EvapUnit1
Type : EvapUnit
Comment :

2. In the **Name** box, type the name that you want to use for the member tagname.
3. In the **Type** box, type the tagname type for the member or, click the **Type** arrow and select the type in the list. In this case, we are using the special template type EvapUnit.
4. In the **Comment** field, type any information that you want to describe the member tagname.
5. Click **OK**.

   Repeat this procedure for EvapUnit2.

Once we have completed the ColdRoom parent template, the TemplateMaker window will display the following template structure hierarchy:



Create Date : Fri, Sep 26, 1997 04:28:45 PM

Modify Date : Fri, Sep 26, 1997 04:28:45 PM

Refrigeration Cold Storage Room. Two Evaporative Condensers and an Ambient Room Temperature.

6. Click **OK**. Now that we have completed the ColdRoom SuperTag template, we can immediately create tagname instances that use the template for their tagname type.

   For more information, see "Defining SuperTag Template Instances."

Editing SuperTag Templates and Member Tagnames

You can modify SuperTag templates or member tagnames at any time. However, if you modify an existing SuperTag template or its members, all existing instances of that template are not affected. (Instances are tagnames defined in the Tagname Dictionary that use a SuperTag for their type.) In other words, modifications that you make to a SuperTag are not retroactive. However, all new instances that you define using the modified SuperTag will use the new composition.

➢ **To edit an existing SuperTag template or member tagname:**

1. In the TemplateMaker window, double-click the SuperTag template name (or member name), or right-click it, and then click **Edit**. The **Edit Template** (or **Edit Member Tag**) dialog box will appear displaying the SuperTag template's (or member's) definition.
2. Make your required edits, and then click **OK**.

➢ **To delete a SuperTag template or member:**

1. In the TemplateMaker window, select the SuperTag template name (or member name) that you want to delete, or right-click it, and then click **Delete**. A message box will appear asking you to confirm the deletion.
2. Click **Yes** to delete the selected name, or click **No** to cancel the deletion.

Note If you press the ESC key to close the TemplateMaker instead **OK**, the template is not deleted. When you delete a template, all its associated member tagnames are also deleted.

Defining SuperTag Template Instances

An important concept in TemplateMaker is distinguishing a SuperTag template from a template instance. A template instance is a specific instantiation of a SuperTag template. The most important difference between a template and an instance is that the parent template name is replaced by the instance tagname. The child template name and the sub-member tagnames do not change.

For example, this could be equated to a literal template that you use for drafting such as a stencil that you use to produce actual drawings. The drawings themselves, in this metaphor, are "template instances," that are patterned after the template or stencil used to create them.

Once again referring to our ColdRoom template scenario, after we have created our template, from it we could create SuperTag instances of "Beef," "Pork," "Chicken" and "Turkey." To do so, we will simply create four tagnames that use ColdRoom for their types. Thus with our one time effort, we will quickly create 60 tagnames in the Tagname Dictionary. A huge time saver!

After we have created the "ColdRoom" SuperTag template and the instances, we can refer to any of its members by using valid SuperTag references in animation link expressions or QuickScripts. For example:

```
Beef\RoomTemp
Chicken\RoomTemp.RawValue
Chicken\EvapUnit1\FanMotor1.OnMsg
Pork\EvapUnit2\EvapStatus
Turkey\EvapUnit2\PrsRegVlv.EngUnits
```

For more information on defining template instances see "Defining SuperTag Member Tagname Details."

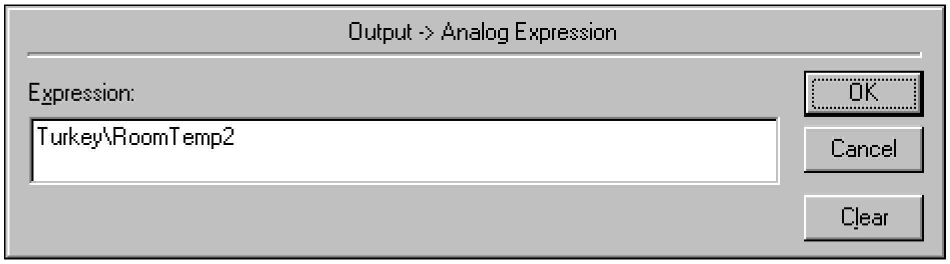
Alternative Methods for Creating SuperTags

In addition to the TemplateMaker, InTouch supports the creation of SuperTags through animation link expressions, InTouch QuickScripts, and external .CSV (Comma Separated Variable) files that you upload into the Tagname Dictionary through the DBLoad utility. However, you can also add a member or sub-member to an existing SuperTag through the Tagname Dictionary, which is the easiest method to use.

For more information on creating SuperTags using DBLoad, see "Creating SuperTags Instances."

Note When you use one of the alternative methods to create the member, it is not reflected in the SuperTag template definition in the TemplateMaker.

When you create a SuperTag through an animation expression or InTouch QuickScript you must use the valid SuperTag format. For example:



Turkey\RoomTemp2

Note The following syntax examples are valid:

ParentInstance\ChildMember
ParentInstance\ChildMember\Submember

The following syntax examples are invalid:

ParentInstance
ParentInstance\ChildMember\

If an invalid format is used, the an error message box will appear informing you that the syntax is invalid.

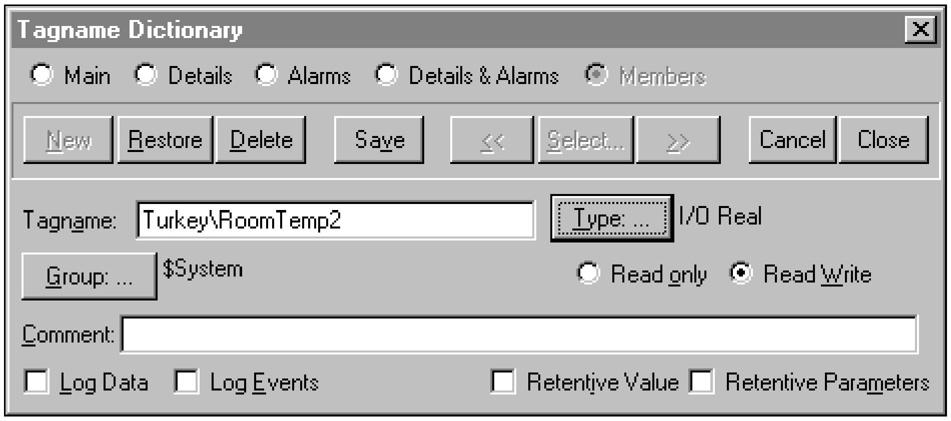
If the SuperTag instance and member tagname you specify in an animation expression or QuickScript are currently not defined, a message box will appear asking you if you want to define it now. Click **OK**. The Tagname Dictionary will appear displaying the SuperTag instance and member tagname that you specified.

Using the Tagname Dictionary to Create SuperTags

The Tagname Dictionary is the easiest alternative method to use for creating a SuperTag instance or member tagnames.

➢ **To create a SuperTag in the Tagname Dictionary:**

1. On the **Special** menu, click **Tagname Dictionary**, or in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box will appear. Click **New**.



Tagname: Turkey\RoomTemp2

Type: ... I/O Real

Group: ... $System

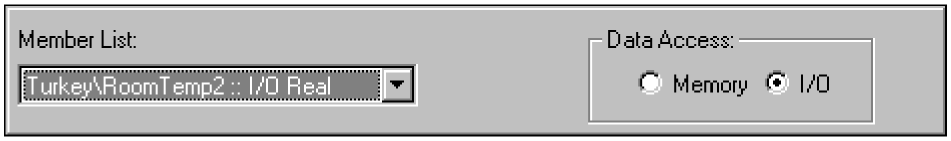
Read only [✓ Read Write]

Comment:

2. In the **Tagname** box, type the exact name of your SuperTag instance followed by the backslash (\) delimiter and the name of the new member tagname. In this case, we would type **Turkey\RoomTemp2**.

Note When you are adding a new member tagname to an existing SuperTag instance, the spelling of the instance name must match exactly. Otherwise, a brand new SuperTag instance and member will be added.

3. Click **Type** and select the type for the SuperTag member. (Selection of the remaining options is not required in this context.) In this case, we have selected I/O Real.
4. Click **Save** or **Close** to add the member.
5. To view the member tagname in the Turkey SuperTag without exiting the Tagname Dictionary, click either the left or right double-arrow buttons. The **Members** details dialog box appears:

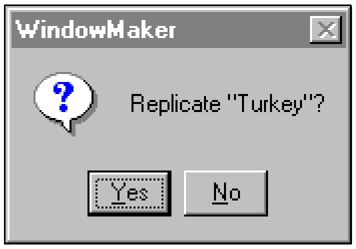


Member List: Turkey\RoomTemp2 :: I/O Real

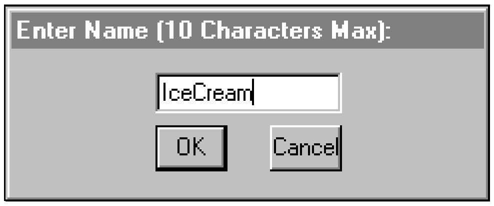
Data Access: Memory [✓ I/O]

6. Click **Close** to close the Tagname Dictionary.

7. If you click **New** when a SuperTag is displayed in the Tagname Dictionary, the following dialog box appears asking you if you want to make an identical copy of the displayed SuperTag instance:



8. Click **Yes** to create another SuperTag instance that is an exact duplicate of the displayed SuperTag instance. The **Enter Name** dialog box appears:



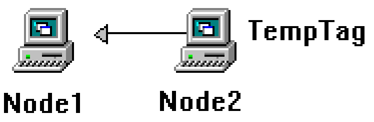
9. Type a new SuperTag instance name.

10. Click **OK**.

The Tagname Dictionary automatically creates all member tagnames and sub-member tagnames for the new SuperTag instance, and they are immediately available for usage in animation links and InTouch QuickScripts.

Remote Tagname Referencing

InTouch allows true client-server architecture for factory automation applications. Client applications can be designed without using any tagnames in the local Tagname Dictionary. This can be achieved by the using "Remote Tagname Referencing." capability of InTouch. For example:



In this example, you can retrieve the value of the tagname "TempTag" on Node2 in two ways:

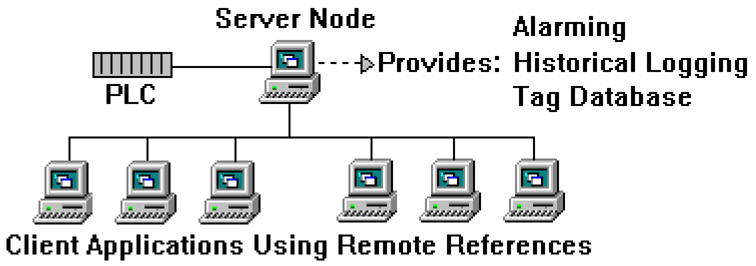
1. Create an I/O type tagname in Node1's Tagname Dictionary that uses "Node2" as the **Node Name** in the Access Name associated with the I/O tagname.
2. Use a remote reference directly to "TempTag." For example, **PLC1:"TempTag"**

In other words, in a window or QuickScript, you can either reference the local tagname or, use *AccessName:"item"* to reference a remote tagname.

☞ For more information on remote tagname reference syntax, see "Remote Tagname Reference Syntax."

When you want to directly reference a remote tagname in any other FactorySuite application, only *AccessName:"item"* is required. You do not have to define the remote tagname in your local Tagname Dictionary. Remote references can also access data from any I/O data source such as, a Wonderware I/O Server or Microsoft Excel. In addition, they support SuperTags (SuperTags). The valid syntax for a remote tagname referencing a SuperTag is *Accessname:"SuperTag"*.

Also, when you use remote tagname references, and you import a window or QuickScript, all you have to do is convert the placeholder tagnames to remote tagname references. You do not have to define tagnames in you local Tagname Dictionary. The remote references are accessible from any FactorySuite application on the network as shown in the following illustration:



Remote Tagname Reference Syntax

The valid syntax for a remote tagname reference is *AccessName:"item"*. The characters tht you can use in a remote reference are the same characters that are valid for a tagname. The valid characters are: A-Z, a-z, 0-9, !, @, -, ?, #, $, %, _, \ and &. If you are sure that you do not use any invalid characters in your remote tagname reference then you do not need to enclose the *item* portion in quotation marks.

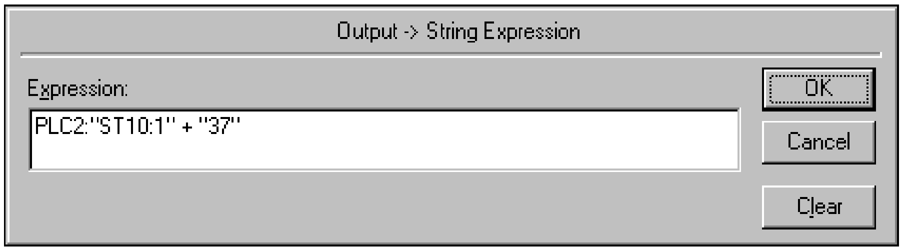
Tagname **.fields** can also be used in the "item" portion of the remote tagname reference. For example, "MyAlarm.HiHi".

In order to use any other characters, you must enclose the *"item"* in quotation marks. For example, if you use; ~, *, /, +, =, ^, |, **, <, >, <=, =>, ==, and <> you must enclose the *"item"* in quotation marks.

However, since some general ASCII I/O Servers accept any character as valid for an item name, we highly recommend that you make a practice of always surrounding the *"item"* portion with quotation marks.

For example, let's assume that you want to get a bit from an Allen-Bradley® PLC integer register and you use **N10:7/3** (third bit from integer 10) as the item name. The system will see **N10:** as an Access Name because the forward slash (/) is not a valid character. However, if you enclose the item name in quotations, **"N10:7/3"**, the system will read entire entry as the item name.

You also cannot string concatenate item names or remote tagname reference item names. For example, let's assume that you created a string output link using the following expression:



PLC2:"ST10:1" + "37"

When the system executes the above expression it will use the Access Name **PLC2** and go through the Allen-Bradley I/O Server to retrieve the string stored in the string file **ST10:1**. Then it will append the string **37** to the end of the string it retrieved in **ST10:1**. If "Green Paint" is stored in **ST10:1**, the string output object linked to the expression will display **Green Paint37**. Therefore, the operator would not see the contents of **ST10:137** as they had intended.

Whenever you use a remote reference (*accessname:"item"*), InTouch validates the Access Name that you specify. If it determines that the Access Name is not defined, you will be prompted to define it. If you select **Yes** when prompted, the **Access Names** dialog box will appear and you can add the new Access Name.

The Access Name is also validated when the remote tagname is activated. If errors are encountered, they will be written to the Wonderware Logger.

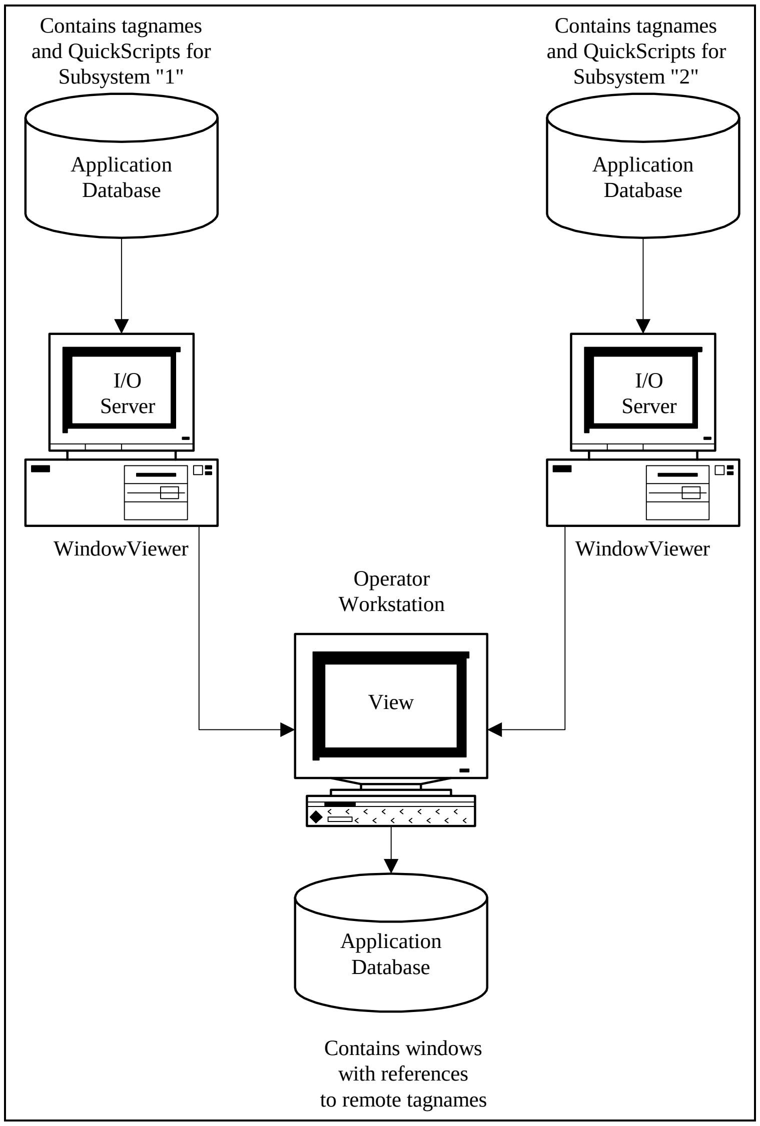
You can delete an Access Name that is used by in a remote reference, as long as a local tagname is not using it.

For more information on defining Access Names, see Chapter 3 - Building a Distributed Application.

For more information on the Wonderware Logger, see your *FactorySuite System Administrator's Guide*.

Creating a Tagname Server Application

By creating an application that contains only InTouch QuickScripts and tagnames, you can establish an instance of WindowViewer that functions as a tagname server. You can create another application that contains only windows (and memory tagnames for window logic processing). If these windows contain only remote tagname references, this application can serve as a repository for all the process windows for a facility. In this case, the remote tagname references are to tagnames in other WindowViewer instances that function as tagname servers. An instance of WindowViewer that connects to this database functions as an operator workstation. This WindowViewer instance can open any window and view data from anywhere on the plant floor. For example:


Contains tagnames and QuickScripts for Subsystem "1"
Contains tagnames and QuickScripts for Subsystem "2"
Application Database
Application Database
I/O Server
I/O Server
WindowViewer
WindowViewer
Operator Workstation
View
Application Database
Contains windows with references to remote tagnames

Remote references are valid for the following:

- Input Links – Discrete User Input, Analog User Input, String User Input, Vertical Slider, Horizontal Slider, and Discrete Value Button
- Discrete Alarm Links – Line Color, Fill Color, and Text Color
- Analog Alarm Links – Line Color, Fill Color, and Text Color
- Expressions – Links and Scripts where a discrete, analog or string tagname can be specified
- Wizards
- Data Change Scripts – "Tagname[.field]"
- ActiveX events, properties, and methods and in all InTouch QuickScript types.

Remote references are <u>not</u> valid for the following:

- Historical Trend Display – "Pen1" through "Pen 8"
- Standard Alarm Display – "Previous Page" and "Next Page"
- Acknowledging an alarm. (Since you cannot see a remote tagname go into alarm, you cannot acknowledge it.)

Notes

1) Implementation of remote tagname references does not require conversion of applications that were created with earlier versions of InTouch that do not support this feature. However, once implemented, the applications will not be backwards compatible with the earlier versions.

2) WindowViewer supports 32767 references to local tagnames and *x* references to active remote references, where *x* = 61,405 minus the number of tagnames defined in the local Tagname Dictionary.

3) Remote references can be a maximum of 95 characters long.

4) The **IOSetAccessName** (**SetDdeTopic** in versions prior to InTouch 7.0) function is also supported for remote references and works that same as it does for local tagnames.

Using Remote Tagname References

There are actually three ways that you can specify a remote tagname reference in a client application:

1. Using the *AccessName:"item"* reference in any animation link tagname or expression or, in an InTouch QuickScript.

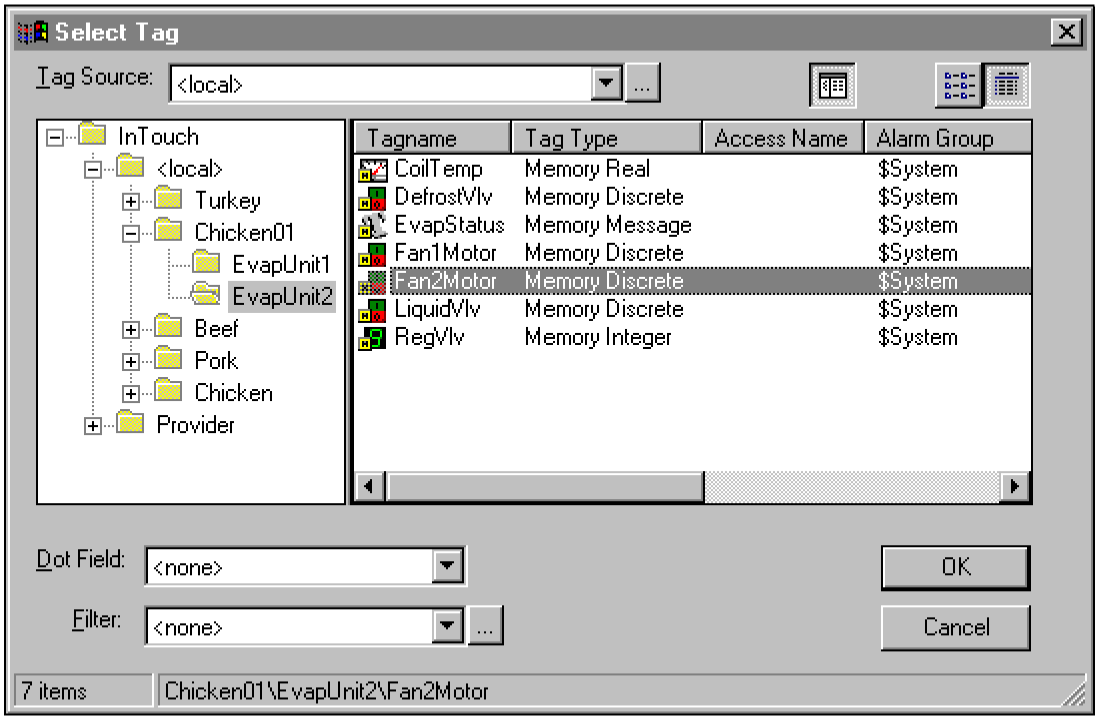
2. Importing a window or QuickScript and converting the placeholder tagnames to remote tagname references by using the **Substitute Tags** command on the **Special** menu in WindowMaker.

   For more information on converting placeholder tagnames, see "Converting Tagnames to Remote References."

   One of the powerful features of InTouch is the ability to import a window from another application. When you import a window, all of its scripts and animation links are imported with it. However, all of the tagnames used in the animation links and scripts are automatically converted to placeholders. You can convert all the placeholder tagnames to remote tagname references and, if desired, design an application with no local tagnames.

   For more information on importing windows or scripts, see Chapter 2 - Using WindowMaker.

3. Selecting the remote tagname that you want to use for an object or QuickScript by configuring the remote application as the tag source in the Tag Browser. For example:



Tagname	Tag Type	Access Name	Alarm Group
CoilTemp	Memory Real		$System
DefrostVlv	Memory Discrete		$System
EvapStatus	Memory Message		$System
Fan1Motor	Memory Discrete		$System
Fan2Motor	Memory Discrete		$System
LiquidVlv	Memory Discrete		$System
RegVlv	Memory Integer		$System

7 items Chicken01\EvapUnit2\Fan2Motor

For more information on selecting remote tagnames from the Tag Browser, see "Defining Tag Sources"

Dynamic Reference Addressing (DRA)

Dynamic Reference Addressing allows you to address multiple data sources with a single tagname. By assigning a valid reference to the **.Reference** field of an I/O type tagname, you can dynamically change the address of the data source for the tagname. Each I/O type tagname has a reference associated with for the address of its data source. The valid syntax for the **.Reference** field includes:

Tagname.Reference="accessname.item"	Changes Access Name and item.
Tagname.Reference="[.]item"	Same Access Name, different item.
Tagname.Reference="accessname."	Changes Access Name.
Tagname.Reference=""	Deactivates the tagname. If the Access Name or Item is not specified, the current value for that field is assumed.

Note Dynamic Reference Addressing is not valid for remote tagname references.

Using Dynamic References

Dynamic references are used to view data points whose values are only needed temporarily, such as in diagnostic applications. Since the data source of a tagname can be changed, dynamic references should not be used for any data that needs to be permanently stored or continuously monitored for alarm conditions.

A good example of a traditional use of dynamic references is the diagnostic application. In this application, a single tagname is used to view the input value of any analog point in a PLC. This allows a maintenance person to immediately view the status of any point for trouble-shooting purposes.

➢ **To create a diagnostic application:**

1. Create an I/O Integer type tagname. In this example, the tagname is called "AnalogSpy." It has an initial reference to **PLC1** for the Access Name and **WX001** for the item name.
2. Create a text object by typing a # sign.
3. Double-click the # sign to open the animation links dialog box.
4. Click **String** in the **User Inputs** section. The **Input -> String Tagname** dialog box will appear.
5. In the **Tagname** box, type **AnalogSpy.Reference**
6. Click **OK**.
7. Start WindowViewer to compile and run the application.
8. Click on the text object, and enter a new value for the Access Name and item name assigned to the tagname.

   For example, to view item **WX031** from Access Name **PLC6**, enter **PLC6.WX031** as the reference.
9. If you want to confirm that the new reference is valid, use the **.ReferenceComplete** field described in the next section.

Using IOSetItem Function to Change References

The **IOSetItem** (**SetDdeItem** in versions prior to InTouch 7.0), function is used to set an I/O type tagname's **.Reference** field. The basic format of this function is:

```
IOSetItem(TagName, AccessName, Item)
```

The tagname, Access Name, and item values can be specified as literal strings, or they can be string values provided by other InTouch tagnames or functions. For example, the **.Reference** field of tagname "MyTag1" can be changed to point to the "Excel" Access Name and the "R1C1" item by:

```
IOSetItem("MyTag1", "Excel", "R1C1");
```

or by,

```
Number = 1;
TagNameString = "MyTag" + Text(Number, "#");
IOSetItem(TagNameString, "Excel", "R1C1");
```

If an empty string ("") is specified for both the Access Name and item values, then the tagname is deactivated. For example, the tagname "MyTag2" is deactivated by:

```
IOSetItem("MyTag2", "", "");
```

If an empty string is specified only for an Access Name value, then the tagname's current Item value is retained and its Access Name value is updated. For example, the following changes the Access Name for tagname "MyTag3" to "Excel2" without affecting its current Item value:

```
IOSetItem("MyTag3", "Excel2", "");
```

Likewise, if an empty string is specified only for an Item, then the tagname's current Item value is retained and its Access Name value is updated. For example, the following changes the Item for tagname "MyTag3" to "R1C2" without affecting its current Access Name value:

```
IOSetItem("MyTag4", "", "R1C2");
```

🕮 For more information on **IOSetItem**, see your *InTouch Reference Guide.*

Using the .ReferenceComplete to Verify References

Each I/O type tagname has a **.ReferenceComplete** field. This discrete field provides confirmation that the item requested in the reference field is reflected in the **.Value** field.

The **.ReferenceComplete** field initializes to false (0) at startup of WindowViewer. When it is confirmed that the **.Value** field is being updated by the source specified in the **.Reference** field, the **.ReferenceComplete** value is set to true (1). If the **.Reference** field is changed, the **.ReferenceComplete** field is automatically set to false (0), and then updated to true(1) when the new value is updated.

The 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. You also use the Tag Browser to access the dialog boxes that allow you to perform tagname editing, replication, and to select tagnames (remote references) in remote tag sources.

The first time you access the Tag Browser, by default, **<local>** will be selected for the tag source. Meaning 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 you use to access it. The following lists the primary methods that you can use to access the Tag Browser in each mode:

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.

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.

  For more information on the Tag Browser modes, see "Tag Browser Selection Modes."

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

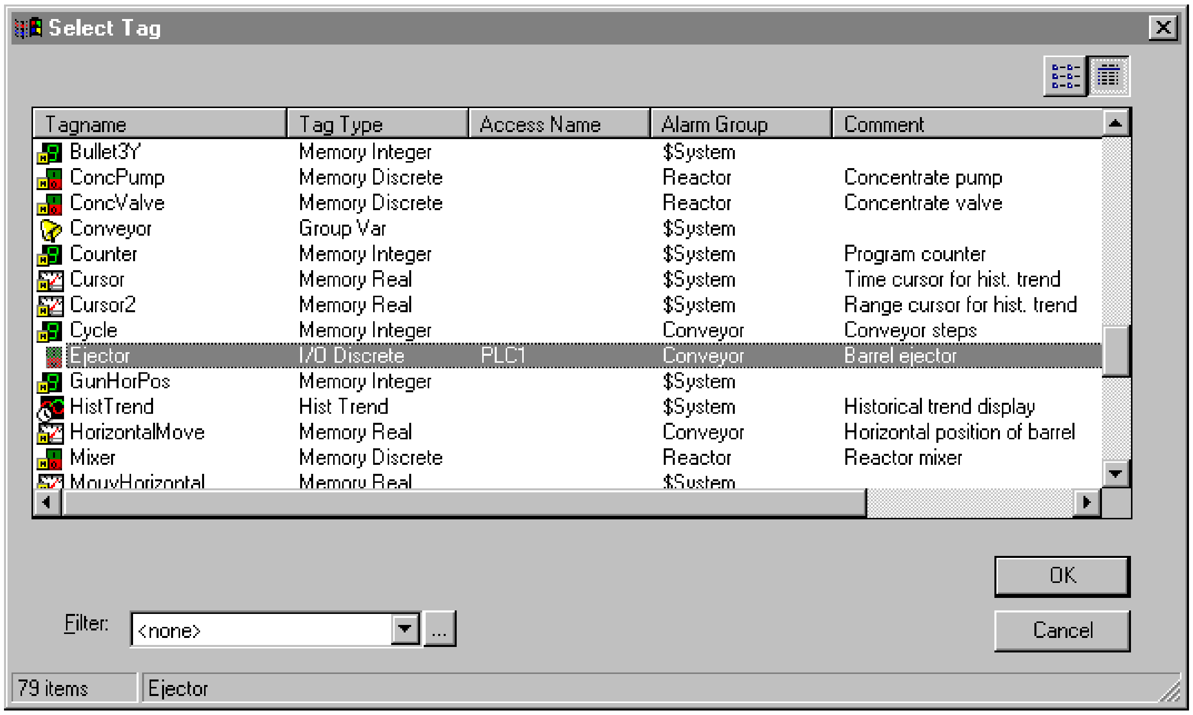
- Total number of items in the application.
- The name of the currently selected item.
- Tagname **.field** selected, if any.
- The Access Name associated with the tag source.

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 or, during runtime (when the operator is allowed to make runtime changes to a historical trend) when selecting a new tagname for a historical trend pen, the tagnames displayed (and available for selecting) will be limited to the current InTouch application. For example:



Tagname	Tag Type	Access Name	Alarm Group	Comment
Bullet3Y	Memory Integer		$System	
ConcPump	Memory Discrete		Reactor	Concentrate pump
ConcValve	Memory Discrete		Reactor	Concentrate valve
Conveyor	Group Var		$System	
Counter	Memory Integer		$System	Program counter
Cursor	Memory Real		$System	Time cursor for hist. trend
Cursor2	Memory Real		$System	Range cursor for hist. trend
Cycle	Memory Integer		Conveyor	Conveyor steps
Ejector	I/O Discrete	PLC1	Conveyor	Barrel ejector
GunHorPos	Memory Integer		$System	
HistTrend	Hist Trend		$System	Historical trend display
HorizontalMove	Memory Real		Conveyor	Horizontal position of barrel
Mixer	Memory Discrete		Reactor	Reactor mixer
MouvHorizontal	Memory Real		$System	

Filter: <none>

79 items | Ejector

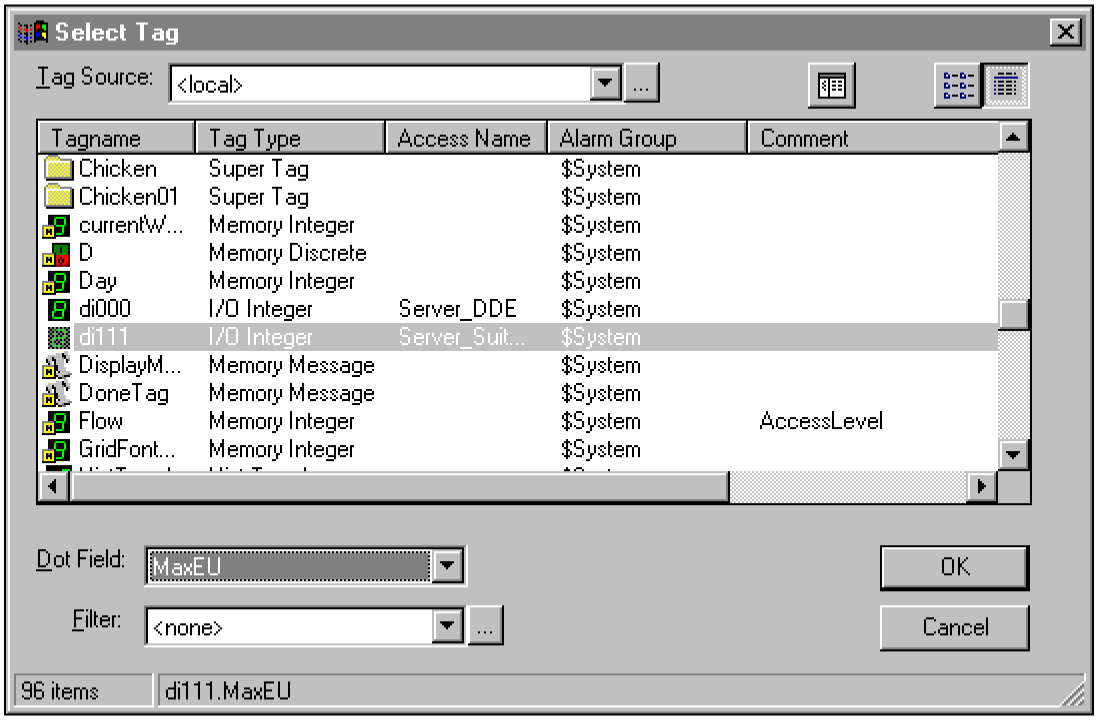
When you access the Tag Browser from the Tagname Dictionary and you select a tagname in this view, it's Tagname Dictionary definition will appear after you click **OK**.

Note Tagname **.fields** cannot be selected in this mode.

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 you select a tagname and/or tagname**.field** 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. For example:



Tag Source: <local>

Tagname	Tag Type	Access Name	Alarm Group	Comment
Chicken	Super Tag		$System	
Chicken01	Super Tag		$System	
currentW...	Memory Integer		$System	
D	Memory Discrete		$System	
Day	Memory Integer		$System	
di000	I/O Integer	Server_DDE	$System	
di111	I/O Integer	Server_Suit...	$System	
DisplayM...	Memory Message		$System	
DoneTag	Memory Message		$System	
Flow	Memory Integer		$System	AccessLevel
GridFont...	Memory Integer		$System	

Dot Field: MaxEU

Filter: <none>

96 items | di111.MaxEU

➢ **To select a .field:**

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

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

   Note Dot Field is not available when you access the Tag Browser from the Tagname Dictionary or, during runtime, when selecting a tagname for a historical trend pen from the **Historical Trend Setup** dialog box. (The historical trend must be configured with the **Allow runtime changes** option selected.)

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

   Not every tagname type has the same **.fields**. For example, a **Discrete** type tagname has **.OnMessage**, whereas an analog does not. If you select a **Discrete** type tagname and you assign **.OnMessage** to it, and then you select another **Discrete** type tagname, the displayed **.field** list will not change. But, if you select another type of tagname in the control view list, for example an analog, the displayed **.field** will revert to **<none>**.

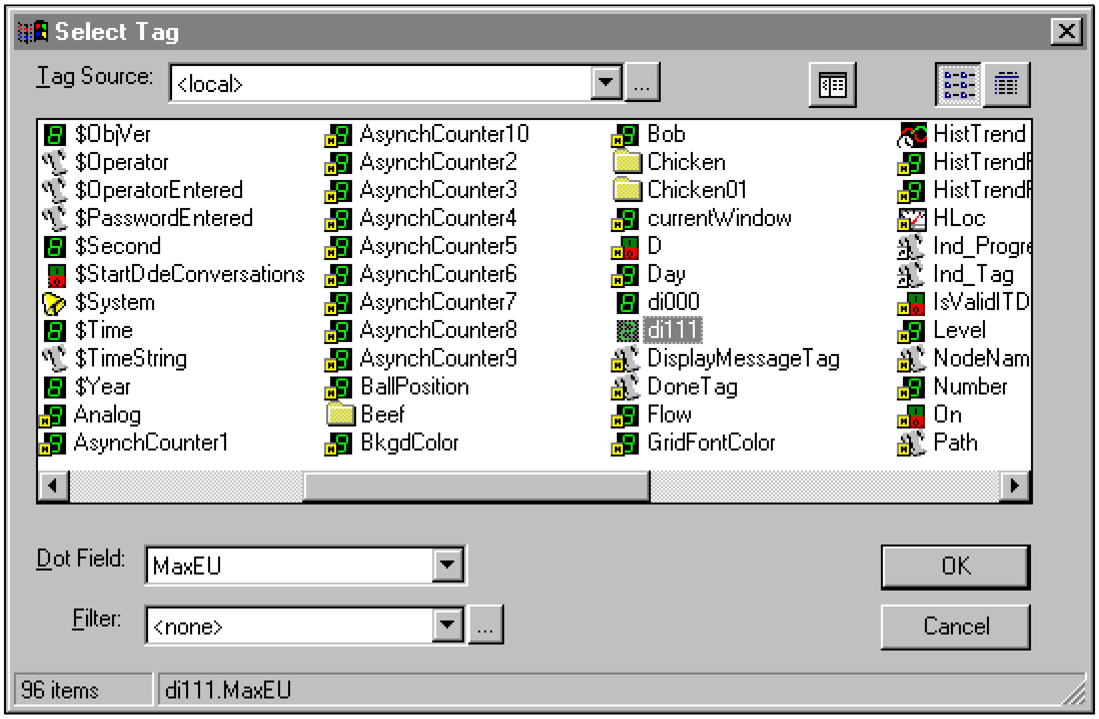
Tag Browser Views

The Tag Browser supports three control views; Tagname List Control, Tagname Details Control and Tagname Tree View Control.

List View

The list view is used to display and select tags within the current selection mode (described above). The Tagname List Control view displays the tagnames in two views depending upon the state of the **List View** and **Details View** buttons:

When you select list view, small icons will be displayed next to the tagnames with icons displayed according to the type of each tagname. No other fields will be displayed in the list view. For example:



Tag Source: <local>

$ObjVer	AsynchCounter10	Bob	HistTrend
$Operator	AsynchCounter2	Chicken	HistTrendI
$OperatorEntered	AsynchCounter3	Chicken01	HistTrendI
$PasswordEntered	AsynchCounter4	currentWindow	HLoc
$Second	AsynchCounter5	D	Ind_Progre
$StartDdeConversations	AsynchCounter6	Day	Ind_Tag
$System	AsynchCounter7	di000	IsValidITD
$Time	AsynchCounter8	di111	Level
$TimeString	AsynchCounter9	DisplayMessageTag	NodeNam
$Year	BallPosition	DoneTag	Number
Analog	Beef	Flow	On
AsynchCounter1	BkgdColor	GridFontColor	Path

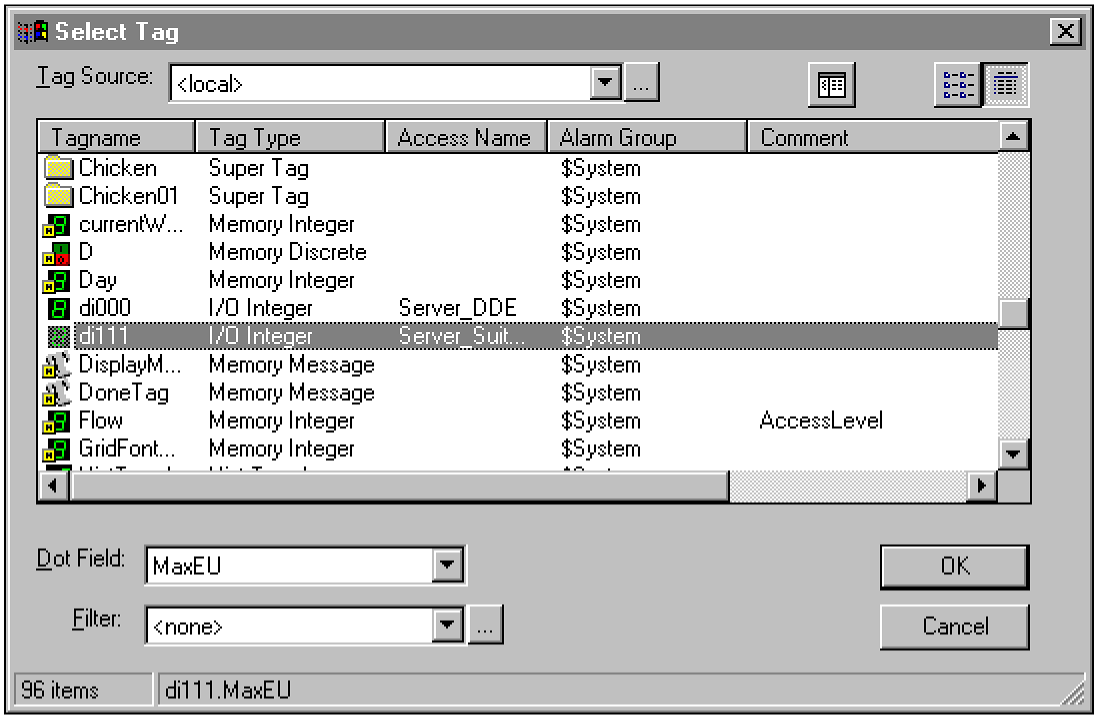
Dot Field: MaxEU

Filter: <none>

96 items | di111.MaxEU

Details View

When you select details view, the tagnames and their details are displayed in a multi-column format. The details displayed are Tagname Name, Tagname Type, Access Name, Alarm Group and Comment. You can sort the list by each detail type by clicking on its column header name. An item can be selected by clicking on any portion of its display, not just the tagname. (The entire row will be highlighted.) For example:



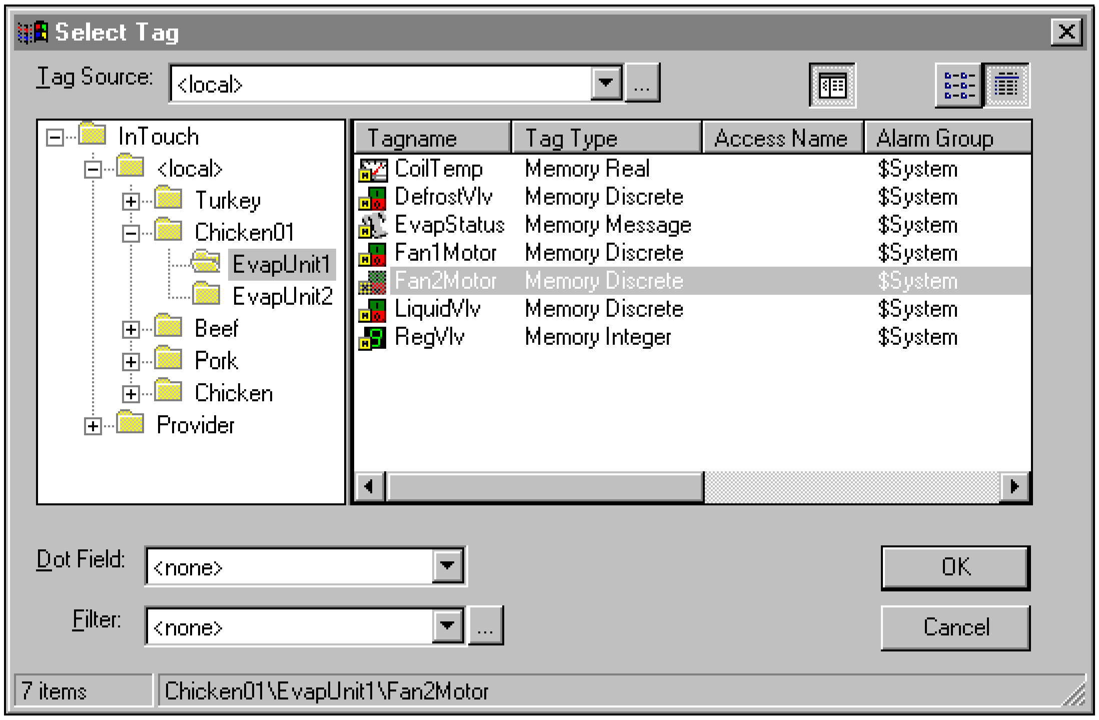
Tagname	Tag Type	Access Name	Alarm Group	Comment
Chicken	Super Tag		$System	
Chicken01	Super Tag		$System	
currentW...	Memory Integer		$System	
D	Memory Discrete		$System	
Day	Memory Integer		$System	
di000	I/O Integer	Server_DDE	$System	
di111	I/O Integer	Server_Suit...	$System	
DisplayM...	Memory Message		$System	
DoneTag	Memory Message		$System	
Flow	Memory Integer		$System	AccessLevel
GridFont...	Memory Integer		$System	

When you switch views, the selected tagname will remain visible and highlighted in the new view.

Tree View

The Tree View displays the tagnames in two views depending upon the state of the **List View** and **Details View** buttons. When you select the tree view, a pane appears on the left side of the dialog box. By using the Tree View you can also access the member tagnames in any SuperTag template.

If the **Details View** mode is active when you select the Tree View, Tag Browser appear as follows:



Tag Source: <local>

InTouch
<local>
Turkey
Chicken01
EvapUnit1
EvapUnit2
Beef
Pork
Chicken
Provider

Tagname	Tag Type	Access Name	Alarm Group
CoilTemp	Memory Real		$System
DefrostVlv	Memory Discrete		$System
EvapStatus	Memory Message		$System
Fan1Motor	Memory Discrete		$System
Fan2Motor	Memory Discrete		$System
LiquidVlv	Memory Discrete		$System
RegVlv	Memory Integer		$System

Dot Field: <none>

Filter: <none>

7 items | Chicken01\EvapUnit1\Fan2Motor

To expand a listing in the Tree View, double-click the application name or, click the ⊞. To collapse a listing, double-click the application name again or, click the ⊟. Double-clicking an application in the tree view pane is the same as selecting it in the **Tag Source** list.

Note When you "drill down" through different levels in the Tag Browser, you can use the BACKSPACE key to "back up" to the previous level.

Defining Tag Sources

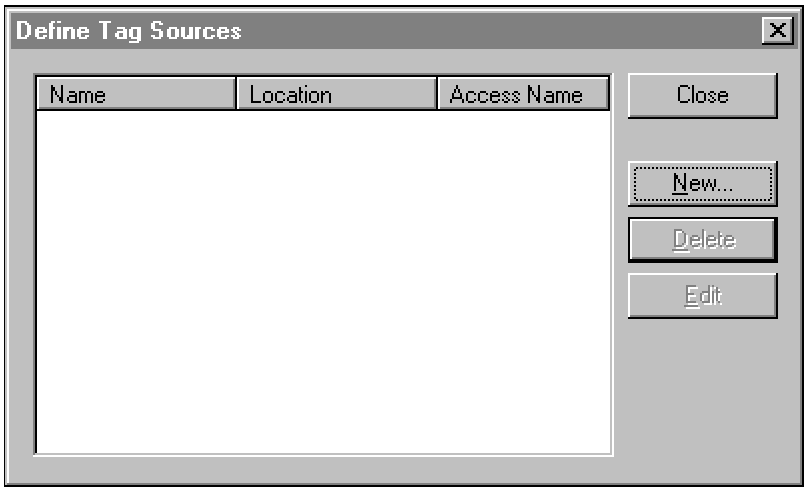
You must define the tag sources for viewing in the Tag Browser. The procedures for adding, deleting or editing tag sources are described in this section. When you add or edit a tag source definition, you will enter information such as the local network Access Name you want to associate with the tag source's tagnames, a user-defined application name, and the data source for the tag source.

Note You will also use these procedures when you are converting placeholder tagnames to remote tagname references.

For more information on remote tagname references, see "Converting Tagnames to Remote References."

➢ ... **To define a tag source:**

1. Open the Tag Browser, and then click the Define Tag Sources button. The **Define Tag Sources** dialog box appears:



Note If tag sources are already defined, they will be listed when the dialog box appears. The list will include the user-defined **Name** for the tag source, the **Location** of the tag source (path) and the local network **Access Name** associated with the application.

To select multiple tag sources, hold down the SHIFT key as you click each name. To select multiple tag sources that are not consecutive in the list, hold down the CTRL key as you click each name.

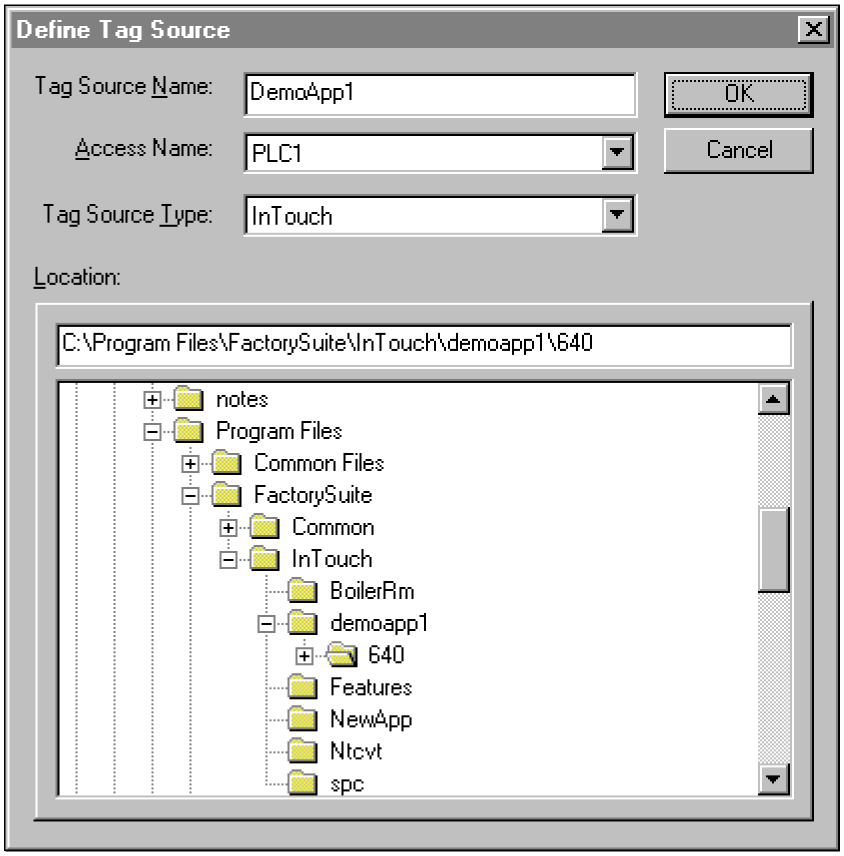
Note When the **Define Tag Sources** dialog box closes, you must click the **Tag Source** arrow in the Tag Browser and select the new tag source in the list. The Tag Browser is then refreshed and tagnames for the selected tag source are displayed.

3. To remove a tag source(s) from the **Tag Source** list in the Tag Browser, click the Define Tag Sources button. The **Define Tag Sources** dialog box will appear. Select the tag source in the list, and then click **Delete**.

4. To edit a defined tag source, select it in the list, and then click **Edit**. The **Define Tag Source** dialog box will appear displaying the configuration for the selected tag source.

5. To define a new tag source, click **New**. The **Define Tag Source** dialog box appears:

Note When you click **New**, if no Access Name is defined in your local application, a message box will appear telling you there are no Access Names defined and you will not be allowed to define a new tag source. (Tag sources must be associated with a local network Access Name.)



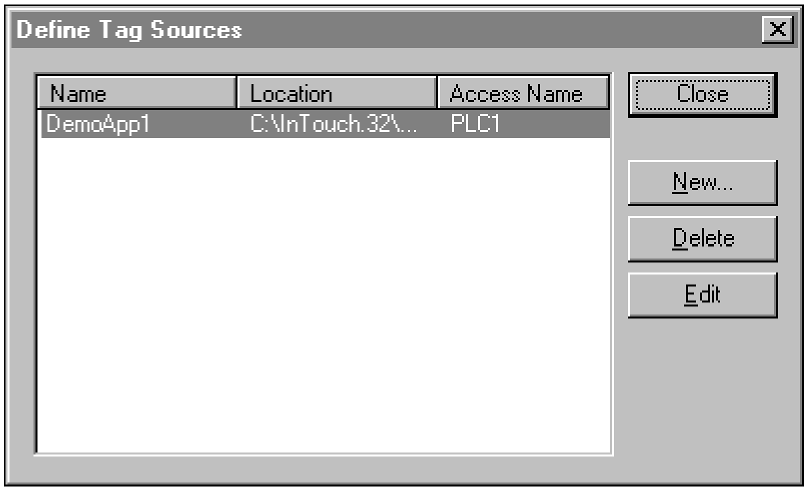
Tag Source Name: DemoApp1

Access Name: PLC1

Tag Source Type: InTouch

Location: C:\Program Files\FactorySuite\InTouch\demoapp1\640

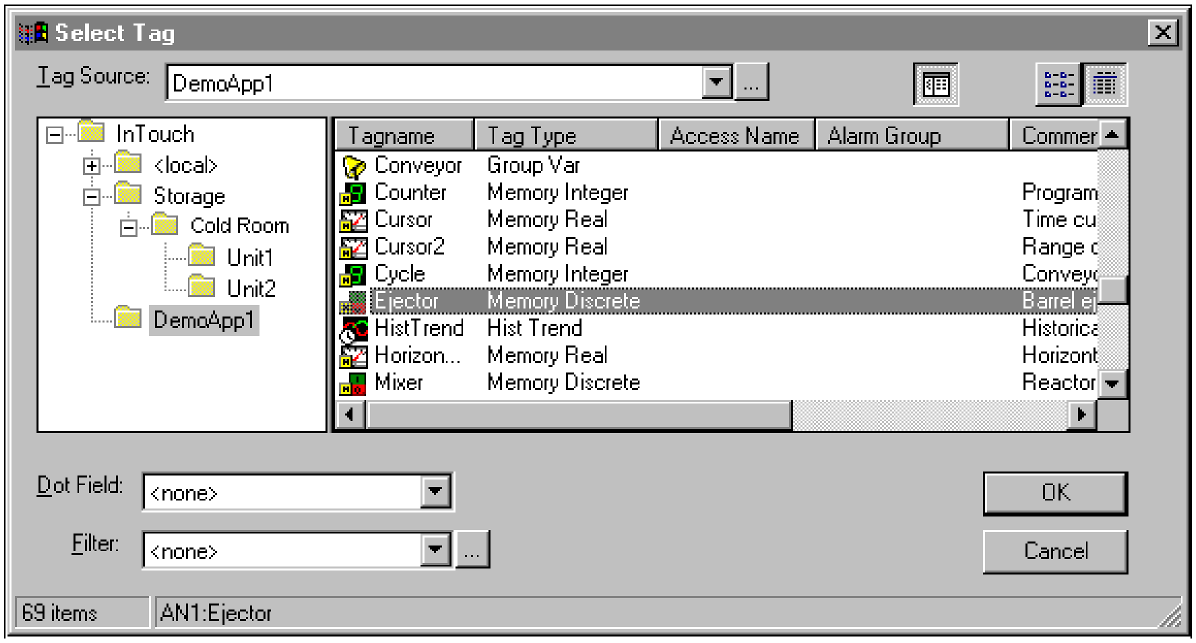
6. In the **Tag Source Name** box, type a name to identify the tag source.
7. Click the **Access Name** arrow and select the Access Name in the local application that you want to associate with the tagnames in the tag source.
8. Click the **Tag Source Type** arrow and select the source for the tag source's tagname database. (By default, **InTouch** is displayed.)
9. The **Location** box displays the full path to the tag source.
10. In the directory tree pane, locate the tag source, and then click **OK**. The **Define Tag Sources** dialog box reappears displaying the selected tag source:



Name	Location	Access Name
DemoApp1	C:\InTouch.32\...	PLC1

11. Click **Close**. The Tag Browser will reappear.

12. Click the tool to display the tree view pane to display all defined tag sources:



Tagname	Tag Type	Access Name	Alarm Group	Commer
Conveyor	Group Var			
Counter	Memory Integer			Program
Cursor	Memory Real			Time cu
Cursor2	Memory Real			Range c
Cycle	Memory Integer			Conveyo
Ejector	Memory Discrete			Barrel e
HistTrend	Hist Trend			Historica
Horizon...	Memory Real			Horizont
Mixer	Memory Discrete			Reactor

If you are not using the tree view mode, click the **Tag Source** arrow and select the name for the tag source that you want to display in the list. The Tag Browser will refresh and the tag sources' tagnames will be displayed.

The first time you access the Tag Browser, by default, **<local>** will be selected for the **Tag Source**. Thereafter, the tagnames for previously accessed tag source will be displayed.

13. Click **OK**.

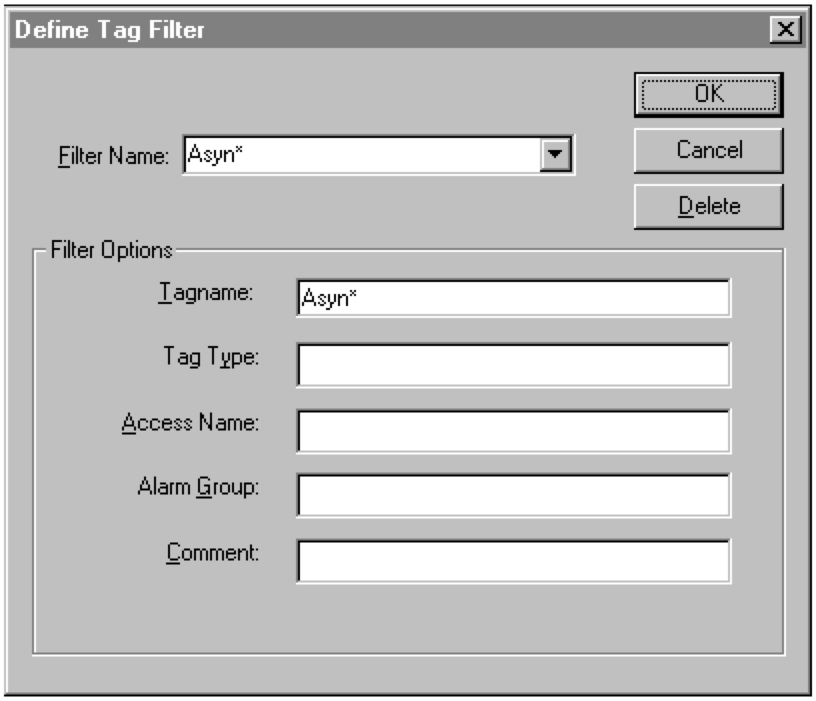
Defining Tag Browser Filters

You will use the procedures described in this section to define the filters (search criteria) you want to use to populate the Tag Browser. By creating filters, you can sort any tagname list and display only the tagnames that meet the criteria you specify. You can sort the tagnames 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 the 20 or so 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.

➢ **To define a search filter:**

1. Click the Define Filter button. The **Define Tag Filter** dialog box appears:



Filter Name: Asyn*

Tagname: Asyn*

Tag Type:

Access Name:

Alarm Group:

Comment:

If you right click the mouse in any of the text entry boxes, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Filter Name** box, type 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 type is added to the list.)

   All of the **Filter Option** controls (**Tagname**, **Tag Type**, **Access Name**, **Alarm Group** and **Comment**) allow you to enter a wildcard expression to limit the scope of your search. If no filter is used, all of the tagnames in the currently displayed tag source will be displayed.

The multiple wildcard is the asterisk (*****). For example, "Asyn*" would search for all tagnames beginning with the character "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", and so on.

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 &.

3. In the **Tagname** box, type the tagname expression that you want to use as a filter. If left blank, the system will ignore this field in the filter definition.
4. In the **Access Name** box, type the local Access Name expression that you want to use as a filter. If left blank, the system will ignore this field in the filter definition.
5. In the **Alarm Group** box, type the name of the Alarm Group expression that you want to use as a filter. If left blank, the system will ignore this field in the filter definition.
6. In the **Comment** box, type the comment expression you want 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.

   The **Filter Name** will now appear in the **Filter** list in the Tag Browser and you can select it to display only the tagnames meeting the criteria specified in the filter.

➢ **To delete a search filter:**

1. Click the **Filter** arrow and select the filter name in the list that you want to delete.
2. Click **Delete**. The filter is immediately deleted.

InTouch Cross Reference Utility

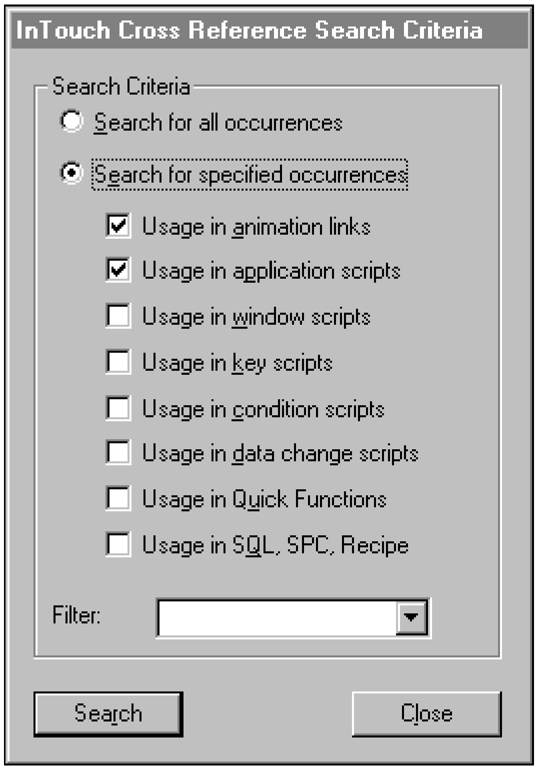
The Tagname Cross Referencing utility allows you to determine your tagname and SuperTag usage in animation links, wizards, InTouch QuickScripts, QuickFunctions, ActiveX controls, scripts and the following InTouch add-on programs, SPC Pro, SQL Access Manager and Recipe Manager. For all objects such as 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.

For convenience, the Tagname Cross Reference utility can remain open in WindowMaker while you perform other tasks.

Note The InTouch Cross Reference utility does not display remote tagname references or Touch Pushbutton Action Scripts.

➢ **To use the Tagname Cross Reference utility:**

1. On the **Special** menu, click **Cross Reference** or, in the Application Explorer double-click **Cross Reference**. The **InTouch Cross Reference Search Criteria** dialog box appears:



InTouch Cross Reference Search Criteria

Search Criteria
- ○ Search for all occurrences
- ● Search for specified occurrences
  - [✓] Usage in animation links
  - [✓] Usage in application scripts
  - [ ] Usage in window scripts
  - [ ] Usage in key scripts
  - [ ] Usage in condition scripts
  - [ ] Usage in data change scripts
  - [ ] Usage in Quick Functions
  - [ ] Usage in SQL, SPC, Recipe

Filter:

2. The **Search Criteria** group allows you to limit the scope of your search. You can easily determine the scope 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, and so on.
Search for specific occurrences	Search for only the tagname or SuperTag only in the specified options . For example, if you only want to search for the usage in window scripts, only select **Usage in window scripts**.

3. In the **Filter** box, type 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 type 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 tilde symbol (**?**). 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", and so on.

   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 &.

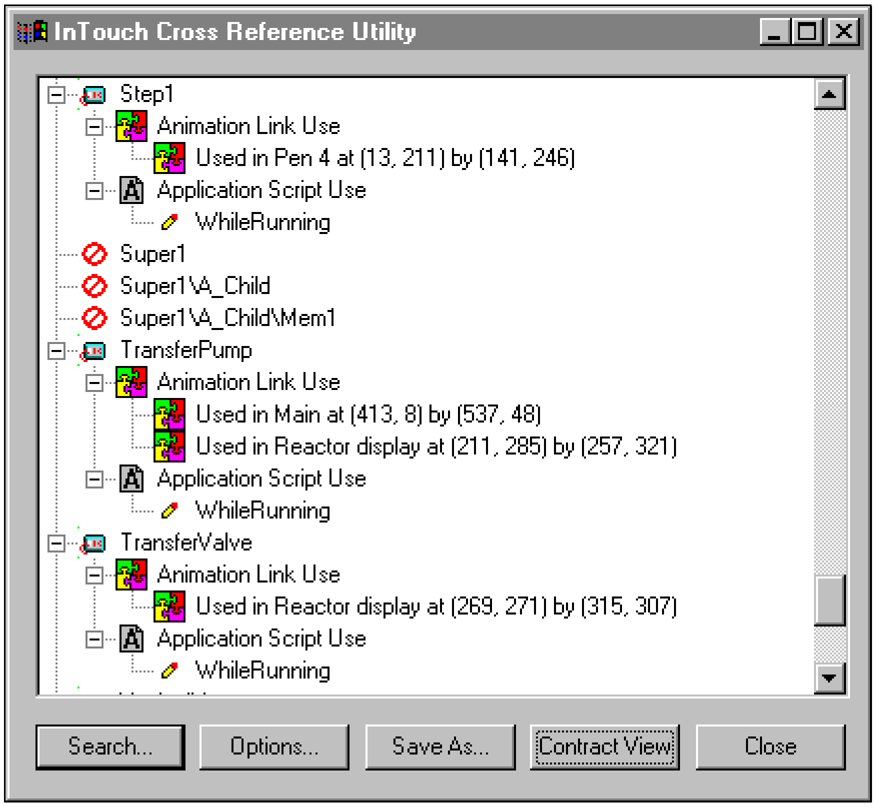
   If you right click the mouse in the **Filter** box, a menu will appear displaying the commands that you can apply to the selected text.

4. Click **Search** to begin the cross reference search based upon your 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:



InTouch Cross Reference Utility

```
Step1
    Animation Link Use
        Used in Pen 4 at (13, 211) by (141, 246)
    Application Script Use
        WhileRunning
Super1
Super1\A_Child
Super1\A_Child\Mem1
TransferPump
    Animation Link Use
        Used in Main at (413, 8) by (537, 48)
        Used in Reactor display at (211, 285) by (257, 321)
    Application Script Use
        WhileRunning
TransferValve
    Animation Link Use
        Used in Reactor display at (269, 271) by (315, 307)
    Application Script Use
        WhileRunning
```

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.

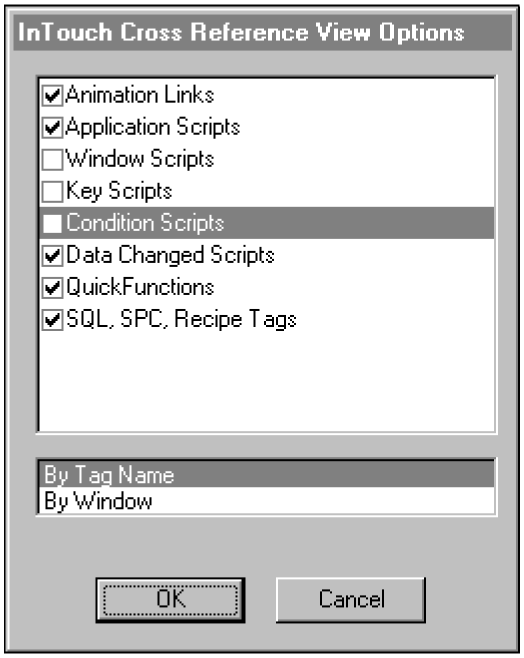
RCP — Displayed tagname or SuperTag is used in a Recipe Manager application.

Changing the Cross Reference Search Criteria

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

➢ **To change the 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:



InTouch Cross Reference View Options

- [✓] Animation Links
- [✓] Application Scripts
- [ ] Window Scripts
- [ ] Key Scripts
- [ ] Condition Scripts
- [✓] Data Changed Scripts
- [✓] QuickFunctions
- [✓] SQL, SPC, Recipe Tags

[✓ By Tag Name]
By Window

2. Select the search criteria options that you want to modify for your new search.

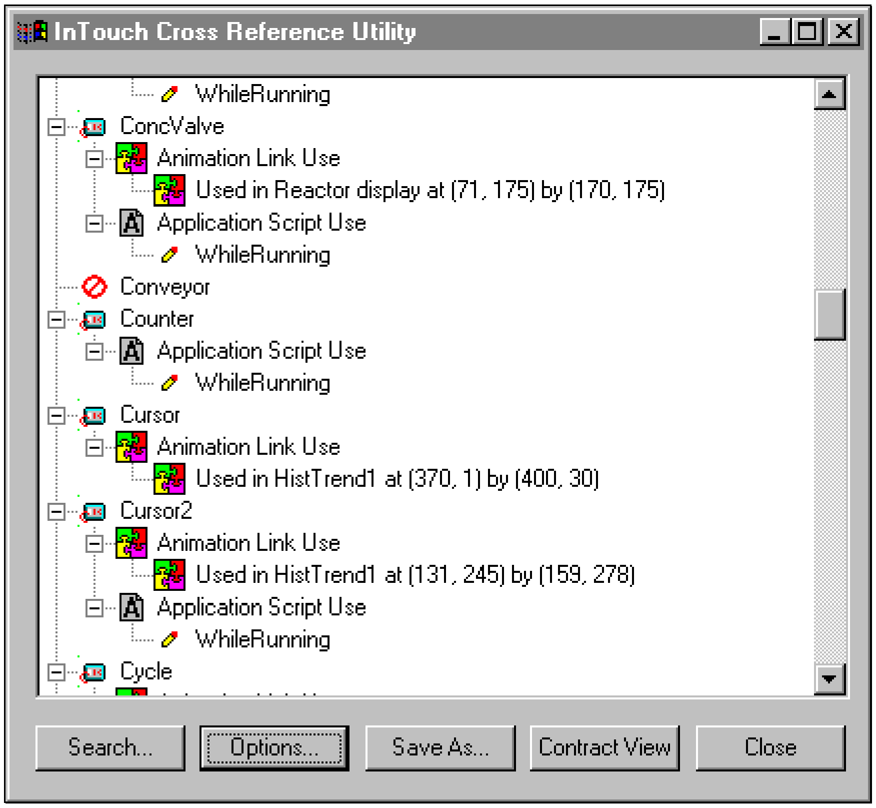
   The options available here are based upon the **Search Criteria** you originally selected in the **InTouch Cross Reference Search Criteria** dialog box. If you selected **Search for all occurrences**, all search criteria options will be available. If you selected **Search for specific occurrences**, only the specific occurrences you originally selected will be available. To change your **Search Criteria** selection, click **Cancel**. The **InTouch Cross Reference Utility** dialog box will reappear. Click **Search**, and select the new **Search Criteria** option.

3. In the list at the bottom of the dialog box, select whether you want the tree view populated by tagname or window name, and then click **OK**.

Cross Referencing By Tagname

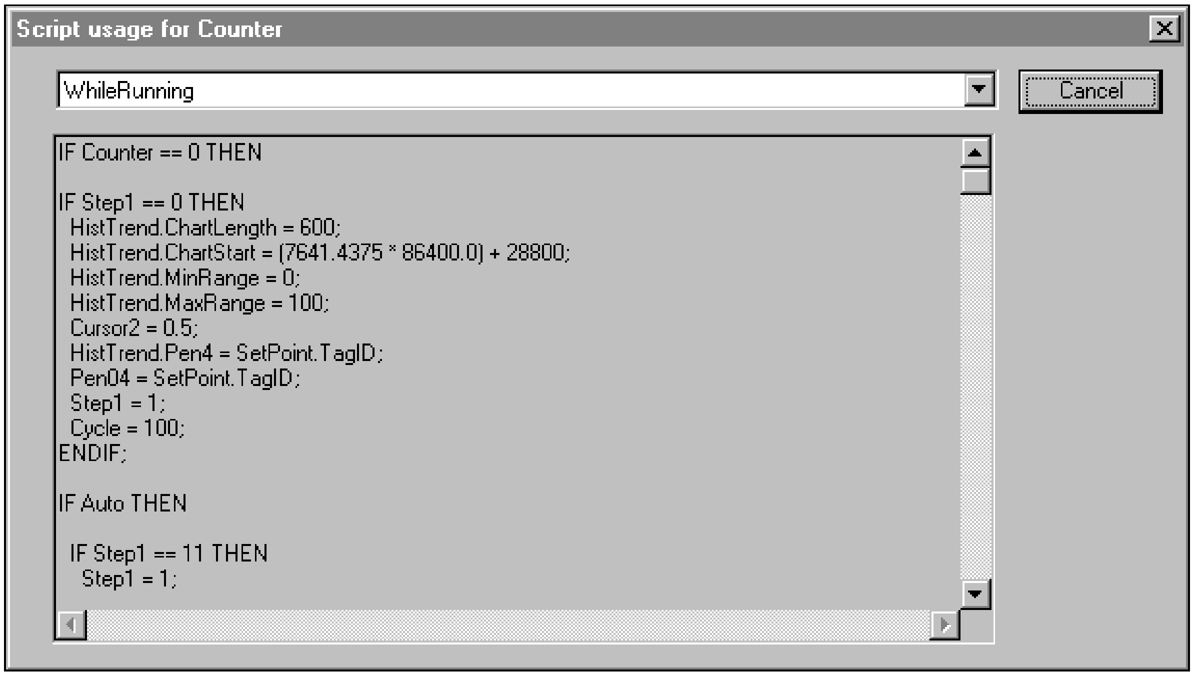
Alphabetically lists all tagnames found for you specified search criteria (default view). Based upon your specified search criteria, this view allows you to view the usage of all tagname found in windows, animation links, scripts and add-on applications.

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



```
      WhileRunning
ConcValve
   Animation Link Use
      Used in Reactor display at (71, 175) by (170, 175)
   Application Script Use
      WhileRunning
Conveyor
Counter
   Application Script Use
      WhileRunning
Cursor
   Animation Link Use
      Used in HistTrend1 at (370, 1) by (400, 30)
Cursor2
   Animation Link Use
      Used in HistTrend1 at (131, 245) by (159, 278)
   Application Script Use
      WhileRunning
Cycle
```

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:



```
IF Counter == 0 THEN

IF Step1 == 0 THEN
  HistTrend.ChartLength = 600;
  HistTrend.ChartStart = (7641.4375 * 86400.0) + 28800;
  HistTrend.MinRange = 0;
  HistTrend.MaxRange = 100;
  Cursor2 = 0.5;
  HistTrend.Pen4 = SetPoint.TagID;
  Pen04 = SetPoint.TagID;
  Step1 = 1;
  Cycle = 100;
ENDIF;

IF Auto THEN

  IF Step1 == 11 THEN
    Step1 = 1;
```

The list box at the top of the screen shows all scripts associated with the selected tagname. Click the arrow to open the list to 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 the QuickFunctions.

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

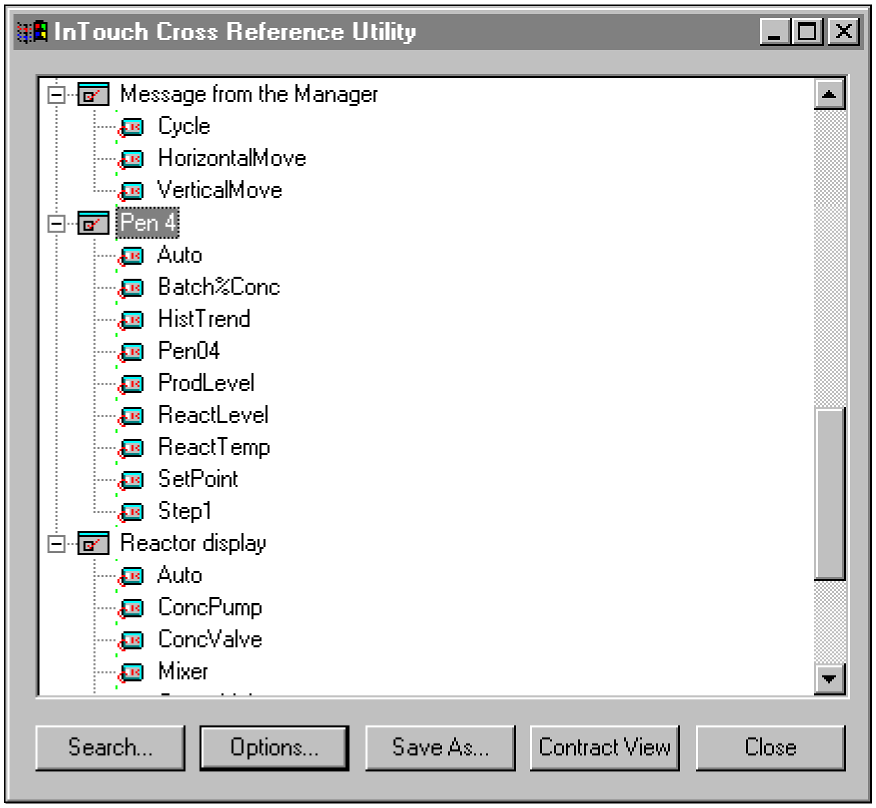
Note This is a read only view of the QuickScript. You cannot edit the QuickScript text in this dialog box. However, you can copy any portion or all of the QuickScript, and then paste it into any InTouch QuickScript editor window.

To copy the QuickScript to the Windows Clipboard, right-click the script, then click **Select All**. Right-click the script again and click **Copy**. You can also execute the Windows copy command (CTRL+C).

To paste the copied script into another InTouch QuickScript, in the Application Explorer under **Scripts**, double-click the type of script that you want to create. The QuickScript editor will appear. On the **Edit** menu, click **Paste**, or right-click the script window, and then click **Paste**. You can also execute the Windows paste command (CTRL+V).

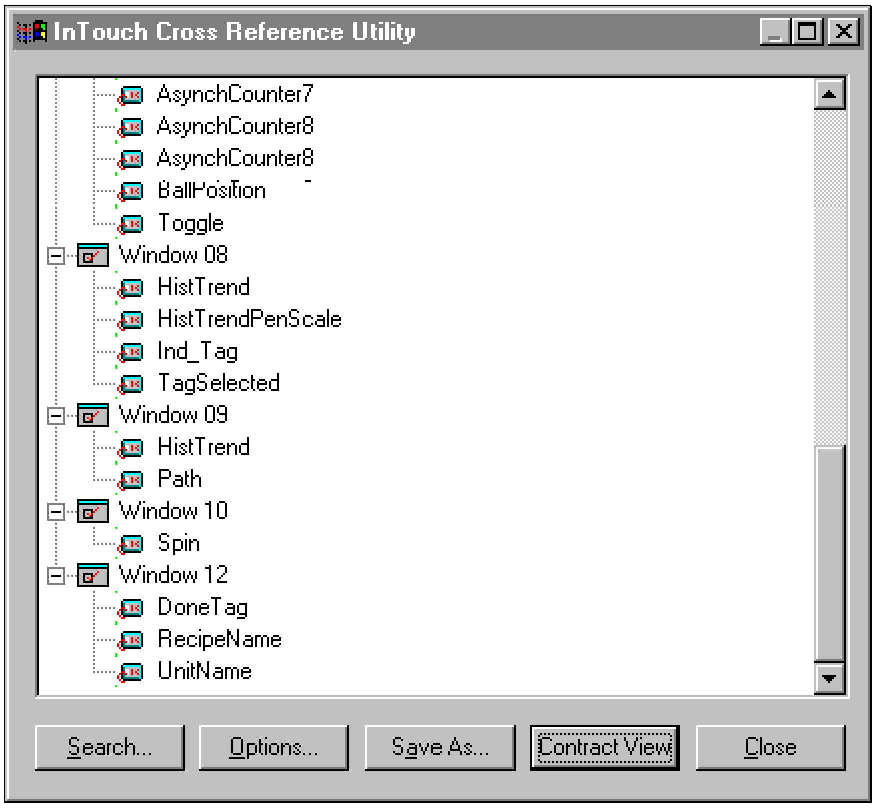
Cross Referencing By Window Name

Sorts the display by window name then the tagnames used in the window. For example:



Note This view only displays the tagnames used in the window, it does not include usage in animation links, scripts, and so on.

Click **Expand View** to display all view levels available for the displayed tagnames or windows. For example:



AsynchCounter7
AsynchCounter8
AsynchCounter8
BallPosition
Toggle
Window 08
HistTrend
HistTrendPenScale
Ind_Tag
TagSelected
Window 09
HistTrend
Path
Window 10
Spin
Window 12
DoneTag
RecipeName
UnitName

Click **Contract View** to return the dialog box to its default mode.

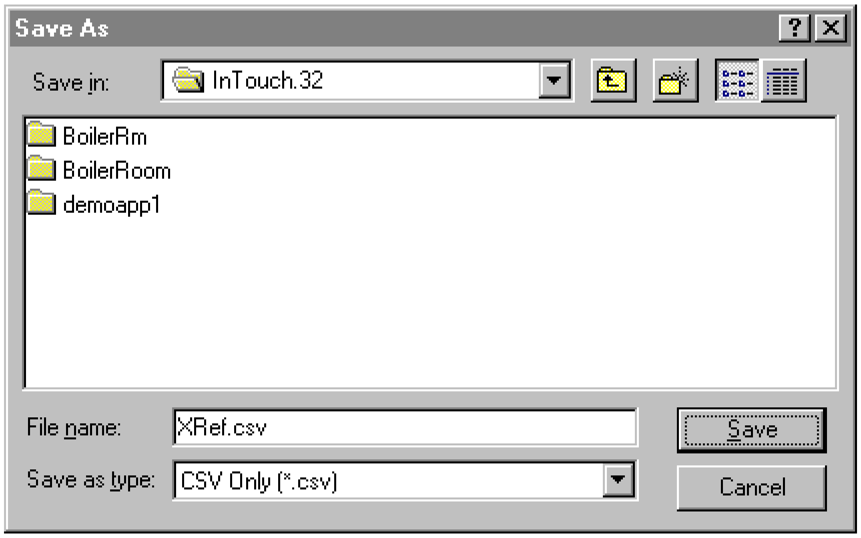
Click **Close** to exit the cross reference utility.

Saving Cross Reference Files

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

➢ **To save a cross reference file:**

1. In the **InTouch Cross Reference Utility** dialog box, click **Save As**. The **Save As** dialog box appears:



2. In the **File name** box, type the name that you want to save the cross reference file under.

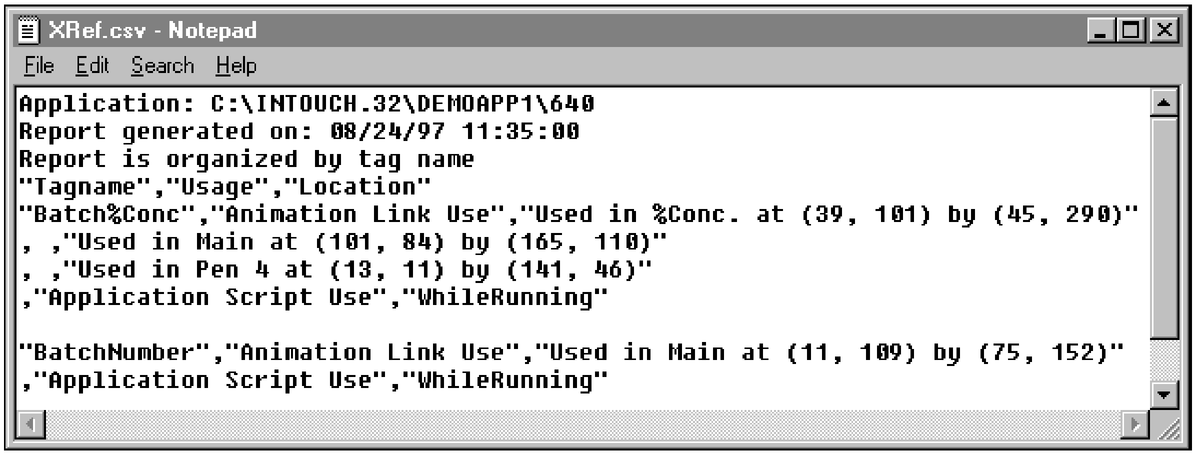
   The file must be save as a .CSV file.

3. Click **Save**.

Printing Cross Reference Files

You can open a cross reference .CSV file in any text editor program that supports the .CSV file format and print the cross reference file as a report.

For example, if you open the file in Notepad, it would appear as follows:



```
Application: C:\INTOUCH.32\DEMOAPP1\640
Report generated on: 08/24/97 11:35:00
Report is organized by tag name
"Tagname","Usage","Location"
"Batch%Conc","Animation Link Use","Used in %Conc. at (39, 101) by (45, 290)"
, ,"Used in Main at (101, 84) by (165, 110)"
, ,"Used in Pen 4 at (13, 11) by (141, 46)"
,"Application Script Use","WhileRunning"

"BatchNumber","Animation Link Use","Used in Main at (11, 109) by (75, 152)"
,"Application Script Use","WhileRunning"
```

To print the file, on the **File** menu, click **Print**.

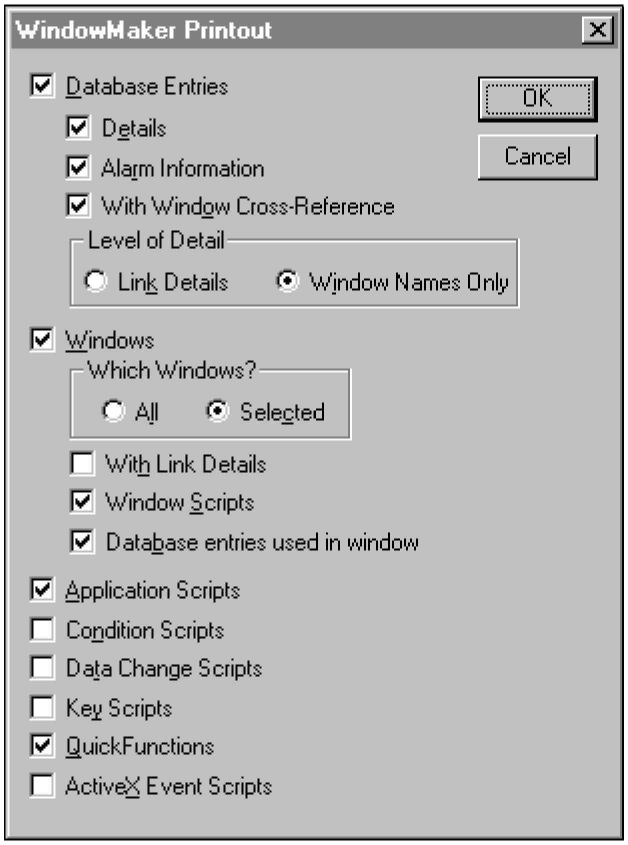
Printing Tagname Dictionary Details

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 you to determine the usage of tagnames.

Note Your Windows default printer will be used to produce the printout which will be 80 columns wide. Your default printer is selected and setup through the Windows Control Panel.

➢ **To print Tagname Dictionary details:**

1. On the **File** menu, click **Print**. The **WindowMaker Printout** dialog box appears:

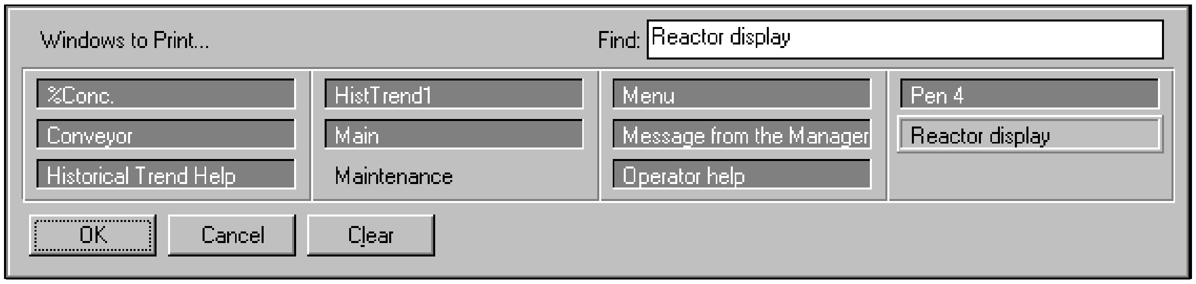


[✓ Database Entries]
[✓ Details]
[✓ Alarm Information]
[✓ With Window Cross-Reference]
Level of Detail: Link Details [✓ Window Names Only]
[✓ Windows]
Which Windows?: All [✓ Selected]
With Link Details
[✓ Window Scripts]
[✓ Database entries used in window]
[✓ Application Scripts]
Condition Scripts
Data Change Scripts
Key Scripts
[✓ QuickFunctions]
ActiveX Event Scripts

2. Select **Database Entries** if you want to print all database information. If you select **Database Entries**, 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).

3. 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:


Windows to Print...

Find: Reactor display

%Conc.	HistTrend1	Menu	Pen 4
Conveyor	Main	Message from the Manager	Reactor display
Historical Trend Help	Maintenance	Operator help	

4. Select the windows you want to print, then 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.
5. Click **OK** to begin printing your report.

Deleting Tagnames from the Dictionary

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, and so on. In these cases, InTouch continues to consider the tagname as being used in the application and will not allow you to delete it. Therefore, you may need to update your use count in order to delete a tagname.

➢ **To delete an unused tagname:**

1. Close WindowViewer if it is running.
2. On the **Special** menu, click **Tagname Dictionary**. The **Tagname Dictionary** dialog box will appear.
3. Click **Select**. The **Select Tagname** (Tag Browser) will appear.
4. Select the tagname that you want to delete then click **OK**. The **Tagname Dictionary** dialog box will appear displaying the selected tagname's definition.
5. Click **Delete**.

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

You can determine where a tagname is being used through the InTouch cross reference utility. (On the **Special** menu, click **Cross Reference**.) Or, you can print a report of all tagname links used in a window. (On the **File** menu, click **Print**.)

For more information on printing reports, see "Printing Tagname Dictionary Details."

Updating Use Counts

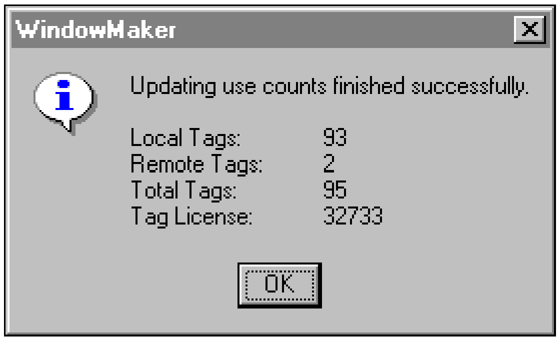
Since InTouch maintains a use count for each item in the database you may need to update the use counts to set all unused tagnames to zero before InTouch will allow you to delete any of them.

➢ **To update tagname use counts:**

1. Close all your windows.
2. On the **Special** menu, click **Update Use Counts**.

   A message box will appear telling you that updating use counts can take quite a while. You may at that time cancel the command or continue.

3. Click **Yes** to continue updating the use counts. Once the system has completed updating the use counts, the following dialog box appears:



Updating use counts finished successfully.

```
Local Tags:      93
Remote Tags:     2
Total Tags:      95
Tag License:     32733
```

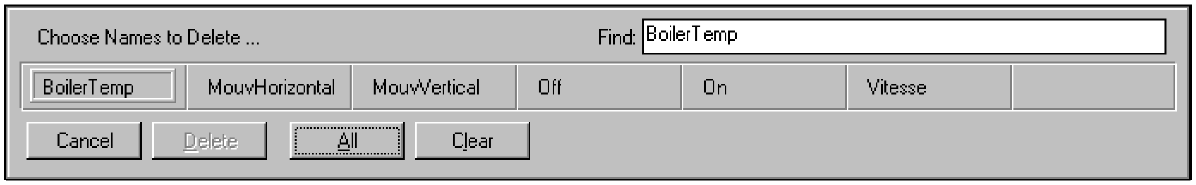
4. Click **OK**.

Deleting Unused Tagnames

After you have updated the use count, InTouch will allow you to delete all unused tagnames. You can either delete them by opening each of them in the **Tagname Dictionary** dialog and clicking **Delete** or, you can delete one or more of them at once by using the **Delete Unused Tags** command.

➢ **To delete multiple unused tagnames:**

1. On the **Special** menu, click **Delete Unused Tags**. The **Choose Names to Delete** dialog box appears:



2. Select the tagnames that you want to delete, then click **Delete**.
3. Click **All** to delete all tagnames displayed.

Warning! Tagnames that are only alarmed have no use count, and can be accidentally deleted. To ensure that alarmed only tagnames are included in the use count, you need to use them in a window or QuickScript.

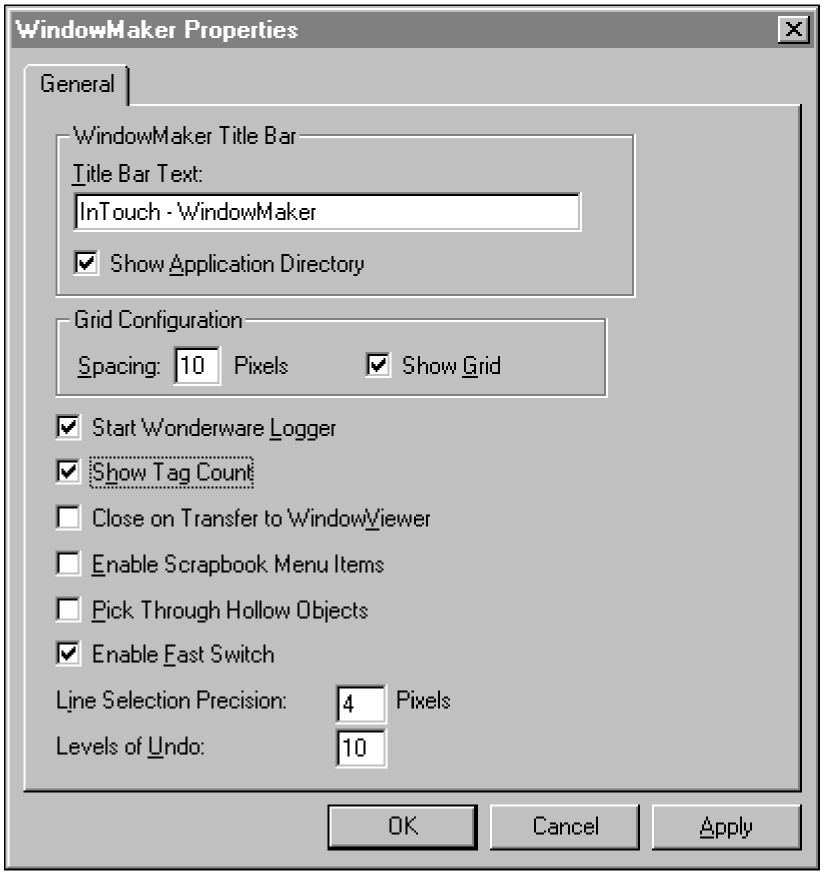
Displaying the Tag Usage Count

You can display the number of local tagnames that are currently defined in your Tagname Dictionary in the menu bar in WindowMaker. (The tagname count does not include internal system tagnames or remote tagname references.)

➢ **To show the tagname count:**

1. On the **Special** menu, point to **Configure**, and then click **WindowMaker**. The **WindowMaker Properties - General** property sheet appears:

   To quickly access the dialog box, in the Application Explorer under **Configure**, double-click **WindowMaker**.


WindowMaker Properties

General

WindowMaker Title Bar

Title Bar Text: InTouch - WindowMaker

[✓] Show Application Directory

Grid Configuration

Spacing: 10 Pixels [✓] Show Grid

[✓] Start Wonderware Logger

[✓] Show Tag Count

[ ] Close on Transfer to WindowViewer

[ ] Enable Scrapbook Menu Items

[ ] Pick Through Hollow Objects

[✓] Enable Fast Switch

Line Selection Precision: 4 Pixels

Levels of Undo: 10

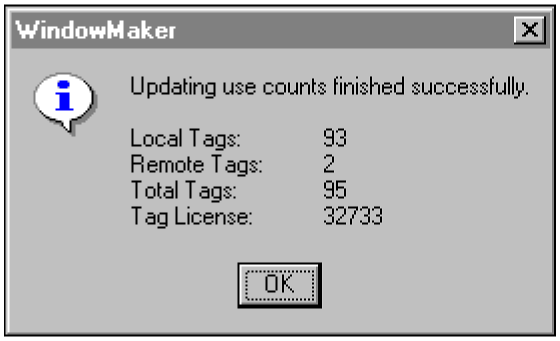
2. Select **Show Tagname Count**.
3. Click **OK**.
4. The total number of local tagnames defined in your Tagname Dictionary will now be displayed at the end of your WindowMaker menu bar.

   The entire Tagname Dictionary must be read in order to update the displayed tagname count. Therefore, when this option is turned on, performance may be degraded when you are making changes to your Tagname Dictionary. If your Tagname Dictionary is large, you should not select this option.

➢ **To determine remote tagname usage:**

1. On the **Special** menu, click **Update Use Counts**.
2. The system will update your tagname usage then display the following dialog box:

   Updating use counts can take a while.



WindowMaker

Updating use counts finished successfully.

```
Local Tags:      93
Remote Tags:     2
Total Tags:      95
Tag License:     32733
```

3. The **Remote Tags** line will display the number of remote tagnames used in your application.
4. Click **OK**.

Substituting Tagnames

When you duplicate an object it is an exact replica of the original including links, animation, scripts and so on. However if you need to use a different tagname on an object that you have duplicated you must change the tagname. In WindowMaker, this is called "substituting a tagname." You can select and change the tagnames for any object at any time and you can select multiple objects and change all their tagnames the same time.

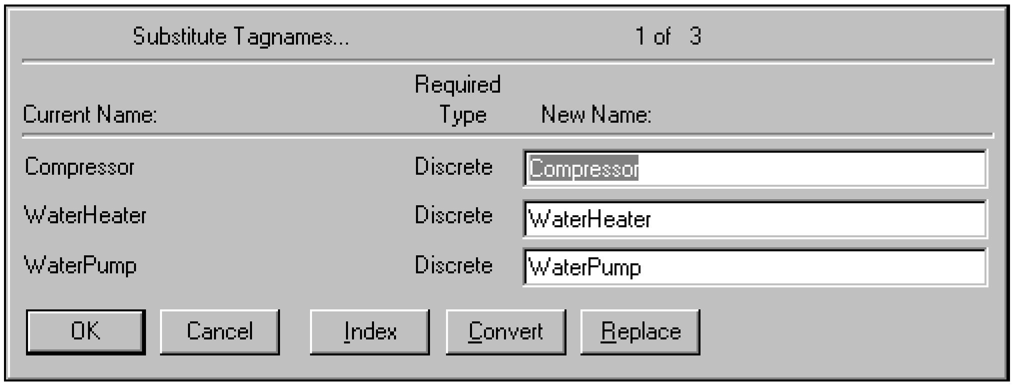
If you change a tagname for an object and WindowViewer is running, you will need to restart WindowViewer for the change to take effect.

If you system's license supports a limited number of tagnames, you can also convert your local tagnames to remote tagname references to reduce the number of tagnames defined in your local Tagname Dictionary.

➢ **To change an object's tagname to another local tagname:**

1. Select the object(s) whose tagname you want to change, and then on the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box appears:

   To quickly access the dialog box, right-click one of the selected objects, point to **Substitute**, and then click **Substitute Tags**.



Substitute Tagnames... 1 of 3

Current Name:	Required Type	New Name:
Compressor	Discrete	Compressor
WaterHeater	Discrete	WaterHeater
WaterPump	Discrete	WaterPump

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.

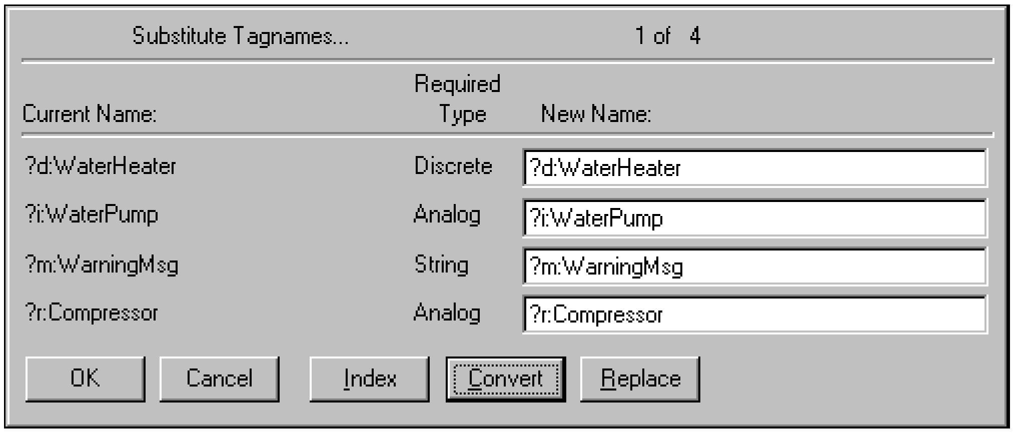
   If right-click the **New Name** box, a menu will appear displaying the commands that you can apply to your text.

   If you double-click a tagname in the **New Name** box, its definition in the Tagname Dictionary will appear.

   If you erase the tagname then double-click in the blank **New Name** box, the Tag Browser will appear.

Converting Placeholder Tagnames

When you index tagnames (to take them out of service) or you import or export a window or QuickScript to or from your current application, all the tagnames associated with that window or QuickScript are transferred with the window, but they are not added to your new application's database. Instead, they are automatically marked as "placeholder" (index) tagnames. You must convert these placeholder tagnames and, if required, define them in your new application Tagname Dictionary. For example:



Current Name:	Required Type	New Name:
?d:WaterHeater	Discrete	?d:WaterHeater
?i:WaterPump	Analog	?i:WaterPump
?m:WarningMsg	String	?m:WarningMsg
?r:Compressor	Analog	?r:Compressor

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

When you import a window, if any of the tagnames (except remote tagnames) are not defined in your local Tagname Dictionary, you will be prompted to define each of them before you can convert them. If this is the case, click **OK**. The **Tagname Dictionary** dialog box will appear and you can define the tagname(s).

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

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

Remote references will not be shown as placeholders but as a remote tagname references, for example, **PLC2:Temperature**.

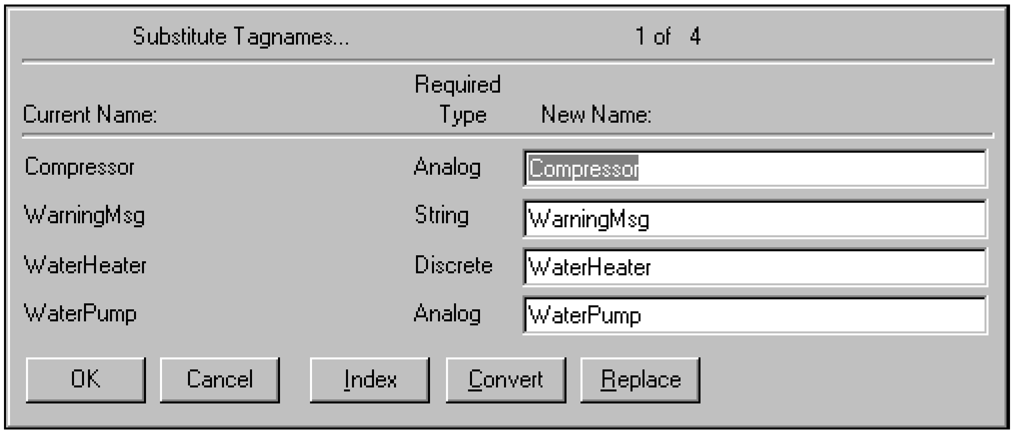
Converting Tagnames to Remote References

There are several methods that you can use in the **Substitute Tagnames** dialog box to convert placeholder (or local) tagnames to remote tagname references. You can directly type in the remote tagname reference, you can convert the placeholder tagnames associated with an imported window or, you can launch the Tag Browser and display the tag source's Tagname Dictionary to select the remote tagname reference.

➢ **To manually convert tagnames to remote tagname references:**

1. Select the object(s) associated with the local tagname that you want to change to a remote tagname reference, and then on the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box appears:

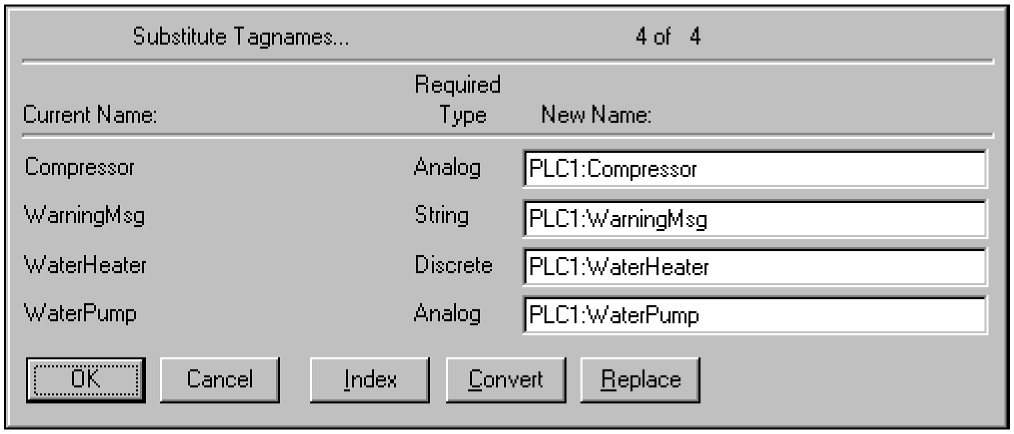
   To quickly access the dialog box, select all the objects, then right-click one of the selected objects, point to **Substitute**, and then click **Substitute Tags**.



Substitute Tagnames... 1 of 4

Current Name:	Required Type	New Name:
Compressor	Analog	Compressor
WarningMsg	String	WarningMsg
WaterHeater	Discrete	WaterHeater
WaterPump	Analog	WaterPump

2. In **New Name** box, select each tagname that you want to change, and then type in the remote tagname reference:



Substitute Tagnames... 4 of 4

Current Name:	Required Type	New Name:
Compressor	Analog	PLC1:Compressor
WarningMsg	String	PLC1:WarningMsg
WaterHeater	Discrete	PLC1:WaterHeater
WaterPump	Analog	PLC1:WaterPump

3. Click **OK**.

If you use this method and you no longer need the original tagnames to be defined in the local Tagname Dictionary, you can update the tagname use counts and then delete the unused tagnames.

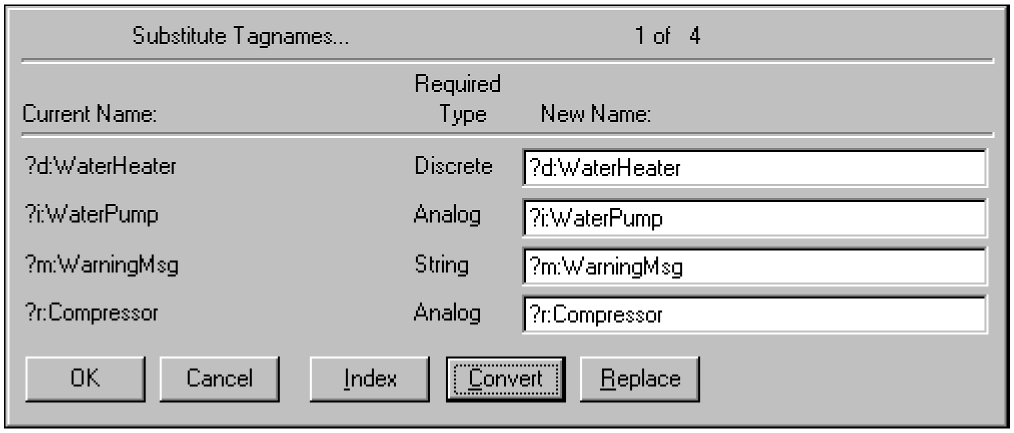
For more information, see "Deleting Tagnames from the Dictionary."

➢ **To convert an imported window's tagnames to remote references:**

1. Open the imported window and select all the object(s), and then on the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box appears:

   For more information on importing windows and scripts see, Chapter 2 - Using WindowMaker.

   To quickly access the dialog box, press the F2 key (to select all the objects in the window), right-click one of the selected objects, then point to **Substitute**, and then click **Substitute Tags**.



Substitute Tagnames... 1 of 4

Current Name:	Required Type	New Name:
?d:WaterHeater	Discrete	?d:WaterHeater
?i:WaterPump	Analog	?i:WaterPump
?m:WarningMsg	String	?m:WarningMsg
?r:Compressor	Analog	?r:Compressor

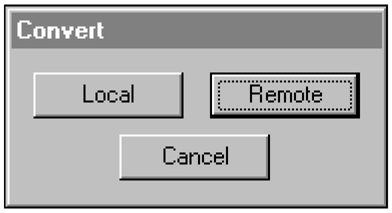
Note The **Index** button turns links into placeholders which disables the animation links to local tagnames. Thus, freeing them for possible deletion.

You can also use the **Substitute Tags** command to convert local tagnames to remote tagname references. To do so, select the objects associated with the local tagnames, and then select the **Substitute Tags** command to display them in the **Substitute Tagnames** dialog box. Click **Index** to change them to placeholder tagnames, and then click **Convert** and follow the steps below.

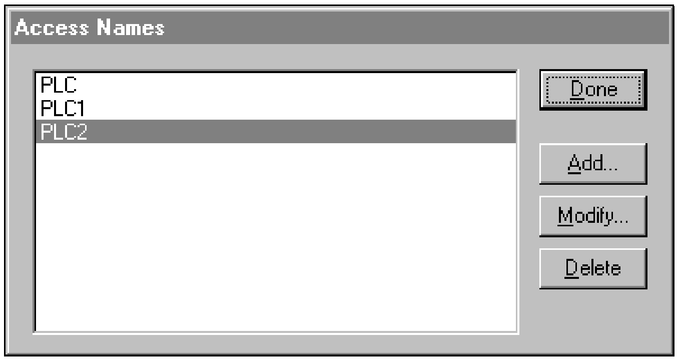
If you use the **Index** button for either of the above and, you no longer need the original tagnames to be defined in the local Tagname Dictionary, you can update the tagname use counts and then delete the unused tagnames.

For more information, see "Deleting Tagnames from the Dictionary."

2. Click **Convert**. The **Convert** dialog box appears:



3. Click **Remote**. The **Access Names** dialog box appears listing all of the Access Names that you have defined in your local application:

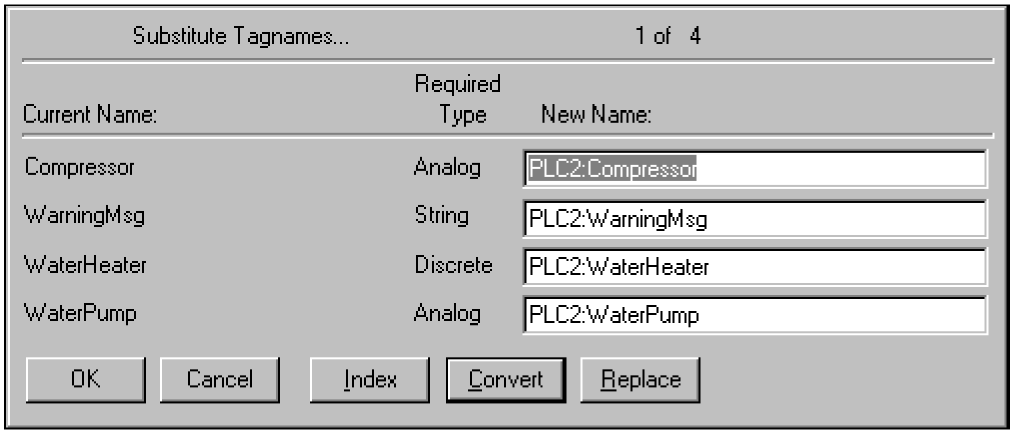


4. Double-click the Access Name in the list or, select it, and then click **OK**.

   To verify that the Access Name correct, you can click **Modify** to view it.

   If you currently do not have an Access Name defined that points to the tag source, click **Add** and define it now. The Access Name **must** include the name of the remote node where the application resides.

5. **All** tagnames displayed in the **Substitute Tags** dialog box will automatically be converted to remote tagname references (prefixed with Access Name that you selected). For example:



Substitute Tagnames... 1 of 4

Current Name:	Required Type	New Name:
Compressor	Analog	PLC2:Compressor
WarningMsg	String	PLC2:WarningMsg
WaterHeater	Discrete	PLC2:WaterHeater
WaterPump	Analog	PLC2:WaterPump

6. Click **OK**.

   This procedure works the same for an imported QuickScript except, you open the QuickScript in the InTouch QuickScript editor, and then click **Convert**.

   By importing a window or QuickScript from another application, and converting all of the tagnames associated with the animation links or QuickScript(s) to remote tagname references, you can instantly be receiving data from hundreds of remote tagnames without defining a single tagname in your local Tagname Dictionary.

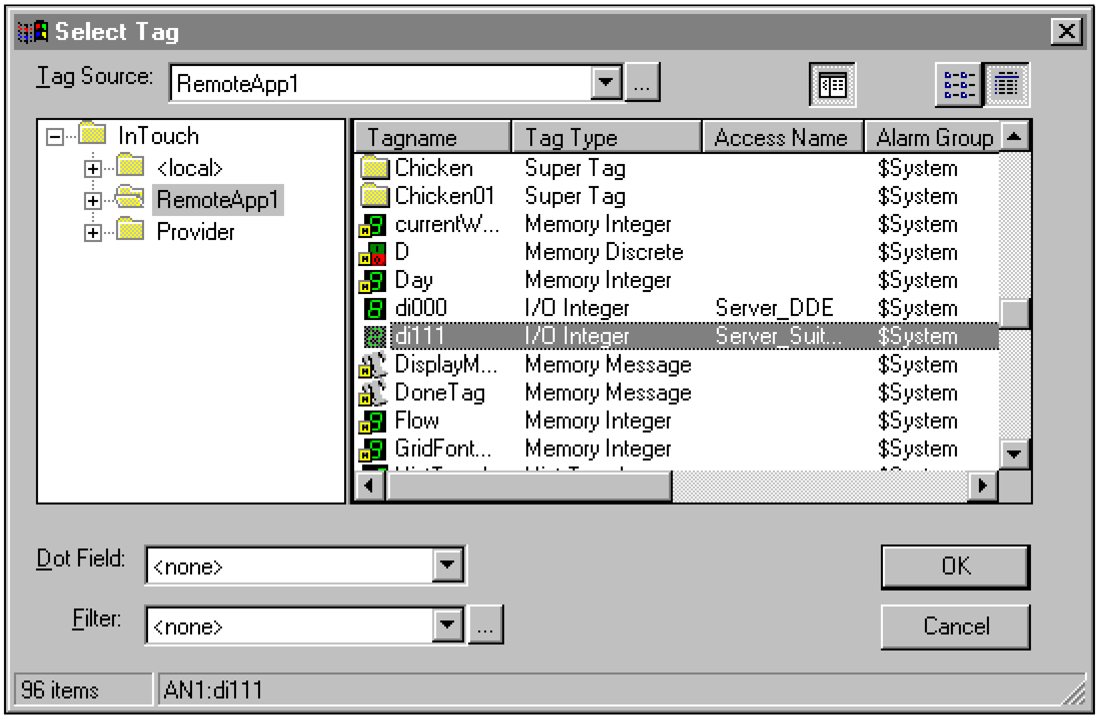
➢ **To select a remote tagname reference in the Tag Browser:**

1. Select the object(s) associated with the local tagname that you want to change to a remote tagname reference, and then on the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box will appear.

   To quickly access the dialog box, select all objects, then right-click one of the selected objects, point to **Substitute**, and then click **Substitute Tags**.

2. Erase the tagname in the **New Name** box that you want to replace with a remote tagname reference, and then double-click the **New Name** box. The **Select Tag** dialog box will appear.

3. Click the tool to display the tree view pane:



Select Tag

Tag Source: RemoteApp1

InTouch
- <local>
- RemoteApp1
- Provider

Tagname	Tag Type	Access Name	Alarm Group
Chicken	Super Tag		$System
Chicken01	Super Tag		$System
currentW...	Memory Integer		$System
D	Memory Discrete		$System
Day	Memory Integer		$System
di000	I/O Integer	Server_DDE	$System
di111	I/O Integer	Server_Suit...	$System
DisplayM...	Memory Message		$System
DoneTag	Memory Message		$System
Flow	Memory Integer		$System
GridFont...	Memory Integer		$System

Dot Field: <none>

Filter: <none>

OK

Cancel

96 items AN1:di111

4. If the tag source is already defined in the Tag Browser, select it in the tree view to displays its tagnames.

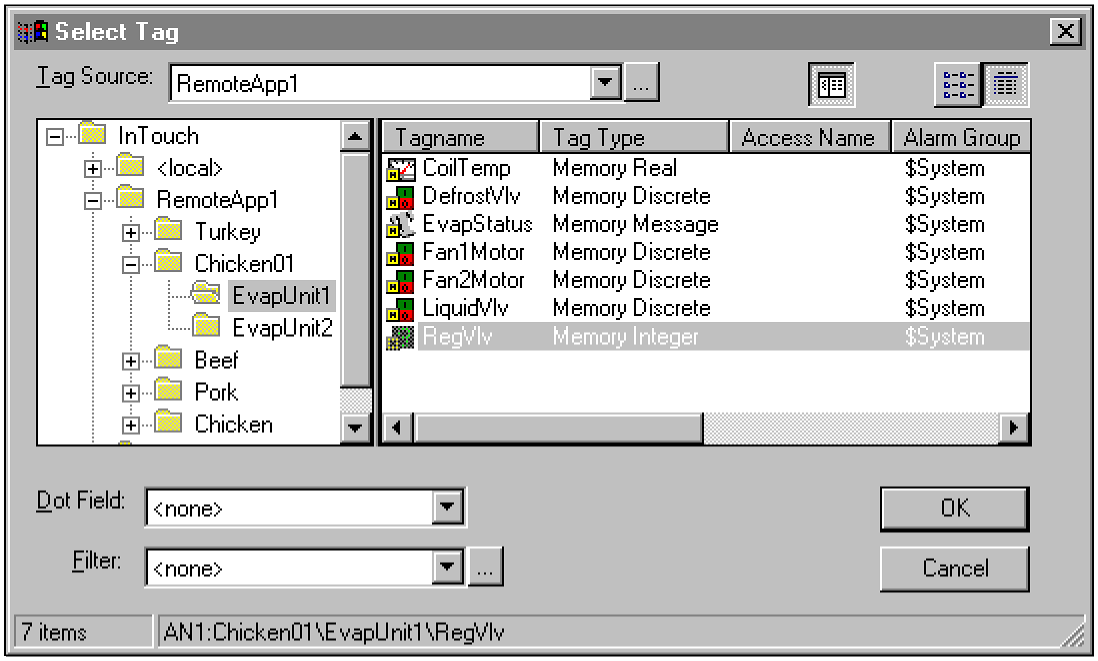
   To expand the tree view, double-click the application name or, click ⊞.

   Note If the tag source is not currently defined, you can define it now.

   For more information on defining tag sources, see "Defining Tag Sources."

5. Double-click the remote tagname that you want to use or, select it, and then click **OK**.

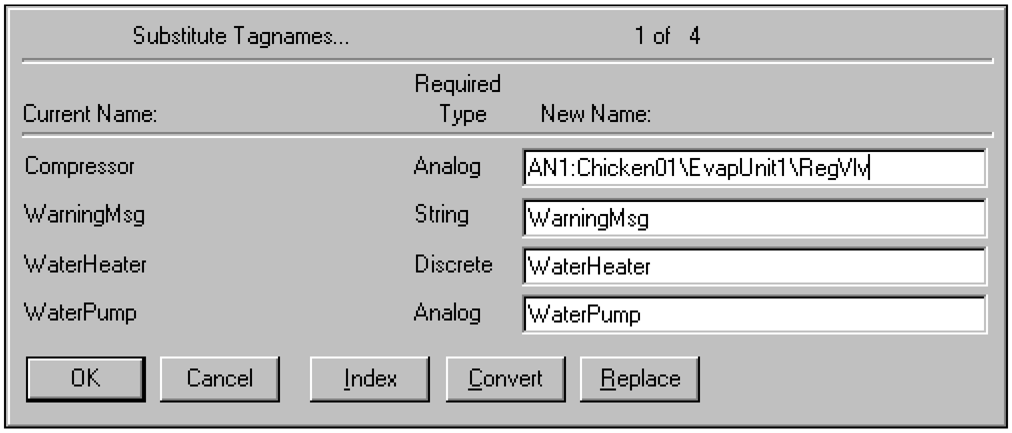
You can also select SuperTag member tagnames. For example:



Tagname	Tag Type	Access Name	Alarm Group
CoilTemp	Memory Real		$System
DefrostVlv	Memory Discrete		$System
EvapStatus	Memory Message		$System
Fan1Motor	Memory Discrete		$System
Fan2Motor	Memory Discrete		$System
LiquidVlv	Memory Discrete		$System
RegVlv	Memory Integer		$System

7 items AN1:Chicken01\EvapUnit1\RegVlv

The **Substitute Tags** dialog box appears displaying the remote tagname reference:



Current Name:	Required Type	New Name:
Compressor	Analog	AN1:Chicken01\EvapUnit1\RegVlv
WarningMsg	String	WarningMsg
WaterHeater	Discrete	WaterHeater
WaterPump	Analog	WaterPump

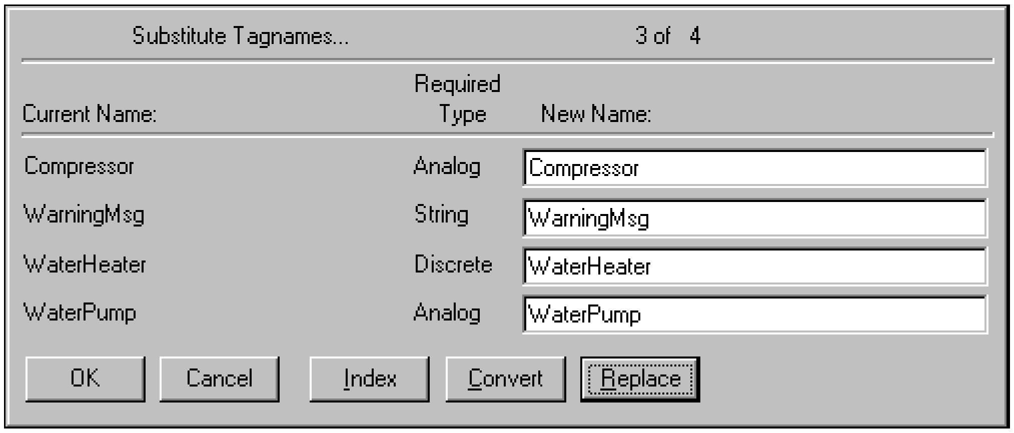
6. Click **OK** to close the dialog box and associate the remote tagname with the selected object(s).

   Repeat this procedure for each tagname that you want to change in the **Substitute Tagnames** dialog box.

➤ **To replacing a tagname:**

1. Select all objects whose tagname you want to change.
2. On the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box appears:

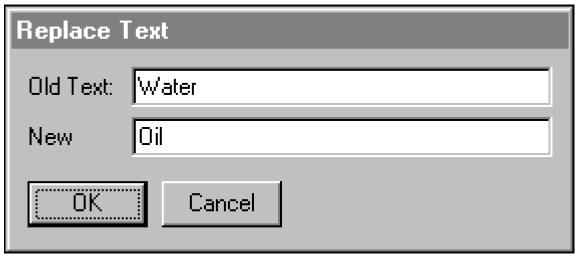
   To quickly access the dialog box, right-click the object, point to **Substitute,** nd then click **Substitute Tags**.



Substitute Tagnames... 3 of 4

Current Name:	Required Type	New Name:
Compressor	Analog	Compressor
WarningMsg	String	WarningMsg
WaterHeater	Discrete	WaterHeater
WaterPump	Analog	WaterPump

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

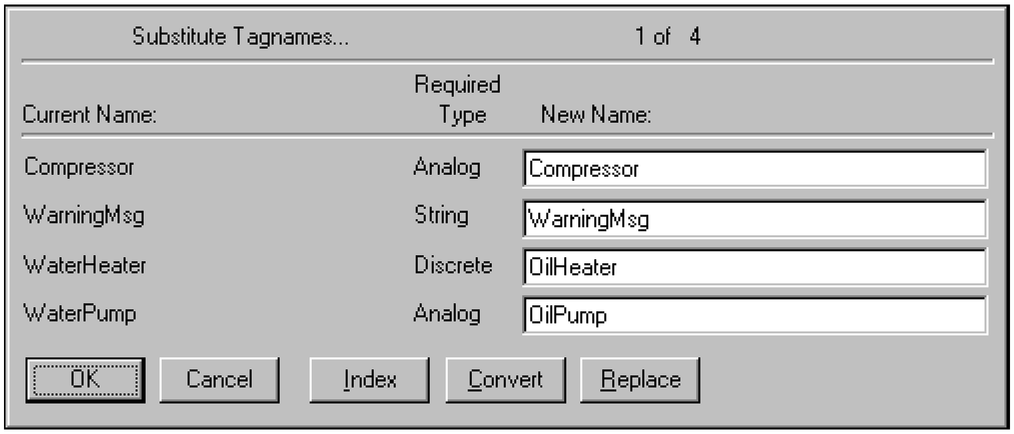


Replace Text

Old Text: Water

New: Oil

4. In the **Old Text** box, type the portion of the tagname that you want to replace.
5. In the **New** box, type the replacement text.
6. Click **OK**. The **Substitute Tagnames** dialog box reappears showing the change made to the tagname:



Substitute Tagnames... 1 of 4

Current Name:	Required Type	New Name:
Compressor	Analog	Compressor
WarningMsg	String	WarningMsg
WaterHeater	Discrete	OilHeater
WaterPump	Analog	OilPump

7. Click **OK**. All tagnames for the selected object(s) containing the text that you replaced will automatically be changed.

Scaling I/O Tagnames

All I/O type tagnames receive their values from other Windows application programs such as Excel and I/O Servers. This value is referred to as the "raw" value. When you define a tagname in the Tagname Dictionary, you must enter values for the "Min Raw" and "Max Raw." These values are used by the database as clamps on the actual raw value received from the I/O device. For example, if you set the "Min Raw" value to 50 and the actual value received from a I/O Server is 0, database will force the Raw value to 50.

InTouch does not display raw values. Instead, it displays engineering units (EU). When you define an I/O type tagname in the Tagname Dictionary, you must specify values for the "Min EU" and "Max EU." These values are used to scale the raw value to the displayed value. If you do not want to do scaling or your I/O device does the scaling for you, set the Min/Max EU values equal to the Min/Max Raw values.

For example, let's assume that a flow transmitter wired to a PLC register generates a value of zero at no flow and a value of 9999 at 100% flow. The following values would be entered:

```
Min EU = 0       Max EU = 100

Min Raw=0 Max Raw = 9999
```

A raw value of 5000 would be displayed as 50.

Let's also assume that a flow transmitter wired to a PLC register generates a value of 6400 at no flow and a value of 32000 at 300 GPM.

```
Min EU = 0       Max EU = 300

Min Raw = 6400   Max Raw = 32000
```

In this case, a raw value of 12800 would be displayed as 150. A raw value of 6400 would be displayed as 0 and a Raw value of 0 would be displayed as 0 (all values outside the boundaries set by the Min Raw and Max Raw values are clamped).

The above scaling works in reverse when the I/O tagname data is written from the InTouch Tagname Dictionary to other Windows applications.

For example, an operator could enter a setpoint value of 0-300 GPM in a data entry window and have a value of 6400-32000 transmitted to the PLC register.

This is always valid when the InTouch program is acting as the client (either requesting data from or sending data to another Windows program). However, when InTouch acts as data source (another Windows program is requesting data from InTouch) it will return the EU value to the requesting program.

For example, if a cell in an Excel spreadsheet contained the remote reference formula:

```
=view|tagname!speed
```

The value displayed in the cell would be the current EU value for the tagname speed.

Instrument Failure Monitoring

Beginning with Version 7.0, InTouch supports three tagname **.fields** (**.RawValue**, **.MinRaw** and **.MaxRaw**) that you can use in InTouch QuickScripts to monitor instrument values to determine out-of-range, out-of-calibration or, failure.

Internal System $Tagnames

InTouch provides you with numerous predefined internal system tagnames that you can use to perform a variety of actions. For example, if you want to display the current time, you could link the system tagname **$TimeString** to a value display string. All internal tagnames are preceded with a dollar sign (**$**). The internal system tagnames are accessed through the Tag Browser.

For more information, see "The Tag Browser."

The following section briefly describes the internal system tagnames:

System Tagname	Description
$AccessLevel	Read only integer security tagname used in expressions or scripts to control the operator's ability to perform specific functions.
$AlarmLogging	Read/write only discrete tagname that is set to 1 to restart alarm logging and printing during runtime. Equal to the **Restart Alarm Log** command on the WindowViewer **Special** menu.
$AlarmPrinterError	Read only discrete tagname that is equal to 1 if there is a printer error.
$AlarmPrinterNoPaper	Read only discrete tagname that is equal to 1 if the printer is out of paper.
$AlarmPrinterOffline	Read only discrete tagname that is equal to 1 if the printer is offline.
$AlarmPrinterOverflow	Read only discrete tagname that is equal to 1 if there is printer overflow.
$ApplicationChanged	Read only real tagname that reflects whether or not the remote application has changed in distributed systems. This number is incremented each time the **Notify Clients** command is selected on the Server node's WindowViewer **Special** menu.
$ApplicationVersion	Read only real tagname that reflects the current version number of the application. This number changes each time a tagname or QuickScript is changed, added or deleted.
$ChangePassword	Write only discrete security tagname that allows the operator to set the value of the **$ChangePassword** tagname to 1, causing the generic Change Password dialog box to be displayed for the operator.
$ConfigureUsers	Write only discrete security tagname that can be used on a discrete button to allow the operator to set the value of the **$ConfigureUsers** tagname to 1**,** causing the generic **Configure Users** dialog box to be displayed for editing the security user name list.

System Tagname	Description
$Date	Read only integer tagname that displays the whole number of days which have passed since 1/1/70.
$DateString	Read only memory message tagname that displays the date in the same format set in the WIN.INI file, for example, 4/18/1992. (This date format is set through the Windows Control Panel.)
$DateTime	Read only real tagname that displays the fractional number of days which have passed since 1/1/70.
$Day	Read only integer tagname that displays the current day (value may be 1-31).
$HistoricalLogging	Read/write discrete tagname that monitors/controls starting and stopping of historical logging. This is a global command for the whole application.
$Hour	Read only integer tagname that displays the current hour of the day (value may be 0-23).
$InactivityTimeout	Read only discrete security tagname that equals 1 when the time configured for automatic log off of the operator has elapsed.
$InactivityWarning	Read only discrete security tagname that equals 1 when the time configured for warning the operator that log off is about to occur has elapsed.
$LogicRunning	Read/write discrete tagname used to monitor and/or control the running of scripts.
	Note Asynchronous User Defined Function scripts that are currently executing cannot be stopped. However, you can prevent new scripts from executing.
$Minute	Read only integer tagname that displays the current minute (value may be 0-59).
$Month	Read only integer tagname that displays the current month (value may be 1-12).
$Msec	Read only integer tagname that displays milliseconds (value may be 0-999).
$NewAlarm	Read/write discrete tagname that is equal to 1 each time a new alarm occurs.
$ObjHor	Read only integer tagname used to display the horizontal pixel location of the center of a selected object.
$ObjVer	Read only integer tagname used to display the vertical pixel location of the center of a selected object.

System Tagname	Description
$Operator	Read only security message tagname that can be used in an expression or QuickScript to control the operator's ability to perform specific functions.
$OperatorEntered	Read/write security message tagname that sets the "User Name" for the operator.
$PasswordEntered	Write only security message tagname that sets the "Password" for the operator.
$Second	Read only integer tagname that displays the current seconds (value may be 0-59).
$StartDdeConversations	Read/write discrete tagname used to start uninitiated conversations during runtime when the **Special** menu has been disabled.
$System	Read only Alarm Group type tagname for the alarm root group. If a tagname is not assigned to a specific Alarm Group name, it is automatically assigned to this root group by default. All defined Alarm Groups are descendants of **$System**.
$Time	Read only integer tagname that displays the time in milliseconds since midnight.
$TimeString	Read only memory message tagname that displays the time in the same format set in the WIN.INI file, e.g., 12:01:59 PM. (This time format is set through the Windows Control Panel.)
$Year	Read only integer tagname that displays the year in four digits, e.g., 1990.

📖 For more information on system tagnames, see your *InTouch Reference Guide.*

Tagname .Fields

Most discussions concerning InTouch refer to the concept of objects. The concept of objects is very widespread and complex. For our discussion we will limit our definition of an object to a collection of information about a graphical object on the screen or information about a tagname in the Tagname Dictionary.

For example, if a rectangle is drawn on the screen it has certain "attributes" such as the line width, line and fill color, pixel location on the screen, links associated with, etc. Tagnames work in much the same way. For example, if an analog, alarmed tagname is created called "Analog_Tagname," it will have "attributes" associated with it such as the name of the tagname, the tagname's **HiHi** alarm setpoint, and so on. Some of these "attributes" are accessible within InTouch through 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**.

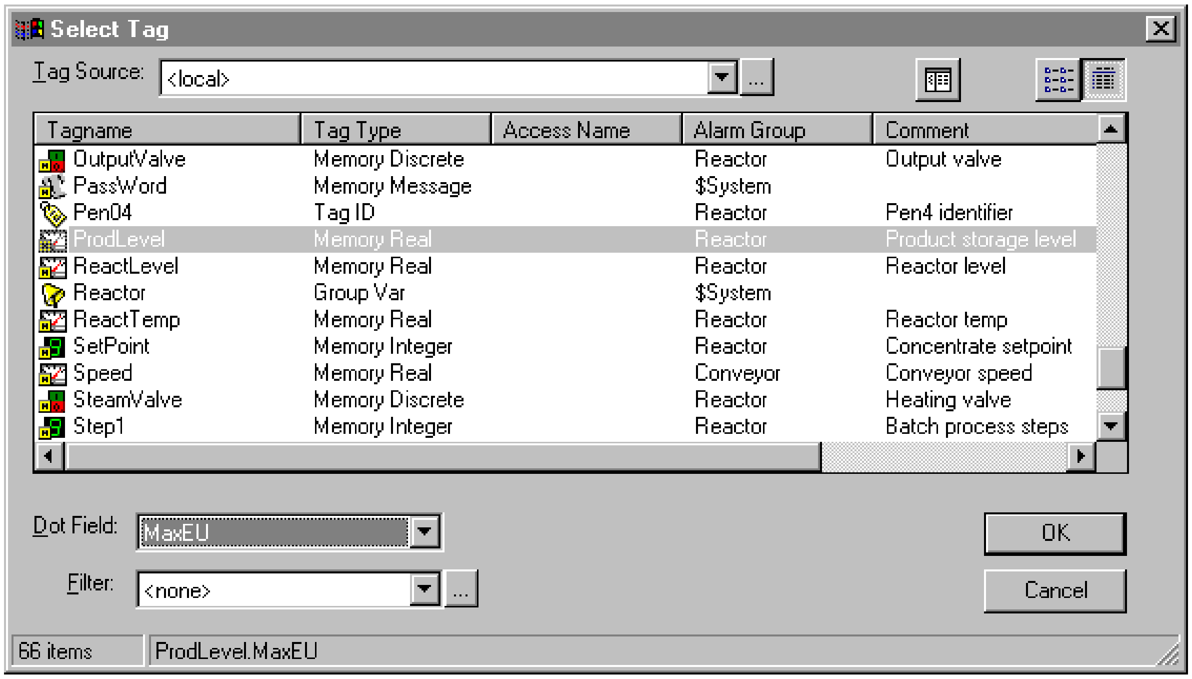
For example, to allow runtime changes to the **HiHi** alarm limit on tagname "Analog_Tagname," you could create an **Analog - User Input** touch link and apply it to a button that is defined with the expression **Analog_Tagname.HiHiLimit**. In runtime, the operator would simply click the button and type in a new value for the **HiHi** alarm limit being used for "Analog_Tagname."

You can use **.fields** to allow input and output of data associated with a tagname and you can use the historical **.fields** to allow the user to dynamically modify the historical trend being displayed. For example you can allow the user to control the scrolling, lock or reposition the scooters on the trend, or reassign the pens to new tagnames.

📖 For more information, **.fields**, see your *InTouch Reference Guide.*

➤ **To access tagname .fields:**

1. Type a tagname plus a period (**tagname.**) in any InTouch QuickScript, or animation link tagname or expression input box, and then double-click to the right of the period (**.**). The Tag Browser appears displaying the tagnames defined for the current tag source:



Select Tag

Tag Source: <local>

Tagname	Tag Type	Access Name	Alarm Group	Comment
OutputValve	Memory Discrete		Reactor	Output valve
PassWord	Memory Message		$System	
Pen04	Tag ID		Reactor	Pen4 identifier
ProdLevel	Memory Real		Reactor	Product storage level
ReactLevel	Memory Real		Reactor	Reactor level
Reactor	Group Var		$System	
ReactTemp	Memory Real		Reactor	Reactor temp
SetPoint	Memory Integer		Reactor	Concentrate setpoint
Speed	Memory Real		Conveyor	Conveyor speed
SteamValve	Memory Discrete		Reactor	Heating valve
Step1	Memory Integer		Reactor	Batch process steps

Dot Field: MaxEU

Filter: <none>

66 items | ProdLevel.MaxEU

➢ **To select a .field:**

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

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

   Note Dot Field is not available when you access the Tag Browser from the Tagname Dictionary or, during runtime, when selecting a tagname for a historical trend pen from the **Historical Trend Setup** dialog box. (The historical trend must be configured with the **Allow runtime changes** option selected.)

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

   Note Not every tagname type has the same **.fields**. For example, a **Discrete** type tagname has **.OnMessage**, whereas an analog does not. If you select a **Discrete** type tagname and you assign **.OnMessage** to it, and then you select another **Discrete** type tagname, the displayed **.field** list will not change. But, if you select another type of tagname in the control view list, for example an analog, the displayed **.field** will revert to **<none>**.

   For more information on the Tag Browser, see "The Tag Browser."

The following section briefly describes the tagname **.fields**:

.Field	Description
.Ack	Read/write discrete .field that monitors/controls the alarm acknowledgment status of tagnames, Alarm Groups and/or Group Variables. **Note .Ack** has an inverse .field called **.Unack**. When an unacknowledged alarm occurs, **.Unack** will be set to 1. **.Unack** can then be used in animation links or condition scripts to trigger annunciators for any unacknowledged alarms.
.Alarm	Read only discrete .field that is equal to 1 when an alarm condition exists for the specified tagname, Alarm Group Name or Group Variable.
.AlarmDevDeadband	Read/write analog (only valid for integer or real tags) .field that monitors/controls the deviation percentage deadband for both minor and major deviation alarms.
.AlarmEnabled	Read/write discrete .field that disables/enables events and alarms for a tagname, Alarm Group or Group Variable.
.AlarmValDeadband	Read/write analog (only valid for integer or real tags) .field that monitors and/or controls the value of an alarm's deadband. This field is valid for Alarm Groups and Group Variables as well as ordinary tags.

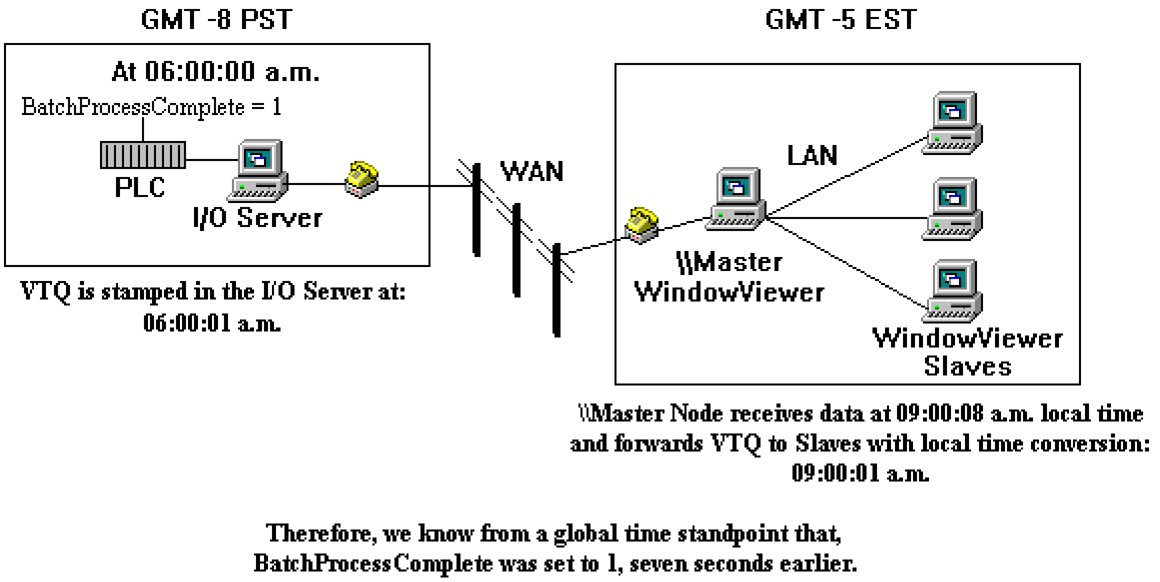
.Field	Description
.ChartLength	Read/write integer tagname .field used to control the length of time displayed in a Historical Trend graph. **.ChartLength** displays the length of the chart in seconds.
.ChartStart	Read/write integer tagname .field used to control the starting time and/or to scroll the corresponding historical trend chart. **.ChartStart** displays the number of elapsed seconds since 12:00 a.m., 1/1/70.
.Comment	Read only message tagname .field that is used to display the comment field entered for a tagname in the Tagname Dictionary.
.DevTarget	Read/write analog (only valid for integer or real tags) .field that monitors and/or controls the target for minor and major deviation alarms.
.DisplayMode	Read/write analog tagname .field used to determine the method to be used in displaying values on the trend.
.EngUnits	Read/write analog tagname .field used to access the engineering units of an analog tagname as specified in the tagname dictionary.
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	Read/write analog tagname .fields that monitors/controls 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 determines 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.
.MaxEU, .MinEU	Read only real tagname .fields that display the maximum and minimum values for the tagname.

.Field	Description
.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.
.MaxRaw, .MinRaw	The high/low clamp setting for the actual raw value received from an I/O Server by WindowViewer as a client. The value for .**MaxRaw/.MinRaw** field comes from the Max/Min Raw value field in tagname database for the specified I/O tagname. Any raw value which exceeds or falls below this setting is clamped to this value.
.MinorDevPct	Read/write integer tagname .field used to monitors and/or controls 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.
.OffMsg, **.OnMsg**	Read/write message tagname .fields used to display the on message and off message specified in the Tagname Dictionary for a discrete tagname.

.Field	Description
.Pen1 - .Pen8	Read/write TagID type tagname .fields used to control the tagname being historically trended by each pen.
	Note An easier method of dynamically assigning Tagnames to Pens is to use the Historical functions **HTSetPenName** and **HTGetPenName**.
	For more information, see Chapter 8 - Real-time and Historical Trending.
.Quality	Read/write message tagname allows the user to access the quality of an I/O tagname as provided by an I/O server.
.QualityLimit	Read only integer tagname .field used to display the quality limit of an I/O value provided by data source when the I/O connection is valid.
.QualityLimitString	Read only message tagname .field used to display the quality limit string of an I/O value provided by an I/O server when the I/O connection is valid.
.QualityStatus	Read only integer tagname .field used to display the quality status of an I/O value provided by an I/O server when the I/O connection is valid.
.QualityStatusString	Read only message tagname .field used to display the quality status string of an I/O value provided by an I/O server when the I/O connection is valid.
.QualitySubstatus	Read only integer tagname .field used to display the quality substatus of an I/O value provided by an I/O server when the I/O connection is valid.
.QualitySubstatusString	Read only message tagname .field used to display the quality substatus string of an I/O value provided by an I/O server when the I/O connection is valid.
	Note If the I/O connection becomes invalid, the quality .fields are automatically reset to the initial value of zero. The **.ReferenceComplete** flag is also set to zero at this time.
.RawValue	Read only tagname .field that is used to display the actual discrete or analog I/O value before InTouch applies scaling.
.Reference	Read/write .field used with I/O type tagnames to dynamically change the address of the tagname's source.

.Field	Description
.ReferenceComplete	Discrete field that provides a confirmation that the item requested is the same reflected in the **.Value** field. For more information on the .Reference .field, see "Dynamic Reference Addressing."
.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.
.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 that represents the position of the left scooter (range 0.0 to 1.0).
.ScooterPosRight	Read/write real field that represents the position of the right scooter (range 0.0 to 1.0).
.TagID	Read only TagID tagname .field used in conjunction with the Historical Trend **.Pen1-.Pen8** TagID tagnames to monitor and/or control the tagname being trended by a pen (see previous "Pen" .field description).

The following illustrates how the **.Time** fields below receive their data:



VTQ is stamped in the I/O Server at: 06:00:01 a.m.

\\Master Node receives data at 09:00:08 a.m. local time and forwards VTQ to Slaves with local time conversion: 09:00:01 a.m.

Therefore, we know from a global time standpoint that, BatchProcessComplete was set to 1, seven seconds earlier.

.Field	Description
.TimeDate	Read only integer tagname .field used to display the whole number of days which have passed since an I/O value was provided by an I/O server when the I/O connection was valid.
.TimeDateString	Read only message tagname .field used to displays the date in the same format set in the WIN.INI file. For example, 10/31/1997 that an I/O value was provided by an I/O server when the I/O connection was valid.
.TimeDateTime	Read only real tagname .field used to display the fractional number of days which have passed since an I/O value was provided by an I/O server when the I/O connection was valid.
.TimeDay	Read only integer tagname .field used to display the day an I/O value was provided by an I/O server when the I/O connection is valid. (value may be 1-31).
.TimeHour	Read only integer tagname .field used to display the hour of the day that an I/O value was provided by an I/O server when the I/O connection was valid. (value may be 0-23).

.Field	Description
.TimeMinute	Read only integer tagname .field used to display the minute that an I/O value was provided by an I/O server when the I/O connection was valid. (value may be 0-59).
.TimeMonth	Read only integer tagname .field used to display the month that an I/O value was provided by an I/O server when the I/O connection was valid. (value may be 1-12).
.TimeMsec	Read only integer tagname .field used to display the time in milliseconds that an I/O value was provided by an I/O server when the I/O connection was valid.
.TimeSecond	Read only integer tagname .field used to display the time in seconds that an I/O value was provided by an I/O server when the I/O connection was valid. (value may be 0-59).
.TimeTime	Read only integer tagname .field used to display the time in milliseconds (since midnight) that an I/O value was provided by an I/O server when the I/O connection was valid.
.TimeTimeString	Read only message tagname .field used to display the time and day of an I/O value provided by an I/O server when the I/O connection is valid.
.TimeYear	Read only integer tagname .field used to display the time in the same format set in the WIN.INI file (for example, 12:09:45) that an I/O value was provided by an I/O server when the I/O connection was valid. (This time format is set through the Windows Control Panel.)
.Unack	Read/write discrete .field used to control the alarm unacknowledgement status of local alarm(s).
.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	Write only discrete tagname .field that can be set to 1 to cause a Historical Trend chart to update using all current values. Historical Trends do not automatically update themselves. A change must be made to either the chart start, chart length, etc. in order for the chart to update and display the current values for the specified tagnames. Using this .field in a button QuickScript will allow the operator to update the chart whenever necessary during runtime.
.Value	Read or Read/write analog tagname that displays the value of the specified tagname.

📖 For more information on **.field**s, see your *InTouch Reference Guide.*

Addressing Bit Fields for Analog Tagnames

Single bits within an integer tagname can be addressed by using the bit .field. These are all considered to be discrete (0/1) and if written, the analog tagname is promptly updated. You can use the bit .fields anywhere that a discrete tagname can be used. For example, in I/O, scripts, expressions, and so on.

.00	Least significant bit
.01	next more significant bit
.02	etc.
.	
.	
.	
.31	Most significant bit in the 32-bit integer

The following is an example of how to use the bit fields in an expression:

```
Temperature.08 == 1;
```

The following is an example of how to use the bit fields in a QuickScript:

```
IF Temperature.29 THEN
   Temperature.29 =0;
ENDIF;
```

Tagname Dictionary Utilities

There are two Tagname Dictionary utility programs; DBDump and DBLoad. DBDump is used to export an InTouch application Tagname Dictionary as a text file that can be viewed or edited in another program such as Microsoft Excel. DBLoad allows a properly formatted Tagname Dictionary file (created in another program such as Excel or a DBDump file from another InTouch application) to be loaded into an existing InTouch application. These two programs allow a database (Tagname Dictionary) to be copied, modified or developed in separate portions and then merged into one application.

The DBLoad utility can also be used as an alternative to the InTouch TemplateMaker to create SuperTag instances.

For more information, see "Creating SuperTag Instances."

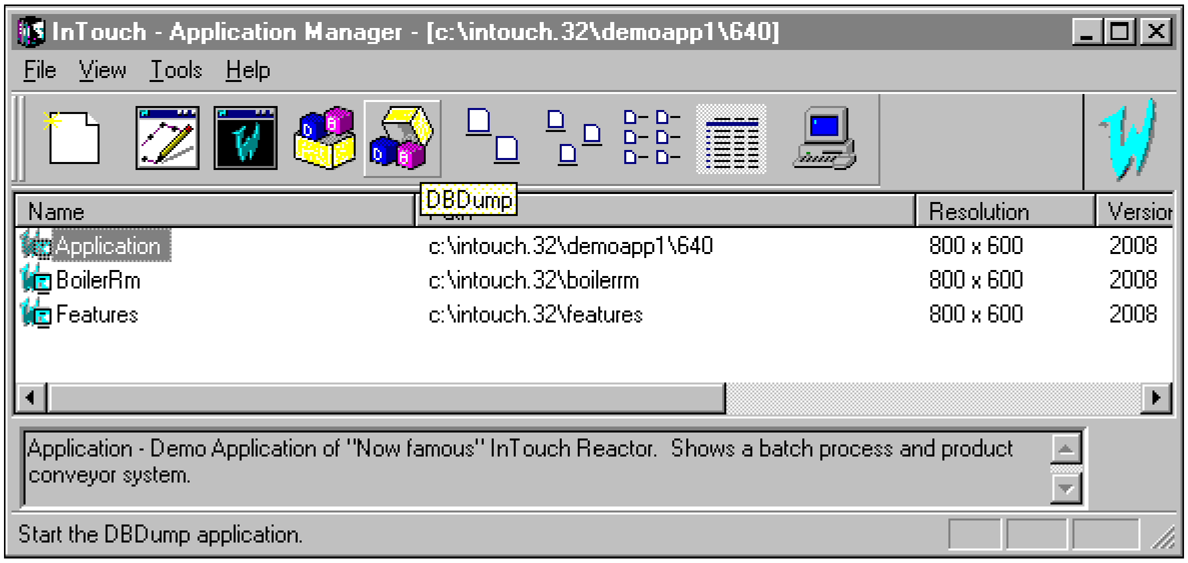
Note Both the DBDump and DBLoad utilities are launched from the InTouch Application Manager (INTOUCH.EXE). Also, you must convert an application created in an earlier version of InTouch before its Tagname Dictionary can be extracted.

For more information on creating database files, see "Creating a Database Input File."

DBDump Utility Program

➢ **To extract an existing InTouch application's Tagname Dictionary:**

1. Close WindowMaker and WindowViewer if they are running.
2. Start InTouch. The **InTouch Application Manager** dialog box appears:

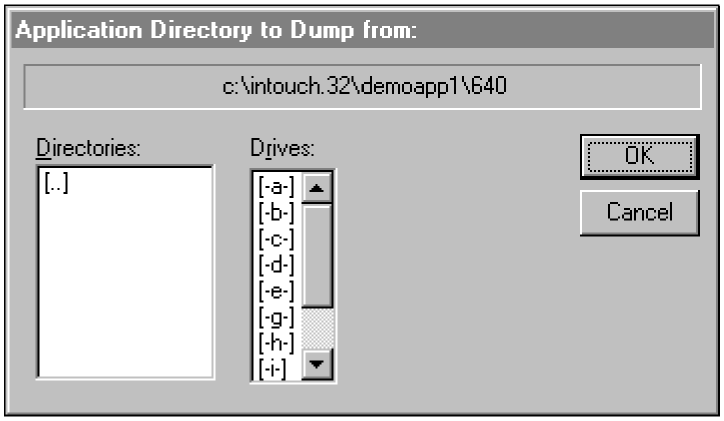

InTouch - Application Manager - [c:\intouch.32\demoapp1\640]
File View Tools Help
DBDump

Name	Path	Resolution	Version
Application	c:\intouch.32\demoapp1\640	800 x 600	2008
BoilerRm	c:\intouch.32\boilerrm	800 x 600	2008
Features	c:\intouch.32\features	800 x 600	2008

Application - Demo Application of "Now famous" InTouch Reactor. Shows a batch process and product conveyor system.

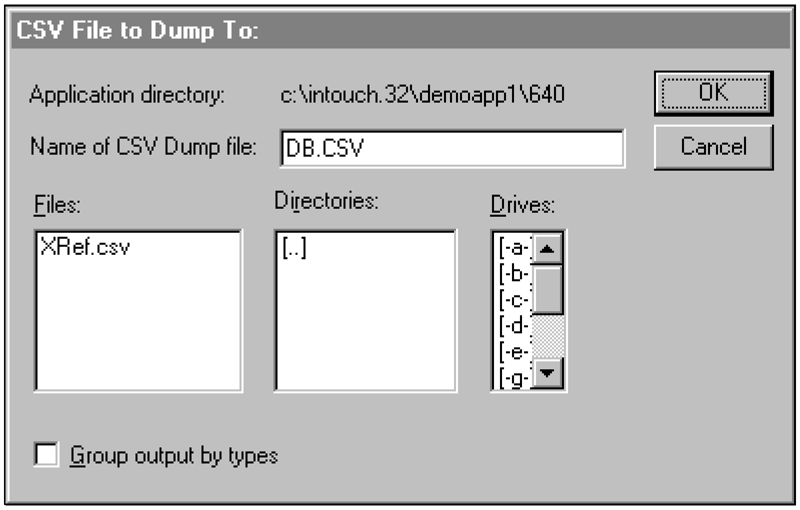
Start the DBDump application.

3. On the **File** menu, click **DBDump** or, click the DBDump tool. The **Application Directory to Dump from** dialog box appears:



4. Select the InTouch application directory into which you want to load the database. (It must replace the default directory shown at the top of the dialog box to be properly selected (in this example **c:\intouch.32**). By default, the last directory you accessed will be displayed when the dialog box initially appears.)

5. Click **OK**. The **CSV File to Dump To:** dialog box appears:



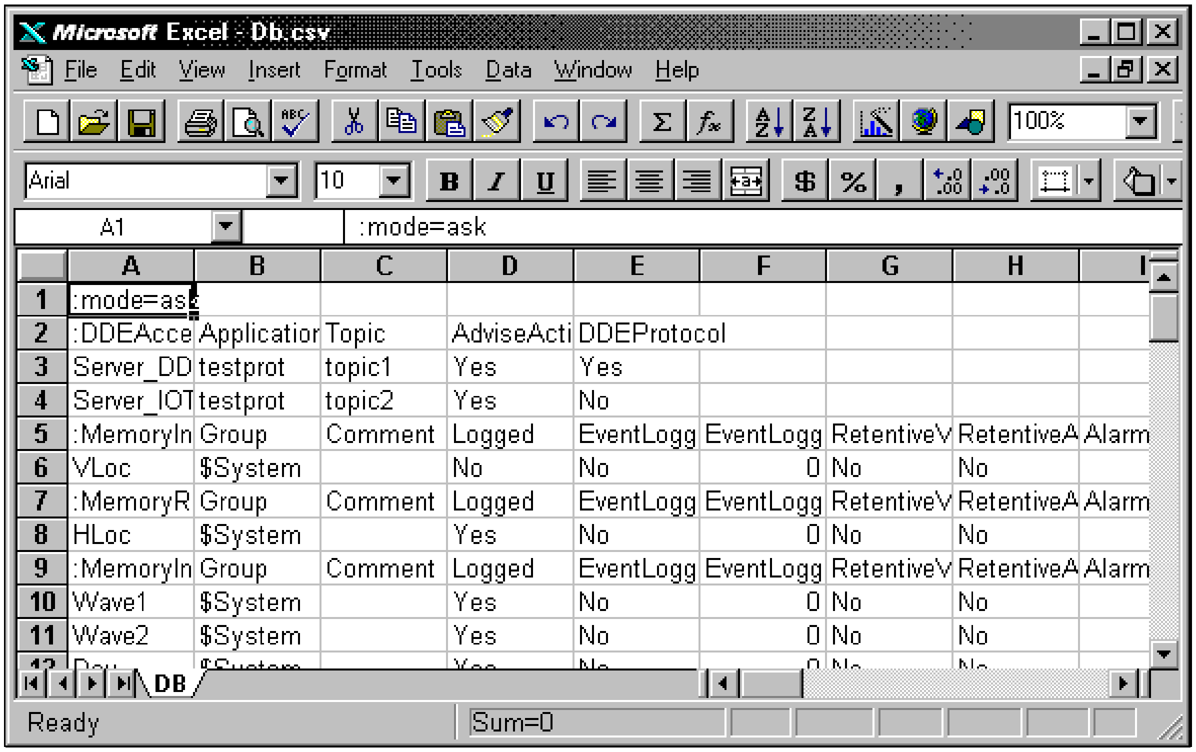
6. In the **Name of CSV Dump file** box, type a name for the file that ends with the .CSV (Comma Separated Variable) extension. (If the name already exists, a message box will appear.)

7. Select **Group output by types** to group the extracted tagnames by tagname type instead of alphabetically by tagname (default.)

Note In order to extract the database items by groups, the system must read the entire file for each tagname type. Therefore, it takes longer to extract the data. However, when the output file is opened in a .CSV supported application such as Microsoft Excel, the grouping makes it much easier to read or edit. A placeholder for each tagname type is included in the dumped file whether tagnames exist for the type or not.

8. Click **OK**. The database information from the selected application directory will be downloaded to the specified filename.

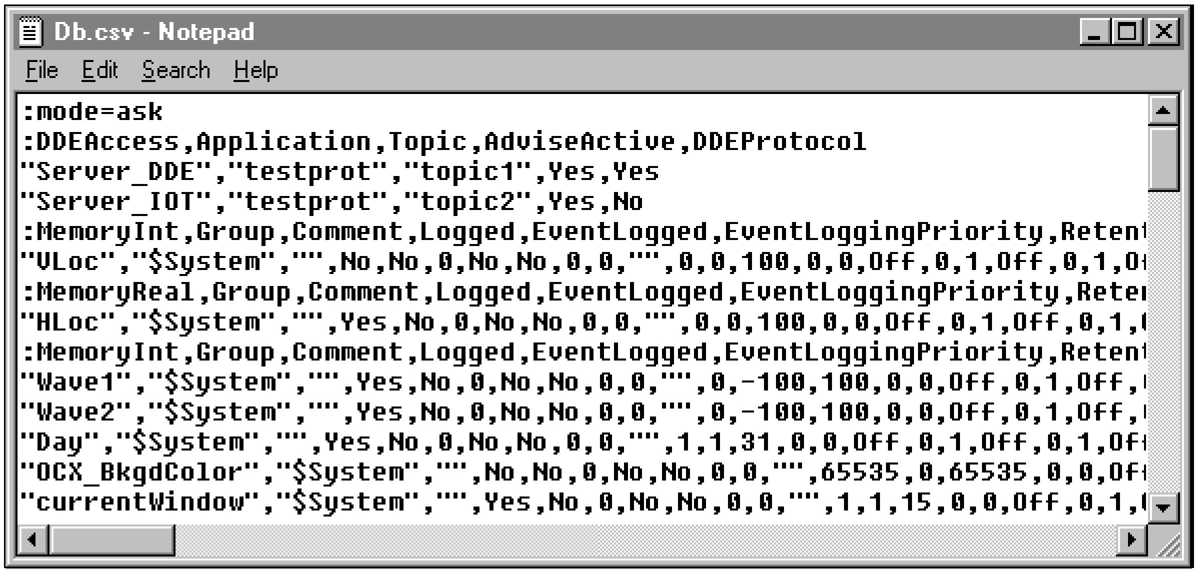
If you open the .CSV file in Microsoft Excel, it sees the comma as a delimiter and automatically separates the data records into columns. For example:



:mode=ask

	A	B	C	D	E	F	G	H	I
1	:mode=as								
2	:DDEAcce	Applicatior	Topic	AdviseActi	DDEProtocol				
3	Server_DD	testprot	topic1	Yes	Yes				
4	Server_IOT	testprot	topic2	Yes	No				
5	:MemoryIn	Group	Comment	Logged	EventLogg	EventLogg	RetentiveV	RetentiveA	Alarm
6	VLoc	$System		No	No	0	No	No	
7	:MemoryR	Group	Comment	Logged	EventLogg	EventLogg	RetentiveV	RetentiveA	Alarm
8	HLoc	$System		Yes	No	0	No	No	
9	:MemoryIn	Group	Comment	Logged	EventLogg	EventLogg	RetentiveV	RetentiveA	Alarm
10	Wave1	$System		Yes	No	0	No	No	
11	Wave2	$System		Yes	No	0	No	No	

If you open the .CSV file in Notepad, each data record is separate by a comma. For example:

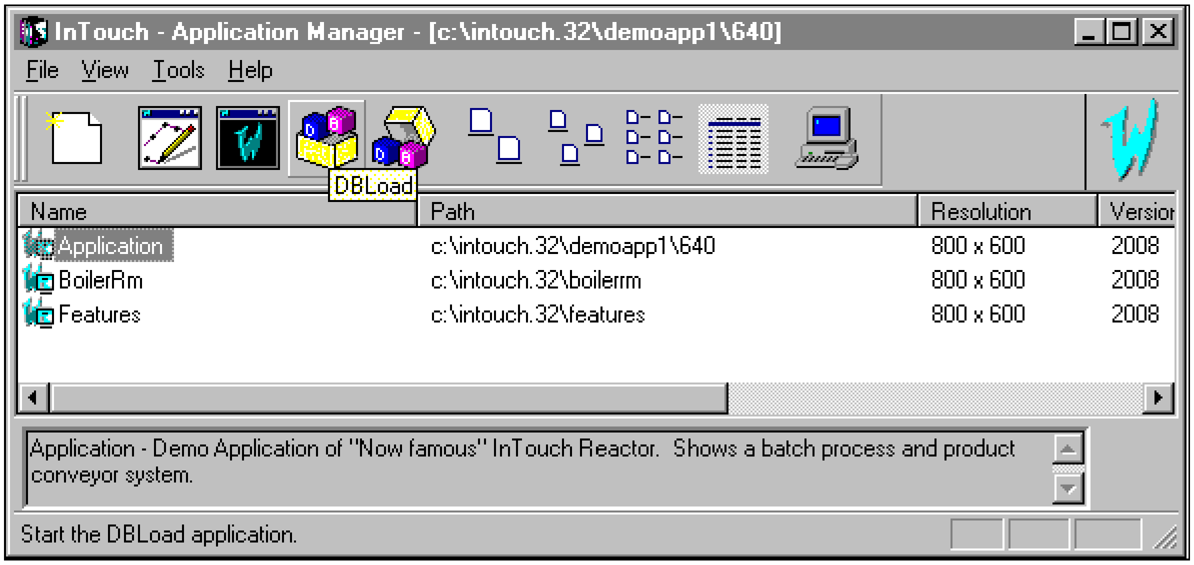


```
:mode=ask
:DDEAccess,Application,Topic,AdviseActive,DDEProtocol
"Server_DDE","testprot","topic1",Yes,Yes
"Server_IOT","testprot","topic2",Yes,No
:MemoryInt,Group,Comment,Logged,EventLogged,EventLoggingPriority,Reten
"VLoc","$System","",No,No,0,No,No,0,0,"",0,0,100,0,0,Off,0,1,Off,0,1,0
:MemoryReal,Group,Comment,Logged,EventLogged,EventLoggingPriority,Rete
"HLoc","$System","",Yes,No,0,No,No,0,0,"",0,0,100,0,0,Off,0,1,Off,0,1,
:MemoryInt,Group,Comment,Logged,EventLogged,EventLoggingPriority,Reten
"Wave1","$System","",Yes,No,0,No,No,0,0,"",0,-100,100,0,0,Off,0,1,Off,
"Wave2","$System","",Yes,No,0,No,No,0,0,"",0,-100,100,0,0,Off,0,1,Off,
"Day","$System","",Yes,No,0,No,No,0,0,"",1,1,31,0,0,Off,0,1,Off,0,1,Of
"OCX_BkgdColor","$System","",No,No,0,No,No,0,0,"",65535,0,65535,0,0,Of
"currentWindow","$System","",Yes,No,0,No,No,0,0,"",1,1,15,0,0,Off,0,1,
```

DBLoad Utility Program

➤ **To load/merge a database input file into an existing InTouch:**

1. Close WindowMaker and WindowViewer if they are running.
2. Start InTouch. The **InTouch Application Manger** dialog box appears:

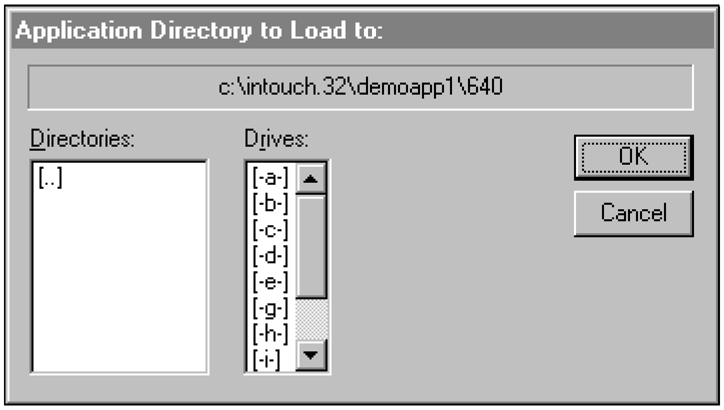

InTouch - Application Manager - [c:\intouch.32\demoapp1\640]
File View Tools Help
DBLoad

Name	Path	Resolution	Version
Application	c:\intouch.32\demoapp1\640	800 x 600	2008
BoilerRm	c:\intouch.32\boilerrm	800 x 600	2008
Features	c:\intouch.32\features	800 x 600	2008

Application - Demo Application of "Now famous" InTouch Reactor. Shows a batch process and product conveyor system.

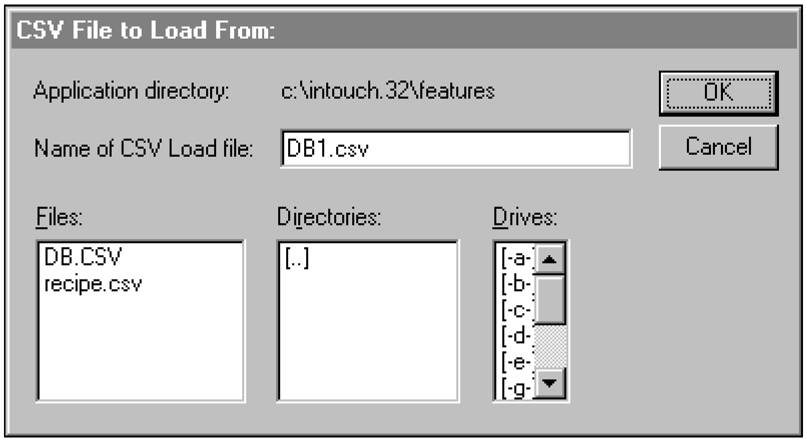
Start the DBLoad application.

3. On the **File** menu, click **DBLoad** or, click the DBLoad tool. The **Application Directory to Load to** dialog box appears:



4. Select the InTouch application directory into which you want to load the database. (It must replace the default directory shown at the top of the dialog box to be properly selected (in this example **c:\intouch.32**). By default, the last directory you accessed will be displayed when the dialog box initially appears.)

5. Click **OK**. A message box will appear asking if you have backed up your application. Click **Yes** to continue. The **CSV File to Load From:** dialog box appears:



6. In the **Name of CSV Load file** box, type the path to the .CSV file that you want to load or, locate the file using the **Directories** and **Drive** list boxes. (Once the file is properly selected its name will be the box.)

7. Click **OK**. The database information contained in the selected file will begin uploading to the selected application's Tagname Dictionary.

Creating a Database Input File

The DBDump and DBLoad database utilities are the tools that you use for performing batch type operations on a Tagname Dictionary. Database input files can be created in any program that supports the comma separated variable file format (**.CSV**). (The database input file must be created in the comma separated variable format.) For example, WordPad, Notepad and Microsoft Excel. Once an input file is created, the DBLoad program is used to load/merge the data contained in the file into an existing InTouch application's database.

You can create a database input file "template" by creating a new InTouch application and then running the DBDump program to dump its database to create a correctly formatted **.CSV** file. This makes entering your modifications easier than creating the input file from scratch.

For more information, see "Creating Database Record Templates."

Database Input File Format

The first line of a database input file should specify the operating **:mode** for the file when it is loaded/merged into an application via **DBLoad**.

If you do not specify a mode, **:mode=test**, by default, **Ask** will be used.

For more information on valid mode keywords, see "Database Input File Operating Modes."

All data records must begin with the valid keyword for the tagname's **:type**, followed by the valid keyword for each data record (separated by commas):

:mode=test
:IOMsg,Group,Comment,Logged,Event Logged,Event Logging Priority, . . .

There is a valid **keyword** for each tagname type and data record

For more information on valid tagname types and the data record entries, "Type and Keyword Entries."

The actual tagname is then entered, followed by the values (separated by commas) for each data record:

:mode=test
:IOMsg,Group,Comment,Logged,EventLogged,EventLoggingPriority, . . .
Ingredient_1,$System,"",No,No,999, . . .

In the above example, **IOMsg** = Ingredient_1, **Group** = $System, **Logged** = No, **EventLogged** = No, and **EventLoggingPriority** = 999. The data record **Comment** will be blank since **""** was entered.

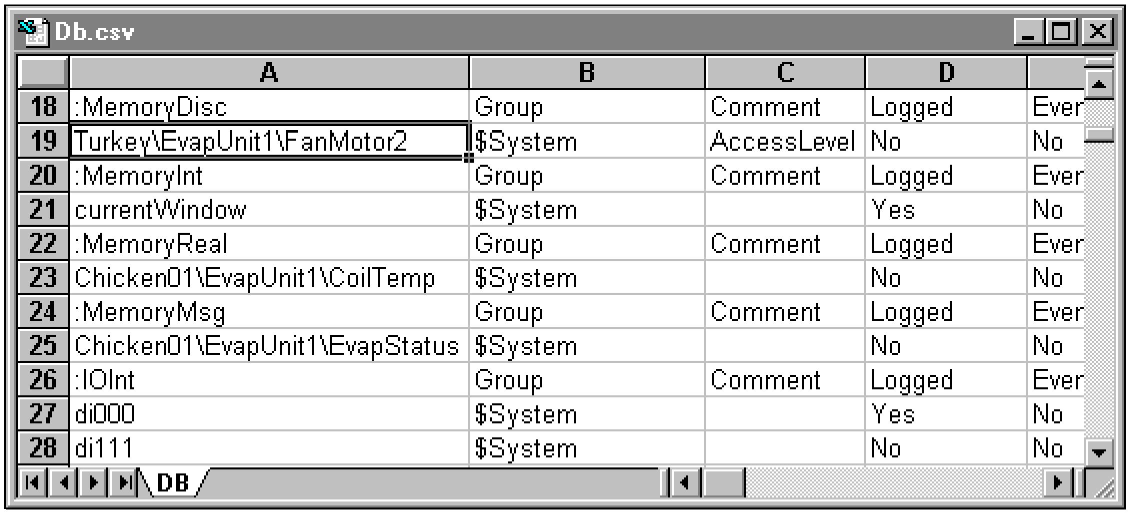
A colon (**:**) must precede the mode and type keywords. To continue a line, a backslash (\) is entered at the end of the line. Comments may be entered by preceding them with a semi-colon (**;**).

Creating SuperTag Instances

In addition to the TemplateMaker, animation links, InTouch QuickScripts and the Tagname Dictionary, InTouch also supports the creation of SuperTags through the DBLoad utility.

Note When you use DBLoad to create SuperTag instances, they are not reflected in the SuperTag template definition in the TemplateMaker.

When you create a SuperTag through DBLoad you must use the valid SuperTag format. and the SuperTag instance data records <u>must</u> begin with the valid keyword for the tagname's **:type**. For example:



Db.csv

	A	B	C	D	
18	:MemoryDisc	Group	Comment	Logged	Ever
19	Turkey\EvapUnit1\FanMotor2	$System	AccessLevel	No	No
20	:MemoryInt	Group	Comment	Logged	Ever
21	currentWindow	$System		Yes	No
22	:MemoryReal	Group	Comment	Logged	Ever
23	Chicken01\EvapUnit1\CoilTemp	$System		No	No
24	:MemoryMsg	Group	Comment	Logged	Ever
25	Chicken01\EvapUnit1\EvapStatus	$System		No	No
26	:IOInt	Group	Comment	Logged	Ever
27	di000	$System		Yes	No
28	di111	$System		No	No

DB

Note The following syntax examples are valid:

ParentInstance\ChildMember
ParentInstance\ChildMember\Submember

The following syntax examples are invalid:

ParentInstance
ParentInstance\ChildMember\

If an invalid format is used, the an error message box will appear informing you that the syntax is invalid.

When you upload the .CSV file containing SuperTag instances, they are automatically added to the Tagname Dictionary, and are immediately available for use in animation links and InTouch QuickScripts.

Blank Strings vs. No Entry

There is a difference between string fields which are blank and fields which have no entry. For example:

:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System

where:

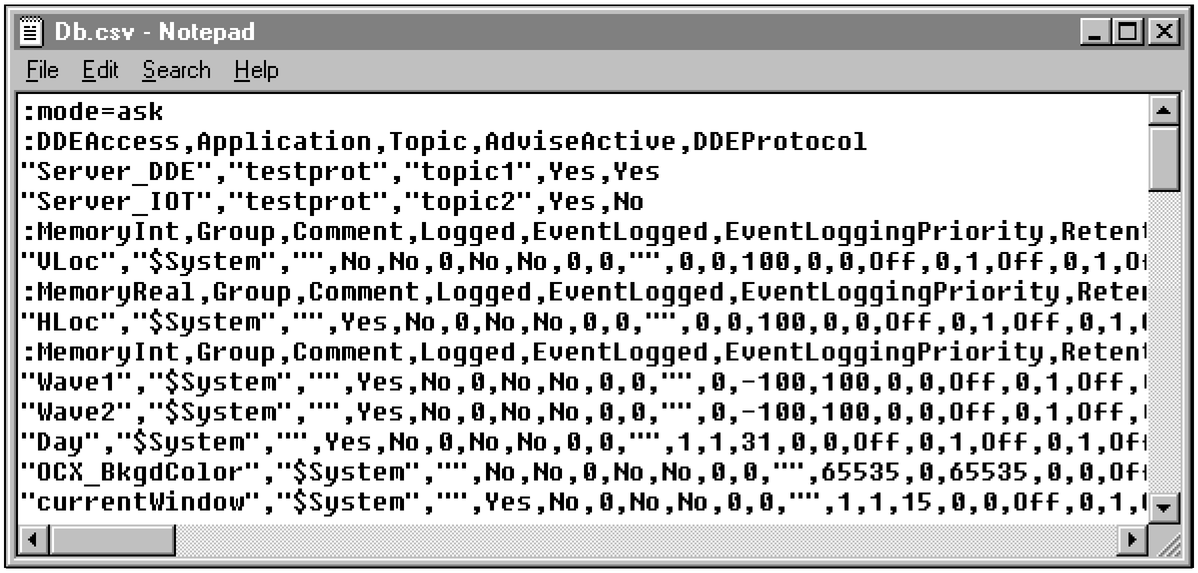
The value of Tagname1's **Comment** field will be **"Hi"** and Tagname2 will have a blank comment. Excel will read in a **.CSV** file such as the one above, but when saving the file, it will save it as follows:

:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,,$System

Thus, to ensure a **blank** string, a space (using the spacebar) must be entered in the cell that is to be **blank**. The following keyword fields are affected in this way:

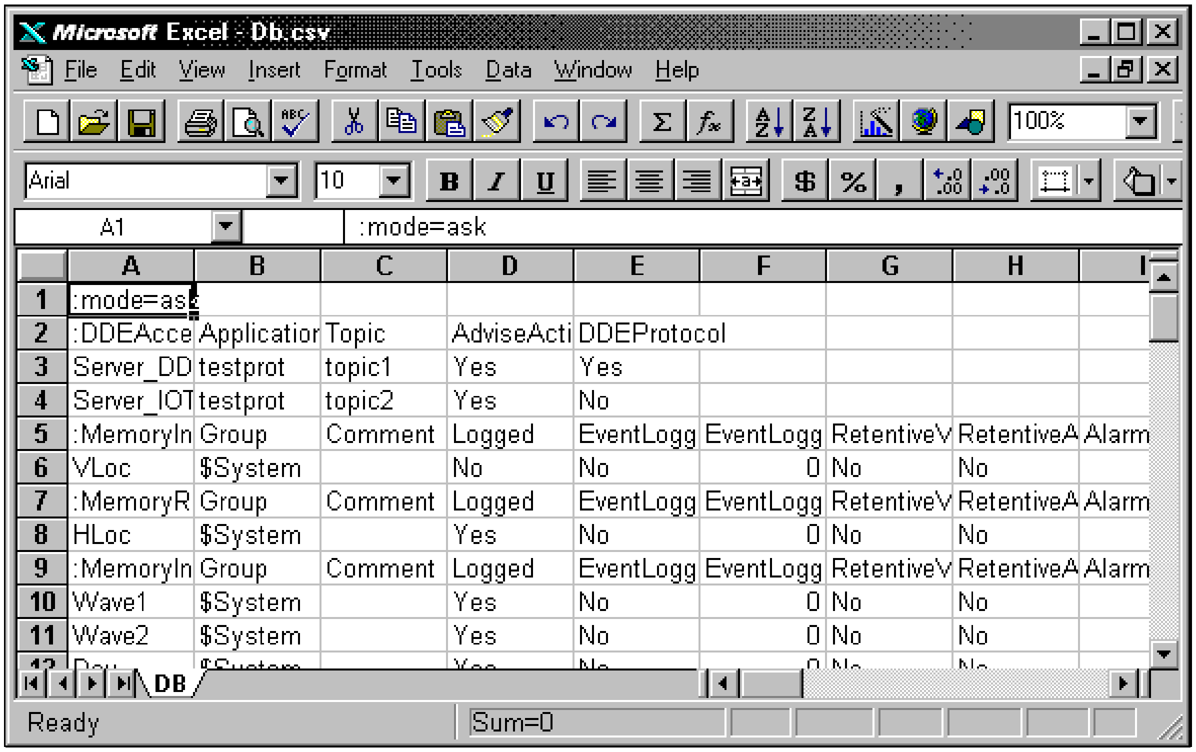
Comment	**Eng Units**
OffMsg	**Initial Message**
OnMsg	**Application**
ItemName	**Topic**

The following illustrates an input file which was created using Windows Notepad:



```
:mode=ask
:DDEAccess,Application,Topic,AdviseActive,DDEProtocol
"Server_DDE","testprot","topic1",Yes,Yes
"Server_IOT","testprot","topic2",Yes,No
:MemoryInt,Group,Comment,Logged,EventLogged,EventLoggingPriority,Retent
"VLoc","$System","",No,No,0,No,No,0,0,"",0,0,100,0,0,Off,0,1,Off,0,1,Of
:MemoryReal,Group,Comment,Logged,EventLogged,EventLoggingPriority,Reten
"HLoc","$System","",Yes,No,0,No,No,0,0,"",0,0,100,0,0,Off,0,1,Off,0,1,(
:MemoryInt,Group,Comment,Logged,EventLogged,EventLoggingPriority,Retent
"Wave1","$System","",Yes,No,0,No,No,0,0,"",0,-100,100,0,0,Off,0,1,Off,(
"Wave2","$System","",Yes,No,0,No,No,0,0,"",0,-100,100,0,0,Off,0,1,Off,(
"Day","$System","",Yes,No,0,No,No,0,0,"",1,1,31,0,0,Off,0,1,Off,0,1,Off
"OCX_BkgdColor","$System","",No,No,0,No,No,0,0,"",65535,0,65535,0,0,Off
"currentWindow","$System","",Yes,No,0,No,No,0,0,"",1,1,15,0,0,Off,0,1,(
```

When you use a program such as Excel to create an input file, the same formatting requirements apply except, each entry is made in a separate column. This makes it much easier to read and there is less chance of error. For example:



	A	B	C	D	E	F	G	H	I
1	:mode=ask								
2	:DDEAcce	Applicatior	Topic	AdviseActi	DDEProtocol				
3	Server_DD	testprot	topic1	Yes	Yes				
4	Server_IOT	testprot	topic2	Yes	No				
5	:MemoryIn	Group	Comment	Logged	EventLogg	EventLogg	RetentiveV	RetentiveA	Alarm
6	VLoc	$System		No	No	0	No	No	
7	:MemoryR	Group	Comment	Logged	EventLogg	EventLogg	RetentiveV	RetentiveA	Alarm
8	HLoc	$System		Yes	No	0	No	No	
9	:MemoryIn	Group	Comment	Logged	EventLogg	EventLogg	RetentiveV	RetentiveA	Alarm
10	Wave1	$System		Yes	No	0	No	No	
11	Wave2	$System		Yes	No	0	No	No	
12	Day	$System		Yes	No	0	No	No	

Database Input File Operating Modes

The following lists the valid operating mode **keywords** and the action which occurs in each mode when a duplicate tagname is encountered during the load:

:MODE=REPLACE
:MODE=UPDATE
:MODE=ASK
:MODE=IGNORE
:MODE=TERMINATE
:MODE=TEST

:MODE=REPLACE

Delete the existing entry and replace it with the new entry.

:MODE=UPDATE

Overwrite the existing definition with only the fields that are explicitly defined in the input file.

Fields are considered explicitly defined if the field is in the record and entered by you or is set by the ":**KEYWORD=value**" mechanism. If a field is not specified in the record and the keyword has been reset via the ":**KEYWORD=**" command, the current field value will not be updated.

The following is an example of what will occur when an input file in the update mode is loaded/merged into an application's database via DBLoad:

:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
; Tagname1's Group updated to Group2 only
Tagname2,,
; Tagname2's Group updated to Group1 and the DConversion left as is
Tagname3,,Reverse
; Tagname3's Group updated to Group1 and the DConversion to "Reverse"
; the following line "resets" the Group field to its default value
:Group=
; Data field "Group" is reset to its default value
Tagname4,,
; Tagname4 will be left alone

Comments are allowed in the .CSV file. They must be identified with an opening semi-colon (**;**).

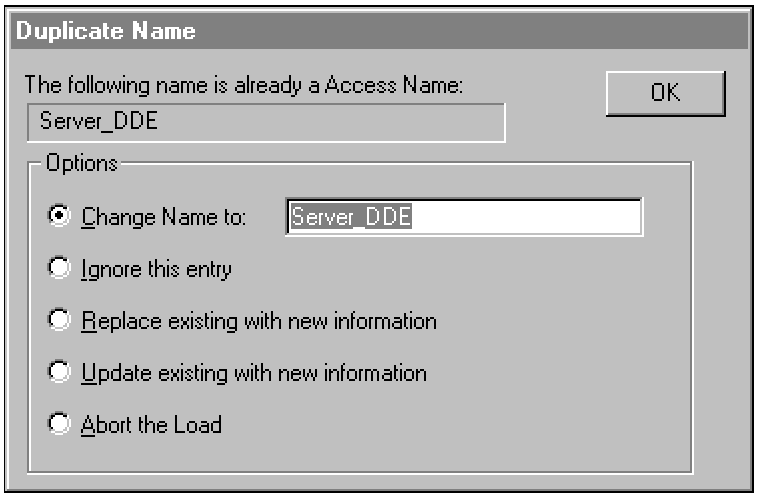
Note The tagname types must be compatible if the type is being changed and the tagname is in use. For example, an existing historical trend tagname cannot be changed to an I/O Integer if the tagname is in use by the application. Also, a tagname cannot be changed to **ReadOnly=yes** if the tagname is being used on an input link in the application. Because of these restrictions, you should update the target application's tagname use counts before running DBLoad.

:MODE=ASK

Change the tagname of the input or the existing entry to your specified tagname and then insert the new definition into the Tagname Dictionary.

Note This is the default input mode if no mode is specified. Each time DBLoad comes across a duplicate tagname, a dialog box will appear asking whether the existing tagname is to be replaced.

When the operating mode is set to **ASK**, and a tagname in the input file is a duplicate of a tagname in the target application's Tagname Dictionary, the **Duplicate Name** dialog box will appear:


Duplicate Name
The following name is already a Access Name:
Server_DDE
OK
Options
Change Name to:
Server_DDE
Ignore this entry
Replace existing with new information
Update existing with new information
Abort the Load

Select an option, then click **OK**. A confirmation message box will appear when the load is completed.

If a problem occurs that causes the load to fail, a message box will appear. (The error messages are written to the Wonderware Logger program.)

Option	Description
Change Name to	Replaces the name for the specified tagname with the name typed in the box.
Ignore this entry	Ignores the displayed tagname and processing continues.
Replace existing with new information	Replaces the existing tagname record with the new record.
Update existing with new information	Overwrites the existing tagname record with only the fields that are explicitly defined in the input file.
Abort the Load	Cancels the load function.

:MODE=IGNORE

Ignore the new record and continue processing.

:MODE=TERMINATE

Terminate processing. Do not update target file.

:MODE=TEST

In this mode, DBLoad will act as if it were in the replace mode, but will not modify the database. It will report errors as found in the Wonderware Logger program and continue with the load. This is useful for verifying the syntax of the input file before actually processing it.

For more information on the Wonderware Logger program, see your *FactorySuite System Administrator's Guide*.

Note Once the input file is set to the test mode, all other modes are ignored. You can enter :**mode=test** as the first line of the file and not be concerned about other mode changes in the file.

For more information on operating modes, see "Creating a Database Input File."

Creating Database Record Templates

KEYWORDs can be used to create template records that supply a "global" entry for the respective data fields in subsequent data records. (There is a keyword for each field value that can be set for the tagname except for the **TAGNAME** and **TYPE** fields.)

Note A template record must follow the formatting requirements previously described for creating a comma separated input file. For example the file must begin with a **:Type** keyword entry.

Setting Field Value Defaults

The keywords can also be used to set the default values for specific fields of a record. For example:

:KEYWORD=value
This will set the default value of the referenced field for all subsequent data records. This feature can be used to set the default value for fields that will remain unchanged for a number of records. If a field has a default value defined, the default value is used if there is no data in the record for the value. For example, the result of **:GROUP=Reactor_Site** would be that all tagnames having a blank entry for the **GROUP** column will be assigned to the **Reactor_Site** Alarm Group. If the tagname has, for example, **$System** entered for the **GROUP**, it remains assigned to the Alarm Group **$System**.

Resetting Field Value Defaults

The individual keywords can be reset to their original default values by not specifying a value. For example, **:GROUP=**.

Resetting All Field Value Defaults

To reset all keywords, use the **:RESET** command. This command is entered "as is" with no arguments (**:RESET**) and will affect all entries below the command.
Note Default values are the original InTouch values for that tagname type. For example, a memory discrete uses the **Group=$System**, **EventLogging=Off**, **InitialValue=Off**, etc. for its default values.

:Type and Keyword Entries

The table below lists the valid keywords for each tagname **:Type** followed by the keyword for each field value for that type.

:Type & Keywords	Acceptable Values	Default
:MemoryDisc		
AlarmPri	1 to 999	1
AlarmState	None, On, or Off	None
Comment	Any Text String	""
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
InitialDisc	1, 0, On, Off, True, False, Yes, or No	0
Logged	On, Off, Yes, or No	No
OffMsg	Any Text String	None
OnMsg	Any Text String	None
RetentiveValue	1, 0, On, Off, True, False, Yes, or No	No

:Type & Keywords	Acceptable Values	Default
:IODisc		
AlarmPri	1 to 999	1
AlarmState	None, On, or Off	None
Comment	Any Text String	None
DConversion	Direct or Reverse	Direct
AccessName	Valid Access Name	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
InitialDisc	1, 0, On, Off, True, False, Yes, or No	0
ItemName	Valid Item Name	None
ItemUseTagname	True, False, Yes, or No	Yes
Logged	On, Off, Yes, or No	No
OffMsg	Any Text String	None
OnMsg	Any Text String	None
ReadOnly	Yes or No	No
RetentiveValue	1, 0, On, Off, True, False, Yes, or No	No
:MemoryInt & :Memory Real		
AlarmDevDeadband	Valid Deviation Percentage	0
AlarmValueDeadband	Valid Integer or Real Value	0
Comment	Any Text String	None
Deadband	Valid Integer of Real Value	0
DevTarget	Valid Integer or Real Value	0
EngUnits	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
HiAlarmPri	1 to 999	1
HiAlarmState	Yes, No, On, or Off	No
HiAlarmValue	Valid Integer or Real Value	0
HiHiAlarmPri	1 to 999	1
HiHiAlarmState	Yes, No, On, or Off	No
HiHiAlarmValue	Valid Integer or Real Value	0
InitialValue	Valid Integer or Real Value	0
LoAlarmPri	1 to 999	1
LoAlarmState	Yes, No, On, or Off	No
LoAlarmValue	Valid Integer or Real Value	0
LogDeadband	Valid Integer or Real Value	0
Logged	On, Off, Yes, or No	No
LoLoAlarmPri	1 to 999	1
LoLoAlarmState	Yes, No, On, or Off	No
LoLoAlarmValue	Valid Integer or Real Value	0
MajorDevAlarmPri	1 to 999	1
MajorDevAlarmState	Yes, No, On, or Off	No
MajorDevAlarmValue	Valid Integer or Real Value	0
MaxValue	Valid Integer or Real Value	9999

:Type & Keywords	Acceptable Values	Default
MinorDevAlarmPri	1 to 999	1
MinorDevAlarmState	Yes, No, On, or Off	No
MinorDevAlarmValue	Valid Integer or Real Value	0
MinValue	Valid Integer or Real Value	0
RetentiveAlarmParameters	On, Off, Yes, or No	No
RetentiveValue	On, Off, Yes, or No	No
ROCAlarmPri	1 to 999	1
ROCAlarmState	Yes, No, On, or Off	No
ROCAlarmValue	Any Valid Percentage	0
ROCTimeBase	Sec, Min, or Hr	Min
:IOInt & :IOReal		
AlarmDevDeadband	Valid Deviation Percentage	0
AlarmValueDeadband	Valid Integer or Real Value	0
Comment	Any Text String	None
Conversion	Linear or Square Root	Linear
AccessName	Valid Access Name	None
Deadband	Valid Integer or Real Value	0
DevTarget	Valid Integer or Real Value	0
EngUnits	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
HiAlarmPri	1 to 999	1
HiAlarmState	Yes, No, On, or Off	No
HiAlarmValue	Valid Integer or Real Value	0
HiHiAlarmPri	1 to 999	1
HiHiAlarmState	Yes, No, On, or Off	No
HiHiAlarmValue	Valid Integer or Real Value	0
InitialValue	Valid Integer or Real Value	0
ItemName	Valid Item Name	None
ItemUseTagname	True, False, Yes, or No	No
LoAlarmPri	1 to 999	1
LoAlarmState	Yes, No, On, or Off	No
LoAlarmValue	Valid Integer or Real Value	0
LogDeadband	Valid Integer or Real Value	0
Logged	On, Off, Yes, or No	No
LoLoAlarmPri	1 to 999	1
LoLoAlarmState	Yes, No, On, or Off	No
LoLoAlarmValue	Yes, No, On, or Off	0
MajorDevAlarmPri	1 to 999	1
MajorDevAlarmState	Yes, No, On, or Off	No
MajorDevAlarmValue	Valid Integer or Real Value	0
MaxEU	Valid Integer or Real Value	9999
MaxRaw	Valid Integer or Real Value	9999
MinEU	Valid Integer or Real Value	0
MinorDevAlarmPri	1 to 999	1
MinorDevAlarmState	Yes, No, On, or Off	No
MinorDevAlarmValue	Valid Integer or Real Value	0
MinRaw	Valid Integer or Real Value	0

:Type & Keywords	**Acceptable Values**	**Default**
ReadOnly	Yes or No	No
RetentiveAlarmParameters	On, Off, Yes, or No	No
RetentiveValue	On, Off, Yes, or No	No
ROCAlarmPri	1 to 999	1
ROCAlarmState	Yes, No, On, or Off	No
ROCAlarmValue	Any Valid Percentage	0
ROCTimeBase	Sec, Min, or Hr	Min
:MemoryMsg		
Comment	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
InitialMessage	Any Text String	None
Logged	On, Off, Yes, or No	No
MaxLength	1-131 characters	131
RetentiveValue	On, Off, Yes, or No	No
:IOMsg		
Comment	Any Text String	None
AccessName	Valid Access Name	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
InitialMessage	Any Text String	None
ItemName	Valid Item Name	None
ItemUseTagname	True, False, Yes, or No	No
Logged	On, Off, Yes, or No	No
MaxLength	1-131 characters	131
ReadOnly	Yes or No	No
RetentiveValue	On, Off, Yes, or No	No
:GroupVar		
Comment	Any Text String	None
Group	Valid Group Name	$System
EventLogged	On, Off, Yes, or No	No
:Type & Keywords	Acceptable Values	Default
:HistoryTrend		
Comment	Any Text String	None
Group	Valid Group Name	$System
:TagID		
Comment	Any Text String	None
Group	Valid Group Name	$System
:IndirectDisc		
Comment	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
RetentiveValue	On, Off, Yes, or No	No

:Type & Keywords	Acceptable Values	Default
:IndirectAnalog		
Comment	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
RetentiveValue	On, Off, Yes, or No	No
:IndirectMsg		
Comment	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System
RetentiveValue	On, Off, Yes, or No	No
:Access		
AdviseActive	Yes or No	Yes
NodeName	Valid network node name	
Application	Valid I/O application name	None
Topic	Valid I/O topic name	None
DDEProtocol	Yes specifies the DDE protocol No specifies the SuiteLink protocol	Yes
:AlarmGroup		
Comment	Any Text String	None
EventLogged	On, Off, Yes, or No	No
EventLoggingPriority	1 to 999	999
Group	Valid Group Name	$System

C H A P T E R 5

Creating Animation Links

Once you create a graphic object or symbol you can "bring it to life" by animating it. By attaching animation links, you can make the 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 or touch (when using a touch screen) 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.

This chapter describes the procedures you use to create each type of animation link.

Contents

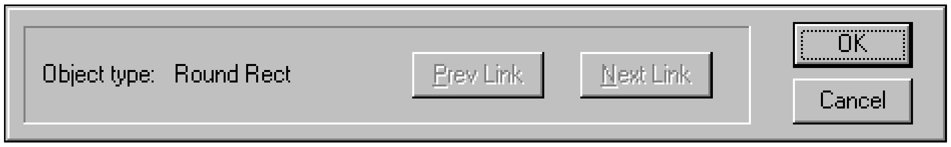
Common Animation Link Features

Many of the 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 you can apply 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 that is 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**.

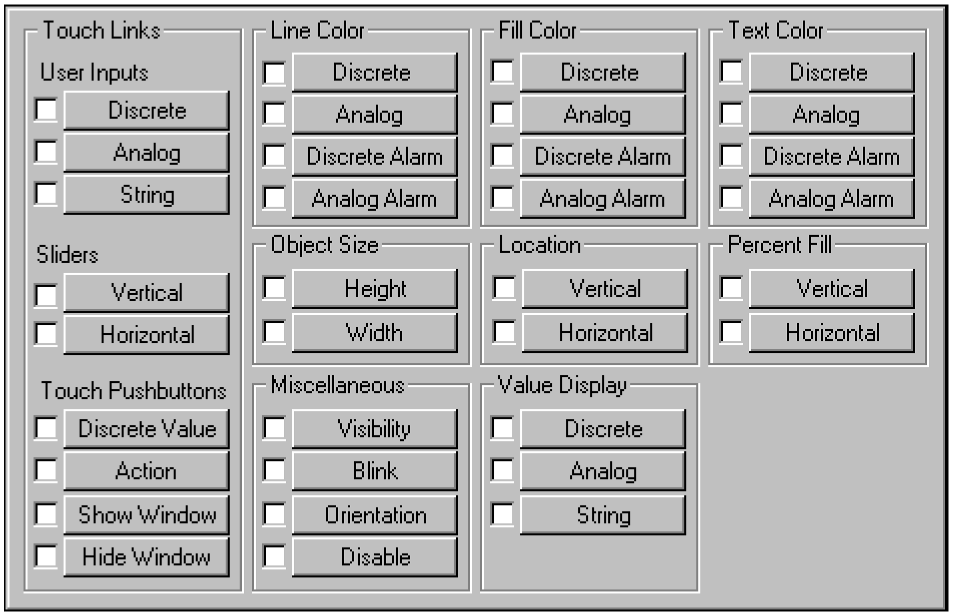

Object type: Round Rect

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.

Links are stored in the order in which they were originally attached to the object.

Animation Link Selection Dialog Box

You can define multiple links 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, visibility, fill level, and so on.



Assigning Key Equivalents

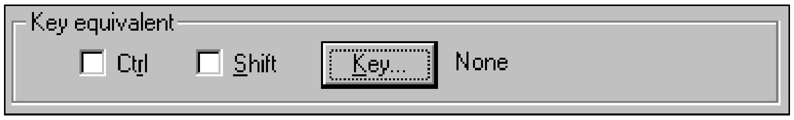
You can assign a specific key on the keyboard to activate some animation links. The key equivalent is only operational when the object with the link is visible or selected.
If the object has a visibility or disable link, the key equivalent is not active when the object is invisible or disabled.

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.

Note If any object or action pushbutton in the active window is assigned to the same key used for a **Key Action Script**, the key equivalent link on the key in the active window will take precedence over the execution of the **Key Action Script**.

For more information on **Key Action Scripts**, see Chapter 6 - Creating InTouch QuickScripts.

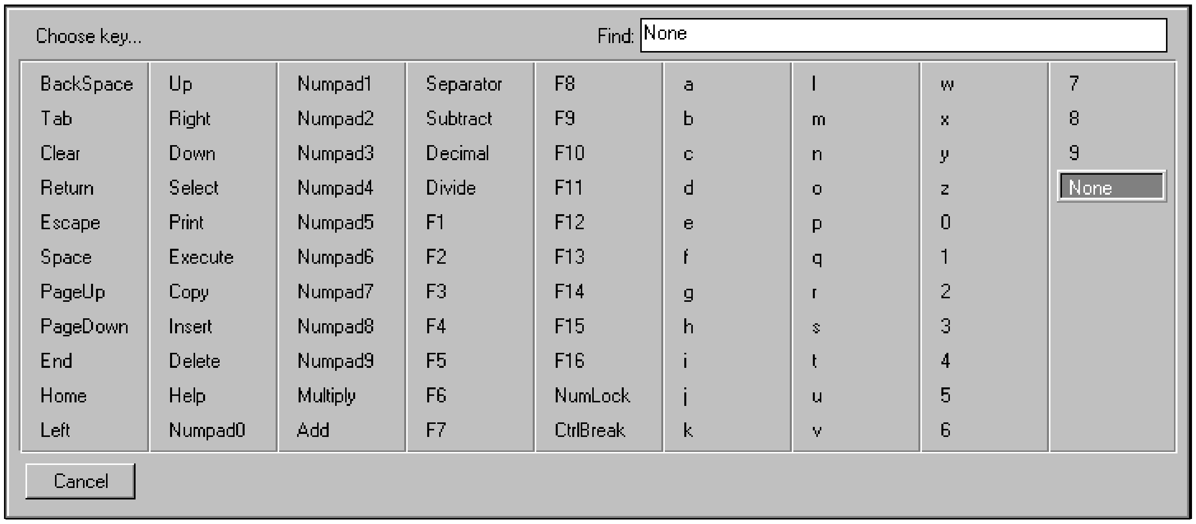
The animation links that support key equivalents will display the **Key Equivalent** group in their link dialog boxes. For example:



To assign a key to a link:

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

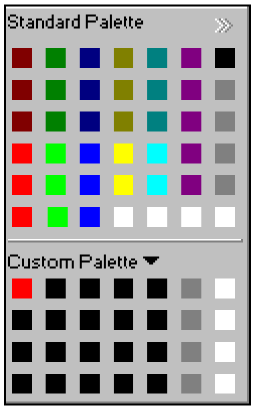
Cancel

3. Click the key that you want 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.

Applying Color Links

You can apply color to the dynamic properties of lines, rectangles, round rectangles, ellipses, polylines, polygons and text. When you create color links for lines, fill or text objects, you will use the color palette to select the colors that you want linked to the value of the tagname, the tagname's alarm state, and so on.

You must use solid colors for line and text color links. If you select a dithered (mixed) color, WindowMaker, by default, will select the closest solid color. You can create custom color palettes and load them into the standard WindowMaker color palette.



When you attach a color link to an object or symbol and you click on a color box in a link's dialog box, the color palette appears.

Click the color you want to use for the link. The color palette automatically closes and the color you selected will appear in the color box in the link detail dialog box.

For more information on customizing the color palette, see Chapter 1 - WindowMaker Program Elements.

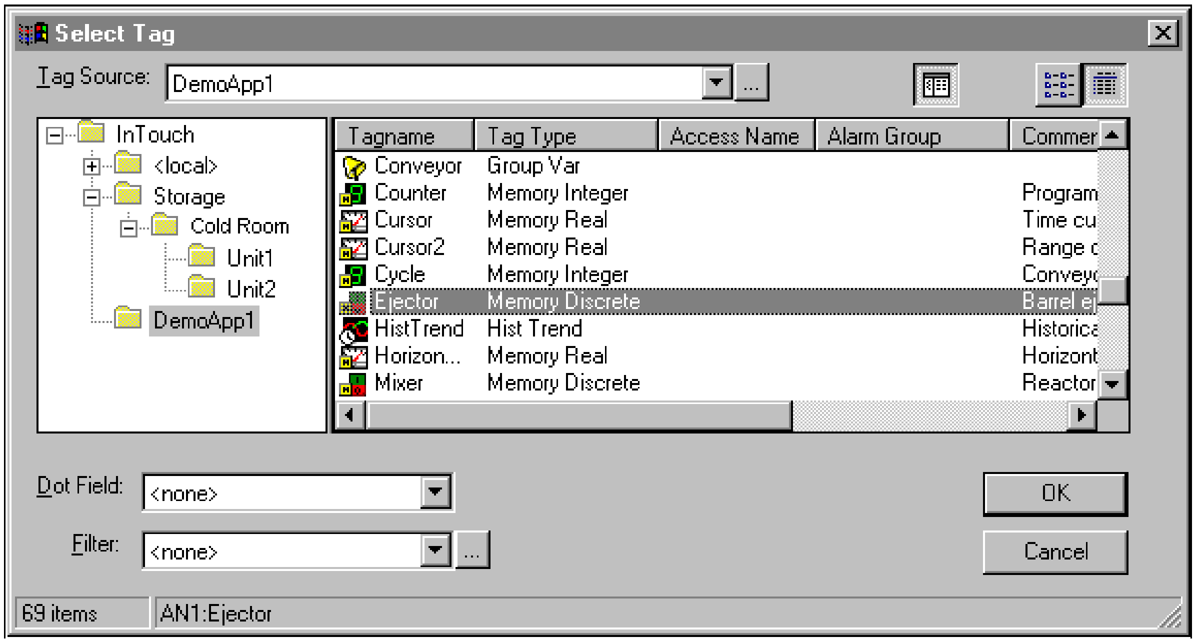
Accessing the Tag Browser

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

➢ **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 tool to show the tree view pane displaying all defined tag sources:



Select Tag

Tag Source: DemoApp1

InTouch
- <local>
- Storage
  - Cold Room
    - Unit1
    - Unit2
- DemoApp1

Tagname	Tag Type	Access Name	Alarm Group	Commer
Conveyor	Group Var			
Counter	Memory Integer			Program
Cursor	Memory Real			Time cu
Cursor2	Memory Real			Range c
Cycle	Memory Integer			Conveyc
Ejector	Memory Discrete			Barrel e
HistTrend	Hist Trend			Historica
Horizon...	Memory Real			Horizont
Mixer	Memory Discrete			Reactor

Dot Field: <none>

Filter: <none>

69 items AN1:Ejector

If you are not using the tree view mode, click the **Tag Source** arrow and select the name for the tag source that you want to display in the list. The Tag Browser will refresh and the selected tag source's tagnames will be displayed.

3. Select the tagname you want to use for the link then 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 and select the **.field** that you want to use in the list, and then click **OK**.

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

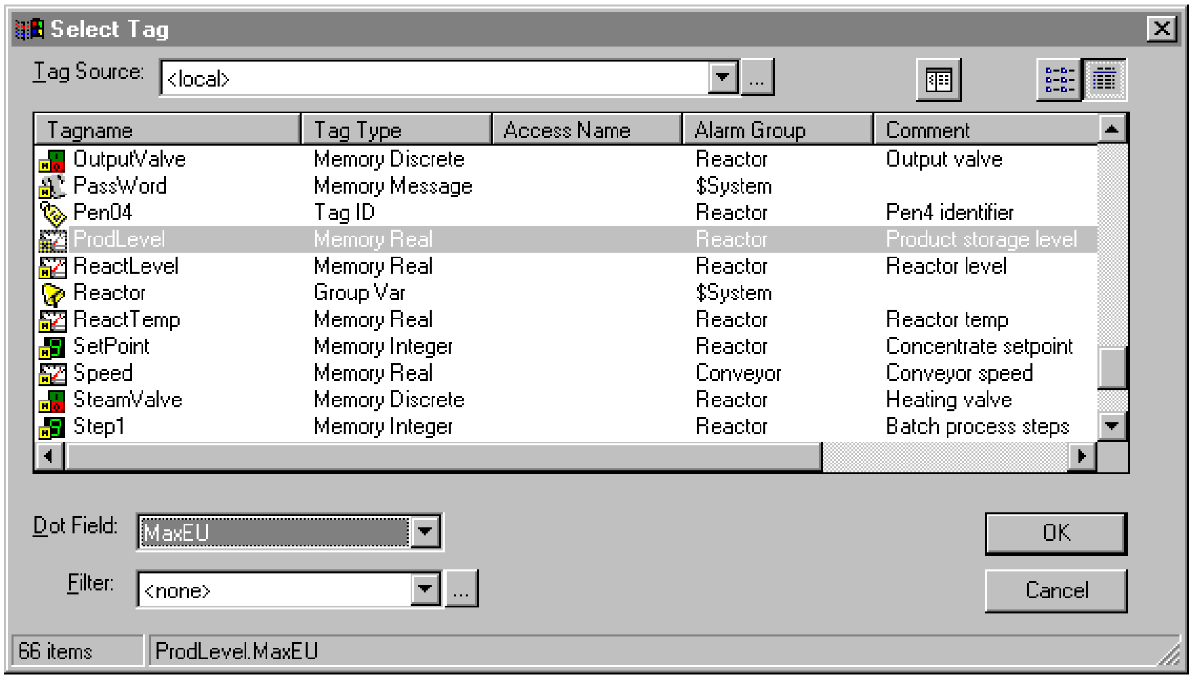
   For more information on the Tag Browser, see Chapter 4 - Tagname Dictionary.

Accessing Tagname .Fields

There are two methods that you can use to access tagname **.fields** from an animation link **Tagname** or **Expression** input box. The two methods are described below.

➢ **To access 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:



Select Tag

Tag Source: <local>

Tagname	Tag Type	Access Name	Alarm Group	Comment
OutputValve	Memory Discrete		Reactor	Output valve
PassWord	Memory Message		$System	
Pen04	Tag ID		Reactor	Pen4 identifier
ProdLevel	Memory Real		Reactor	Product storage level
ReactLevel	Memory Real		Reactor	Reactor level
Reactor	Group Var		$System	
ReactTemp	Memory Real		Reactor	Reactor temp
SetPoint	Memory Integer		Reactor	Concentrate setpoint
Speed	Memory Real		Conveyor	Conveyor speed
SteamValve	Memory Discrete		Reactor	Heating valve
Step1	Memory Integer		Reactor	Batch process steps

Dot Field: MaxEU

Filter: <none>

66 items ProdLevel.MaxEU

2. Click the **Dot Field** arrow to open the list of **.fields** that you can associate with the type of tagname currently selected.

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

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

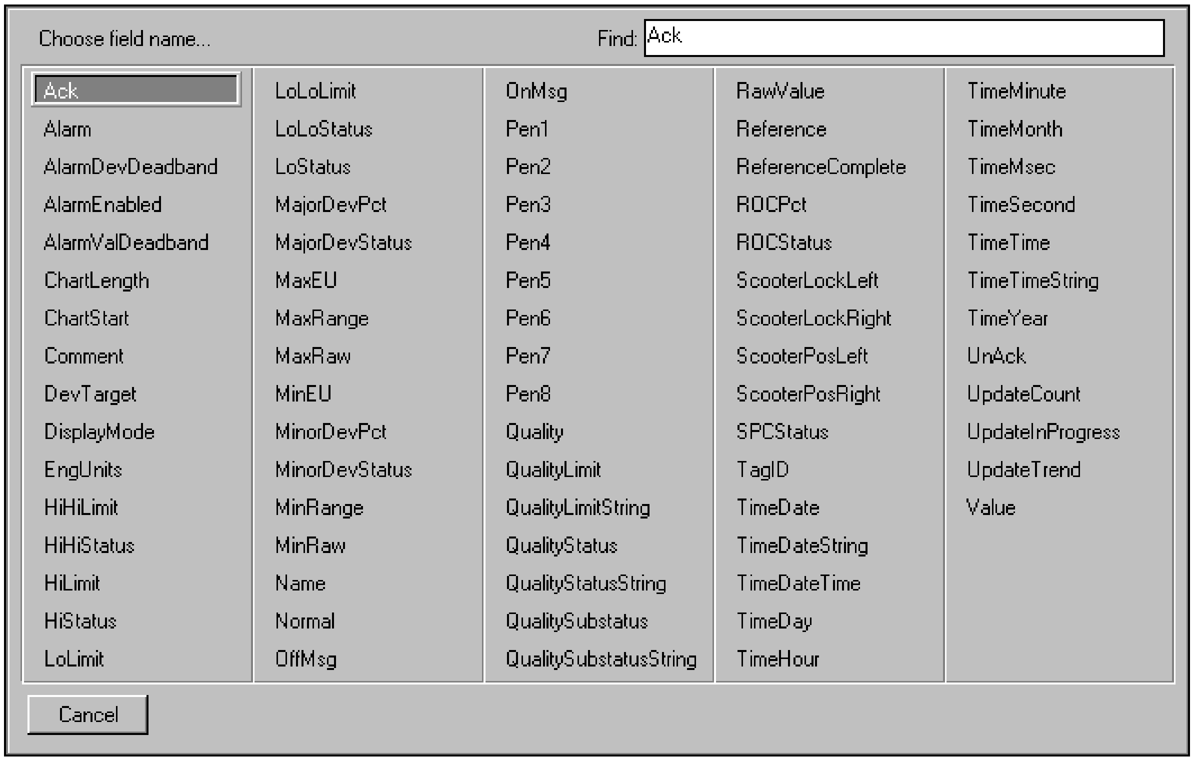
Note Not every tagname type has the same **.fields**. For example, a **Discrete** type tagname has **.OnMessage**, whereas an analog does not. If you select a **Discrete** type tagname and you assign **.OnMessage** to it, and then you select another **Discrete** type tagname, the displayed **.field** list will not change. But, if you select another type of tagname in the control view list, for example an analog, the displayed **.field** will revert to **<none>**.

For more information on the Tag Browser, see Chapter 4 - Tagname Dictionary.

For more information on tagname **.fields**, see your *InTouch Reference Guide*.

➢ **To access the tagname .fields through the Choose field name dialog box:**

1. In **Tagname** or **Expression** input box, type a tagname plus a period (**tagname.**) and then double-click to the right of it, or type just a period, and then double-click to the right of it. The **Choose field name** dialog box appears displaying all tagname .fields:


Choose field name... Find: Ack

Ack	LoLoLimit	OnMsg	RawValue	TimeMinute
Alarm	LoLoStatus	Pen1	Reference	TimeMonth
AlarmDevDeadband	LoStatus	Pen2	ReferenceComplete	TimeMsec
AlarmEnabled	MajorDevPct	Pen3	ROCPct	TimeSecond
AlarmValDeadband	MajorDevStatus	Pen4	ROCStatus	TimeTime
ChartLength	MaxEU	Pen5	ScooterLockLeft	TimeTimeString
ChartStart	MaxRange	Pen6	ScooterLockRight	TimeYear
Comment	MaxRaw	Pen7	ScooterPosLeft	UnAck
DevTarget	MinEU	Pen8	ScooterPosRight	UpdateCount
DisplayMode	MinorDevPct	Quality	SPCStatus	UpdateInProgress
EngUnits	MinorDevStatus	QualityLimit	TagID	UpdateTrend
HiHiLimit	MinRange	QualityLimitString	TimeDate	Value
HiHiStatus	MinRaw	QualityStatus	TimeDateString	
HiLimit	Name	QualityStatusString	TimeDateTime	
HiStatus	Normal	QualitySubstatus	TimeDay	
LoLimit	OffMsg	QualitySubstatusString	TimeHour	

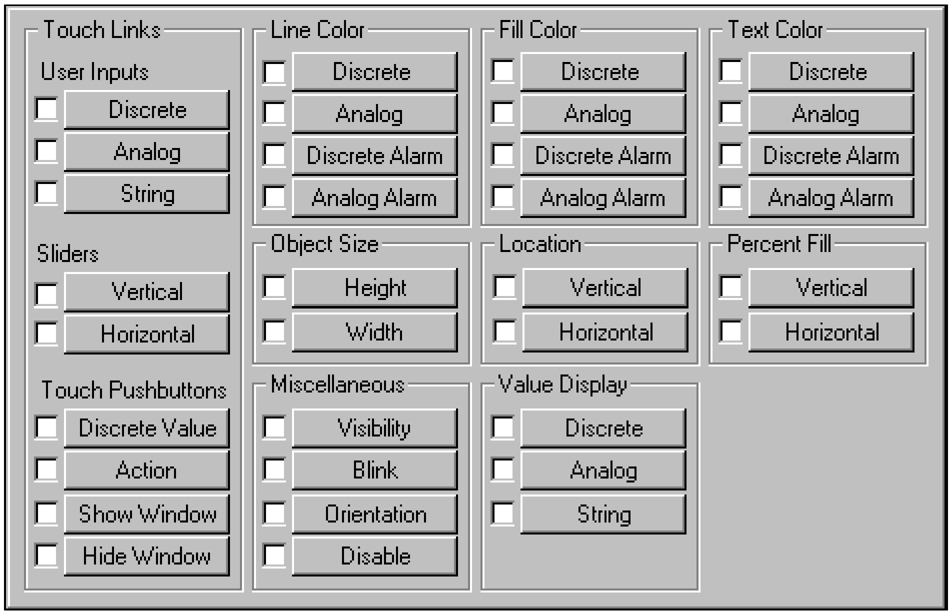
2. Select the **.field** that you want to use. The dialog box will close and the selected **.field** will automatically be inserted into the **Tagname** or **Expression** input box.

Animating Objects

➢ **To animate an object or symbol:**

1. Create and select the object (line, filled shape, text, button or symbol) that you want to animate.
2. On the **Special** menu, click **Animation Links** or, double-click on the object. The dialog box containing all animation links appears:

   You can also right-click the object, and then click **Animation Links**.



3. Click the button for the link that you want to attach 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.

   Note You will not be able to modify the definition of the default link unless you click the button.

4. Enter the details for the link definition, and then click **OK**. The **Link Selection** dialog box will reappear and if desired, you can create another link for the object.
5. Click **OK** to accept all links for the object and close the dialog box.

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

Creating Touch Links

You use **Touch Links** on objects or symbols that you want to be "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 or log on using text strings, and so on.

Touch links are easily identified in runtime because a "frame" surrounds a touch-sensitive object when you pass the cursor over it or you press the TAB key 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 that you can create:

Touch Link	Types
User Inputs	Discrete, Analog, String
Sliders	Vertical, Horizontal
Touch Pushbuttons	Discrete Value, Action, Show Window, Hide Window

The following sections describe how to create each of the touch links.

Note If the object or symbol used for these links (with the exception of 3-D buttons) contains a text field, all attributes currently set for text, (justification, style, font, and so on) will be applied when the text object is displayed in WindowViewer. When a text field is used for inputting the value, the output value will also be displayed unless the **Input Only** option in the respective tagname's detail dialog box is enabled.

Creating User Input Touch Links

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

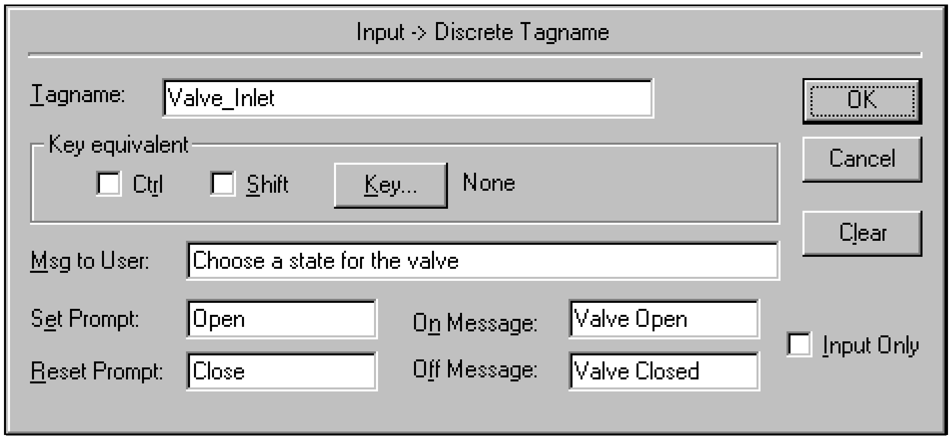
User Input	Description
Discrete	Used to control the value of a discrete tagname. When this link is activated, a dialog box will appear prompting the operator to make a selection.
Analog	Used to input the value of an analog (integer or real) tagname. When the link is activated, an input box will appear 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 will appear for entering the message value or an optional on screen keyboard.

➢ **To create a discrete input link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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



Tagname: Valve_Inlet
Key equivalent: None
Msg to User: Choose a state for the valve
Set Prompt: Open
Reset Prompt: Close
On Message: Valve Open
Off Message: Valve Closed

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

   Right-click the **Tagname** box, to access the commands that you can apply to the selected text.

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

   For more information on assigning keys, "Assigning Key Equivalents."

5. In the **Msg to User** box, type the message that you want to appear in the input dialog box when the input link is activated.

6. In the **Set Prompt** and **Reset Prompt** boxes, type the messages that you want to display on the buttons the operator will click in the input dialog box to turn the discrete value on and off.

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

8. Select **Input Only** if you want 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.

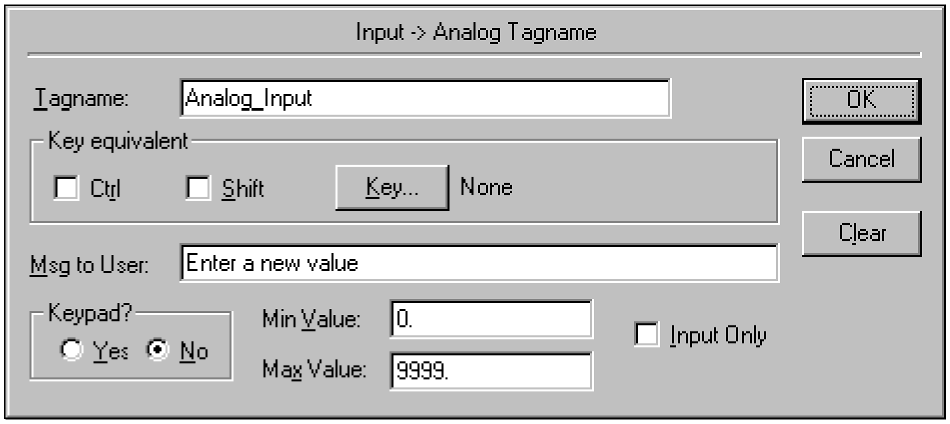
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create an analog input link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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



Input -> Analog Tagname

Tagname: Analog_Input

Key equivalent: ☐ Ctrl ☐ Shift Key... None

Msg to User: Enter a new value

Keypad? Yes [✓ No]

Min Value: 0.

Max Value: 9999.

☐ Input Only

Note If a text field is being used for this link, it must be formatted properly in order to display the analog (integer or real) value's output correctly.

For more information on formatting text fields, see Chapter 2 - Using WindowMaker.

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

   Right-click the **Tagname** box, to access the commands that you can apply to the selected text.

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

   For more information on assigning keys, see "Assigning Key Equivalents."

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

6. If you want to display an on-screen numeric keypad for inputting the new value of the string, select **Yes**.

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

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

9. Select **Input Only** if you want 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 if desired.

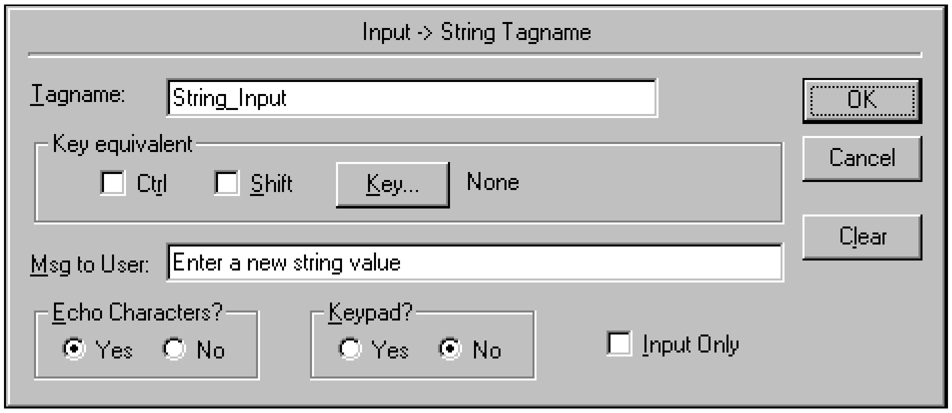
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create a string input link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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


Input -> String Tagname

Tagname: String_Input

Key equivalent: ☐ Ctrl ☐ Shift Key... None

Msg to User: Enter a new string value

Echo Characters? [✓ Yes] No

Keypad? Yes [✓ No]

☐ Input Only

3. In the **Tagname** box, type a message type tagname.

   Right-click the **Tagname** box, to access the commands that you can apply to the selected text.

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

   For more information on assigning keys, see "Assigning Key Equivalents."

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

6. If you want the input string to appear on the screen as it is typed, select **Yes** for the **Echo Characters?** option. If the data is sensitive (for example, a password) and should not be visible on the screen, select **No**.

7. If you want to display an on-screen keyboard for inputting the new value of the string, select **Yes** for the **Keypad?** option.

8. Select **Input Only** if you want 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.)

   Note When WindowViewer is initially started, the string will display the text you typed in the **Initial Value** box when you defined the tagname linked to this object.

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.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Slider Touch Links

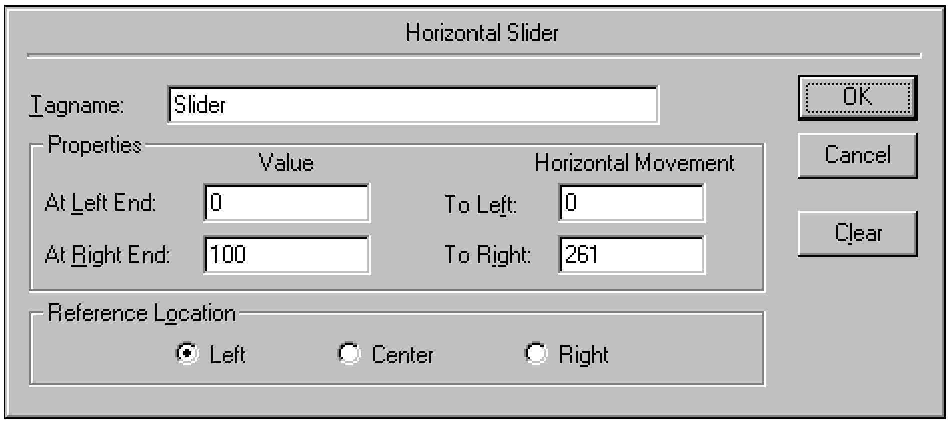
You use **Slider Touch Links** to create objects or symbols that can be moved around the window with the mouse or other pointing devices such as a finger on a touch screen. As the object or symbol is moved, it alters the value of the tagname linked to it. This provides the ability to create devices for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

An object may have a horizontal or 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.

Note The horizontal and vertical slider links are created the same way. This procedure describes the **Horizontal Slider** link.

➢ **To create a horizontal (or vertical) slider link:**

1. Double-click the object or select it, and then on the **Special** menu, click **Animation Links**. The link selection dialog box will appear.
   - To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.
2. In the **Slider** section, click **Horizontal**. The **Horizontal Slider** dialog box appears:



Horizontal Slider

Tagname: Slider

Properties

	Value		Horizontal Movement
At Left End:	0	To Left:	0
At Right End:	100	To Right:	261

Reference Location: [✓ Left] Center Right

3. In the **Tagname** box, type an analog (integer or real) type tagname.
   - Right-click the **Tagname** box, to access the commands that you can apply to the selected text.
4. In the **At Left End** box, type the value for the tagname when the slider is in its farthest left position.
5. In the **At Right End** box, type the value for the tagname when the slider is in its farthest right position.
6. In the **To Left** box, type 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, type 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 on for moving the object.
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.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Touch Pushbuttons Touch Links

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

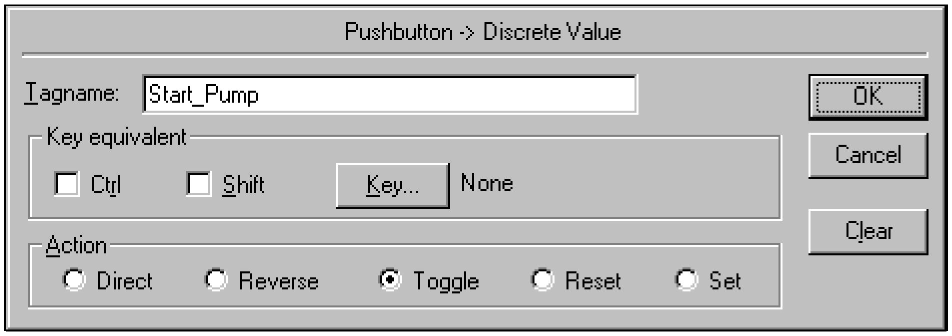
Touch Pushbuttons	Description
Discrete Value	Used to make any object or symbol into a pushbutton that controls the state of a discrete tagname. Pushbutton actions can be set, reset, toggle, momentary on (direct) and momentary off (reverse) types.
Action	Allows any object, symbol or button to have up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

➢ **To create a discrete value touch pushbutton link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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


Pushbutton -> Discrete Value

Tagname: Start_Pump

Key equivalent: Ctrl [ ] Shift [ ] Key... None

Action: Direct, Reverse, [✓ Toggle], Reset, Set

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

   Right-click the **Tagname** box, to access the commands that you can apply to the selected text.

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

5. For more information on assigning keys, see "Assigning a Key to an Animation Link."

6. Select the **Action** option that you want 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.

7. 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.

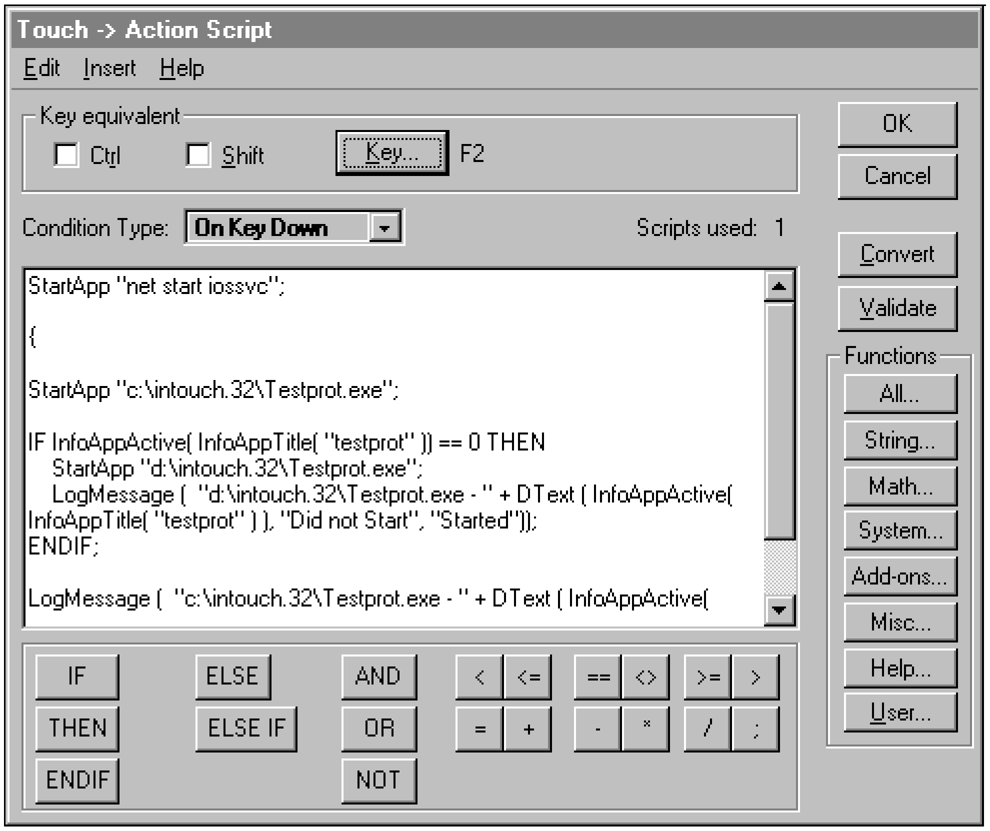
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create an action touch pushbutton link**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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



```
StartApp "net start iossvc";

{

StartApp "c:\intouch.32\Testprot.exe";

IF InfoAppActive( InfoAppTitle( "testprot" )) == 0 THEN
   StartApp "d:\intouch.32\Testprot.exe";
   LogMessage (  "d:\intouch.32\Testprot.exe - " + DText ( InfoAppActive(
InfoAppTitle( "testprot" ) ), "Did not Start", "Started"));
ENDIF;

LogMessage (  "c:\intouch.32\Testprot.exe - " + DText ( InfoAppActive(
```

For more information on writing QuickScripts, see Chapter 6 - Creating InTouch QuickScripts.

3. Click the **Condition Type** arrow and select the script type that you want 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.

Note If any object or action pushbutton in the active window is assigned to the same key used for a **Key** Script, the key equivalent link on the key in the active window will take precedence over the execution of the **Key** script.

For more information on scripts, see Chapter 6 - Creating InTouch QuickScripts.

4. Click in the script editor's window and type the script that you want executed when the object is activated.
5. Click **OK** to attach the script to the object and return to the animation links dialog box. You can now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

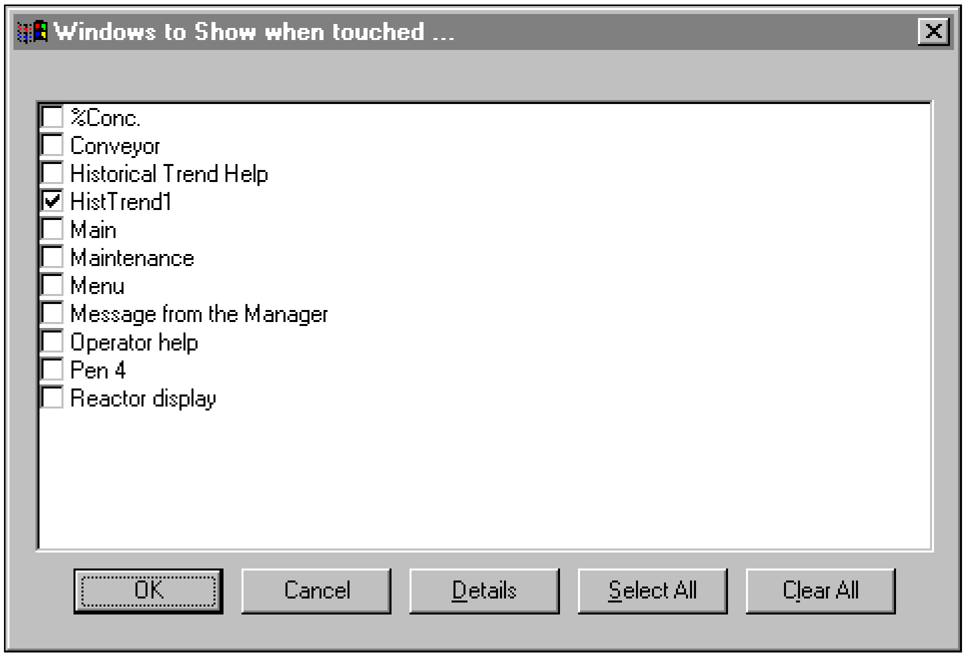
➢ **To create a show (or hide) window touch pushbutton link:**

Note The **Show Window** and **Hide Window** links are created in the same way. This procedure describes the **Show Window** link.

1. Double-click the object or select it, on the **Special** menu, click **Animation Links**. The link selection dialog box will appear.

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Touch Pushbutton** section, click **Show Window**. The **Windows to Show when touched** dialog box appears:



- [ ] %Conc.
- [ ] Conveyor
- [ ] Historical Trend Help
- [x] HistTrend1
- [ ] Main
- [ ] Maintenance
- [ ] Menu
- [ ] Message from the Manager
- [ ] Operator help
- [ ] Pen 4
- [ ] Reactor display

3. Select the windows that you want 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. You can now attach another link to the object if desired.

When showing more than one window where one of the windows is a **Replace** type, if it intersects any other windows, they will close before they are displayed (giving the illusion that your **Show Window** animation link is not working).

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 will appear and you can change the type. (You cannot change the window's type if WindowViewer is running.)

For more information on window properties, see Chapter 2 - Using WindowMaker.

Creating Display Links

You use the various **Display Links** to provide output to the operator. There are eight types of display links that you can create:

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

The following sections describe how to create each of the display links.

Creating Color Links

You use color links to animate the **Line Color**, **Fill Color**, and **Text Color** attributes of an object.

Note You must use solid colors for line and text color links. If you select a dithered (mixed) color, WindowMaker, by default, will select the closest solid color. To avoid dithered colors, your video card must have at least 2MB and your color depth settings must be higher than 256 colors, such as 32K or 65K (sometimes called "high color".)

For more information on the color palette, see "Applying Color Links."

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:

Color Link	Description
Discrete	Used to control the fill, line and text colors attributes of an object or symbol that is linked to the value of a discrete expression.
Analog	The 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	The text, line, and fill color of an object can all 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 it is used with an analog tagname, it responds to any alarm condition of the tagname.
Analog Alarm	The text, line, and fill color of an object can all 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 as well as a separate color for each alarm condition defined for the tagname.

Note Objects will not go into an alarm state when using an Analog Alarm animation link while the animation lik is using a remote tagname that is accessing tagname information from an uncoverted applicated created prior to InTouch Version 7.0.

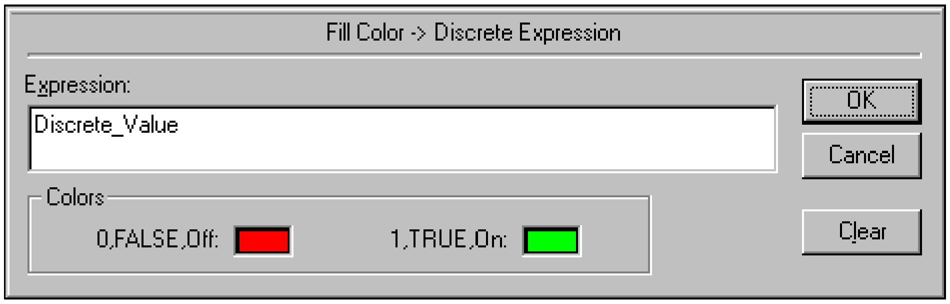
➢ **To create a discrete fill color link:**

Note All of the **Line Color** and **Text Color** links are created in the same way as the **Fill Color** links. The following procedure describes creating a **Fill Color** link.

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Fill Color** (**Line Color** or **Text Color**) section, click **Discrete**. The **Fill Color -> Discrete Expression** dialog box appears:



Fill Color -> Discrete Expression

Expression:
Discrete_Value

3. In the **Expression** box, type a discrete tagname or an expression that equates to true or false.

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then 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 you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. In the **Colors** group, click each color box to open the color palette. Click the color in the color palette that you want to use for each tagname state.

   Note You must use solid colors for line and text color links. If you select a dithered (mixed) color, WindowMaker, by default, will select the closest solid color. To avoid dithered colors, your video card must have at least 2MB and your color depth settings must be higher than 256 colors, such as 32K or 65K (sometimes called "high color".)

   For more information on the color palette, see "Applying Color Links."

5. 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.

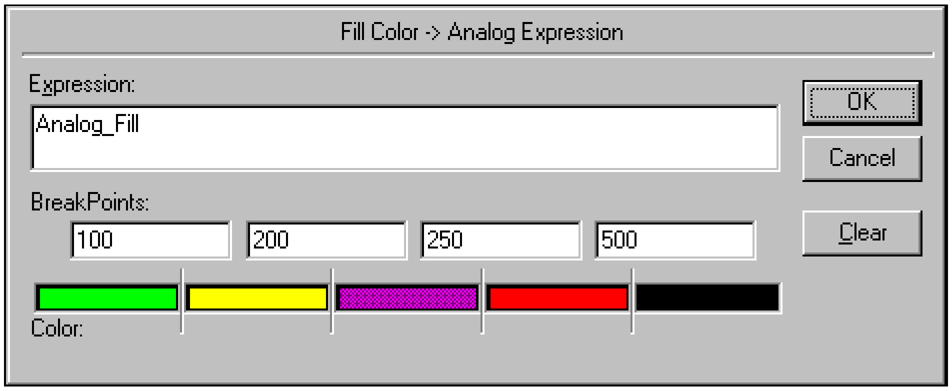
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create an analog expression color link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Fill Color** (**Line Color** or **Text Color**) section, click **Analog**. The **Fill Color -> Analog Expression** dialog box appears:



Expression: Analog_Fill

BreakPoints: 100 | 200 | 250 | 500

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

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

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

   You do not have to use four different values. For example, if you only want the object to change color three times, type three values then use the same color for the third and fourth values.

5. In the **Colors** group, click each color box to open the color palette. Click the color in the color palette that you want to use for each breakpoint.

   Note You must use solid colors for line and text color links. If you select a dithered (mixed) color, WindowMaker, by default, will select the closest solid color. To avoid dithered colors, your video card must have at least 2MB and your color depth settings must be higher than 256 colors, such as 32K or 65K (sometimes called "high color".)

   For more information on the color palette, see "Applying Color Links."

6. 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.

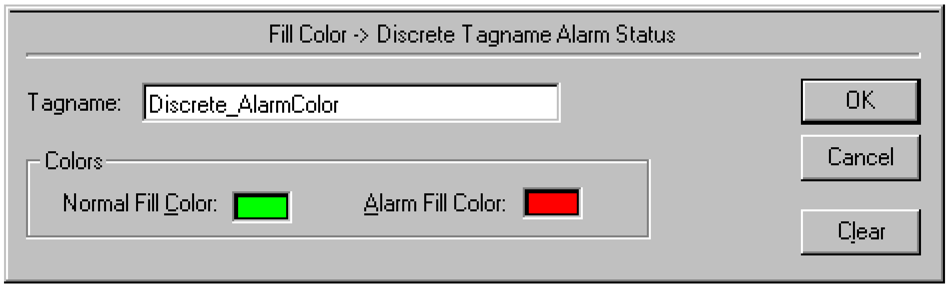
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create a discrete alarm status color link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Fill Color** (**Line Color** or **Text Color**) section, click **Discrete Alarm**. The **Fill Color -> Discrete Tagname Alarm Status** dialog box appears:



Tagname: Discrete_AlarmColor

3. In the **Tagname** box, type the discrete tagname whose alarm status you want associated with the object.

   Right-click the **Tagname** box, to access the commands that you can apply to the selected text.

4. In the **Colors** group, click each color box to open the color palette. Click the color in the color palette that you want to use for each color state.

   Note You must use solid colors for line and text color links. If you select a dithered (mixed) color, WindowMaker, by default, will select the closest solid color. To avoid dithered colors, your video card must have at least 2MB and your color depth settings must be higher than 256 colors, such as 32K or 65K (sometimes called "high color".)

   For more information on the color palette, see "Applying Color Links."

5. 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.

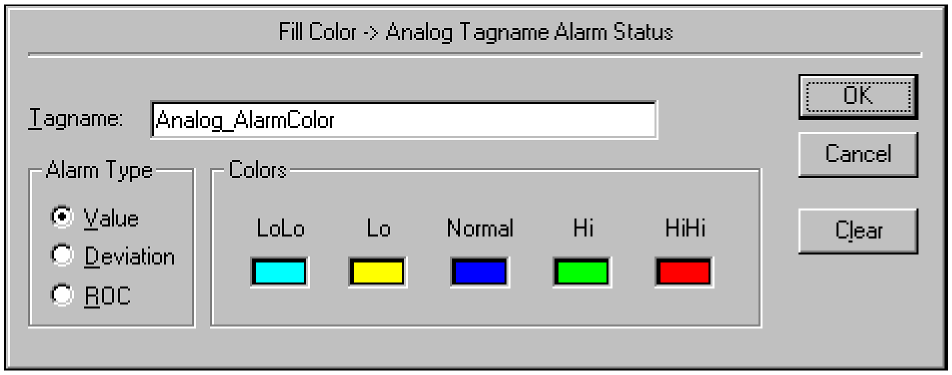
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create an analog alarm status color link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Fill Color** (**Line Color** or **Text Color**) section, click **Analog Alarm**. The **Fill Color -> Analog Tagname Alarm Status** dialog box appears:



Tagname: Analog_AlarmColor

Alarm Type: [✓ Value] Deviation ROC

3. In the **Tagname** box, type the analog (integer or real) tagname whose alarm status you want associated with the object.

   Right-click the **Tagname** box, to access the commands that you can apply to the selected text.

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

   Value Alarm - You can select up to five different colors depending on the status of the value alarms defined for the tagname (example above).

   Deviation - You can select up to three different colors depending on the status of the deviation alarms defined for the tagname (example above).

   ROC (Rate-of-Change) - You can select two different colors depending on the status of the rate-of-change alarm defined for the tagname.

5. In the **Colors** group, click each color box to open the color palette. Click the color in the color palette that you want to use for each color state.

   Note You must use solid colors for line and text color links. If you select a dithered (mixed) color, WindowMaker, by default, will select the closest solid color. To avoid dithered colors, your video card must have at least 2MB and your color depth settings must be higher than 256 colors, such as 32K or 65K (sometimes called "high color".)

   For more information on the color palette, see "Applying Color Links."

6. 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.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Object Size Links

You use **Object Size** links 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.

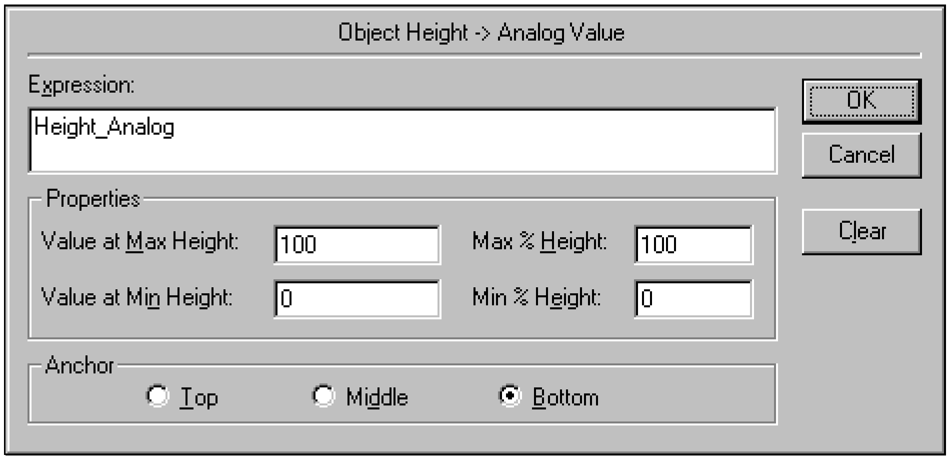
Note The height and width links are created the same way. This procedure describes the **Height** link.

➢ **To create a height (or width) link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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



Expression: Height_Analog

Value at Max Height: 100 Max % Height: 100

Value at Min Height: 0 Min % Height: 0

Anchor: Top [✓ Bottom] Middle

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

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

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

6. In the **Max % Height** box, type 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** field.

7. In the **Min % Height** box, type 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** field.

🖱 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. You can now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Location Links

You use **Location Links** 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.

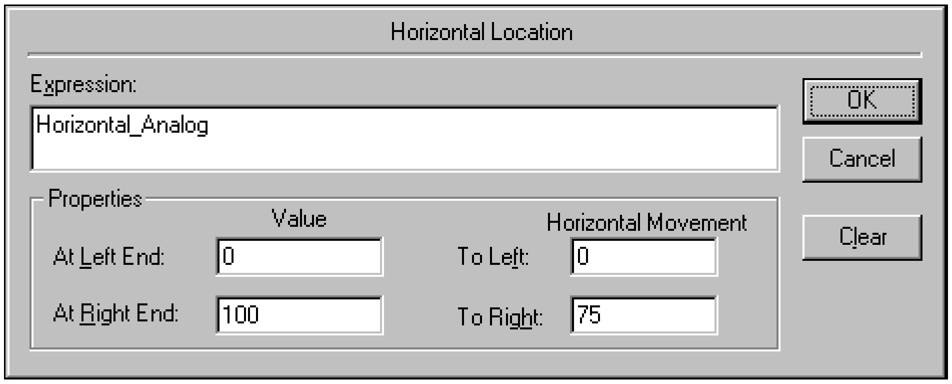
Note The **Horizontal** and **Vertical Location** links are created the same way. This procedure describes the **Horizontal Location** link.

➢ **To create a horizontal location link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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



Horizontal Location

Expression: Horizontal_Analog

Properties

	Value		Horizontal Movement
At Left End:	0	To Left:	0
At Right End:	100	To Right:	75

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

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

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

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

6. In the **To Left** box, type 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, type 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. You can now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Percent Fill Links

You use **Percent Fill Links** to provide the ability 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.

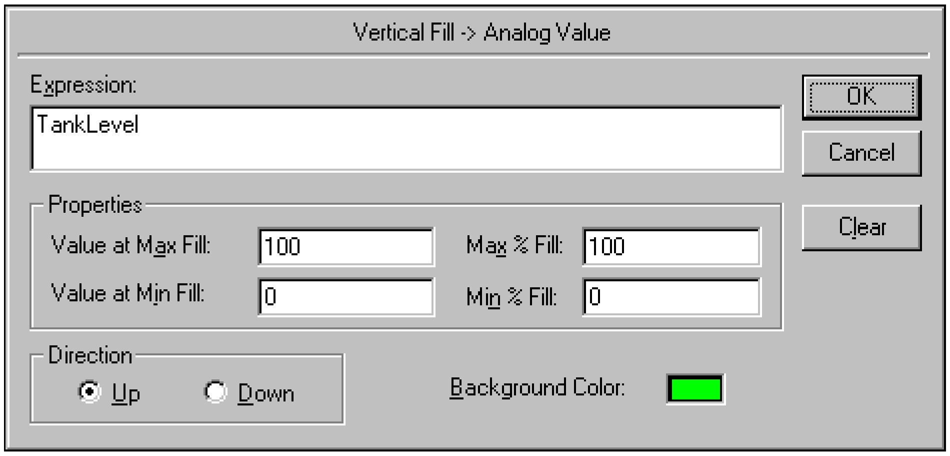
Note The **Horizontal** and **Vertical Percent Fill** links are created the same way. This procedure describes the **Vertical Percent Fill** link.

➢ **To create a vertical percent fill link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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


Vertical Fill -> Analog Value

Expression: TankLevel

Properties
Value at Max Fill: 100
Max % Fill: 100
Value at Min Fill: 0
Min % Fill: 0

Direction: [✓ Up] Down

Background Color: 🟩

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

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

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

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

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

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

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

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

8. Select the **Direction** that you want the object 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. In the **Background Color** box to open the color palette. Click on the desired color. The color palette will be removed from the screen.

   🖰 This **Background Color** selection is for the color of the **unfilled** portion of the object. The actual fill color is the color that you select for the object when you draw it. If you link both **Vertical Percent Fill** and **Horizontal Percent Fill** links, to the same object, the last color you select 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. You can now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Miscellaneous Links

There are four type of miscellaneous links.

Misc Link	Description
Visibility	Use 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.

Often used as part of a security strategy.

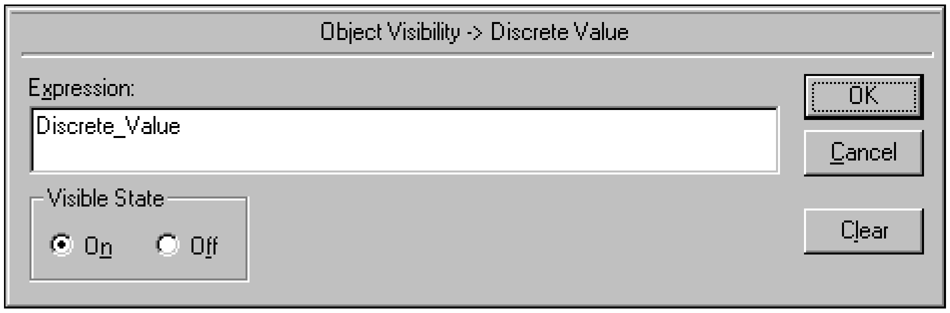
For more information on applying security to your application, see Chapter 3 - Building a Distributed Application.

➢ **To create a visibility link**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Miscellaneous** section, click **Visibility**. The **Object Visibility -> Discrete Value** dialog box appears:



Object Visibility -> Discrete Value

Expression: Discrete_Value

Visible State: [✓ On] Off

3. In the **Expression** box, type a discrete tagname or an expression that equates to a discrete value.

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Discrete expressions can also contain analog tagnames, for example, TankLevel >= 75. In this example, when the value of the tagname, TankLevel is greater than or equal to 75, the object will become visible in the window.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. Select the **Visible State** for the object. If you select **On**, the object will be invisible when the value of the expression is true. If you select **Off**, 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. You can now attach another link to the object if desired.

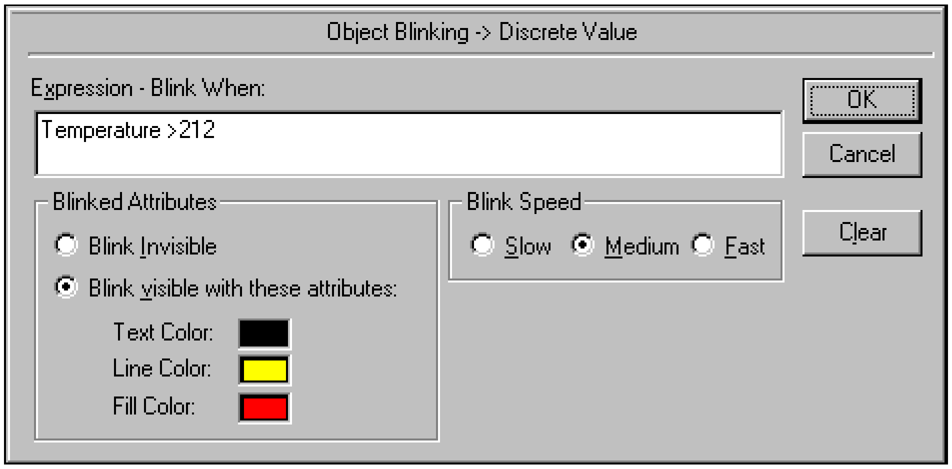
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create a blink link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Miscellaneous** section, click **Blink**. The **Object Blinking -> Discrete Value** dialog box appears:



Object Blinking -> Discrete Value

Expression - Blink When: Temperature >212

Blinked Attributes: ○ Blink Invisible [✓ ● Blink visible with these attributes:]

Text Color: ■
Line Color: 🟨
Fill Color: 🟥

Blink Speed: ○ Slow [✓ ● Medium] ○ Fast

3. In the **Expression - Blink When** box, type a discrete tagname or an expression that equates to a discrete value.

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then 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 object will blink.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. Select the **Blinked Attributes** that you want 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 changing of the color attributes selected creates the blinking effect.

   Click the **Text Color**, **Line Color** and **Fill Color** boxes to open the color palette. Click on the desired color. The color palette will be removed from the screen.

   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** that you want to use for the blinking speed of the object.

Note To configure the blink speed for **Slow**, **Medium**, and **Fast**, on the **Special** menu, point to **Configure** and then click **WindowViewer**. The **WindowViewer General** property sheet will appear. In the **Blink Frequency** group, type the number of milliseconds you want to use for the speeds.

Any changes you make to these settings are global and will effect the blink speeds of all blink links throughout your application.

6. 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.

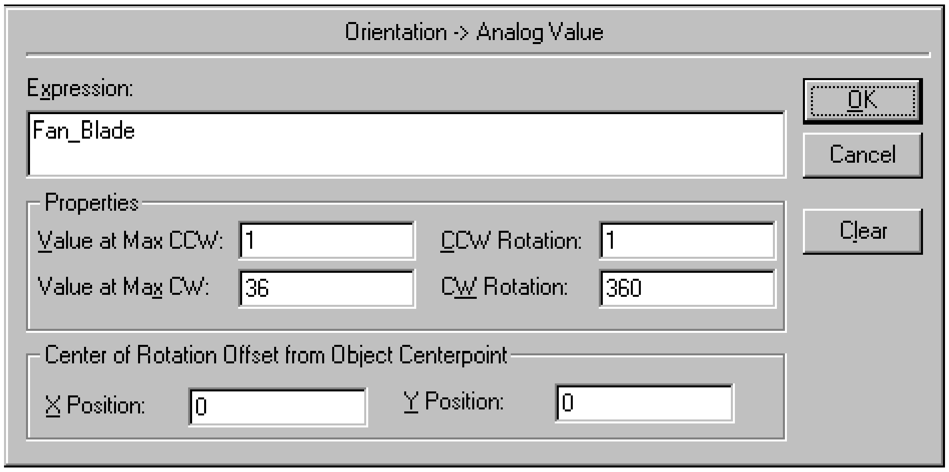
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create an orientation link**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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


Orientation -> Analog Value

Expression: Fan_Blade

Properties
Value at Max CCW: 1
CCW Rotation: 1
Value at Max CW: 36
CW Rotation: 360

Center of Rotation Offset from Object Centerpoint
X Position: 0
Y Position: 0

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

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

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

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

   The object is rotated clockwise or counter-clockwise based on its original drawn position when drawn in WindowMaker.

   To force an object like text to a specific angle, simply set **Value at Max CCW** to 360 and **Value at Max CW** to 0, **CCW Rotation** to 360 and **CW Rotation** to 0, and then in the **Expression** box, type the angle value such as 90 (for 90 degrees). Remember, without a tagname, this expression will never change and the object will always hold its 90 degree position.

   Note Text can be set in WindowMaker, but not rotated in WindowViewer on a tagname value.

8. In the **X Position** box, type 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, type 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. You can now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

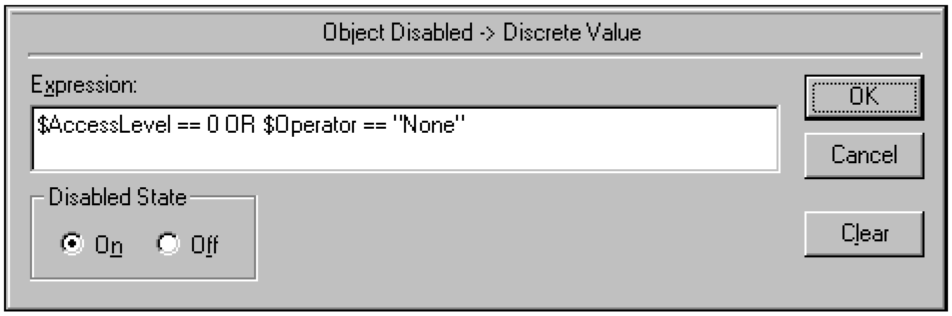
➢ **To create a disable link:**

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

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Miscellaneous** section, click **Disable**. The **Object Disabled -> Discrete Value** dialog box appears:



Expression:
$AccessLevel == 0 OR $Operator == "None"

Disabled State: [✓ On] Off

3. In the **Expression** box, type 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.

Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then 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 object will be disabled.

Right-click the **Expression** box, to access the commands that you can apply to the selected text.

For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. Select the **Disabled State** that will turn off or on functionality of the object 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. You can now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

Creating Value Display Links

Value Display Links provide the ability to use a text object to display the value of a discrete, analog, or string tagname. There are three types:

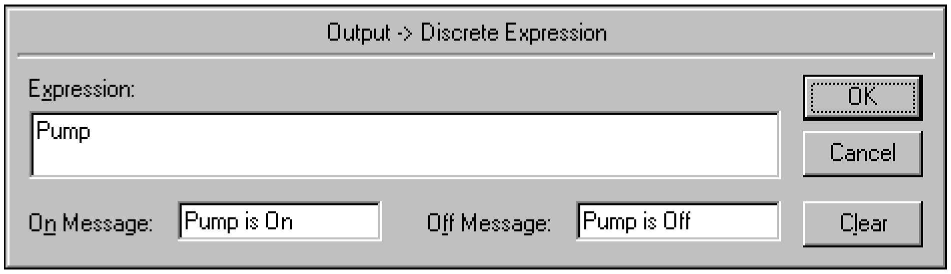
Value Display Type	Description
Discrete	Uses the value of a discrete expression to display an On or Off user defined message in a text object.
Analog	Displays the value of an analog expression in a text object.
String	Displays the value of a string expression in a text object.

➢ **To create a discrete value display link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

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


Output -> Discrete Expression

Expression: Pump

On Message: Pump is On

Off Message: Pump is Off

3. In the **Expression** box, type a discrete tagname or an expression that equates to a discrete value.

   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 appropriate message will be displayed.

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. In the **On Message** box, type the message that you want displayed when the value of the discrete expression equals 1 (True, On, Yes).

5. In the **Off Message** box, type the message that you want 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 now attach another link to the object if desired.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

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 type 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.

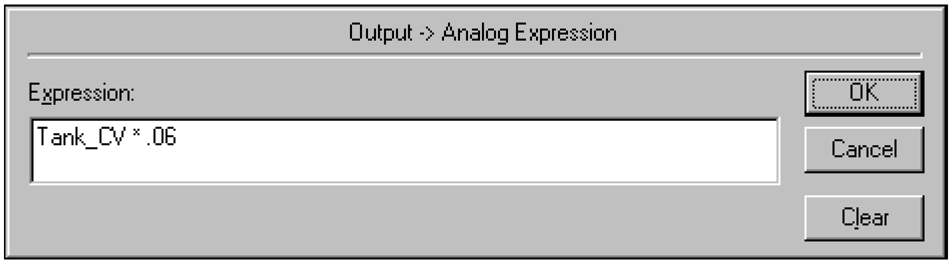
➢ **To create an analog value display link:**

For more information on formatting analog display objects, see Chapter 2 - Using WindowMaker.

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Value Display** section, click **Analog**. The **Output -> Analog Expression** dialog box appears:



Tank_CV * .06

3. In the **Expression** box, type 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.)

   Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. 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.

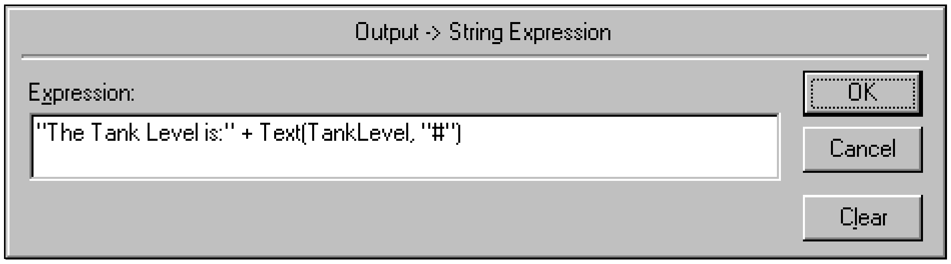
Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

➢ **To create a string value display link:**

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

   To quickly access the animation link selection dialog box, right-click the object, and then click **Animation Links**.

2. In the **Value Display** section, click **String**. The **Output -> String Expression** dialog box appears:



Output -> String Expression

Expression:

"The Tank Level is:" + Text(TankLevel, "#")

3. In the **Expression** box, type 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.

   In Up to 256 characters may be typed for your expression. If you need to use a larger expression, create a QuickFunction then call it in your expression.

   Right-click the **Expression** box, to access the commands that you can apply to the selected text.

   For more information on QuickFunctions, see Chapter 6 - Creating InTouch QuickScripts.

4. 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.

Note If the tagname you entered is not defined in your Tagname Dictionary (except for remote tagnames) you will be prompted to define it now.

C H A P T E R 6

Creating QuickScripts in InTouch

InTouch scripting is one of the most powerful features of an InTouch application. The 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, and so on.

QuickFunctions are scripts that you create that can be called from other scripts and animation link expressions. The reused code is stored in one script and in one location, thereby supporting update of all script instances with one edit session.

By using scripts, a wide variety of customized and automated system functions can be created.

Contents

InTouch QuickScripts

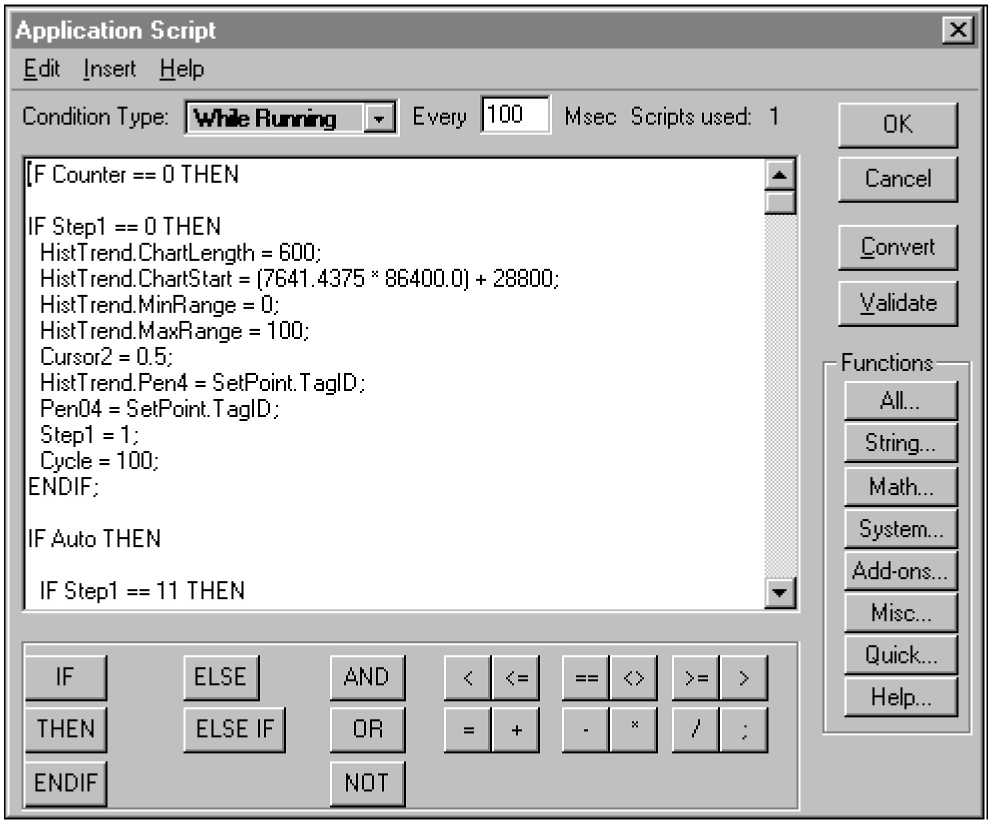
All InTouch QuickScripts are event driven. The event may be a data change, condition, mouse click, timer, and so on. The order of processing is application specific. While it may appear that there is some inherent order in the way multiple scripts initiated by the same event are scheduled, there is no 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 that you can create:

Script Type	Description
Application	Linked to the entire application.
Window	Linked to a specific window.
Key	Linked to a specific key or key combination on the keyboard.
Condition	Linked to a discrete tagname or expression.
Data Change	Linked to a tagname and/or **tagname.field** only.
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.
Action Pushbutton	Associated with an object that you link to an **Touch Link - Action Pushbutton**.
ActiveX Event	Execute ActiveX control events in runtime.

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 chapter.



Condition Type: While Running Every 100 Msec Scripts used: 1

```
IF Counter == 0 THEN

IF Step1 == 0 THEN
  HistTrend.ChartLength = 600;
  HistTrend.ChartStart = (7641.4375 * 86400.0) + 28800;
  HistTrend.MinRange = 0;
  HistTrend.MaxRange = 100;
  Cursor2 = 0.5;
  HistTrend.Pen4 = SetPoint.TagID;
  Pen04 = SetPoint.TagID;
  Step1 = 1;
  Cycle = 100;
ENDIF;

IF Auto THEN

  IF Step1 == 11 THEN
```

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. The procedures that are unique to a script type are described in that script type's respective section later in this chapter.

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

➢ **To indent/unindent text in a script:**

Position the cursor at the beginning of the line that you want to indent, and then press the TAB key. To remove the indent, press hold down the SHIFT key while you press the TAB key.

➢ **To create a new script:**

On the **Script** menu, click **New**.

Note The **Script** menu does not exist for **Application, Window Scripts** or **Touch Pushbutton Action** scripts.

➢ **To delete a script from your application:**

Select the text you want to delete, and then on the **Script** menu, click **Erase**. The script is deleted from your application entirely.

Note Deleted text is not written to the Windows Clipboard.

➢ **To undo your last action:**

On the **Edit** menu, click **Undo**. Your last editing operation, a paste, for example, is reversed.

🖱 To quickly execute this command, right-click the script window, and then click **Undo**. The **Undo** command will not be active unless you have performed an action that can be reversed.

➢ **To select the entire script:**

On the **Edit** menu, click **Select All**. The entire script is selected.

🖱 To quickly execute the command, right-click the script window, and then click **Select All**. You can now copy, cut or delete the entire script.

➢ **To cut selected text from a script:**

Select the text you want to remove, and then on the **Edit** menu, click **Cut**. The cut text is deleted from the script and copied to the Windows Clipboard. You can now paste the cut text at another location in this script or you can paste it in another script.

🖱 To quickly perform this command, right-click the script window, and then click **Cut**. The **Cut** command will not be active unless you have selected text to cut.

➢ **To copy selected text from a script:**

Select the text to be removed, and then on the **Edit** menu, click **Copy**. The copied text is written to the Windows Clipboard. You can now paste the copied text at another location in this script, or you can paste it in another script.

🖱 To quickly perform this command, right-click the script window, and then click **Copy**. The **Copy** command will not be active unless you have selected text to copy.

Note When you cut or copy text, it is automatically written to the Windows Clipboard. This information remains on the Clipboard until you perform a subsequent cut or copy command.

➢ **To paste text into a script:**

On the **Edit** menu, click **Paste**. The contents of the Windows Clipboard is pasted into your script at the cursor location.

🖱 To quickly execute the **Paste** command, right-click the script window, and then click **Paste**. (The **Paste** command will not be active if there is nothing in the Windows Clipboard to paste.)

➢ **To clear the text in a script:**

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

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

➢ **To insert a function into a script:**

1. On the **Insert** menu, point to **Functions**, and then click the name of the function category. The respective **Choose function** dialog box will appear.
2. Click the function that you want to use. 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). 🖱 You can also click the **All** button in the **Functions** group to access these functions.
String	The **Choose function** dialog box appears displaying all available string functions. 🖱 You can also click the **String** button in the **Functions** group to access these functions.
Math	The **Choose function** dialog box appears displaying all available mathematical functions. 🖱 You can also click the **Math** button in the **Functions** group to access these 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, and so on. 🖱 You can also click the **System** button in the **Functions** group to access these functions.
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). 🖱 You can also click the **Add-ons** button in the **Functions** group to access these functions.
Misc	The **Choose function** dialog box appears displaying all available miscellaneous functions. For example, the functions for alarms, historical trending, windows controls, ActiveX controls, and so on. 🖱 You can also click the **Misc** button in the **Functions** group to access these functions.
Help	The **Choose function to Obtain Help for** dialog box appears listing all available functions. Click a function to open its respective Help topic. 🖱 You can also click the **Help** button in the **Functions** group to access these functions.

Quick Functions The **Choose function** dialog box appears listing the names of all the QuickFunctions available for calling from the current script.

You can also click the **Quick** button in the **Functions** group to access all QuickFunctions.

For more information on the individual script functions, see "Script Functions."

➢ **To insert a tagname into a script:**

1. On the **Insert** menu, click **Tagname**. The Tag Browser will appear in the unlimited selection mode.

Note The tagnames defined in the last tag source accessed through the Tag Browser will be displayed. To change the tag source, click the **Tag Source** arrow and select a different tag source in the list.

Click the Define Tag Sources button to add or remove a tag source from the **Tag Source** list.

---

2. Double-click the tagname you want to use or select it, and then click **OK**. The Tag Browser will close and the tagname will automatically be inserted into your Quick Script 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, type the tagname in the QuickScript window, and then double-click it.

   For more information on the Tag Browser, see Chapter 4 - Tagname Dictionary.

➢ **To insert a tagname .field into a script:**

1. On the **Insert** menu, click **Tagname**. The Tag Browser will appear in the unlimited selection mode.

Note The tagnames defined in the last tag source accessed through the Tag Browser will be displayed. To change the tag source, click the **Tag Source** arrow and select a different tag source in the list.

Click the Define Tag Sources button to add or remove a tag source from the **Tag Source** list.

---

2. Select the tagname that you want to use, and then click the **Dot Field** arrow. Select the **.field** that you want 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.

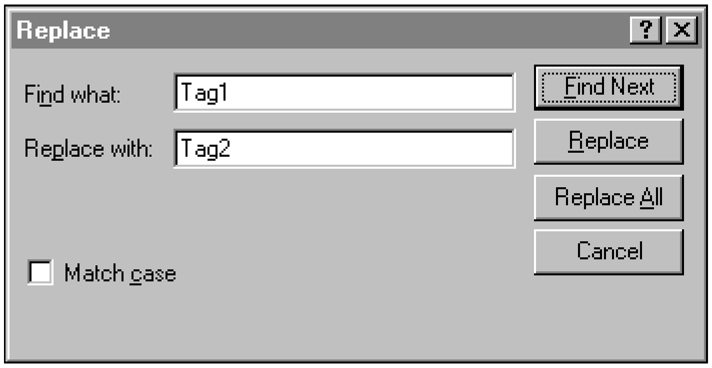
   To quickly insert a tagname **.field**, type the tagname followed by a period (.), and then double-click to the right of the period. The **Choose field name** dialog box will appear. Click the **.field** that you want to use. The dialog box will close and the selected **.field** will automatically be inserted into your QuickScript at the cursor location.

   For more information on the Tag Browser, see Chapter 4 - Tagname Dictionary.

   For more information on the Tagname **.fields**, see your *InTouch Reference Guide*.

➢ **To find or replace a tagname in a script:**

4. On the **Edit** menu, click **Find**. The **Replace** dialog box appears:



5. In the **Find what** box, type the tagname that you want to find (or replace), and then click **Find Next**.

6. In the **Replace with** box, type the new tagname that you want to use to replace the old tagname then click **Replace** or **Replace All**.

   If you only want to replace certain instances of an old tagname, click **Find Next**. **InTouch** will begin searching your script for the old tagname. When the old tagname is found, it will be highlighted. Click **Replace** to replace it with the new tagname or click **Find Next** to skip it and continue searching. If you want to replace all occurrences of a specific tagname, click **Replace All** at any time during the search.

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

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

➢ **To insert a window name into a script:**

1. On the **Insert** menu, click **Window**. The **Window Name to Insert** dialog box will appear displaying the names of all the windows in your application.

2. Click the window name that you want to use. The dialog box will close and the window name will automatically be inserted into your script at the cursor location.

➢ **To validate a script:**

Click **Validate** to verify that your script syntax is accurate at any time while you are writing the script.

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

For more information on script errors, see "Script Editor Error Messages."

➢ **To save a script:**

If you are writing multiple scripts, after you have finished writing one, you can click **Save** to save it, and then on the **Script** menu, click **New** to write another new script.

Note Application and **Window** scripts don't support this function. Otherwise, the save function is automatically performed when you click the **OK** button.

If the system encounters errors when saving your script, a corresponding error message box will appear.

- **To restore a script:**

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

Note You cannot restore a script once you have saved it. **Application** and **Window** scripts don't support this function.

- **To exit the script editor:**

On the **Script** menu, click **Exit**. 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.

Note When you select **Exit**, **OK** or you click the **X** button in the upper right hand corner of the dialog box, the system automatically verifies your script for accuracy.

For more information on script errors, see "Script Editor Error Messages."

- **To specify a script's execution frequency:**

In the **While *Running/Showing/Down* Every 0 Milliseconds** boxes, type the number of milliseconds that you want to elapse before the script executes.

When you create 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.

Note WindowViewer will make every attempt possible to run these types of scripts as fast as the time you specify. However, the performance cannot be guaranteed. Also scripts can never run any faster than the **Tick Interval** setting that you specify when you configure WindowViewer's properties.

Scripts cannot execute faster than every 10 milliseconds on the Windows NT operating system or every 50 milliseconds on Windows 95.

For more information on the **Tick Interval** setting, see Chapter 2 - Using WindowMaker.

Application Scripts

The Application Scripts are linked to the entire application. You can use application scripts to start other applications, create process simulations, calculate variables, and so on. There are three types of Application Scripts that you can apply to an application:

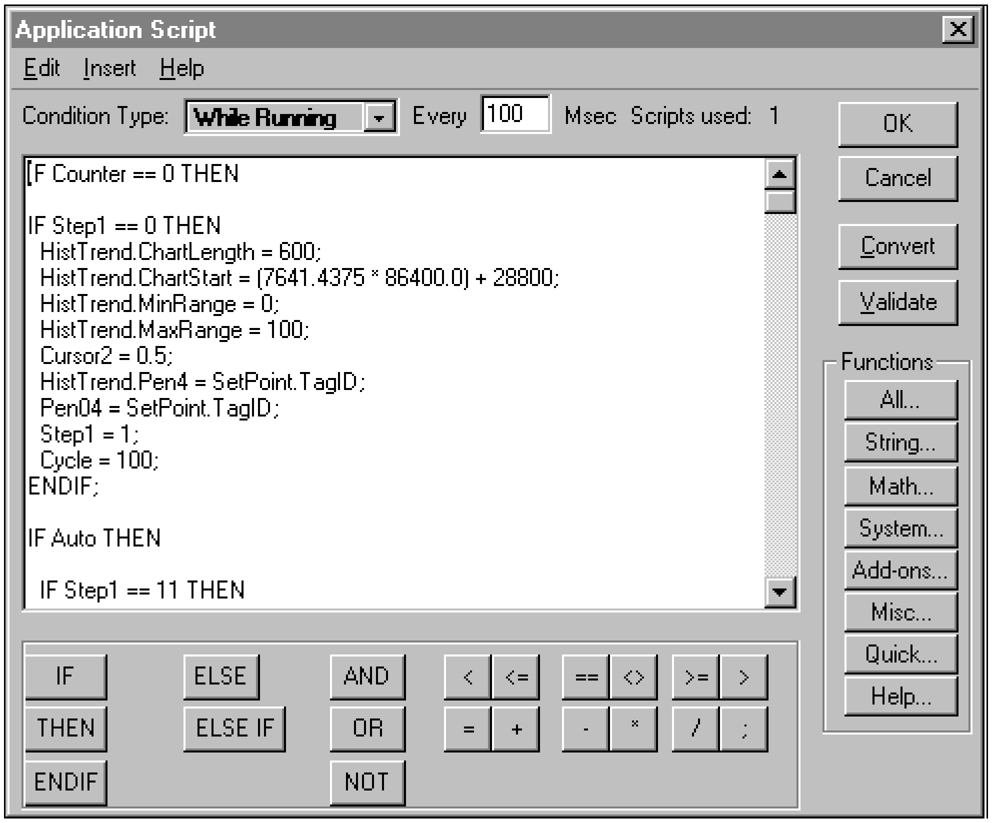
On Startup	Executes one time when the application is initially started up.
While Running	Executes continuously at the specified frequency while the application is running.
On Shutdown	Executes one time when the application is exited.

Note The Application **On Startup** script executes before any window opens or any runtime initialization occurs. Therefore, you cannot refer to ActiveX methods, properties or events in an Application **On Startup** script. Similarly, I/O communications are initialized after the Application **On Startup** script executes. Therefore, you cannot refer to I/O type tagnames or remote tagname references in an Application **On Startup** script. In addition, I/O type tagnames and remote tagname references will not update in an Application **On Shutdown** script. Also, you cannot use an Application **On Shutdown** script to startup other applications.

➢ **To access the Application Script editor:**

On the **Special** menu, point to **Scripts**, and then click **Application Scripts,** or in the Application Explorer under **Scripts**, double-click **Application**. The **Application Script** editor appears:

- In the Application Explorer under **Scripts**, you can also right-click **Application**, and then click **Open**.



Condition Type: While Running Every 100 Msec Scripts used: 1

```
IF Counter == 0 THEN

IF Step1 == 0 THEN
  HistTrend.ChartLength = 600;
  HistTrend.ChartStart = (7641.4375 * 86400.0) + 28800;
  HistTrend.MinRange = 0;
  HistTrend.MaxRange = 100;
  Cursor2 = 0.5;
  HistTrend.Pen4 = SetPoint.TagID;
  Pen04 = SetPoint.TagID;
  Step1 = 1;
  Cycle = 100;
ENDIF;

IF Auto THEN

  IF Step1 == 11 THEN
```

When you select a **While Running** script, the **Every 0 Milliseconds** box becomes active. In the box, type the number of milliseconds that you want to elapse before the script executes. If you want the script to execute immediately, create an identical **On Startup** script. However, 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 you can apply to a window:

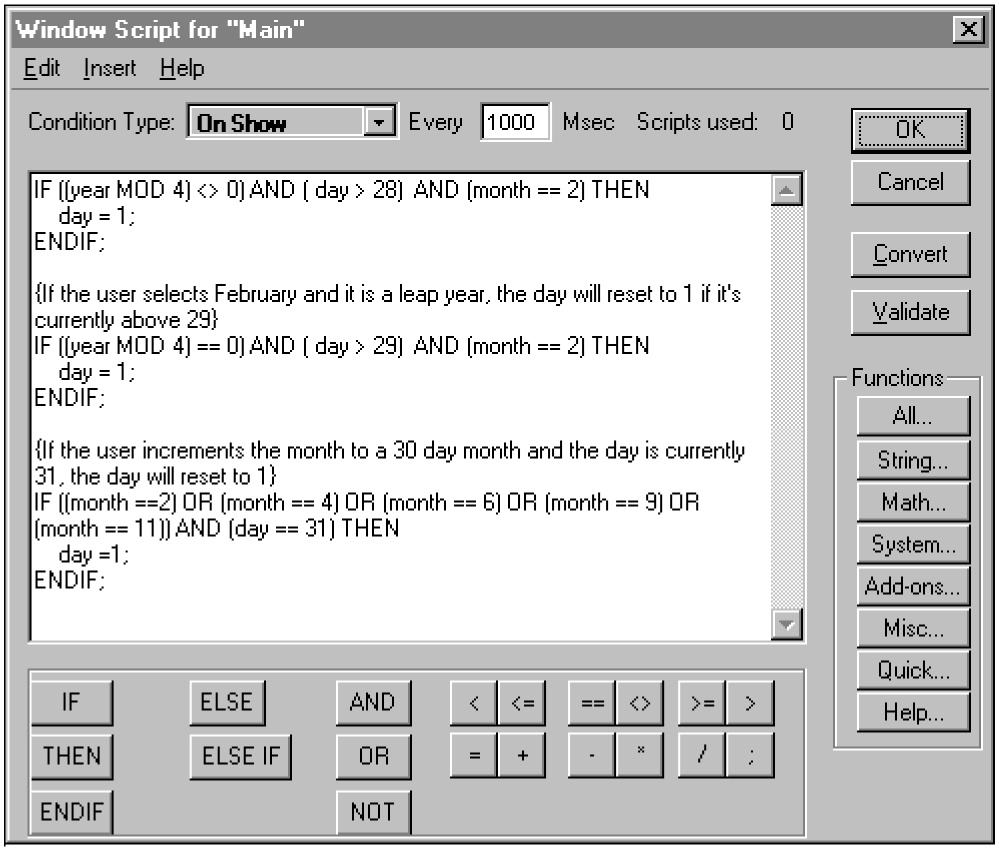
On Show	Executes one time when the window is initially shown.
While Showing	Executes continuously at the specified frequency while the window is showing.
On Hide	Executes one time when the window is hidden.

Note If you attach a Window Script to the active window and then you create a new window, the scripts from the active window can be copied to the new window. A message dialog box will appear asking if you want to copy the script(s).

➢ **To create Window Scripts:**

On the **Special** menu, point to **Scripts**, and then click **Window Scripts**. The **Window Script** editor appears:

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**. You can also right-click a blank area of an open window, and then click **Window Scripts**. If a script exists for the selected window, it will be displayed.



Condition Type: On Show Every 1000 Msec Scripts used: 0

```
IF ((year MOD 4) <> 0) AND ( day > 28)  AND (month == 2) THEN
   day = 1;
ENDIF;

{If the user selects February and it is a leap year, the day will reset to 1 if it's
currently above 29}
IF ((year MOD 4) == 0) AND ( day > 29)  AND (month == 2) THEN
   day = 1;
ENDIF;

{If the user increments the month to a 30 day month and the day is currently
31, the day will reset to 1}
IF ((month ==2) OR (month == 4) OR (month == 6) OR (month == 9) OR
(month == 11)) AND (day == 31) THEN
   day =1;
ENDIF;
```

When you select **While Showing**, the **Every 0 Milliseconds** box becomes active. In the box, type the number of milliseconds that you want to elapse before the script executes. If you want the script to execute immediately, create an identical **On Show** script. However, 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

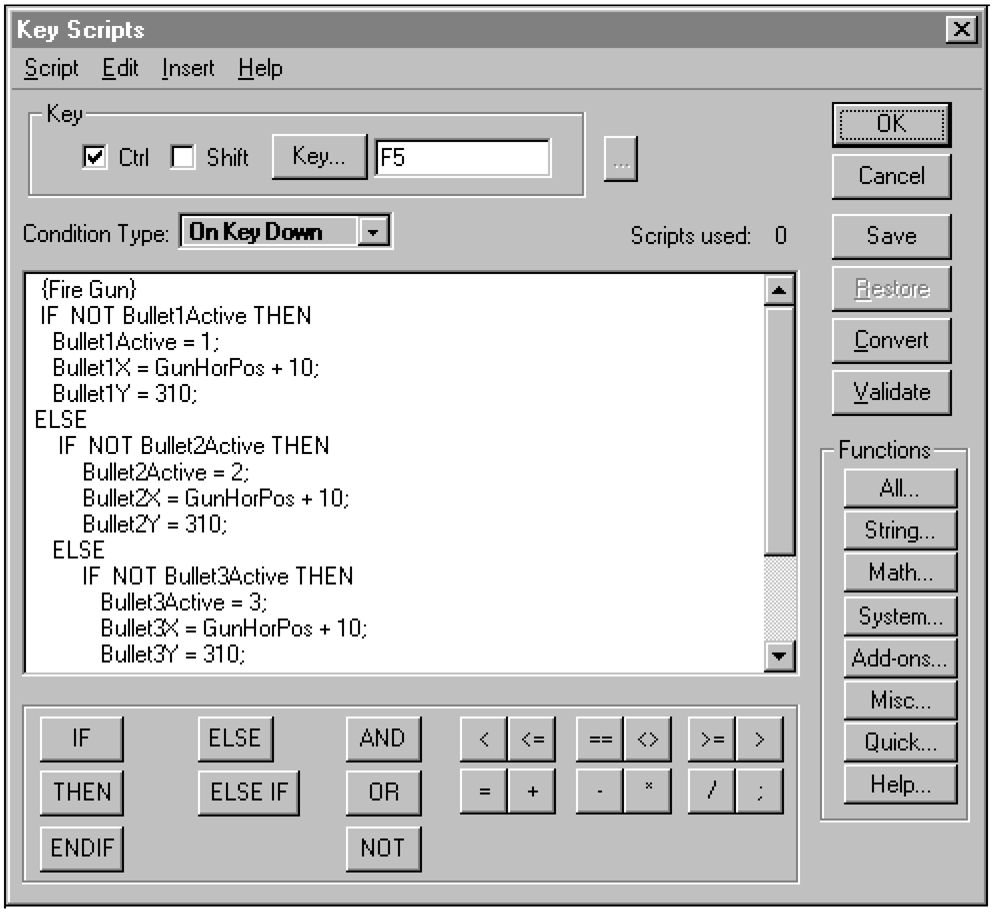
Key Scripts are linked to a specific key or key combination on the keyboard. You can use them to create global keys for the application. For example, returning to a main menu window, logging off the operator, and so on. There are three types of Key Scripts that you can apply to a key:

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.

➢ **To access the Key Script editor:**

On the **Special** menu, point to **Scripts**, and then click **Key Scripts**, or in the Application Explorer under **Scripts**, double-click **Key**. The **Key Script** editor appears:

In the Application Explorer, under **Scripts**, you can also right-click **Key**, and then click **Open**.



Key: ☑ Ctrl ☐ Shift Key... F5

Condition Type: On Key Down Scripts used: 0

```
 {Fire Gun}
 IF  NOT Bullet1Active THEN
   Bullet1Active = 1;
   Bullet1X = GunHorPos + 10;
   Bullet1Y = 310;
ELSE
    IF  NOT Bullet2Active THEN
        Bullet2Active = 2;
        Bullet2X = GunHorPos + 10;
        Bullet2Y = 310;
   ELSE
        IF  NOT Bullet3Active THEN
           Bullet3Active = 3;
           Bullet3X = GunHorPos + 10;
           Bullet3Y = 310;
```

When you select **While Down**, the **Every 0 Milliseconds** box becomes active. In the box, type the number of milliseconds that you want to elapse before the script executes. If you want the script to execute immediately, create an identical **On Key Down** script. However, as long as the condition or event for the **While Down** script is met, the script will repeatedly execute at the specified frequency.

For more information on assigning a key to the script, see "Assigning a Key Equivalent to a Script."

Note The key equivalents used in the local active window's for Touch Pushbutton Action scripts will override any global Key Scripts with the same key equivalents.

Touch Pushbutton Action Scripts

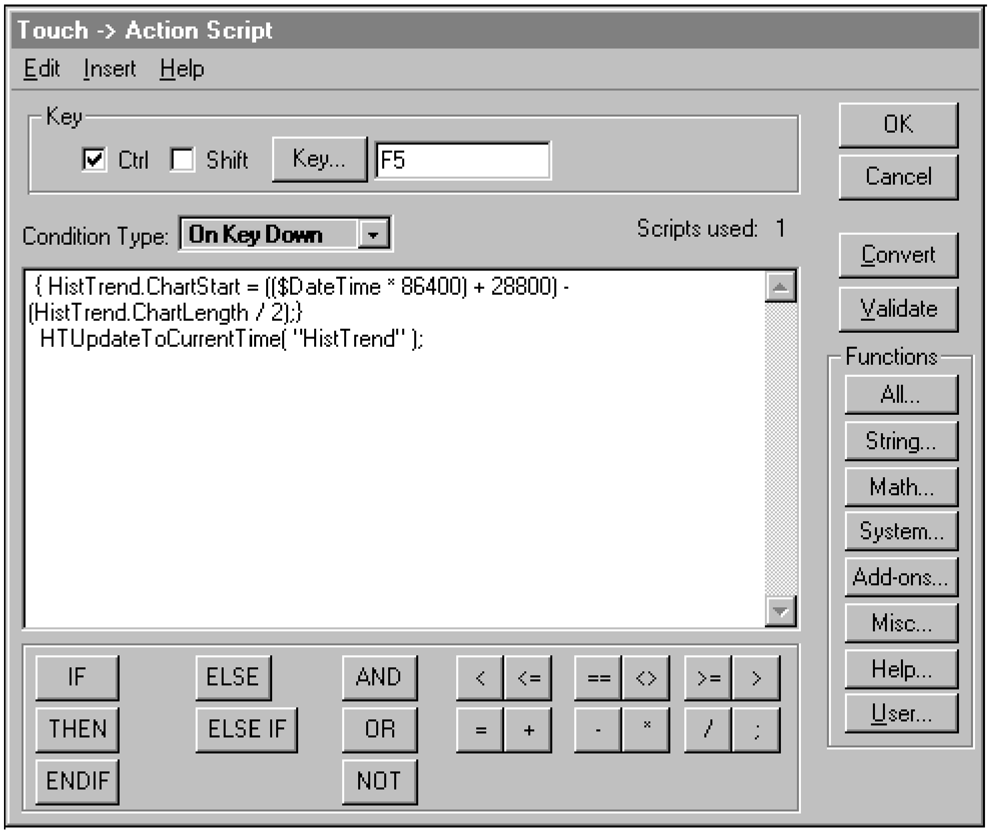
Touch Pushbutton Action Scripts are similar to Key Scripts, except they are associated with an object that you link to a **Touch Link- Action Pushbutton**. (The script editor is accessed through the animation link selection dialog box.) They are executed when the operator clicks or presses the object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

➢ **To create an action pushbutton script:**

1. Draw the object or button that you want to be linked to the script.
2. Double-click the object or select it, and then on the **Special** menu, click **Animation Links**. The animation link selection dialog box will appear.

   🖰 To quickly access the dialog box, right-click the object, and then click **Animation Links**.
3. In the **Touch Pushbutton** section, click **Action**. The **InTouch -> Action Script** editor appears:



```
{ HistTrend.ChartStart = (($DateTime * 86400) + 28800) -
(HistTrend.ChartLength / 2);}
 HTUpdateToCurrentTime( "HistTrend" );
```

When you select **While Down**, the **Every 0 Milliseconds** box becomes active. In the box, type the number of milliseconds that you want to elapse before the script executes. If you want the script to execute immediately, create an identical **On Key Down** script. However, as long as the condition or event for the **While Down** script is met, the script will repeatedly execute at the specified frequency.

For more information on assigning a key to the script, see "Assigning a Key Equivalent to a Script."

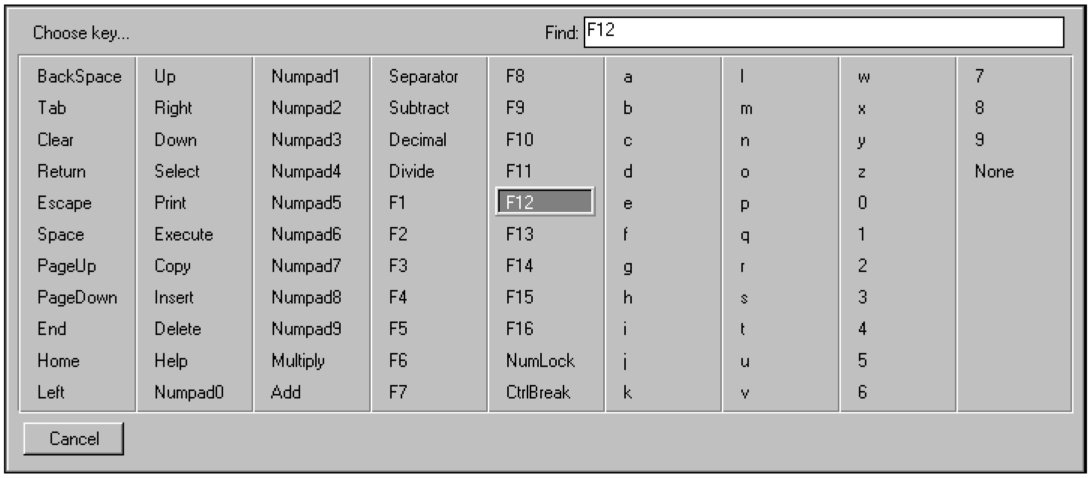
Note The key equivalents used in the local active window's for Touch Pushbutton Action scripts will override any global Key Scripts with the same key equivalents. Also, key equivalents are only active when the window with the object is active.

Assigning a Key Equivalent to a Script

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

➢ **To assign a key to a Key Script:**

1. Select **Ctrl** and/or **Shift**, if you want the operator to have to hold down the CTRL and/or SHIFT keys while pressing the key to execute the script.
2. Click **Key** to select the key you want assigned to the script. The **Choose key** dialog box appears:



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

3. Click the desired key. The dialog box automatically closes and your selection is automatically entered in the **Key** box.

Condition Scripts

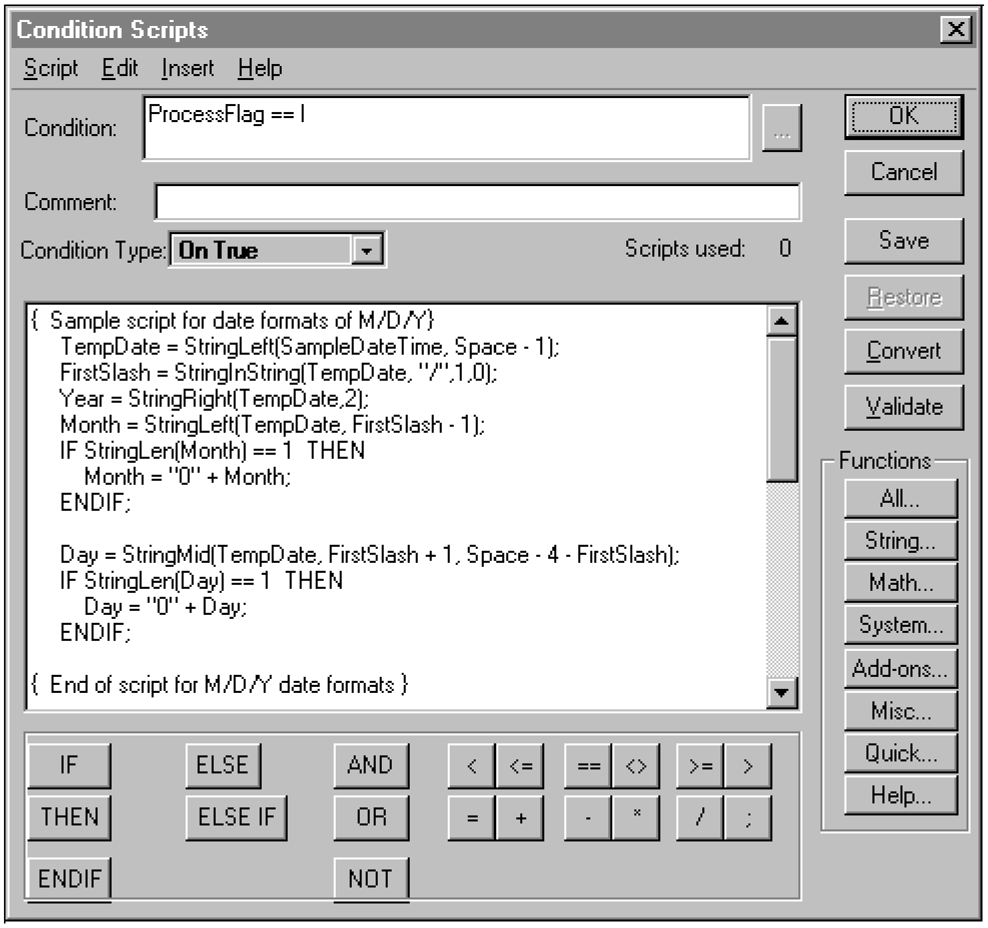
Condition Scripts are linked to a discrete tagname or expression that equates to true or false. You can also use discrete expressions that contain analog tagnames (see example below). There are four types of scripts that you can apply to a condition:

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.

➢ **To access the Condition Script editor:**

1. On the **Special** menu, point to **Scripts**, and then click **Condition Scripts**, or in the Application Explorer under **Scripts**, double-click **Condition**. The **Condition Script** editor appears:

   In the Application Explorer under **Scripts**, you can also right-click **Condition**, and then click **Open**.



Condition: ProcessFlag == I

Comment:

Condition Type: On True

Scripts used: 0

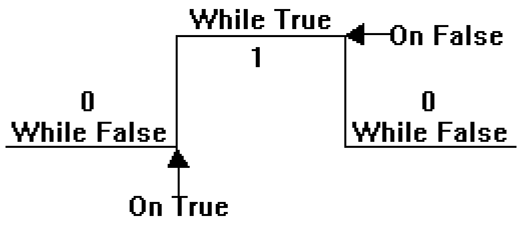
```
{ Sample script for date formats of M/D/Y}
   TempDate = StringLeft(SampleDateTime, Space - 1);
   FirstSlash = StringInString(TempDate, "/",1,0);
   Year = StringRight(TempDate,2);
   Month = StringLeft(TempDate, FirstSlash - 1);
   IF StringLen(Month) == 1  THEN
      Month = "0" + Month;
   ENDIF;

   Day = StringMid(TempDate, FirstSlash + 1, Space - 4 - FirstSlash);
   IF StringLen(Day) == 1  THEN
      Day = "0" + Day;
   ENDIF;

{ End of script for M/D/Y date formats }
```

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

   You can also use a discrete expression that equates an analog tagname to true or false. For example, TankLevel >= 75. In this example, when the value of the tagname TankLevel is greater than or equal to 75, the script will execute.

Note The value for the condition **must transition** to become true or false before the script will execute. For example, if the initial value when WindowViewer starts is true, the value must become false and then true again for an **On True** script to execute.

You can apply all four script types to the same condition. Both **While True** and **While False** scripts will begin executing after the specified number of milliseconds have elapsed. To cause immediate execution, create duplicate **On True** and/or **On False** scripts. For example:



3. In the **Comment** box type any miscellaneous comments that you want on file regarding the script.

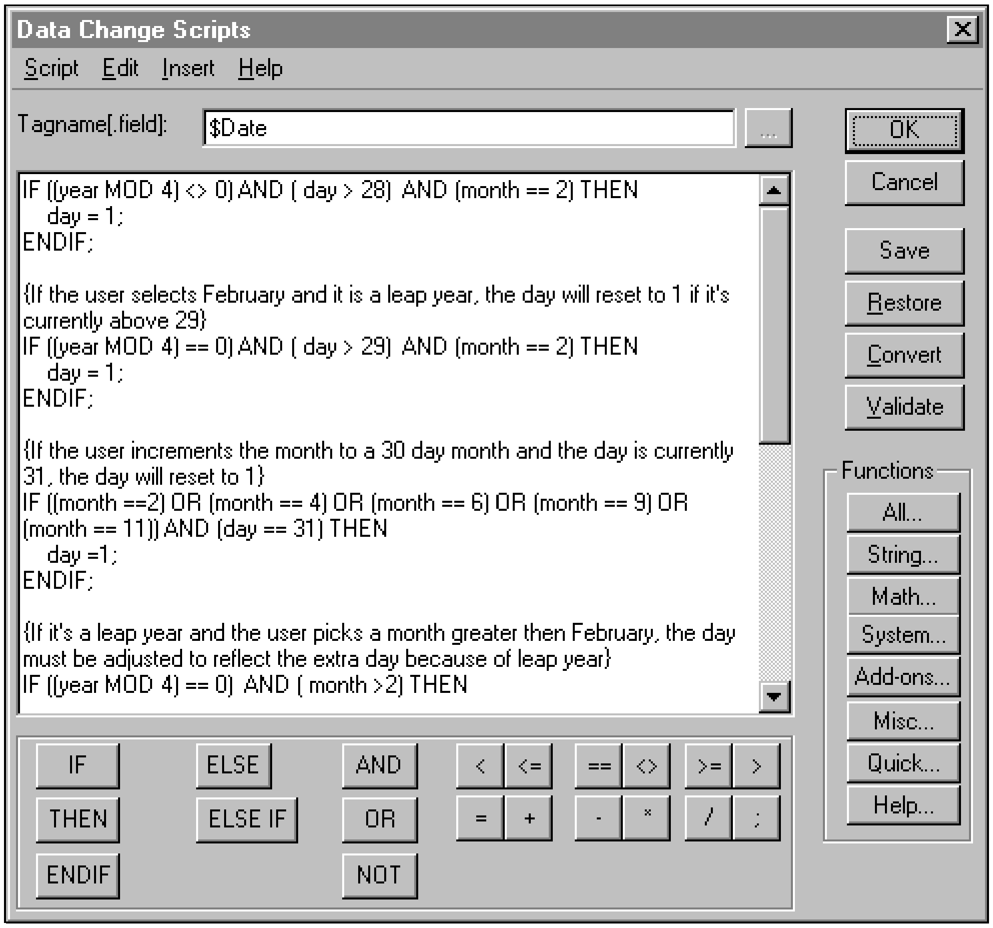
Data Change Scripts

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 that you defined for the tagname in the Tagname Dictionary.

➢ **To access the Data Change Script editor:**

1. On the **Special** menu, point to **Scripts**, and then click **Data Change Scripts**, or in the Application Explorer under **Scripts**, double-click **Data Change**. The **Data Change Script** editor appears:

   In the Application Explorer under **Scripts**, you can also right-click **Data Change**, and then click **Open**.



Tagname[.field]: $Date

```
IF ((year MOD 4) <> 0) AND ( day > 28)  AND (month == 2) THEN
   day = 1;
ENDIF;

{If the user selects February and it is a leap year, the day will reset to 1 if it's
currently above 29}
IF ((year MOD 4) == 0) AND ( day > 29)  AND (month == 2) THEN
   day = 1;
ENDIF;

{If the user increments the month to a 30 day month and the day is currently
31, the day will reset to 1}
IF ((month ==2) OR (month == 4) OR (month == 6) OR (month == 9) OR
(month == 11)) AND (day == 31) THEN
   day =1;
ENDIF;

{If it's a leap year and the user picks a month greater then February, the day
must be adjusted to reflect the extra day because of leap year}
IF ((year MOD 4) == 0)  AND ( month >2) THEN
```

2. Since Data Change Scripts are executed based upon a change in a data value, you must specify a tagname or **tagname.field** in the **Tagname[.field]** box.

3. On the **Insert** menu, click **Tagname** or double-click the script window. The **Select Tag** dialog box will appear.

   - To select a tagname without a **.field**, double-click the tagname or select it, and then click **OK**. The selected tagname will be inserted into your script at the cursor location.

     To quickly access the **Select Tag** dialog box, double-click a blank area in the script editing window. To access a specific tagname's definition in the Tagname Dictionary, type the tagname, and then double-click it.

   - To select a **.field**, first select the tagname that you want to use, then click the **Dot Field** arrow and select the **.field** that you want to associate with the selected tagname. Click **OK**. The selected tagname**.field** will be inserted into your script at the cursor location.

     To quickly insert a tagname **.field**, type the tagname followed by a period (.), and then double-click to the right of the period. The **Choose field name** dialog box will appear. Click the **.field** that you want to use. The dialog box will close and the selected **.field** will automatically be inserted into your script at the cursor location.

   For more information on tagname **.fields**, see your *InTouch Reference Guide*.

   For more information on the Tag Browser, see Chapter 4 - Tagname Dictionary.

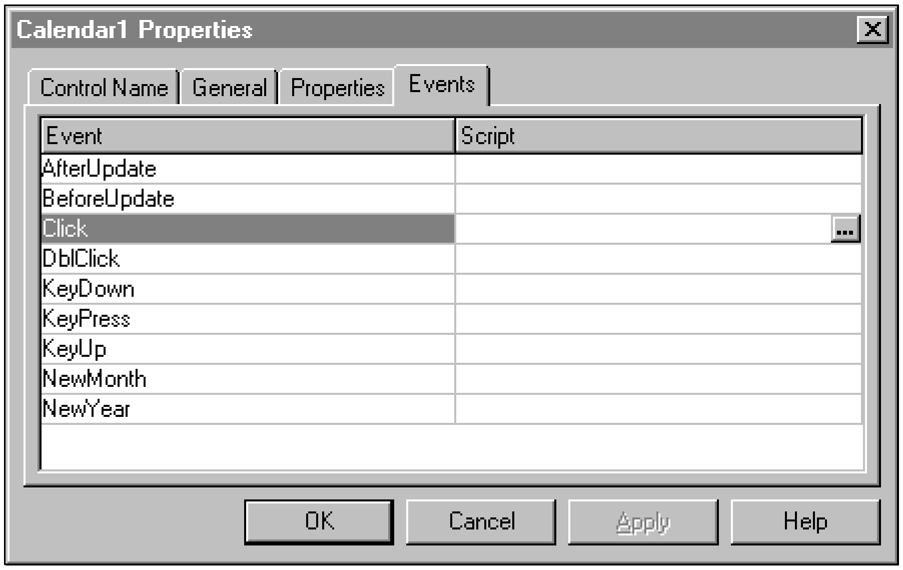
Important Note Tagnames that are modified (written to) in a Condition Script or a Data Change Script should **not** be used as the tagname for a Data Change Script or in the expression of a condition script. For example: A Data Change Script that executes on the value of "A" changing, containing the logic "B=B+1." The tagname "B" should not be used as the tagname for a Data Change Script or be part of the expression for a Condition Script. Otherwise, it will execute only once - the first time.

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. You can associate one ActiveX Event script to each event. You execute ActiveX control events in runtime (WindowViewer).

➢ **To access the ActiveX Event Script editor:**

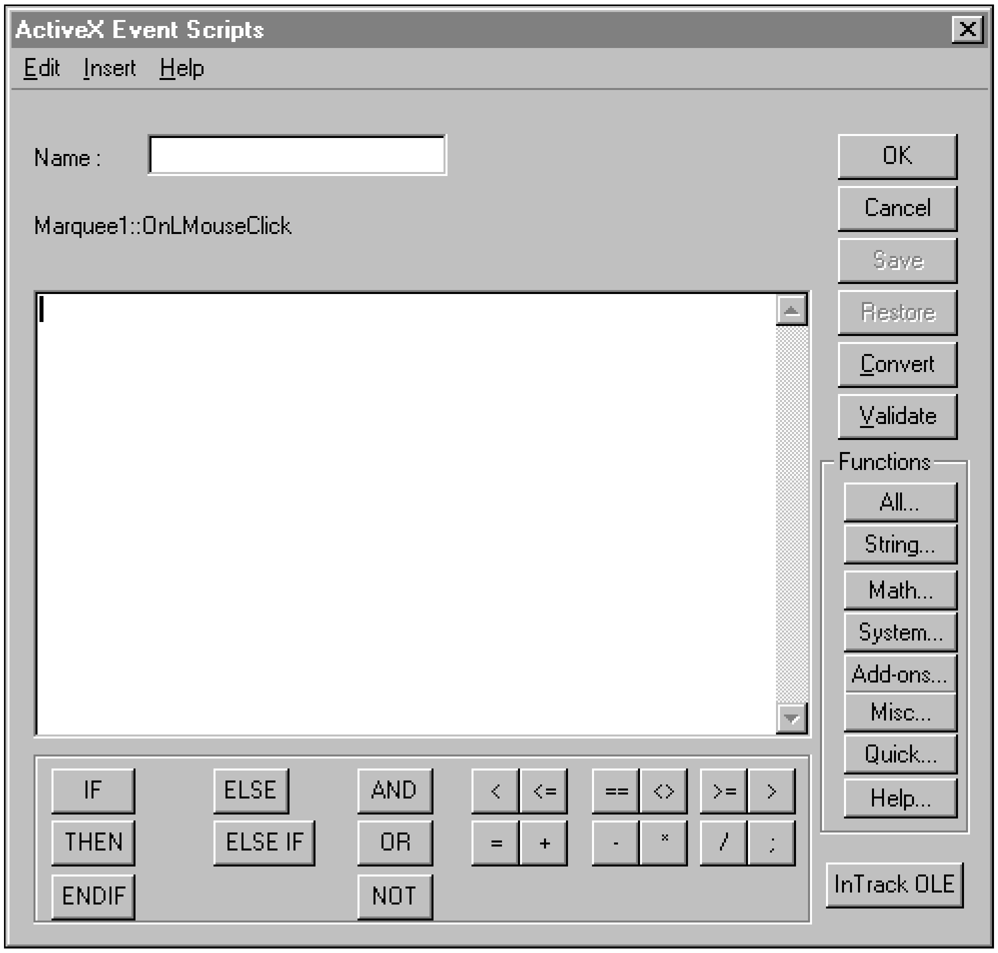
1. Click the **Events** tab in the ActiveX control's **Properties** dialog box to activate the **Events** property sheet. For example:



Event	Script
AfterUpdate	
BeforeUpdate	
Click	
DblClick	
KeyDown	
KeyPress	
KeyUp	
NewMonth	
NewYear	

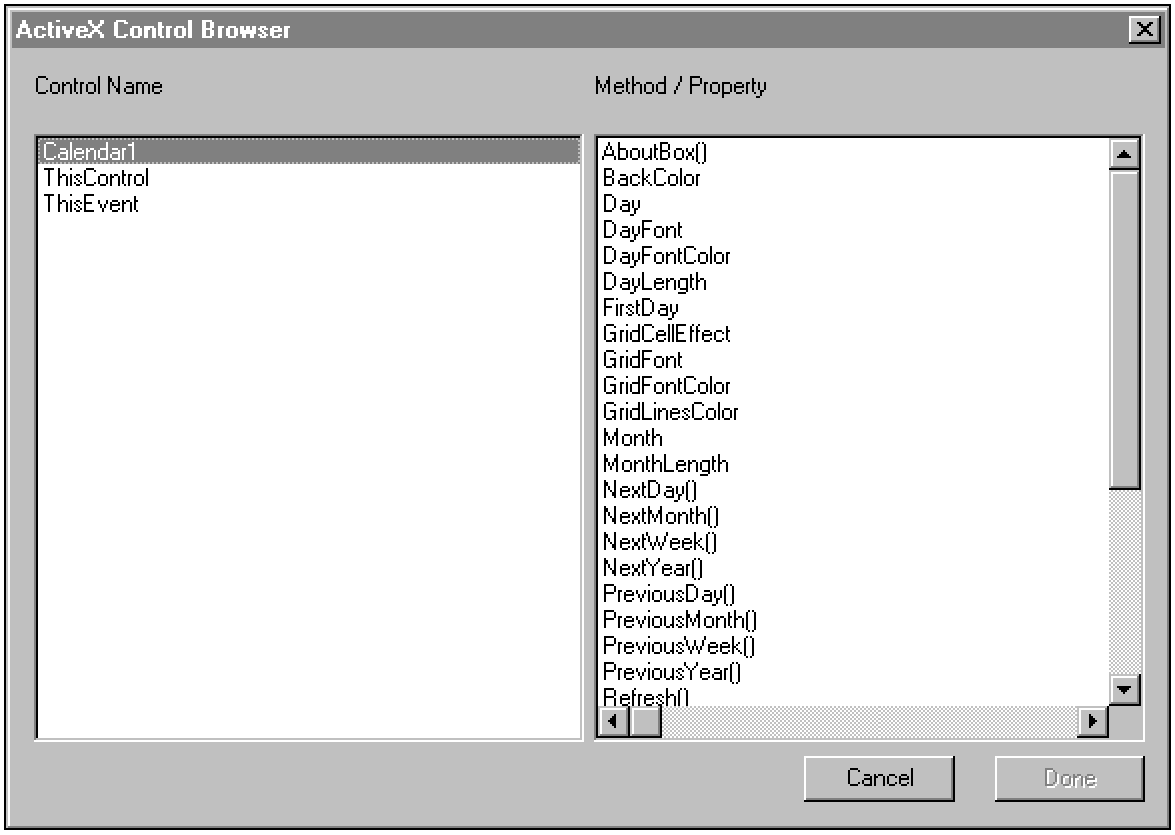
2. Double-click a blank cell in the **Script** column, or type a name for the ActiveX Event script and click **OK**.

3. If an ActiveX Event script for the name you type does not currently exist, a message box will appear asking you if you want to create it now. Click **OK**. The **ActiveX Event Scripts** editor appears:



4. In the **Name** box, type the name that you want to use to identify the ActiveX Event script.

5. ActiveX control methods are similar to ActiveX control properties. To access the ActiveX control methods, on the **Insert** menu, click **ActiveX**. The **ActiveX Control Browser** appears:



Calendar1
ThisControl
ThisEvent

AboutBox()
BackColor
Day
DayFont
DayFontColor
DayLength
FirstDay
GridCellEffect
GridFont
GridFontColor
GridLinesColor
Month
MonthLength
NextDay()
NextMonth()
NextWeek()
NextYear()
PreviousDay()
PreviousMonth()
PreviousWeek()
PreviousYear()
Refresh()

The **ActiveX Control Browser** will display the names of all ActiveX controls being used in your application. When you select a control's name, its respective methods will be displayed. Select the method that you want to insert into your script, and then click **Done**.

QuickFunctions

QuickFunctions are scripts that you can write and call 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 update of all script instances with one edit session.

QuickFunctions can be defined as asynchronous that, when executed, run in the background of the main WindowViewer (runtime) process. This functionality allows WindowViewer to separate time consuming operations such as, SQL database calls, from the main program flow. When these time consuming operations need to be performed, by creating an asynchronous QuickFunction, you provide a way for all animation links and other InTouch functionality to remain active while the operation executes.

For more information on asynchronous scripts, see "Asynchronous QuickFunction Scripts."

The animation link expression boxes are limited to 255 characters. However, you can create a more complex QuickFunction then call it from an animation link expression box. This allows you to use the **CALL** statement to call a complex script that contains a **RETURN** statement that returns the result back to the expression.

For example, when several tagnames of 30 characters each are added (using "**&**" and "**:**") together, you can only use 8 tagnames (plus spaces) into the expression. However, by using the statement, **CALL MYSCRIPT()**, in the expression, you can execute a QuickFunction that might contain hundreds of 30 character tagnames. This QuickFunction will use the RETURN statement to provide a value back to the expression.

Note The QuickFunction RETURN statement cannot be used to return message or string type values.

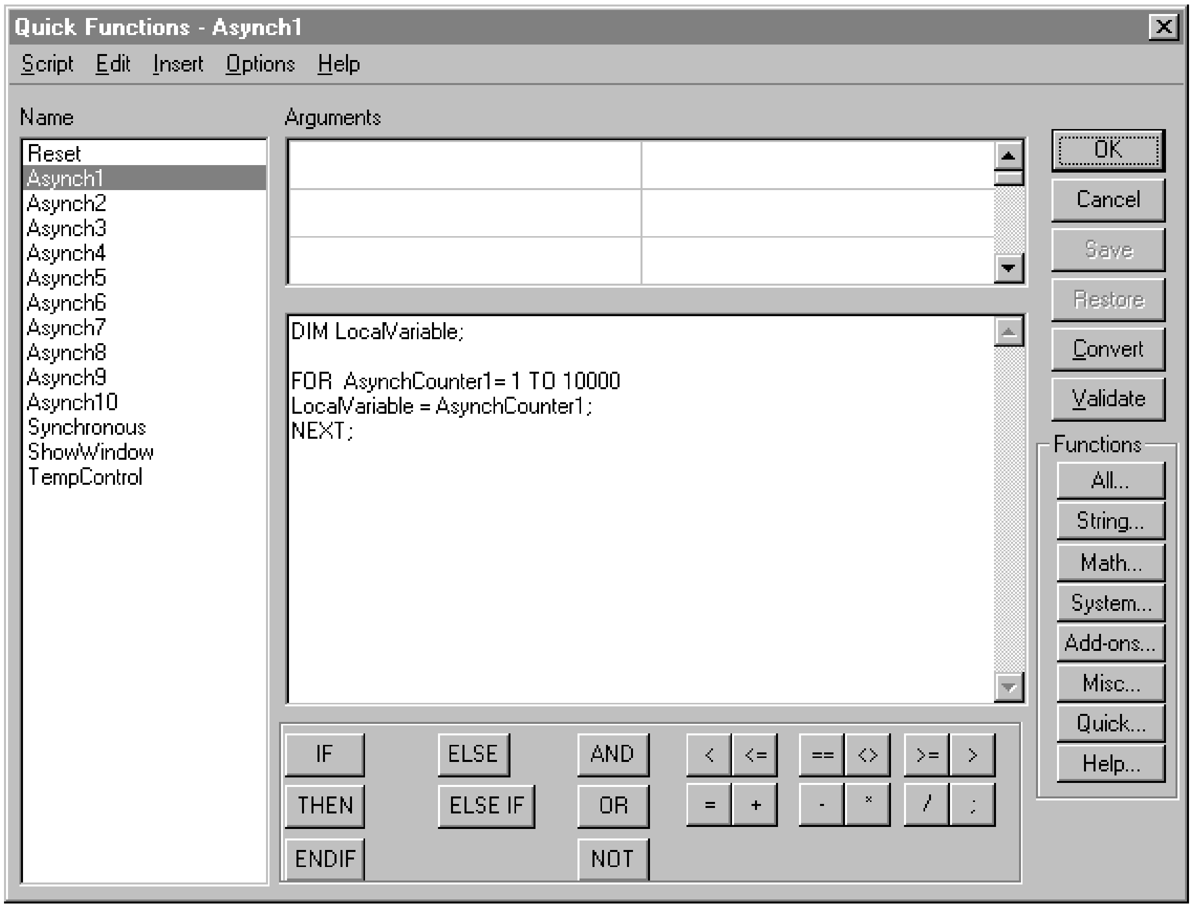
Once you create a QuickFunction and save it, you can immediately call it from any other script or expression by its name.

For more information on the syntax used for QuickFunctions, see your *InTouch Reference Guide*.

➢ **To create a QuickFunction:**

1. On the **Special** menu, point to **Scripts**, and then click **QuickFunctions**, or in the Application Explorer under **Scripts**, double-click **QuickFunctions**. The **QuickFunctions** script editor appears:

   In the Application Explorer, under **Scripts**, you can also right-click **QuickFunctions**, and then click **Open**.



Reset
Asynch1
Asynch2
Asynch3
Asynch4
Asynch5
Asynch6
Asynch7
Asynch8
Asynch9
Asynch10
Synchronous
ShowWindow
TempControl

```
DIM LocalVariable;

FOR  AsynchCounter1= 1 TO 10000
LocalVariable = AsynchCounter1;
NEXT;
```

2. The names of all the currently defined QuickFunctions will be listed in the **Names** list. To view a QuickFunction, click the name in the list.

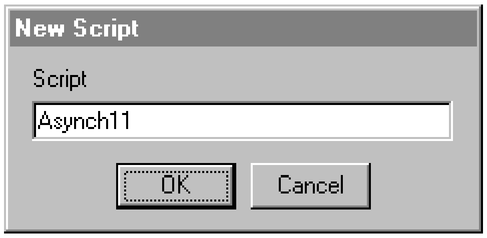
➢ **To rename a QuickFunction:**

1. On the **Script** menu, click **Rename**. The **Rename Script** dialog box will appear.
2. In the **Script Name** box, type the new name for the script.
3. Click **OK**.

Note The application's QuickFunction dictionary maintains a tally of the number of references by QuickFunction name. This value increases each time you call the QuickFunction from another QuickScript or animation link expression. Once the QuickFunction reference tally is greater than zero, you cannot delete it, rename it or modify its argument list unless all references to it in your application are deleted. However, you can modify the script text at any time without restriction.

➢ **To create a new QuickFunction:**

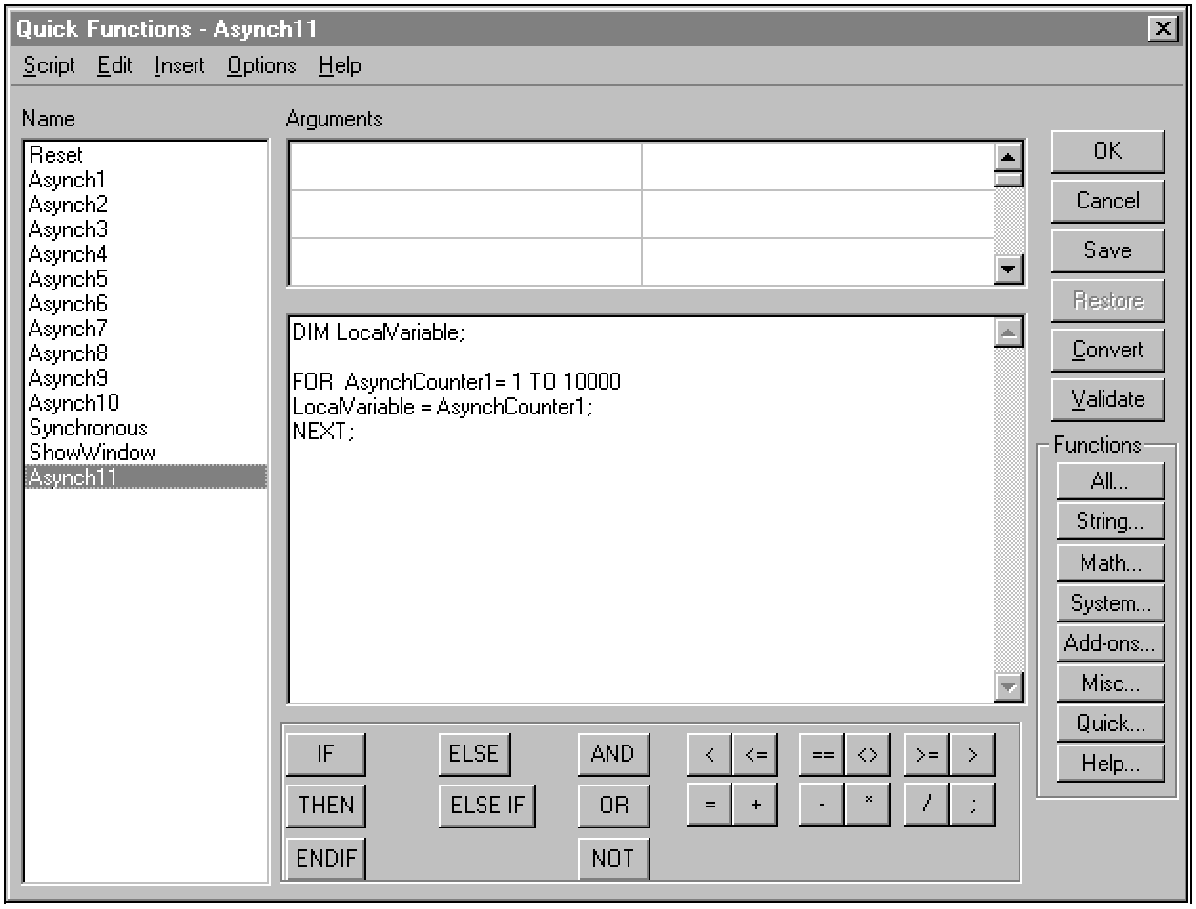
1. On the **Script** menu, click **New**. The **New Script** dialog box appears:



2. In the **Script** box type the unique name that you want to use for the new QuickFunction.

   The name can be up to 31 characters in length. (Blank spaces and duplicate names cannot be used.) This is the name that other QuickScripts or expressions will use to call the QuickFunction. This name will also appear in the **Choose function** dialog box when you click either the **All** or the **Quick** buttons in the script editor, or on the **Insert** menu, under **Functions**, when you click either the **All** or **Quick Functions** command.

3. Click **OK**. The name will appear in the script's **Name** list:



```
DIM LocalVariable;

FOR  AsynchCounter1= 1 TO 10000
LocalVariable = AsynchCounter1;
NEXT;
```

   If a QuickFunction is currently displayed when you create a new QuickFunction, the script is retained for use in the new QuickFunction.

4. Type each argument name for your script in the **Arguments** boxes, and then click the argument's respective arrow and select its data type in the list. The valid data types are:

Data Type	Description
Integer	Used to pass integer variable, tagname or constant values.
Real	Used to pass real variable, tagname or constant values.
Discrete	Used to pass discrete variable, tagname or constant values.
Message	Used to pass string variable, tagname or constant values.

Argument names are local variables that exist only within the QuickFunction in which they are defined. You can use up to 16 arguments per QuickFunction. The argument names can be up to 31 characters in length and spaces cannot be used. The argument names must also begin with an alpha character (A-Z). Duplicate names cannot be used.

Do not use tagnames for argument names. Tagnames take precedence over argument names that are the same name and will cause your script to not execute properly. An argument name does not consume a tagname count because they are treated as local variables.

Note The following are reserved keywords that should not be used as argument names:

Return	Call	Dim	As
Integer	Real	Discrete	Message

5. Once you have typed in the argument names and selected their type, you are ready to write your QuickFunction.

QuickFunction Argument Expressions

Script parameters are passed by value. Argument expressions can be any script expression that returns an integer, real, discrete, message data type value. All argument expression values are resolved by the calling script before executing the QuickFunction. Examples:

```
CALL Stuff (5.6, 237, "PI");
```

In this expression, The Real constant 5.6 as argument1, Integer constant 237 as argument 2, Message constant "PI" as argument 3.

```
CALL Temp (IntegerTag);
```

IntegerTag passed as argument expression value.

```
CALL ValveOpen (Tag.MaxEU -5);
```

Calculated value (Tag.MaxEU -5) passed as argument expression value.

Argument Data Type Matching

There must be a strict left to right correspondence between the data types of the calling statement's argument list and the data types of the saved QuickFunction which is called. There must also be an exact number of arguments to match the number of arguments in the saved QuickFunction argument list as well. Coercion is used to type cast Real values to Integer and Integer values to Real. This ability to modify the type of a value, allows any analog argument to be passed to any other analog types.

For example, If you pass a Real value of 1.23 to an Integer argument it will use only the 1 and the .23 will be lost. Similarly, if you pass an Integer value of 1 to a Real argument it will promote the 1 to 1.0. However, despite this coercion capability we recommend that you make a strict match of calling argument types to the corresponding QuickFunction argument list.

Valid QuickFunction Syntax

QuickFunctions return a value. The QuickFunction statement syntax and form are as follows:

`CALL`*`QuickFunctionName`* `(` *`[arg1, ... arg16]`* `);`

Where:

`CALL`	Required keyword in all QuickScripts and expressions to call a QuickFunction.
`QuickFunctionName`	1 to 31 character string that corresponds to the name assigned to the saved QuickFunction.
`( [arg1, ...arg16] )`	0 to 16 comma separated argument expressions enclosed in parentheses.

📖 For more information on the syntax used for QuickFunctions, see your *InTouch Reference Guide*.

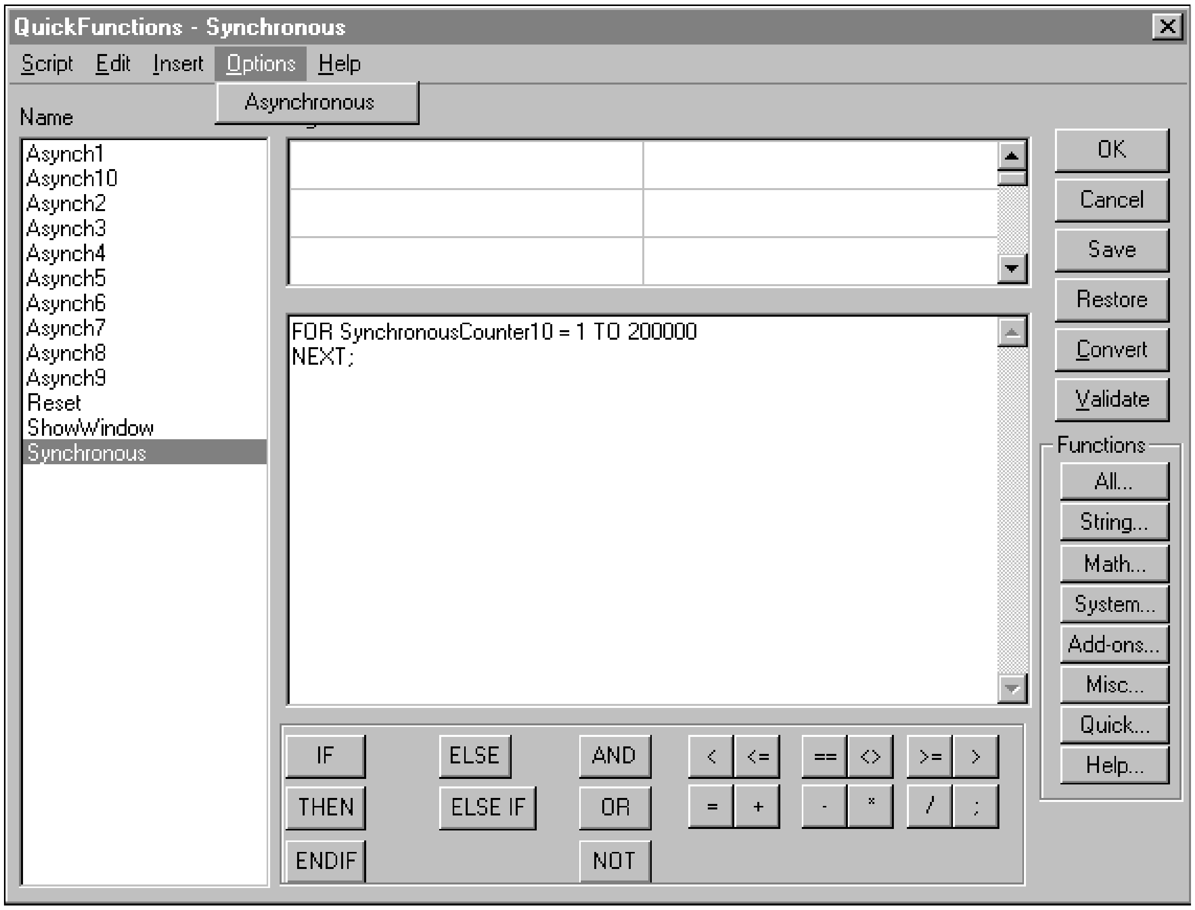
Asynchronous QuickFunction Scripts

You can define your QuickFunctions as asynchronous. (QuickFunctions are the only InTouch script type that support asynchronous functionality.) When WindowViewer encounters a call to an asynchronous QuickFunction, a separate thread gets spawned. Once the new thread is spawned, WindowViewer is free to continue to call more scripts (including more asynchronous QuickFunctions), wait for the asynchronous scripts to end, or do graphic window updates. It is from the newly spawned thread that the asynchronous QuickFunctions will actually be launched. Once the asynchronous script completes execution, the newly spawned thread ends. The threading mechanism is transparent to the operator.

Note Asynchronous QuickFunctions cannot return a value. Therefore, asynchronous QuickFunctions cannot be used in animation link expressions. Additionally, there is no limit to the number of asynchronous QuickFunctions that you can execute simultaneously. However, to ensure adequate system performance, we highly recommend that no more than three be executed simultaneously.

➢ **To create an asynchronous QuickFunction:**

1. Create a QuickFunction.
2. On the **Options** menu, click **Asynchronous**. For example:



```
FOR SynchronousCounter10 = 1 TO 200000
NEXT;
```

Note In runtime, asynchronous QuickFunctions cannot be stopped once they begin executing. However, if the operator stops all scripts from running (through the **Logic** menu **Halt Logic** command or by pressing a button linked to the **$LogicRunning** system tagname) no new asynchronous QuickFunctions will begin executing.

Controlling Asynchronous Scripts

You can use the **IsAnyAsyncFunctionBusy()** function to find out if any asynchronous QuickFunctions are running. This function can be used to make the QuickScript that calls an asynchronous QuickFunction wait for all other asynchronous QuickFunctions to complete processing. This allows the QuickScript to re-synchronize itself.

The valid syntax for this function is:

```
DiscreteTag = IsAnyAsyncFunctionBusy(timeout);
```

Where:

DiscreteTag	A discrete type tagname to which a value is returned as follows: ● If the function times out, waiting all executing QuickFunctions to complete, a 1 (true) is returned to *DiscreteTag*. ● If there are no asynchronous QuickFunctions running, a value of 0 (false) will be returned immediately, or the QuickFunction will wait for the timeout to elapse. It will also return a 0 if after timing out there are no asynchronous QuickFunctions running.
timeout	An integer value representing the number of seconds to wait to determine if any asynchronous QuickFunctions are running.

For example, let's assume you want to connect to several SQL databases using asynchronous QuickFunctions and, you know that it will take 2 minutes to make those connections. First, you would execute the asynchronous QuickFunctions to connect to the SQL databases. Next, you would launch the function, **IsAnyAsyncFunctionBusy(120)** to allow enough time for SQL to make the connections before completing the QuickFunction.

However, if after 2 minutes the connections have not been made and the asynchronous QuickFunctions are still busy trying to make the connections, a value of 1 (true) will be returned by the **IsAnyAsyncFunctionBusy()** function. You can now display an error message telling the operator that the SQL connections were unsuccessful.

For example, you could use the following window **On Show** QuickScript:

```
IF IsAnyAsyncFunctionBusy(120) == 1 THEN
   SHOW "SQL Connection Error Dialog";
ENDIF;
```

Using Local Variables

You can declare local variables within a script to store temporary results and create complex calculations with intermediate scripting values without impacting or decreasing your licensed tagname count and increase performance.

Local variables or tagnames can be used interchangeably within the same script. However, local variables loose their value and meaning once the script ends where, 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 that you can declare within a given script body is limited only by your available memory. Once you have declared a local variable, you can include it 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 hand 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 &.

Warning! If there is a conflict in your script between a declared variable name and a tagname, (both are the same name) the variable name takes precedence over the tagname.

For example, let's assume that you have a tagname defined in your database as "Temp," and you declare "DIM Temp AS Integer;". Within the declaring script, expressions using "Temp" in a statement will refer to the value associated with the local variable "Temp" rather than the tagname "Temp."

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;
```

Notes

1. Data-type keywords are case insensitive.
2. The DIM statement line must be terminated with a semi-colon (;).
3. Cascaded DIM statements are not supported.
4. The DIM statement can be located anywhere within the script body. But must precede the first referencing script statement or expression.
5. If the local variable is referenced before the DIM statement, the script editor will read it as a tagname when the script is validated and you will be asked to define it.

For more information on the syntax used for local variables, see your *InTouch Reference Guide*.

Creating FOR-NEXT Loop Scripts

A FOR-NEXT loop is used to perform a function (or set of functions) within a script several times during a single execution of a script. The general format of the FOR-NEXT loop is as follows:

```
FOR AnalogTag = start_expression TO end_expression [STEP
change_expression]
  ...statements...
IF (condition) THEN
   [EXIT FOR;]
   ENDIF;
   ...statements...
NEXT;
```

Where:

[ ]	Items enclosed in brackets denote optional parameters.
BOLDCASE	Bold items in **UPPERCASE** denote script language reserved keywords.
italics	Items in lowercase *italics* denote variable data.
AnalogTag	An InTouch Analog type tagname.
start_expression	A valid InTouch expression, to initialize *AnalogTag* to a value for execution of the loop.
end_expression	A valid InTouch expression, if *AnalogTag* is greater than *end_expression*, execution of the script jumps to the statement immediately following the **NEXT** statement. (This holds true if loop is incrementing up, otherwise, if loop is decrementing, loop termination will occur if IntegerTag is less than *end_expression.*)
change_expression	A valid InTouch expression, to define the increment or decrement value of *AnalogTag* after execution of the **NEXT** statement.

Note The *change_expression* can be either positive or negative. If *change_expression* is positive, *start_expression* must be less than or equal to *end_expression* or the statements in the loop will not execute. If *change_expression* is negative, *start_expression* must be greater than or equal to *end_expression* for the body of the loop to be executed. If **STEP** is not set, then *change_expression* defaults to 1.

...statements...	One or more valid InTouch script language statements. These could be nested **FOR** loops. Nested loops require different *change_expressions* from outer loops.
FOR	Signals the beginning of the "For" loop.
TO	Signals the beginning of the *end_expression*.
STEP	Signals the beginning of the *change_expression*.
EXIT FOR	Immediately terminates the loop with script execution jumping to the statement immediately following the **NEXT** statement.
NEXT	Signals the end of the loop statement.

When you execute a FOR...LOOP function, InTouch:

1. Sets AnalogTag equal to start_expression.
2. Tests to see if AnalogTag is greater than end_expression. If so, InTouch exits the loop. (If change_expression is negative, InTouch tests to see if AnalogTag is less than end_expression.)
3. Executes the statements.
4. Increments AnalogTag by 1 - or by change_expression if it is specified.
5. Repeats steps 2 through 4.

Nesting FOR-NEXT Loops

FOR-NEXT loops may be nested. Number of levels of nesting possible depend on your system's memory and resource availability.

Screen Updates and Performance Penalties

During execution of the FOR-NEXT loop, the screen update sub-system within InTouch will pause until the loop is complete.

- All animation on the screen will pause during execution of the FOR-NEXT loop. Therefore, you cannot use the FOR-NEXT loop to animate an object to move across the screen, since all movement will occur only after the loop has completed.
- Real-time trends will pause.
- Historical trends will pause, if "updating."
- Displayed values will not update on the screen during loop execution. Although there may be new values present within those variables, those new values will only be displayed after the loop has completed.
- The value of any I/O type tagname modified within the body of a FOR-NEXT loop will be transmitted to the I/O server only after loop execution is finished. Therefore, if you modify the value of a I/O type tagname during each interaction of a FOR-NEXT loop, only the final value of that I/O type tagname will be transmitted to the PLC.

Note FOR-NEXT loops pause all operations in InTouch. While executing, no data moves in or out of WindowViewer, no animation links are updated and, no other scripts are executed including asynchronous QuickFunctions. However, FOR-NEXT loops used inside asynchronous QuickFunctions do not pause other operations.

Loop Execution Time Limit

By default, FOR-NEXT loops must complete execution within 5 seconds. This is a safety limit built into the FOR-NEXT sub-system. The time limit will be enforced for all FOR-NEXT loops. For example:

```
FOR X = 1 TO 1000000
   FileWriteMessage("C:\LOG.TXT","Hello");

NEXT;
```

Note You can extend this limit by adding the following switch to your INTOUCH.INI file in your application directory:

LoopTimeout=20

Where; 20 is the number of seconds before terminating the loop prematurely.

This loop will most likely exceed the time limit of 5 seconds. In the Wonderware Logger a message will appear that indicates the following:

```
95/03/07 07:34:40.550/VIEW    /Exceeded loop time limit of 5 
seconds.

95/03/07 07:34:40.550/VIEW    /FOR-NEXT Timeout at X = 65464
```

📖 For more information on the Wonderware Logger program, see your *FactorySuite System Administrator's Guide.*

This error message indicates that the FOR-NEXT loop has terminated before meeting the *end_condition*, it also provides the value of the loop variable at the time of the loops termination. This information will allow you to track down which FOR-NEXT is having the problem.

Note The 5 second time limit is only evaluated each time the NEXT; statement is reached within the FOR-NEXT loop. For example, if you were executing the following script:

```
FOR Index = 1 to 10
   SQLInsert(ConnectionID,"ORG","list1");
   SQLInsert(ConnectionID,"ORG","list2");
   SQLInsert(ConnectionID,"ORG","list3");
   SQLInsert(ConnectionID,"ORG","list4");
NEXT;
```

If each **SQLInsert()** took 12 seconds to complete, all four inserts would be executed to completion before the loop exited because it had exceeded the 5 second time limit.

Loop Variable Value After Loop Execution

As in Visual Basic (and most other popular Basic programming languages) the value of the loop variable at the end of loop execution is defined as follows:

The Index continues to increase in value, starting at the *Start_Condition*, incremented each iteration by the value of *Step_Expression*, until it reaches the last iteration at which the Index value exceeds that of the *End_Expression*.

Therefore, if you had a loop as follows:

```
FOR Index = 2 TO 25 STEP 7
   { Some script statements }
NEXT;
```

The value of Index would progress as follows:

Iteration	Value	Calculation
1	9	2 + 7
2	16	2 + 7 + 7
3	23	2 + 7 + 7 + 7
4	30	2 + 7 + 7 + 7 + 7

At that point, when the value reached 30, the loop would stop executing because it exceeded the *End_Expression*. The ending value of Index would be 30.

Nested Control Structures

Control structures can be placed inside other control structures (such as an IF...THEN block within a FOR...NEXT loop). A control structure inside another control structure is known as *nested*.

Example:

```
FOR TagX = 1 TO 5
FOR TagY = 1 TO 10
   ...statements...
   IF (condition) THEN
   [EXIT FOR;]
   ENDIF;
   ...statements...
   NEXT;
NEXT;
```

Where:

The first NEXT closes the inner FOR loop and the last NEXT closes the outer FOR loop. Likewise, in nested IF statements, the ENDIF statements automatically apply to the nearest prior IF statement.

Exiting a Control Structure

The EXIT FOR statement allows you to exit directly from a FOR loop. Syntactically, the EXIT FOR statement is simple. EXIT FOR can appear as many times as needed inside a FOR loop:

Example:

```
FOR TagX = 1 TO 10;
...statements...
IF (condition) THEN
   EXIT FOR;
ENDIF;
...statements...
NEXT;
```

Below are some examples of various FOR-NEXT loop scripts:

Example 1 "Simple Math 2"
This loop allows a person to configure the number with which to raise to a power, as well as the power to which they would like to raise it by setting up the value input links for the tagnames NumberToRaise and Power:

```
Product = 1;
NumberToRaise = 4;
Power = 12;
FOR Index = 1 TO Power
   Product = Product * NumberToRaise;
NEXT;
```

Once the above script has completed processing the value of the **Product** will be 16,777,216.

Example 2 "Complex FOR-NEXT using indirect tagnames"
This loop utilizes the "EXIT FOR" and "STEP" portions of the FOR-NEXT construct to perform a search on a set of 100 tagnames, searching for the tagname to which NumberEntered is equivalent.

Note For this example, it is assumed that there are 100 **Memory Integer** tagnames (TAG1 - TAG100) already existing. An operator enters a number into the tagname NumberEntered, and this loop searches TAG1 - TAG100 for a matching value. In addition, there is an indirect analog tagname created: *IndirectTag*

```
Found = 0;

FOR Index = 1 TO 100
   IndirectTag.NAME = "TAGNAME" + TEXT( Index, "#" );
   IF (IndirectTag.NAME == ("TAGNAME"+ Text(NumberEntered,"#")))
      THEN
      Found = 1;
      EXIT FOR;
   ENDIF;
NEXT;

IF (Found==1) THEN
   Show "NumberFound"; {window notifying search was successful}
ELSE
Show "NumberNotFound";
ENDIF;
```

Once the script has completed processing, a window will be displayed either indicating that the number was found, or not.

Note Notice the use of two addition functions within this script: **Show()** and **TEXT()**.

Example 3

This loop performs an odd calculation, but illustrates the use of nested FOR-NEXT loops, as well as the use of the "STEP" portion of the FOR-NEXT construct:

```
MyTag = -1;

FOR Index = 1000 TO -1000 STEP -5
   IF (MyTag > Index) THEN
      FOR Index2 = 1 TO 10 STEP 2
         MyTag = MyTag * (Index + 11);
      NEXT;
   ENDIF;
NEXT;
```

Once the script has completed processing the values will be: MyTag = -7776,
Index = -1005 and Index2 = 11.

Script Editing Styles and Syntax

The InTouch script editor supports two "styles" of scripts: "Simple" and "Complex." Simple scripts allow assignments, comparisons, simple math functions, and so on. Complex scripts provide the ability to perform logical operations in the form of IF-THEN-ELSE type statements. In addition, InTouch also supports the use of built-in complex functions, as well as native QuickFunctions.

An example of a function would be the **StartApp(*ApplicationName*)** function actually started the Windows application identified in the argument, "(*ApplicationName*)". Functions may be used in both Simple and Complex scripts. The following section includes complete descriptions of each style.

Required Syntax for Expressions and Scripts

The syntax used in scripts and expression dialog boxes is similar to the algebraic syntax of a calculator. Most expressions are assignment statements written in the following form:

```
a = (b - c) / (2 + x) * xyz;
```

This statement would cause the value of the expression to the right of the equal sign (=) to be placed in the variable location named "a." Every expression must end with a semicolon (;). The operands in an expression may be constants or variables. A single tagname must appear to the left of the assignment operator =.

Memory or **I/O Message** type tagnames can be concatenated by using the plus (+) operator. For example, tagnames can be concatenated for use in **Indirect** type tagnames. If a Data Change Script such as the one below was created, each time the value of "Number" changed, the indirect tagname "Setpoint" would change accordingly:

```
Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

Where: The result is "Setpoint1."

Simple Scripts

Simple scripts provide the ability to implement logic such as assignments, math and functions. An example of this type of scripting is:

```
React_temp = 150;
ResultTag = (Sample1 + Sample2)/2;
{this is a comment}
Show "Main Menu";
```

In this example, the script will assign the value of "150" to the tagname "React_temp." "Sample1" will be added to "Sample2" and the result divided by "2"and the "Main Menu" window will appear on the screen when the script is run.

Note Notice that each logical statement must end with a semicolon (;) and that several logical statements may be included in one script. Also notice that comments are allowed within the script editor. Comments are identified by a pair of braces {}. The function **Show** was also used with the argument "Main Menu" (WindowName) to cause the specified window to open.

In addition to simple assignments, mathematical operators and functions, InTouch supports several other "Operations" for use on "Operands," that is, tagnames, number constants, and so on.). **Discrete**, **Integer** and **Real** tagname types are supported for all operations listed below. **Message** tagname types may be used with comparison and assignment operations only. The following is a list of InTouch supported operations:

Operations that Require 1 Operand:

~	Complement
-	Negation
NOT	Logical NOT

Operations that Require 2 Operands:

*****	Multiplication
/	Division
+	Addition and Concatenation
-	Subtraction
=	Assignment
MOD	Modulo
SHL	Left Shift
SHR	Right Shift
&	Bitwise AND
^	Exclusive OR
|	Inclusive OR
******	Power
<	Less than
>	Greater than
<=	Less than or Equal to
>=	Greater than or Equal to
==	Equivalency ("is equivalent to")
<>	Not Equal to
AND	Logical AND
OR	Logical OR

Precedence of Operators

The following list shows the order of precedence for evaluation of operators. The first operator in the list is evaluated first, the second operator is evaluated second, and so on. Operators in the same line in the list have equivalent precedence. Operators are listed from highest precedence to lowest precedence.

( )	Highest Precedence
- , NOT, ~	
**	
* , /, MOD	
+, -	
SHL, SHR	
<, >, <=, > =	
==, <>	
&	
^	
|	
=	
AND	
OR	Lowest Precedence

Precedence Examples

Since * has higher precedence than +,
B + C * D; is equivalent to B + (C * D);

Since ***** and **/** have equivalent precedence,
B / C * D; is equivalent to (B / C) * D;

Some other examples to note:
B * - D; is equivalent to B * (-D);
B or C and D; is equivalent to B or (C and D);

Descriptions of Operators

Arguments for the previously listed operators can be numbers or tagnames. Putting parentheses around the arguments is optional, and the operator names are not case-sensitive.

Parentheses ( )

Parentheses are used to ensure the correct order of evaluation for the operations. They can also make a complex expression easier to read. Operations in parentheses are evaluated first (preempting the other rules of precedence that would apply in the absence of parentheses). If the precedence is in question or needs to be overridden, use parentheses. In the example below parentheses are used to force B and C to be added together before multiplying by D:

```
( B + C ) * D;
```

Negation (-)

Negation is an unary operator that converts a positive integer or real number into a negative number.

Complement (~)

This operator yields the one's complement of a 32-bit integer. In other words, it converts each zero-bit to a one-bit and each one-bit to a zero-bit. The one's complement operator is an unary operator that accepts an integer operand.

Power (**)

This binary operator returns the result of a number (the base) raised to the power of a second number (the power). The base and the power can be any real or integer numbers, subject to the following restrictions:

- A zero base and a negative power are invalid.
  Example: **"0 ** - 2"** and **"0 ** -2.5"**
- A negative base and a fractional power are invalid.
  Example: **"2 ** 2.5"** and **"-2 ** -2.5"**

Invalid operands yield a zero result. Moreover, the result of the operation should not be so large or so small that it cannot be represented as a real number. Example:

```
1 ** 1 = 1.0
3 ** 2 = 9.0
10 ** 5 = 100,000.0
```

Multiplication (*), Division (/), Addition (+), Subtraction (-)

These binary operators perform basic mathematical operations. The plus (+) is also used to concatenate **Memory** or **I/O Message** type tagnames. For example, tagnames can be concatenated for use in **Indirect** tagnames. If a Data Change Script such as the one below were created, each time the value of "Number" changed, the indirect tagname "Setpoint" would change accordingly:

```
Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

Where: The result would be "Setpoint1."

Modulo (MOD)

MOD is a binary operator that divides an integer quantity to its left by an integer quantity to its right. The remainder of this division is the result of the MOD operation. Example:

```
97 MOD 8 yields 1
63 MOD 5 yields 3
```

Shift Left (SHL), Shift Right (SHR)

SHL and **SHR** are binary operators that use only integer operands. The binary content of the 32-bit word referenced by the quantity to the left of the operator is shifted (right or left) by the number of bit positions specified in the quantity to the right of the operator. Bits shifted out of the word are lost. Bit positions vacated by the shift are zero-filled. (The shift is an unsigned shift.)

Bitwise AND (&)

This is a bitwise binary operator which compares 32-bit integer words with each other, bit for bit. A common use of this operator is to mask a set of bits. The operation in this example would "mask out" (set to zero) the upper 24 bits of the 32-bit word. For example:

```
result = name & 0xff;
```

Exclusive OR (^) and Inclusive OR (|)

The **OR**s are bitwise logical operators which compare 32-bit integer words to each other, bit for bit. The Exclusive **OR** looks for opposite bit status's in corresponding locations. If the corresponding bits are the same, a zero is the result. If the corresponding bits differ, a one is the result. Example:

0 ^ 0 yields 0
0 ^ 1 yields 1
1 ^ 0 yields 1
1 ^ 1 yields 0

The Inclusive **OR** examines the corresponding bits for a one condition. If either bit is a one, the result is a one. Only when both corresponding bits are zeros is the result a zero. For example:

0 | 0 yields 0
0 | 1 yields 1
1 | 0 yields 1
1 | 1 yields 1

Assignment (=)

Assignment is a binary operator which accepts integer or real operands. Each statement may contain only one assignment operator. Only one name can be on the left side of the assignment operator. Read the equal sign (=) of the assignment operator as "is assigned to" or "is set to."

Note Do not confuse the equal sign with the equivalency sign (==) used in IF-THEN-ELSE clauses and relational contacts.

Comparisons (<, >, <=, >=, ==, <>)

These operators are used in IF-THEN-ELSE statements to execute various instructions based on the state of an expression.

AND, OR, and NOT

These operators only work on discrete tagnames. Although, if these operators are used on integers or reals, they are *converted* as follows:

Real to Discrete: If real is 0.0, discrete is 0, otherwise discrete is 1.
Integer to Discrete: If integer is 0, discrete is 0, otherwise discrete is 1.

Thus, if the statement were: "Disc1 = Real1 AND Real2;" and Real1 was 23.7 and Real2 was 0.0, Disc1 would have 0 assigned to it, since Real1 would be converted to 1 and Real2 would be converted to 0.

Complex Scripts

Complex scripts provide the ability to perform logical operations in the form of IF-THEN-ELSE type scripts and the ability to process loops using FOR-NEXT script structures. The following is an example of an IF-THEN-ELSE script:

```
IF React_temp > 200 THEN
   React_temp_sp = 150;
   PRValve = 1;
   PlaySound("c:\alert.wav",1);
ELSE
   PRValve = 0;
   PlaySound("c:\All_Ok.wav",1);

ENDIF;
```

In this example, the script checks if the reactor temperature is higher than "200." If so, then the "React_temp_sp" tagname is assigned the value of "150," the "PRValve" is turned on and the "alert.wav" file is played by calling the **Playsound()** function. Else, the reactor temperature is lower than "200," the "PRValve" is turned off and the "All_Ok.wav" file is played.

Note Notice that each IF statement requires an ENDIF statement. Also be aware that an ELSE statement is not required if unnecessary for the script's functionality. Note the use of the Function **PlaySound(*path_text,number*)** in this complex script.

Simple Math

This loop performs a simple iterative mathematical calculation. When executed, Product equal the value of NumberToRaise raised to the power of 10, that is, **Product=NumberToRaise**^{10}.

```
Product = 1;
NumberToRaise = 4;
FOR Index = 1 TO 10
   Product = Product * NumberToRaise;
NEXT;
```

Once the script executes, the value of the "Product" will be "1048576."

Note FOR-NEXT loops pause all operations in InTouch. While executing, no data moves in or out of WindowViewer, no animation links are updated and, no other scripts are executed including asynchronous QuickFunctions. However, FOR-NEXT loops used inside asynchronous QuickFunctions do not pause other operations.

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 various 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;
```

Note Each IF must have a matching ENDIF and a semicolon must be entered at the end of each statement line.

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 = l;
   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;
```

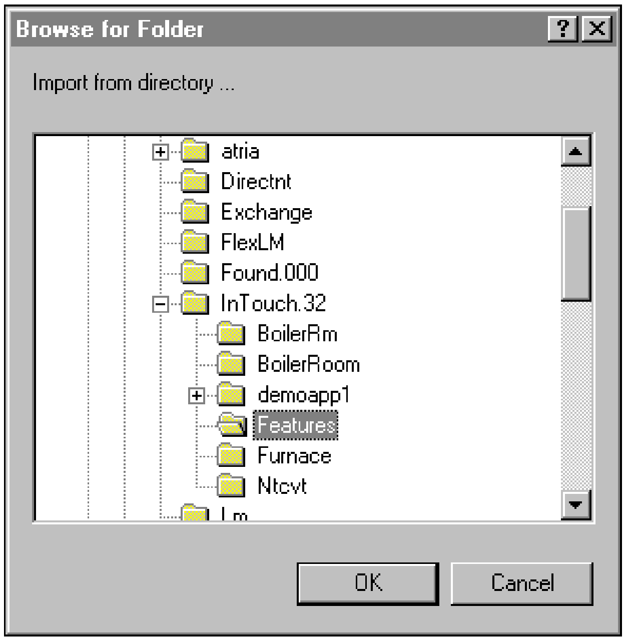
Importing QuickScripts

Importing QuickScripts from one InTouch application to your current application, can save you a considerable amount of 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 QuickScripts. When you move QuickScripts from one InTouch application to another, you must use the **Import** command on the **File** menu.

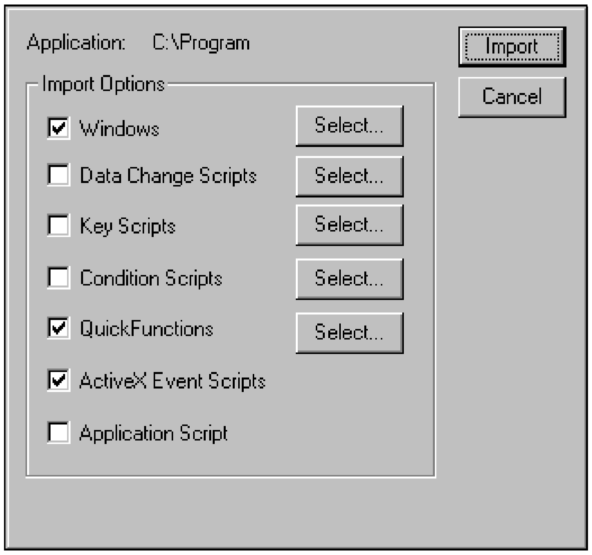
For more information on remote tagname references, see Chapter 4 - Tagname Dictionary.

➢ **To import a QuickScript:**

1. Close all windows in your current application
2. On the **File** menu, click **Import**. The **Browse for Folder** dialog box appears:

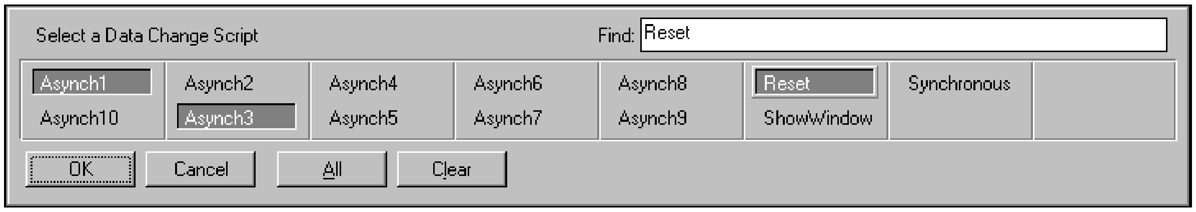


3. Locate and select the application directory (folder) containing the QuickScript(s) the you want to import.
4. Click **OK**. The following dialog box appears:


Application: C:\Program

Import Options
- [✓] Windows
- [ ] Data Change Scripts
- [ ] Key Scripts
- [ ] Condition Scripts
- [✓] QuickFunctions
- [✓] ActiveX Event Scripts
- [ ] Application Script

5. Select the QuickScript type(s) that you want to import.
6. Click **Select**. The **Select a *ScriptType* Script** dialog box appears:



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

Note When you import ActiveX Event scripts, from one application to another, all ActiveX Events scripts are imported. Additionally, in order for an imported ActiveX Event script to function properly in the new application, the same ActiveX control and the same event for which the script was originally created, must also be used in the new application, and it must be loaded into memory. If the window containing an ActiveX control is closed, its ActiveX Event scripts, or any other InTouch QuickScripts containing script functions associated with that ActiveX control, will not execute properly.

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

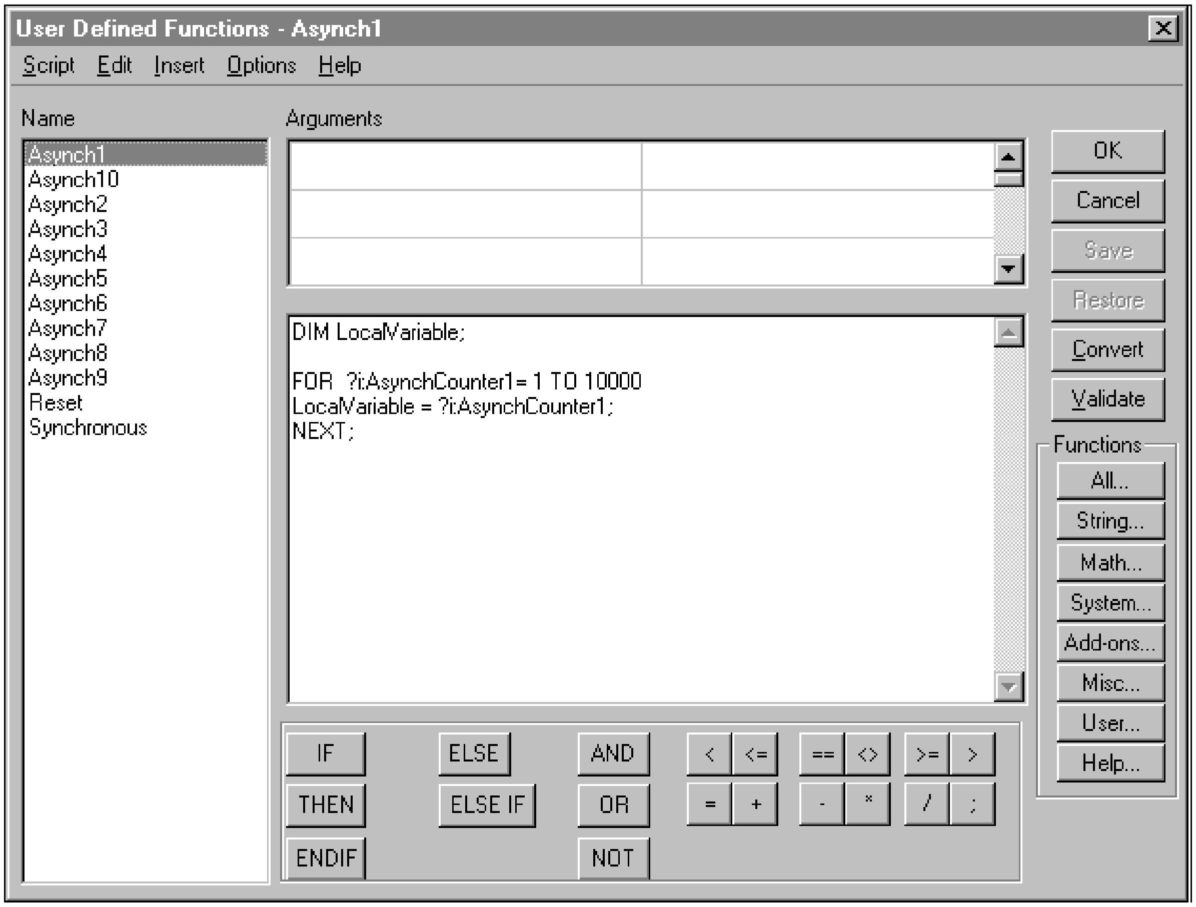
When you import a QuickScript into a new application, all the tagnames in the QuickScript are imported with the QuickScript, but they are not added to your Tagname Dictionary. Instead, they are automatically converted to "placeholder" tagnames. You must convert the placeholder tagnames in order to use them and, 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 not 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 only for placeholder tagnames. This increase is tagname length is not supported for standard tagnames.

➢ **To convert placeholder tagnames in an imported script:**

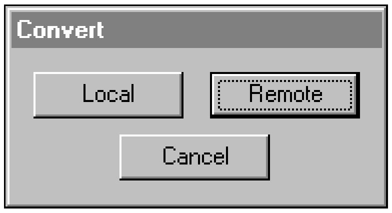
1. On the **Special** menu, point to **Scripts**, and then click the type of QuickScript you imported, or in the Application Explorer under **Scripts**, double-click the QuickScript type that you imported. The QuickScript editor will appear displaying the first QuickScript on file for the type of script you selected. For example, if you imported QuickScript functions, the **QuickFunctions** script editor appears:

   To quickly open the imported QuickScript, double-click **Scripts** in the Application Explorer, and then double-click the QuickScript type.



```
DIM LocalVariable;

FOR  ?i:AsynchCounter1= 1 TO 10000
LocalVariable = ?i:AsynchCounter1;
NEXT;
```

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.

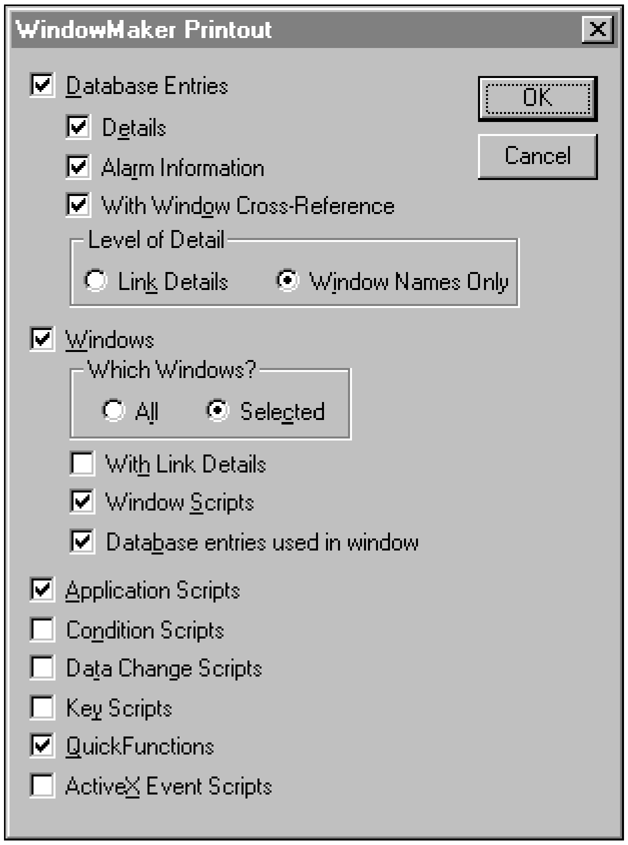
   For more information on converting to remote tagname references, see Chapter 4 - Tagname Dictionary.

Printing Scripts

You can print all scripts in each InTouch QuickScript category.

➢ **To print a script:**

1. On the **File** menu, click **Print**. The **WindowMaker Printout** dialog box appears:



WindowMaker Printout

- [✓] Database Entries
  - [✓] Details
  - [✓] Alarm Information
  - [✓] With Window Cross-Reference
  - Level of Detail: ○ Link Details [● Window Names Only]
- [✓] Windows
  - Which Windows?: ○ All [● Selected]
  - [ ] With Link Details
  - [✓] Window Scripts
  - [✓] Database entries used in window
- [✓] Application Scripts
- [ ] Condition Scripts
- [ ] Data Change Scripts
- [ ] Key Scripts
- [✓] QuickFunctions
- [ ] ActiveX Event Scripts

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 will appear. Select the window(s), whose script you want to print, and then click **OK**.

Note If you select a window that does not have a script linked to it, the following will be printed on the report: "Window Scripts for *Window Name*: none."

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

Script Functions

InTouch provides you with numerous built-in functions that you can be link to objects or pushbuttons or use in scripts to perform a multitude of tasks. For example, acknowledging alarms, hiding windows, changing the tagname being trended by a pen, and so on.

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, you highlight the argument you want to modify and type in the new value.

String Functions

String functions are used on string variables. The following briefly describes each string script function.

📖 For more information on the valid syntax for each function and examples of how you use each function, see your *InTouch Reference Guide.*

Function	Description
DText	Changes a message tagname based on the value of a discrete tagname.
StringASCII	Returns the ASCII value of the first character in a specified message tagname.
StringChar	Returns the character corresponding to a specified ASCII code.
StringFromIntg	Converts an integer value into its string representation in another base.
StringFromReal	Converts a real value into its string representation, either as a floating-point number or in exponential notation.
StringFromTime	Converts a time value (in seconds since Jan-01-1970) into a particular string representation.
StringInString	Returns the position in *Text* where *Search For* first occurs.
StringLeft	Returns the number of characters specified by Chars starting with the leftmost character of text.
StringLen	Returns the length of text to integer result.
StringLower	Converts all the uppercase characters in text to lower case, and places the resulting sting in MessageResult.
StringMid	Extract from text the specific numbers of characters specified by Chars, starting at the position *StartChar*. This function is slightly different from its counterparts **StringLeft()** function and **StringRight()** function in that it allows the user to specify both the start and end of the string which is to be extracted from the message tag.
StringReplace	Replaces or changes specific parts of a provided string..
StringRight	Returns the number of character specified by Chars starting at the rightmost character of text..
StringSpace	Generates a string of spaces either within a message tagname or an expression..

Function	Description
StringTest	Tests the first character of text to determine whether it is of a certain type..
StringToIntg	Converts the numeric value of a message tagname to an integer value to which mathematical calculations can be applied..
StringToReal	Converts the numeric value of a message tagname to a real (floating point) value to which mathematical calculations can be applied. .
StringTrim	Removes unwanted spaces from text..
StringUpper	Converts all the lowercase character in text to uppercase..
Text	Causes a message type tagname to display the value of an analog (integer or real) tagname based upon the specified ***Format_Text***..

Math Functions

Math functions are used on integer or real tagnames. In the following math functions, the **ResultNumericTags** and **InputNumericTags** can be either **Real** or **Integer** and freely intermixed. Keep in mind, however, that returning a non-integer result of a function to an **Integer** tagname will result in the truncation of the result. (The portion to the right of the decimal point will be lost.) The following examples assume that ***ResultNumericTag*** has been defined as either a **Memory Real** or **I/O Real** type tagname.

The following briefly describes each math script function.

🕮 For complete details on the valid syntax for each function and examples of how you each function, see your *InTouch Reference Guide.*

Function	Description
Abs	Returns the absolute value (unsigned equivalent) of a specified number.
ArcCos	Given a number between -1 and 1 (inclusive), returns an angle between 0 and 180 degrees whose *cosine* is equal to that number.
ArcSin	Given a number between -1 and 1 (inclusive), returns an angle between -90 and 90 degrees whose *sine* is equal to that number.
ArcTan	Given a number, returns an angle between -90 and 90 degrees whose *tangent* is equal to that number.
Cos	Returns the *cosine* of an angle given in degrees.
Exp	Returns the result of ***e*** raised to a power**.**
Int	Returns the next integer less than or equal to a specified number.
Log	Returns the log of a number.
LogN	Returns the values of the *logarithm* of ***x*** to base ***n***.
Pi	Returns the value of *Pi*.
Round	Rounds a real number to a specified precision.

Function	Description
Sgn	Determines the sign of a value (whether it is positive, zero, or negative).
Sin	Returns the *sine* of an angle given in degrees.
Sqrt	Returns the *square root* of a number.
Tan	Returns the *tangent* of an angle given in degrees.
Trunc	Truncates a real (floating point) number by simply eliminating the portion to the right of the decimal point.

System Functions

System functions are used to perform actions on your system such as activating another Windows application, copying, deleting or moving files and retrieving information regarding your application. There are two types of system functions; File and Info. The System File functions are used to read and write data from files. They each have two common parameters, ***Filename*** and ***FileOffset***.

The ***Filename*** refers to the name of the file which will be read from or written to. This filename must include the full path. The ***FileOffset*** refers to the position in the file where the read or write operation will begin (measured in bytes from the beginning of the file). The first byte of the file is ***FileOffset*** 0. Upon completion, each function returns the byte position directly following the data that was just read from or written to the file. For example, if the function reads 5 bytes of data starting at byte position 10, the function will return 15.

The ***FileOffset*** tagname can be used as both a parameter to the functions and as the return tagname. This can facilitate continuous operations.

Example:

FileOffset**=FileReadMessage(*****Filename,FileOffset,Message_Tagname,0*****);**
In the previous example, a line of text is read from ***Filename***. The starting location is specified by the original value of ***FileOffset*** (0 for instance, being the beginning of the file). The position where the next read will begin is then returned to ***FileOffset*** in preparation for the next call to **FileReadMessage()**. Every time the function is called, ***FileOffset*** gets larger and larger as the **FileReadMessage()** reads through the file.

The following briefly describes each system script function.

🕮 For complete details on the valid syntax for each function and examples of how you use each function, see your *InTouch Reference Guide.*

Function	Description
ActivateApp	Activates another currently running Windows application.
FileCopy	Copies a *SourceFile* to a *DestFile*, similar to the DOS Copy command or the Copy function in Windows File.
FileDelete	Deletes unnecessary or unwanted files.
FileMove	This is similar to **FileCopy()** except that it moves the file from one location to another instead of making a copy.
FileReadFields	Reads a Comma Separated Variable (CSV) record from a specified file.

Function	Description
FileReadMessage	Reads a specified number of bytes (or a whole line) from a specified file.
FileWriteFields	Writes a Comma Separated Variable (CSV) record to a specified file.
FileWriteMessage	Writes a specified number of bytes (or a whole line) to a specified file.
InfoAppActive	Tests whether an application is active.
InfoAppTitle	Returns the Application Title or Windows Task list name of a specified program which is currently running.
InfoDisk	Returns information about a specific local (or network) disk drive.
InfoFile	Returns information about a specific file or subdirectory.
InfoInTouchAppDir	Returns the current InTouch application directory.
InfoResources	Returns various system resource values as follows: **Case 1 and Case 2:** GDI and USER are hard-coded to return 50% on Windows NT and Windows 95. **Case 3:** On Windows NT and Windows 95 returns "free bytes of paging file." **Case 4:** On Windows NT and Windows 95 returns the result of search of all the top level windows. It only counts the windows that are visible and do not have any owners. This is not the actual "number of tasks currently running" in the system. Its closest approximation would be the count of items on **Applications** tab when you run the **Task Manager** in Windows NT or the **Close Program** window which comes up when you press CTRL+ALT+DEL in Windows 95.
IsAnyAsynchFunctionBusy	Used to find out if any asynchronous QuickFunctions are running. This function can be used to make the QuickScript that calls an asynchronous QuickFunction wait for all other asynchronous QuickFunctions to complete processing. This allows the QuickScript to re-synchronize itself.
StartApp	Automatically starts another Windows application.

Misc Functions

Miscellaneous functions are used to perform miscellaneous actions such as, hiding a window, monitoring and controlling historical trends, printing windows, sending keys to other applications, and so on.

The following briefly describes each miscellaneous script function.

& For complete details on the valid syntax for each function and examples of how you use each function, see your *InTouch Reference Guide.*

The function names that begin with "alm" are used for distributed alarm systems only. Function names beginning with "wc" are used for Windows Controls objects (list boxes, text boxes, combo boxes, and so on.). Function names beginning with "HT" are used for historical trend objects only.

Function	Description
Ack	Acknowledges any unacknowledged alarm. This function can be applied to a tagname, Alarm Group or Group Variable.
almAckAll	Acknowledge all alarms in current query including those not currently displayed in the alarm display object.
almAckDisplay	Acknowledge only those alarms currently visible in the alarm display object.
almAckRecent	Acknowledge the most recent alarm that has occurred.
almAckSelect	Acknowledge only those alarms selected in the alarm display object.
almDefQuery	Performs a query to update an alarm display object using the default properties.
almMoveWindow	Scrolls the alarm display object window.
almQuery	Performs a query to update an alarm display object.
almSelectAll	Toggles the selection of all the alarms in an alarm display object.
almSelectItem	Toggles the selection of the item that is highlighted in an alarm display object.
almShowStats	Displays the alarm display object statistics screen.
ChangePassword	Displays the **Change Password** dialog box allowing the logged on operator to change his/her password.
DialogStringEntry	Displays an alphanumeric keyboard on the screen, allowing the operator to change the current string value of a message tagname in the Tagname Dictionary.
DialogValueEntry	Displays the numeric keypad on the screen, allowing the operator to change the current value of a discrete, integer or real tagname in the Tagname Dictionary.
GetNodeName	Returns the NetDDE node name to a string variable.
GetPropertyD	Retrieves the specified property’ s discrete value during runtime.

Function	Description
GetPropertyI	Retrieves the specified property' s integer value during runtime.
GetPropertyM	Retrieves the specified property' s message value during runtime.
Hide	Hides various windows from within a script. A **Hide()** function must precede the name of each window to be hidden.
HideSelf	Hides the currently active window.
HTGetLastError	Determines if there was an error during the last retrieval of a specified pen.
HTGetPenName	Returns the tagname 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*.
HTGetTimeStringAtScooter	Returns the string containing the time/date for the sample at the scooter location specified by *ScootNum* and *ScootLoc*.
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 #.
HTGetValueAtZone	Returns a value of the requested type 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.
HTSelectTag	Displays the **Select Tag** dialog box and the operator can select a different tagname for specified pen. (The dialog box only lists the tagnames that have been defined for historical logging (**Log Data** option selected) in the Tagname Dictionary.)
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.

Function	Description
HTZoomIn	Calculates a new chart width and start time. If the trend's **.ScooterPosLeft** is 0.0 and the **.ScooterPosRight** is 1.0, then the new chart width equals the old chart width divided by two.
HTZoomOut	Calculates a new chart width and start time. The new chart width is the old chart width multiplied by two.
IOSetAccessName	Modifies the *application* or *topic* name portions of an Access Name during runtime which allows implementing of hot-backup strategies for InTouch.
IOSetItem	Changes the access name and/or item in a I/O type tagname ***.Reference*** field.
LogMessage	Writes a user-defined message to the Wonderware Logger.
PlaySound	Plays a wave form sound specified by a .wav filename or an entry in the **[sounds]** section of the WIN.INI file through the Windows sound device (if installed).
PrintHT	Can be used in a button for printing the Historical Trend chart associated with the specified Hist Trend type tagname. The Historical Trend must be visible on the screen when using this function.
PrintWindow	Prints the specified window.
RestartWindowViewer	Allows the user control over shutting down and restarting WindowViewer.
SendKeys	Sends keys to an application.
SetDdeAppTopic	This function is replaced by **IOSetAccessName** beginning with InTouch Version 7.0. See **IOSetAccessName**.
SetDdeItem	This function is replaced by **IOSetItem** beginning with InTouch Version 7.0. See **IOSetItem**.
SetPropertyD	Specifies the property's discrete value that is to be written during runtime.
SetPropertyI	Specifies the property's integer value that is to be written during runtime.
SetPropertyM	Specifies the property's message value that is to be written during runtime.
Show	Displays a specified window. (Window name must be enclosed in quotation marks.)
ShowAt	Specifies the horizontal and vertical pixel location of a window when it is shown. When the window opens, it will be centered on the horizontal and vertical coordinates.

Function	Description
ShowHome	Displays the "home" window(s). Home windows are those you selected to open automatically when WindowViewer is started. (The home windows are selected in the **WindowViewer Properties - Home Windows** property sheet.) For more information on home windows, see Chapter 2 - Using WindowMaker.
ShowTopLeftAt	Specifies the horizontal and vertical pixel location of the top left corner of a window when it is shown. When the window opens, its top left corner will be positioned where the horizontal and vertical coordinates meet.
wcAddItem	Adds the supplied string to the list box or combo box.
wcClear	Removes all items from the list box or combo box.
wcDeleteItem	Deletes the item associated with the item index argument in both list or combo boxes.
wcDeleteSelection	Deletes the currently selected item from the list. Applies to list boxes and combo boxes
wcErrorMessage	Given an error number, **wcErrorMessage()**, returns a string message describing the error. Applies to list boxes, text boxes, combo boxes, radio buttons and check boxes.
wcFindItem	Determines the corresponding index of the first item in the list box or combo box that matches the supplied string.
wcGetItem	Returns the value property of an item string associated with a corresponding index in a list box or combo box.
wcGetItemData	Retrieves the integer value associated with a list item in a list box or combo boxes.
wcInsertItem	Inserts a string into a list box or combo box.
wcLoadlist	Replaces the contents of the list box or combo box with new items.
wcLoadText	Replaces the contents of the text box with a new string.
wcSavelist	Replaces the contents of a filename with the items in a list object.
wcSaveText	Saves the text contained in a text box to a filename.
wcSetItemData	Assigns an integer value to an item in a list box.

WW DDE Functions

You should not use the WW DDE functions as a replacement for normal InTouch DDE communications. Whenever possible, you should create an DDE type tagname to get data from or send data to an external application. The WW DDE functions are intended to support applications that cannot communicate using the DDE Advises supported by InTouch. For example, some applications support only DDE Executes or Pokes.

The **WWExecute()**, **WWPoke()** and **WWRequest()** functions use the same Windows functions as MS Visual Basic (DDEML). A single function actually does several things. For example, a **WWPoke()** will perform a DDE Initiate, a DDE Poke and a DDE Terminate all in one function. This makes WW DDE functions less error prone, but also less efficient in processing many DDE messages. As general guidelines for the use of these functions, you should never:

- Loop these functions (call them over and over).
- Call several of the DDE functions in a row and in the same script.
- Use them to call a lengthy task in another DDE application.

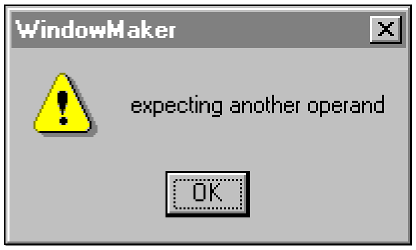
If the DDE command executes a lengthy task in another application it can use up all of the available processor time. However, even if communication difficulties occur, no loss of data will occur. Even if the I/O Server cannot send messages to InTouch, it will continue to try.

The following briefly describes each script function. For details on the valid syntax for each function and examples of how to use each function, see your *InTouch Reference Guide.*

Function	Description
WWControl	Allows you to Restore, Minimize, Maximize, or Close an application from InTouch.
WWExecute	Sends a command (using a DDE Execute) to a specified *Application* and *Topic*.
WWPoke	Pokes a value (using an DDE Poke) to a specified *Application*, *Topic*, and *Item*.
WWRequest	Makes a one-time request for a value (using an DDE Request) from a particular *Application*, *Topic*, and *Item*.

Script Editor Error Messages

If the script editor detects any errors in the script when it is validated a corresponding message box will be displayed. For example:



expecting another operand

In most cases, when an error is encountered, InTouch will place the cursor at the position in the script where the error has occurred. However, in some cases, such as a missing ENDIF, the cursor will be at the end of the script. All errors must be corrected before the system will accept the script.

Error Message	Definition
Can only compare alarm groups for equality	Cannot compare alarm groups for <, >, <=, >=.
Cannot add, subtract, multiply or divide with strings	These operations are not supported for strings.
Cannot mix another type with alarm group	Trying to compare an Alarm Group with another type (e.g. integer) or trying to use something other than an Alarm Group somewhere where an Alarm Group is expected.
Cannot mix another type with string	Trying to compare a string with another type (e.g. integer) or trying to use something other than a string somewhere where a string is expected.
Cannot negate alarm groups	Minus sign (-) has been used.
Cannot negate Access Name	A "-" or "~" is not allowed prior to a DDE Access Name.
Cannot negate strings	Minus sign (-) has been used.
Cannot negate TagID	Minus sign (-) has been used.
Cannot negate window	A "-" or "~" is not allowed prior to a window name.
Cannot use Access Name in this manner	A DDE Access Name cannot be used in this context.
Cannot use HistTrendTag in this manner	Trying to compare a string with another type (e.g. integer) or trying to use something other than a string somewhere where a string is expected.
Cannot use TagID in this manner	A TagID type variable cannot be used in this context.
Cannot use window name in this manner	A window name cannot be used in this context.

Error Message	Definition
E Format must be between -38 and +38	The maximum "e" format number is between e-38 and e+38.
E format must have digit following E	Valid "e" format is n.nnen, 1.e is not legal.
Expecting ")" after function arguments for	A matching right parenthesis is required to match the left parenthesis following the function name.
Expecting "(" after function name for	A left parenthesis is required after this function name.
Expecting a number after 0x	Hexadecimal (base 16) numbers can be entered in **InTouch**. To start a hexadecimal number, start with 0x and follow with digits.
Expecting an expression after IF	Missing discrete expression.
Expecting analog argument for function	This argument for this function requires an analog value.
Expecting another argument for	The function requires more arguments than are present.
Expecting another operand	If "a + " is entered, **InTouch** will display this error message.
Expecting assignment	In an action script, a tagname was entered and the next logical operation would be an assignment.
Expecting comma and other argument(s) to function	More arguments are required for this function.
Expecting DLL Name	A DLL name must be used in this context.
Expecting ENDIF	Must have an ENDIF for each IF.
Expecting ENDIF or ELSE	Every IF/THEN must be matched with an ENDIF or ELSE.
Expecting expression after assignment (=)	In an action script, a tagname and assignment were entered and no value was given for the assignment. This can also happen if => is entered instead of >=.
Expecting Function Name	A Function Name must be used in this context.
Expecting name	Expecting a tagname for this argument.
Expecting name in statement	Missing name in statement.
Expecting right parentheses	A matching ")" was not found.
Expecting semicolon	Semicolon is missing at end of line.
Expecting string	The given argument must be a string expression (i.e. the name of a string tagname or a constant string (text surrounded by double quotes ("')).

Error Message	Definition
Expecting THEN	THEN missing after IF statement.
Expecting window name - must be string expression	The given argument must be a string expression (i.e. the name of a string tagname or a constant string (text surrounded by double quotes (")).
Extra expressions	For example, the expression "a b" is not legal, "a + b" is OK.
Function only legal in action scripts or logic	Some functions are only legal in scripts, not in expressions.
IF expression must be discrete (use == instead of =)	This error is received because of using the assignment (=) instead of comparison (==). For example, "IF a = b THEN ..." should be "IF a == b THEN ...". May also be received if "IF x THEN..." is used and x is not a discrete tagname.
Invalid or missing operand	The operand required for an operator is either invalid or missing.
Invalid placeholder name - must have chars follow ?x:	Describing characters must follow ?x: in placeholder name.
Invalid placeholder name - second char must be d, i, a, r, m, v, g, h, t	Describing 2nd digit character is invalid for placeholder name.
Invalid placeholder name - third char must be ":"	Describing 3rd digit character is invalid for placeholder name.
Logical AND/OR must use discrete	Using AND/OR operators must be done with discrete expressions. Thus "x AND y" is OK, if x and y are discrete tagnames; if they are of any other type, this error message will be received.
Logical NOT must use discrete	Using the NOT operator must be done with discrete expressions. Thus, "NOT x" is OK if x is a discrete tagname, but if x is of any type other than discrete, this error message will appear.
Maximum string 131 characters	The string is longer than the max allowed.
Must assign the return value of function	Certain functions require function evaluation of the return value.
Must have a digit after decimal point	The syntax "1." is not legal.
Must have hist trend variable for this argument	A Hist Trend type variable must be used in this context.
Must have writeable analog variable or name.field for this argument	The argument must be an integer or real variable or integer or real **.field** of a variable.

Error Message	Definition
Must have writeable discrete variable for this argument	The argument for this function must be a tagname whose type is discrete and for which read-only is NOT checked.
Must have writeable integer variable for this argument	The argument for this function must be a tagname whose type is integer and for which read-only is NOT checked.
Must have writeable message variable for this argument	The argument for this function must be a tagname whose type is message and for which read-only is NOT checked.
Must have writeable real variable for this argument	The argument for this function must be a tagname whose type is real and for which read-only is NOT checked.
Must have writeable variable for this argument	The argument for this function must be a tagname for which read-only is NOT checked.
Must not assign the return value of function	Certain functions do not return a value and a return value cannot be evaluated for them.
Name too long	Tagnames must be <= 32.
No closing comment	Opening comment delimiters ({) must be matched with closing comment delimiters (}).
No closing quote	Missing closing quotation mark (").
Not enough room in display buffer	Not enough memory for this operation. Clear up memory and this operation will complete.
Not enough room in expression buffer	Not enough memory for this operation at this time. Clear up memory and this operation could complete.
Number too large	Absolute value of numbers must be < 2e38.
Number too small	Absolute value of numbers must be > 2e-38.
Only 8 digits allowed after 0x	When entering a hexadecimal number, only 8 digits are allowed.
Too many arguments	Supplied more arguments to this function than are necessary.
Trying to assign to a read-only name	It is not legal to assign a value to a tagname for which read-only is checked.
Undefined field name	The .field name is not defined, most likely due to a spelling error.
Unrecognized character	The highlighted character is not a legal character for expressions or action scripts.
Using a reserved field name (e.g., SP) for normal tagname	Can't use a field name for tagname.

Error Messages for Windows Controls and Distributed Alarms

The Window Controls and distributed alarm script functions return a value based on the result of processing the script function. The return value, used for error diagnostics, can be assigned to an Integer tagname. For example:

```
ErrorNumber = wcGetItem("ControlName", Number, Tagname);
```

Where:

ErrorNumber is an **Integer** tagname type that will hold the returned error value. The return value of the function can be passed to the **wcErrorMessage()**. The **wcErrorMessage()** will return a string description of the error. For example:

```
ErrorMsg = wcErrorMessge(ErrorNumber);
```

Where:

ErrorMsg is a **Message** type tagname that holds the text description of the returned error. The following table identifies the numeric value and its description:

Error Message	Description
0	Success
-1	General failure
-2	Insufficient memory available
-3	Property is read-only
-4	Specified item already present
-5	Object name unknown
-6	Property name unknown
-*x**	Unknown error

* -*x* represents any other number.

C H A P T E R 7

Alarms/Events

InTouch provides a notification system to inform operators of process and system conditions. This system supports the displaying, logging, and printing of process alarms and system events. Alarms represent warnings of process conditions, while events represent normal system status messages.

InTouch includes two alarm systems: a standard system and a distributed system. The standard system is used to display and acknowledge events and alarms generated by the local InTouch application. The distributed system expands this scope to allow the display and acknowledgment of alarms generated by alarm systems of other networked InTouch applications.

This chapter describes the two alarm systems, the various types of alarm conditions, and the grouping hierarchies. Specific sections cover how you add, modify and delete Alarm Groups, assign tagnames to Alarm Groups, define the alarm conditions for a tagname, display, log and print alarms, and how you configure both alarm systems.

Contents

Alarms and Events

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 print it out to a printer. Once the operator acknowledges the alarm, the system returns to an *acknowledged* state.

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. If configured to do so, InTouch can log an event to a disk-based file and print it out to a printer.

You can configure any tagname to do event monitoring while you are defining it in the Tagname Dictionary. 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, whether the change was initiated by the operator, I/O, scripts or the system.

For more information on configuring a tagname to do event monitoring, see Chapter 4 - Tagname Dictionary.

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 distributed alarm system provides further categorization of these alarms into *Class* and *Type*. The table below summarizes the classification for both systems:

Alarm Condition	Standard Type	Distributed Class	Distributed Type
Discrete	DISC	DSC	DSC
Deviation - Major	LDEV	DEV	MAJDEV
Deviation - Minor	SDEV	DEV	MINDEV
Rate-of-Change	ROC	ROC	ROC
SPC	SPC	SPC	SPC
Value - LoLo	LOLO	VALUE	LOLO
Value - Low	LO	VALUE	LO
Value - High	HI	VALUE	HI
Value - HiHi	HIHI	VALUE	HIHI

You can associate each alarm with an InTouch tagname. Depending upon a tagname' s type, you can define one or more of the alarm classes or types for it. You define your alarm conditions in the Tagname Dictionary.

For information on defining alarm conditions, see Chapter 4 - Tagname Dictionary.

You can also configure alarm logging, printing, and a standard display to show the alarm *Type* field. The distributed display can also show the alarm *Class* field.

Event Types

InTouch also classifies events into general categories based on their characteristics. These categories are known as *Event Types*. Both the standard and distributed alarm systems use the same *Event Types*. The table below summarizes the classification for both systems:

Event	Condition
ACK	Alarm was acknowledged
ALM	Alarm has occurred
EVT	An alarm event occurred
RTN	Tagname returned from an alarm state to a normal state
SYS	A system event occurred
USER	$Operator changed
DDE	The tagname value was poked from a DDE client
LGC	A QuickScript modified the tagname value
OPR	The operator modified the tagname value using the Value Input

The first six events listed are configured automatically when event logging is enabled. The remaining three you must define for each tagname in the Tagname Dictionary.

For more information, on events, see "Alarms and Events."

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 ones. You can also create animation links, acknowledgment scripts, and filtered viewing and printing all based on the priority range.

For example, if a process plant has determined that they need four levels of severity, they could establish ranges as shown below:

Alarm Severity	Priority Range
Critical	0 - 249
Major	250 - 499
Minor	500 - 749
Advisory	750 - 999

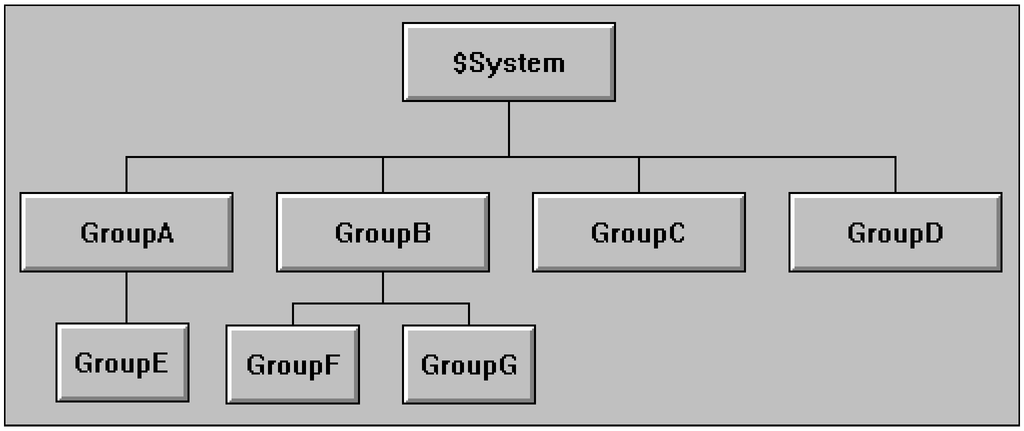
As the plant engineers create InTouch tagnames and alarm conditions, each alarm will be assigned to one of these severity levels by choosing a priority number within that range. With these ranges configured, the plant operators may now easily display and print only certain severity levels.

Alarm Groups

Each InTouch alarm is assigned to a logical Alarm Group. These groups are user-definable and can be arranged into 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 you choose. Alarm Groups are useful for filtering alarm displays, alarm printers, and acknowledgment scripts.

Every tagname is associated with an Alarm Group. If you do not associate an Alarm Group name to a tagname, by default, InTouch automatically associates it with the root group, **$System**. Any Alarm Group may have both tagnames and other Alarm Group names associated with it. Alarm Groups are organized into a hierarchical tree structure with the root group, **$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.



This illustration displays only Alarm Groups, not the tagnames within each group. This 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 it’s Alarm Group Lists.

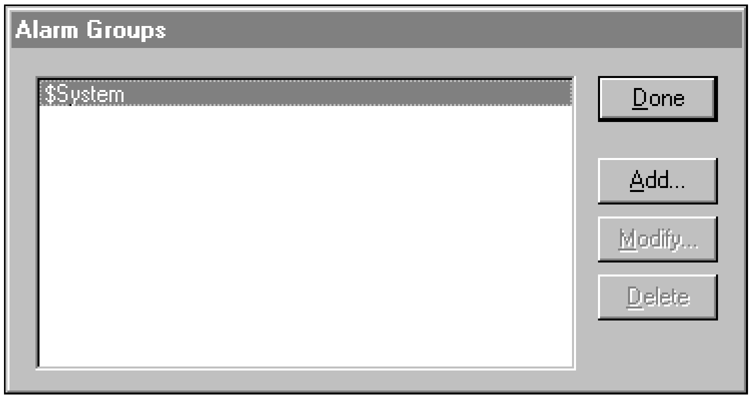
For more information, see "Distributed Alarm Group Lists."

Note While Alarm Groups do not count as tagnames with InTouch licensing, they do count as tagnames in the database. Therefore, the total number of Alarm Groups plus actual tagnames cannot exceed 61,405.

➢ **To create an Alarm Group:**

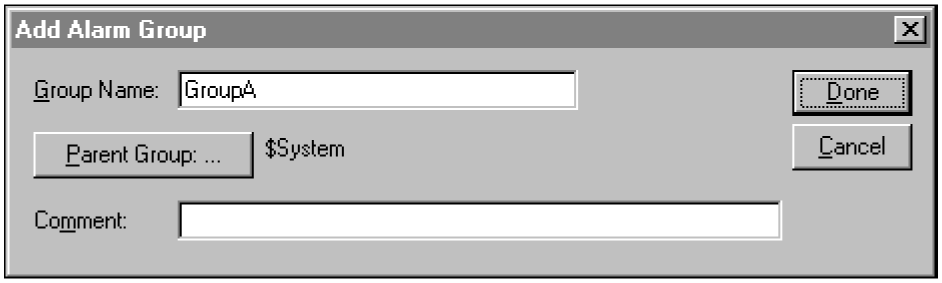
1. On the **Special** menu, click **Alarm Groups**. The **Alarm Group Definition** dialog box appears:

   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.

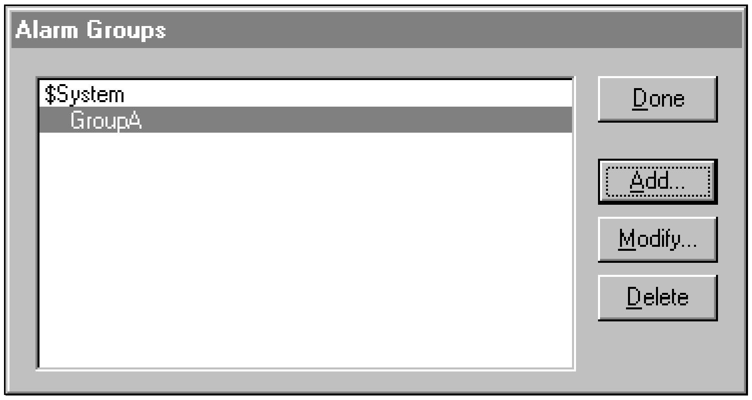


   If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

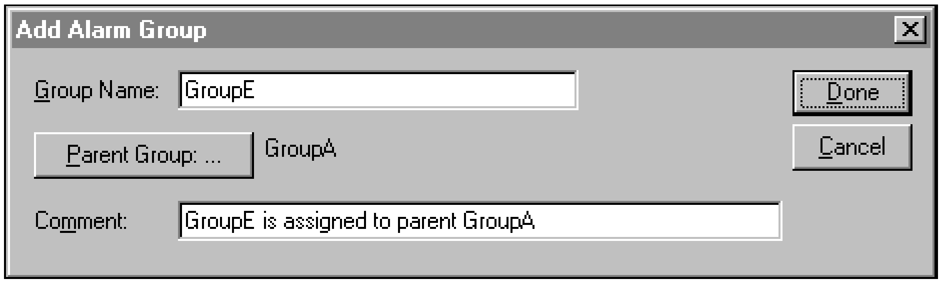
3. In the **Group Name** box, type the name for the new Alarm Group. Since this is the first Alarm Group you have created, it is automatically assigned to the **$System** Parent Group.

   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:

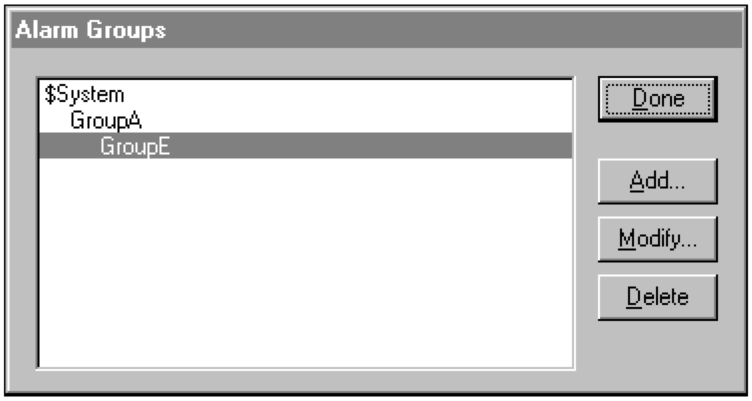


5. In the **Select an Alarm Group** list, double-click the name of the Alarm Group that you want to use as the Parent Group (or select it, then click **Done**) for the new Alarm Group. The **Add Alarm Group** dialog box reappears displaying the selected Parent Group. For example:



Group Name: GroupE
Parent Group: GroupA
Comment: GroupE is assigned to parent GroupA

6. In the **Comment** box, type any comment for the new Alarm Group.
7. Click **Close**. The **Alarm Group Definition** dialog box appears displaying your Alarm Group hierarchy:



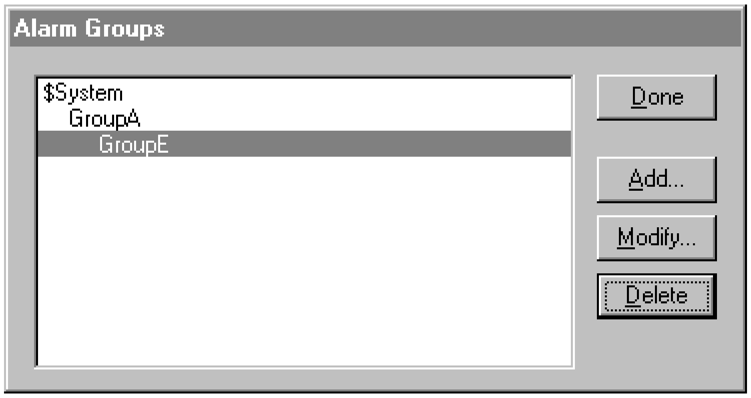
$System
GroupA
GroupE

8. Click **Done**.

➢ **To delete an Alarm Group:**

1. On the **Special** menu, click **Alarm Groups**. The **Alarm Group Definition** dialog box appears:

   You can also delete Alarm Groups while you are defining your tagnames in the Tagname Dictionary.



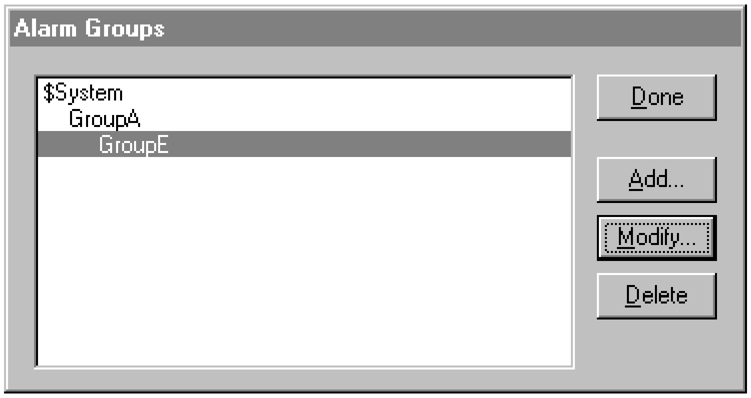
$System
GroupA
GroupE

2. Select the Alarm Group that you want 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, or click **No**, to cancel the deletion.
3. Click **Done**.

➢ **To modify an Alarm Group:**

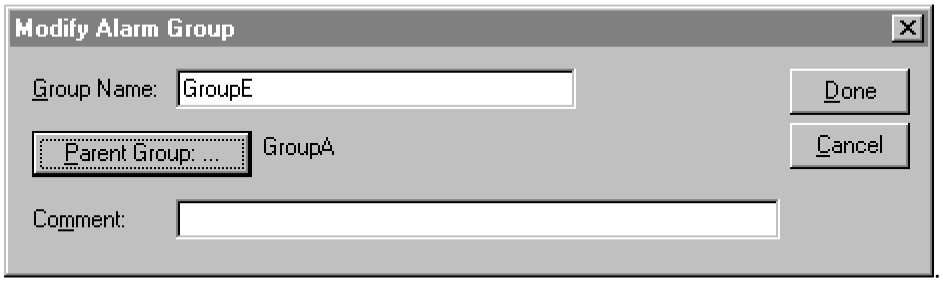
1. On the **Special** menu, click **Alarm Groups**. The **Alarm Group Definition** dialog box appears:

   You can also modify Alarm Groups while you are defining your tagnames in the Tagname Dictionary.



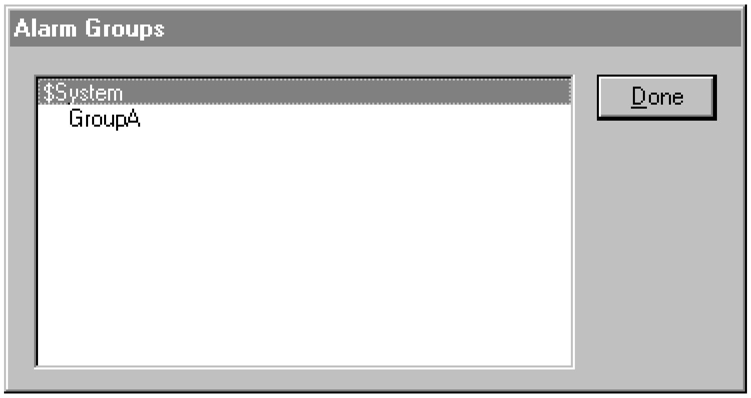
$System
  GroupA
    GroupE

2. Select the Alarm Group that you want to modify in the list, and then click **Modify**. The **Modify Alarm Group** dialog box appears:



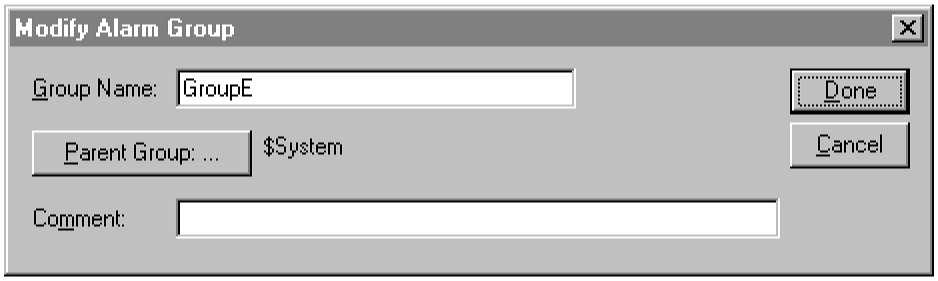
Group Name: GroupE
Parent Group: GroupA
Comment:

3. Make the required changes to the Alarm Group. If you want to change the parent group for the Alarm Group, click **Parent Group**. The **Alarm Groups** dialog box appears:



$System
  GroupA

4. Select the new parent group, and then click **Done**. The **Modify Alarm Group** dialog box reappears displaying the new parent group.



Group Name: GroupE

Parent Group: $System

Comment:

5. Click **Done**.

Defining Tagname Alarm Conditions

You define alarm conditions in the Tagname Dictionary. Therefore, you can define them at the same time that you are defining the tagname. You can define alarm conditions for discrete type tagnames and analog (integer or real) type tagnames.

For more information on defining alarm conditions, see Chapter 4 - Tagname Dictionary.

The Standard Alarm Display

The standard alarm system provides you with a unique display object that you use to show locally generated alarms. While the distributed alarm system provides you with a display object that can show alarms generated both locally and remotely.

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.

👓 For more information, see the "Configuring the Standard Alarm System."

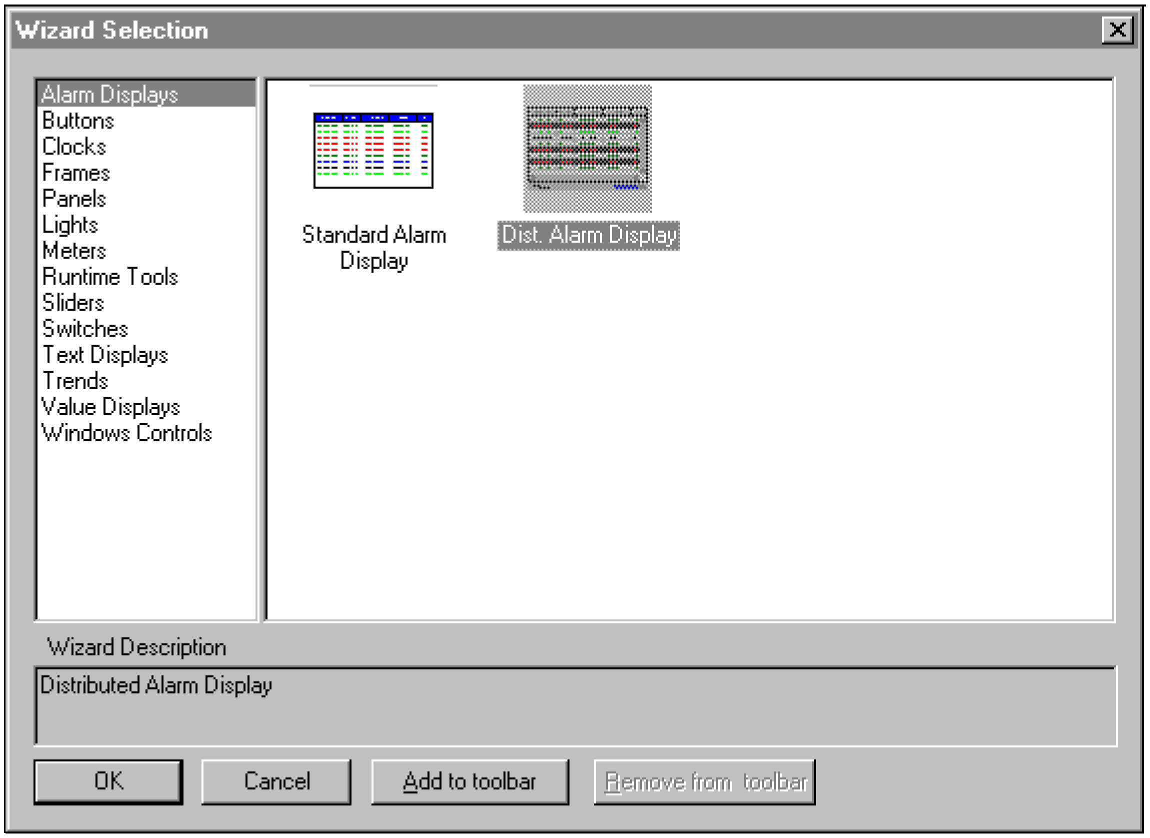
In both the Alarm Summary and the Alarm History display objects, each entry is shown as a separate line. The number of entries displayed is determined by the size you have drawn the object and the size of the font that you are using. The standard alarm display lists all active alarms or subsets of active 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 how many alarms are stored for the Alarm History object, 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

➤ **To create a standard alarm display:**

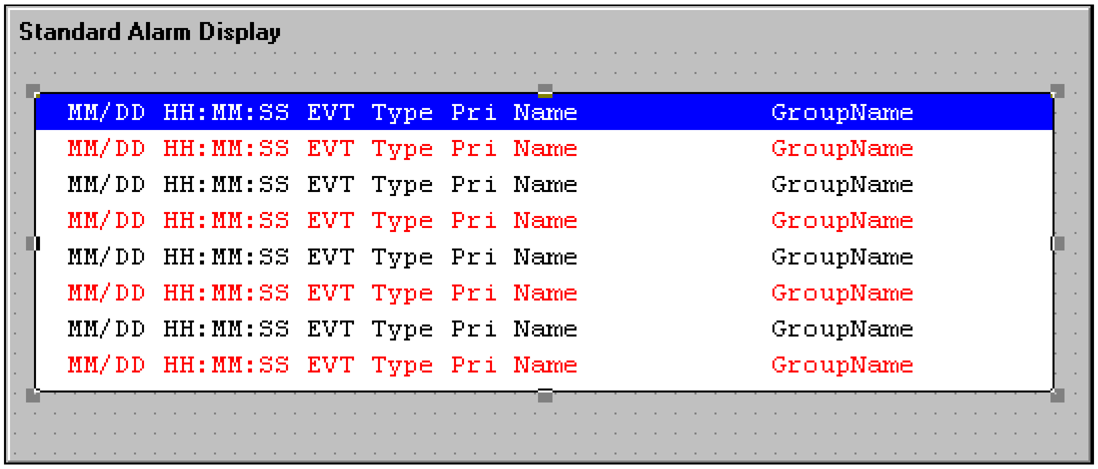
1. Click the wizard tool in the **Wizard/ActiveX Toolbar**. The **Wizard Selection** dialog box appears:



Distributed Alarm Display

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:



```
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
MM/DD HH:MM:SS EVT Type Pri Name          GroupName
```

- An alarm display object is like any other object drawn in WindowMaker. It can be moved by grabbing it with the mouse or it can be resized by grabbing one of the object "handles." The text that appears when the alarm display is initially drawn is sample text to show you how the actual display will look at runtime. You can place multiple alarm displays in a window.

4. You are now ready to configure the alarm display object.

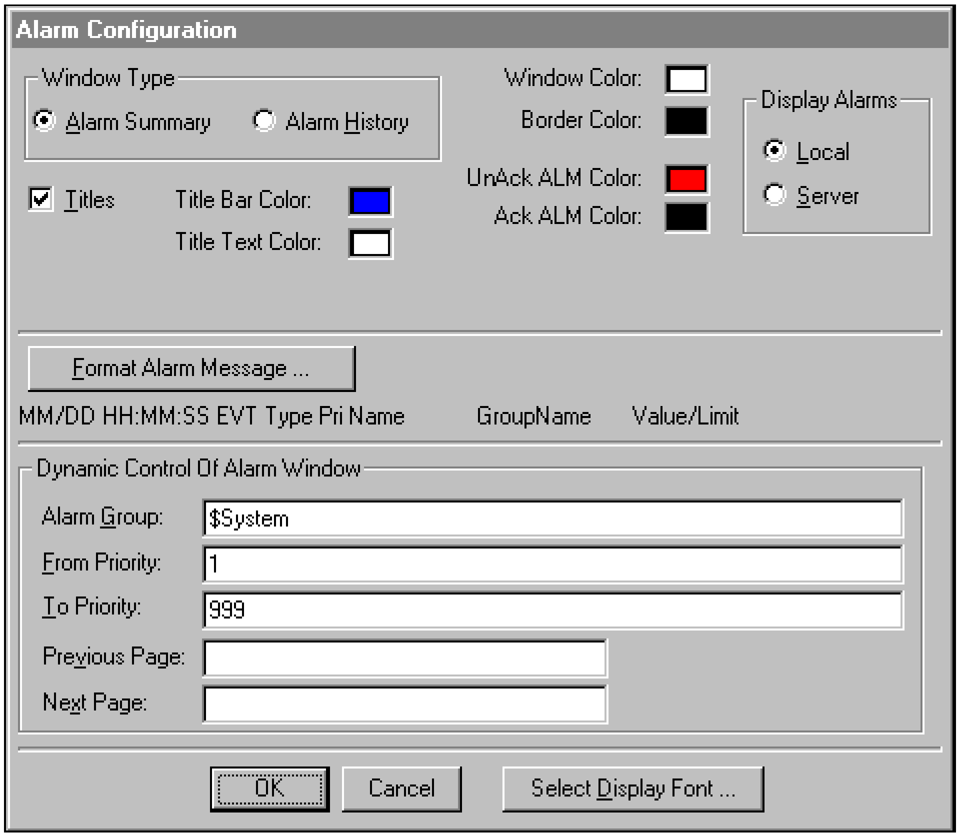
Configuring a Standard Alarm Display

The first time you paste an alarm object, the system default configuration settings are used. Once you have configured an alarm object, the next one you create will, by default, be configured with the same settings.

➢ **To configure a standard alarm display object:**

1. Double-click the alarm display or, with the alarm display selected, on the **Special** menu, click **Animation Links**. The **Alarm Configuration** dialog box appears:

   - To quickly access the dialog box, right-click the alarm display object, and then click **Properties**.


Alarm Configuration

Window Type: [✓ Alarm Summary] Alarm History

Window Color: ⬜
Border Color: ⬛
UnAck ALM Color: 🟥
Ack ALM Color: ⬛

Display Alarms: [✓ Local] Server

[✓ Titles] Title Bar Color: 🟦 Title Text Color: ⬜

Format Alarm Message ...

MM/DD HH:MM:SS EVT Type Pri Name GroupName Value/Limit

Dynamic Control Of Alarm Window

Alarm Group: $System
From Priority: 1
To Priority: 999
Previous Page:
Next Page:

OK Cancel Select Display Font ...

   - If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Select the **Window Type** for the alarm display that you want 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** if you want 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 open the InTouch color palette. Click the color in the palette that you want to use for the object's title bar.

   Repeat this process for all displayed color selections.

5. Select the **Display Alarms** type that you want to use:

   Local - To display locally generated alarms and events.
   Server - To display the alarms/events collected by the server node.

   You define the server node in the **WindowViewer Properties - General** property sheet.

   To quickly access the **WindowViewer Properties**, in the tree view pane, under **Configure**, select **WindowViewer**.

   For more information on remote alarming, see "Using the Standard Alarm System for Remote Alarming."

6. If you want to specify a specific group of alarms that you want to be logged, in the **Alarm Group** box, type an Alarm Group name or the name of a Group Variable. If you want to log all Alarm Groups, type **$System**.

   Note A Group Variable is a tagname that is defined as a **Group Var** type with the name of an Alarm Group assigned to it.

   If you want to control the choice of alarms logged at runtime, create a Group Variable type tagname, for example, **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, type the highest and lowest alarm priority level for the range of priorities that you want shown in the alarm display.

   Note The lower the number, the higher the priority of the alarm. For example, 1 is the highest priority. If you use the default values, all priorities will be used.

   You can type an analog tagname or an expression in either of the priority boxes if you want a tagname's value to determine the priority level to be logged. You can control the alarm priority level being 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, type the discrete tagname that you want to use to page up and down through the list of alarm messages when there are more alarms than the window can display.

   Warning! Do not use remote tagname references as **Previous Page** or **Next Page** discrete tagnames. This will create a problem for importing and exporting windows to other applications.

9. Click **Select Display Font** to open the **Font** dialog box to choose the font, its style and size that you want used for the messages displayed in the alarm display object.

   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 you want shown for each alarm message in the alarm display.

    For more information on formatting alarm messages, see "Standard Alarm/Event Message Format."

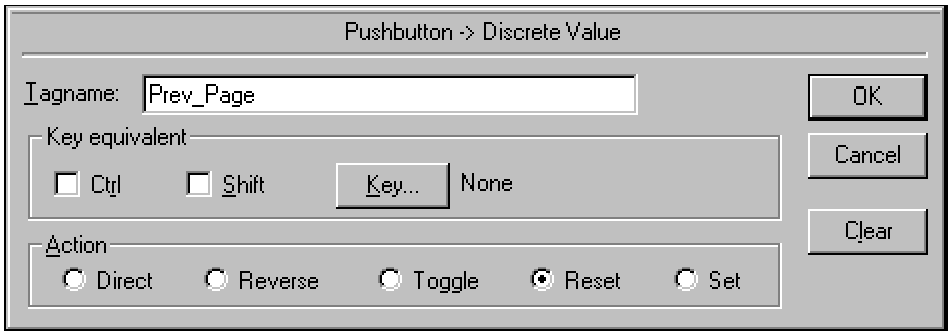
    After you format your alarm messages, check your display object to make sure that you have drawn it large enough to show all of your choices. If it is not big enough, the text at the far right side will be truncated.

11. Click **OK** to save your settings and close the **Alarm Configuration** dialog box.

Previous Page and Next Page Buttons

➢ **To create alarm object page up button:**

1. Create an object such as a 3-D button.
2. Double-click the object and select **Discrete Value Pushbutton** in the animation link selection dialog box. The **Pushbutton -> Discrete Value** dialog box appears:



Pushbutton -> Discrete Value

Tagname: Prev_Page

Key equivalent: [ ] Ctrl [ ] Shift Key... None

Action: ( ) Direct ( ) Reverse ( ) Toggle (● Reset) ( ) Set

3. In the **Expression** box type the discrete tagname that you typed in the **Previous Page** box when you configured your alarm object.
4. In the **Action** group, select **Reset**.
5. Click **OK**.

    Whenever the value of the discrete tagname transitions from On (1, True) to Off (0, False), the alarm display object will display the previous page. Once the previous page is displayed, the discrete tagname will automatically reset to On (1, True), unless the top of the list has been reached. In this case, the value of the variable remains Off and the previous page action is disabled.

    You can also add a visibility link on the button to hide it when the value of the discrete tagname is 1 (Off).

6. To create a "Next Page" button, repeat this procedure using the discrete tagname that you enter in the **Next Page** box when configuring the alarm object.

Warning! Do not use remote tagname references as **Previous Page** or **Next Page** discrete tagnames. This will create a problem for importing and exporting windows to other applications.

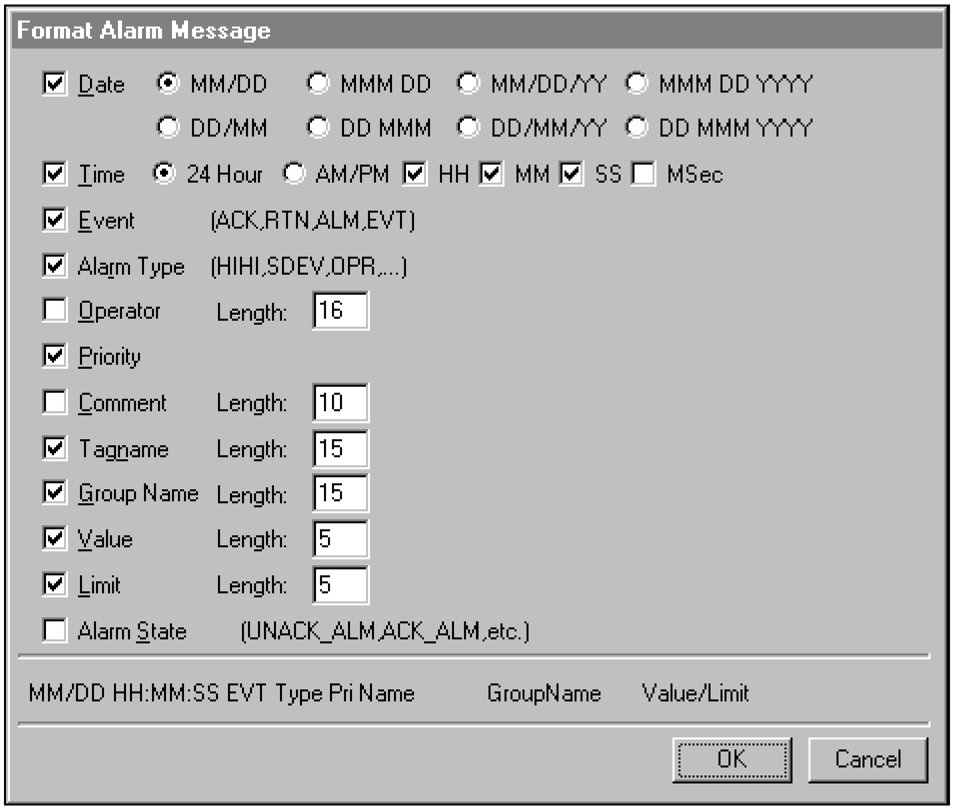
Standard Alarm/Event Message Format

The information shown in an alarm display object, logged to disk or printed is configurable. This configuration process involves selecting what information you want displayed and, in some cases, how many characters you want to display.

The order of appearance of the items in the message is fixed and cannot be altered.

➢ **To format alarm messages:**

1. Double-click the alarm display or right-click it, and then click **Properties**. The **Alarm Configuration** dialog box will appear.
2. Click **Format Alarm Message**. The **Format Alarm Message** dialog box appears:



Format Alarm Message

[✓] Date [● MM/DD] ○ MMM DD ○ MM/DD/YY ○ MMM DD YYYY ○ DD/MM ○ DD MMM ○ DD/MM/YY ○ DD MMM YYYY

[✓] Time [● 24 Hour] ○ AM/PM [✓] HH [✓] MM [✓] SS [ ] MSec

[✓] Event (ACK,RTN,ALM,EVT)

[✓] Alarm Type (HIHI,SDEV,OPR,...)

[ ] Operator Length: 16

[✓] Priority

[ ] Comment Length: 10

[✓] Tagname Length: 15

[✓] Group Name Length: 15

[✓] Value Length: 5

[✓] Limit Length: 5

[ ] Alarm State (UNACK_ALM,ACK_ALM,etc.)

MM/DD HH:MM:SS EVT Type Pri Name GroupName Value/Limit

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

Note The preview area (at the bottom of the dialog) displays an example of the alarm message as currently configured. This example will show the message using the font selected, but not color.

3. Select **Date** if you want to display the date in the alarm message, and then 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** if you want to display the time in the alarm message, and then select the format for the time as follows:

24 Hour	Selects the 24 hour military time format. For example, three o'clock in the afternoon is displayed as 15:00.
AM/PM	Selects the AM/PM format. For example, 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** if you want to display the event type. The event types are:

ACK	Displayed when the alarm has been acknowledged.
RTN	Displayed when the alarm condition returns to normal.
ALM	Displayed when the tagname is in alarm state.
EVT	Displayed 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** if you want to display the alarm type. The alarm types are:

HIHI, etc.	Displayed for Alarm Value conditions.
SDEV	Displayed for Minor Deviation Alarm conditions.
LDEV	Displayed for Major Deviation Alarm conditions.
OPR	Displayed if an operator change caused the alarm condition.

7. Select **Operator** if you want to display the logged-on operator's ID associated with the alarm condition. Enter a value in the Length field to control the number of characters displayed (16 characters maximum).
8. Select **Priority** if you want to display the alarm priority.
9. Select **Comment** if you want to display the tagname's comments. These comments were typed in the **Comments** box when the tagname was defined in the database. In the **Length** box, type the number of characters that you want displayed (50 characters maximum).
10. Select **TagName** if you want to display the tagname. In the **Length** box, type the number of characters that you want displayed (32 characters maximum).
11. Select **Group Name** if you want to display the Alarm Group name. In the **Length** box, type the number of characters that you want displayed (32 characters maximum).
12. Select **Value** if you want to display the value of the tagname when the alarm occurred. In the **Length** box, type the number of characters that you want displayed.

    The value should be large enough to provide the desired level of precision (15 characters maximum).

13. Select **Limit** if you want to display the alarm limit value of the tagname. In the **Length** box, type the number of characters that you want displayed.

    The size of this field should be large enough to provide the desired level of precision (32 characters maximum).

14. Select **Alarm State** if you want to display the state (unacknowledged, acknowledged, etc.) of the alarm.

Configuring the Standard Alarm System

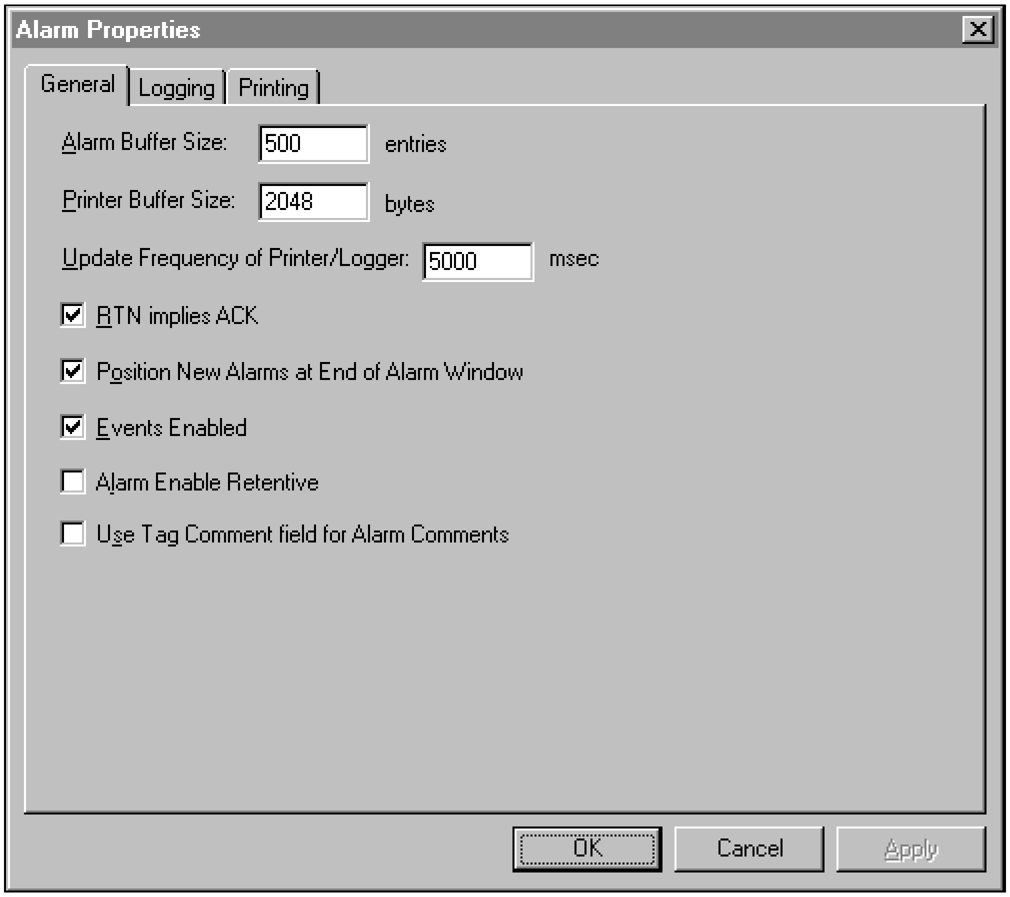
You can configure various parameters for the alarm system, such as the printer and logger buffer size, whether to enable events, the position of new alarms in the alarm display and so on. You can also control, logging and printing properties for alarms and events.

Note The configuration dialog box behaves like any standard Windows property sheet in that no settings are recorded until you click **OK**. The options are verified for proper entries, however, when you change from one property sheet (tab) to another, if an entry verification fails, the property sheet containing the failed entry is brought back into focus, and a message box appears indicating the error. If you click **Cancel**, all input is ignored and the dialog box closes.

Alarm/Event General Properties

➢ **To configure alarms/events general properties:**

1. On the **Special** menu, point to **Configure**, and then click **Alarms**, or in the Application Explorer under **Configure**, double-click **Alarms**. The **Alarm Properties** appears with the **General** properties sheet active:


Alarm Properties

General | Logging | Printing

Alarm Buffer Size: 500 entries

Printer Buffer Size: 2048 bytes

Update Frequency of Printer/Logger: 5000 msec

☑ RTN implies ACK

☑ Position New Alarms at End of Alarm Window

☑ Events Enabled

☐ Alarm Enable Retentive

☐ Use Tag Comment field for Alarm Comments

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Alarm Buffer Size** box, the number of "in-memory" alarm events you want WindowViewer 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.

Note If you set this value too high, it can slow down the performance of your system. If you are using the standard alarm system, a value of 500 is recommended. If you are using the distributed alarm system, a value of 300 is recommended.

3. In the **Printer Buffer Size** box, type the number of bytes of the buffer that will be used by WindowViewer for parallel printers.

   If you are using a serial printer for alarm printing, this entry will have no effect. Only increase the default number (2048) when experiencing problems with printer overflow.

4. In the **Update Frequency of Printer/Logger** box, type the number of milliseconds that WindowViewer wait before trying again to print alarm messages when a printer is offline.
5. Select **RTN implies Ack** if you want alarmed tagnames that return to the "normal" state (RTN) to automatically be acknowledged (ACK). Do not select this option if you want the operator to acknowledge an alarm after it returns to normal.
6. Select **Position New Alarms as End of Alarm Window** if you want new alarms to be displayed at the end of the alarm display object. Alarm windows provide the ability to page back and forth through the alarm queue. Therefore, 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.

   Note This setting will only affect the standard display. The distributed display uses a similar setting in each display object.

7. Select **Events Enabled** if you want to turn on event logging of all data changes that are initiated by the operator, I/O, QuickScripts or, the system. (Only tagnames with **Log Events** selected will be effected.)

   For more information on events, see "Alarms and Events."

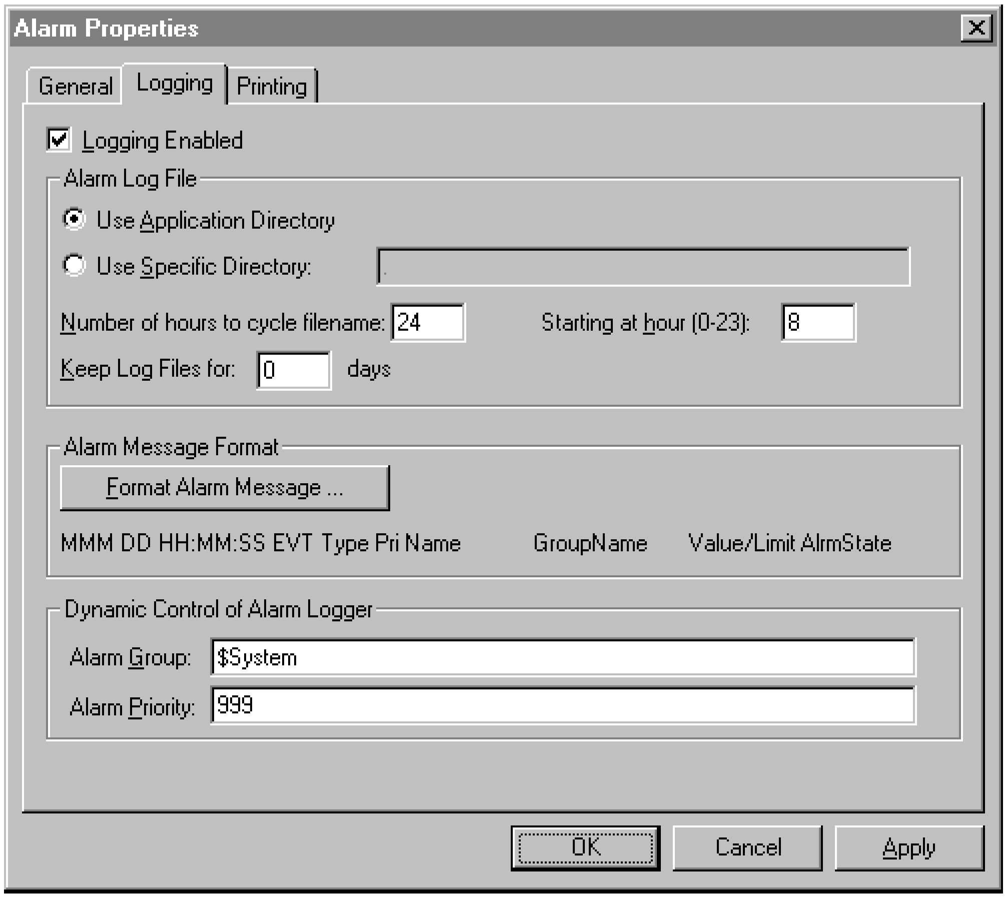
8. Select **AlarmEnable Retentive** if you want the state of the **.AlarmEnable** variable to be retained when WindowViewer is closed.
9. Select **Use Tag Comment field for Alarm Comments** if you are doing distributed alarming and you want the distributed alarm system to use the comments in the Tagname Dictionary for alarm acknowledgment comments.
10. Click **OK** to save your settings and close the dialog box.

Alarm/Event Logging Properties

In addition to displaying and printing alarms, InTouch allows you to log alarms to the computer's hard disk. The log file created is an ASCII file and can be read from most text editors. You can configure various parameters such as when you want the system to cycle filenames, how long you want the files to be stored and what information you want logged.

➢ **To configure alarms/events logging:**

1. On the **Special** menu, point at **Configure**, and then click **Alarms**, or in the Application Explorer under **Configure**, double-click **Alarms**. The **Alarm Properties** dialog box will appear.
2. Click the **Logging** tab to activate the **Logging** property sheet:



[✓] Logging Enabled

Alarm Log File

[●] Use Application Directory
[○] Use Specific Directory:

Number of hours to cycle filename: 24 Starting at hour (0-23): 8

Keep Log Files for: 0 days

Alarm Message Format

MMM DD HH:MM:SS EVT Type Pri Name GroupName Value/Limit AlrmState

Dynamic Control of Alarm Logger

Alarm Group: $System

Alarm Priority: 999

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

3. Select **Logging Enabled** to turn on alarm logging.
4. Select **Use Application Directory** if you want the alarm log file to be saved in your application directory. Or, select **Use specific Directory** and type the complete path to the directory that you want to use.

   This entry must be either a DOS path such as C:\ALARMLOG. If you want to store alarm log files on another node, a Universal Naming Convention (UNC) path can be used. For example:

   \\node\share\directory

   The UNC method is recommended if you are using NAD.

   For more information, on distributed alarms, "The Distributed Alarm System."

For more information on NAD, see Chapter 3 - Building a Distributed Application.

By default, alarm log files are named as follows:

YYMMDDHH.ALG

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)
HH equals the hour the file was created (00-23)

For example, if files were created for three 8-hour shifts beginning at 6:00 a.m. on April 30, 1996, they would be named as follows:

97121506.ALG
97121514.ALG
97121522.ALG

5. In the **Number of hours to cycle filename** box, type the number of hours worth of alarm data that you want stored in each log file. Valid entries are 1 through 24.

Note A "Return to Normal" message may appear in the log file for a tagname that went into alarm during the previous day. Manually adding the **AlarmLogCarryover** parameter in the INTOUCH.INI file and setting it equal to 1 (**AlarmLogCarryover=1**) will carry over the alarm messages that were still active from the previous day and insert them into the current day's log file. The alarm messages from the previous day will still retain the original date/time stamp (displaying when they actually occurred).

6. In the **Starting at hour (0-23)** box, type the hour that you want the first log to begin. Valid entries must be between 0 (midnight) to 23 (for 11:00 PM).

   Example 1: The plant operates three shifts. The first shift begins at 6:00 a.m. The alarms are to be logged by individual shifts. To do this, enter 8 in the **Number of hours to cycle filename** field and 6 in the **Starting at hour (0-23)** field. Now, a file is created from 6:00 a.m. to 2:00 p.m. another from 2:00 p.m. to 10:00 p.m. and a third from 10:00 p.m. to 6:00 a.m.

   Example 2: The plant operates three shifts. The alarms are to be logged by days with the log file starting at midnight. To do this, enter 24 in the **Number of hours to cycle filename** field and **0** in the **Starting at hour (0-23)** field.

7. In the **Keep Log Files for** box, type the number of days (prior to the current day) worth of log files that you want to keep on disk. For example, if you type 10 and it is the 12th day of the month, the log files for the 2nd through the 12th (10 days plus the current day) will be kept and the file for the 1st will automatically be deleted. To keep the files indefinitely, use 0 (zero).

8. If you want to configure the contents of the alarm message that is written to the log file, click **Format Alarm Message**. The **Format Alarm Message** dialog box will appear.

Note Displayed, printed and logged alarm messages are all formatted in the same manner.

For more information on configuring the alarm message format, see "Standard Alarm/Event Message Format."

9. If you want to specify a specific group of alarms that you want to be logged, in the **Alarm Group** box, type an Alarm Group name or the name of a Group Variable. If you want to log all Alarm Groups, type **$System**.

   Note A Group Variable is a tagname that is defined as a **Group Var** type with the name of an Alarm Group assigned to it.

   - If you want to control the choice of alarms logged at runtime, create a Group Variable type tagname, for example, **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";
   ```

10. In the **Alarm Priority** box, type the lowest priority level that you want to be logged. Valid entries are 1 to 999 with 1 being the highest priority. To log all priorities at all times, enter 999 (lowest priority). This will cause all alarms with a priority of less than or equal to 999 to be written to the log file.

    - You can type an analog tagname or an expression if you want a tagname's value to determine the priority level to be logged. You can control the alarm priority level being logged by assigning a value to this tagname through an analog input link or QuickScript.

11. Click **OK** to save your settings and close the dialog box.

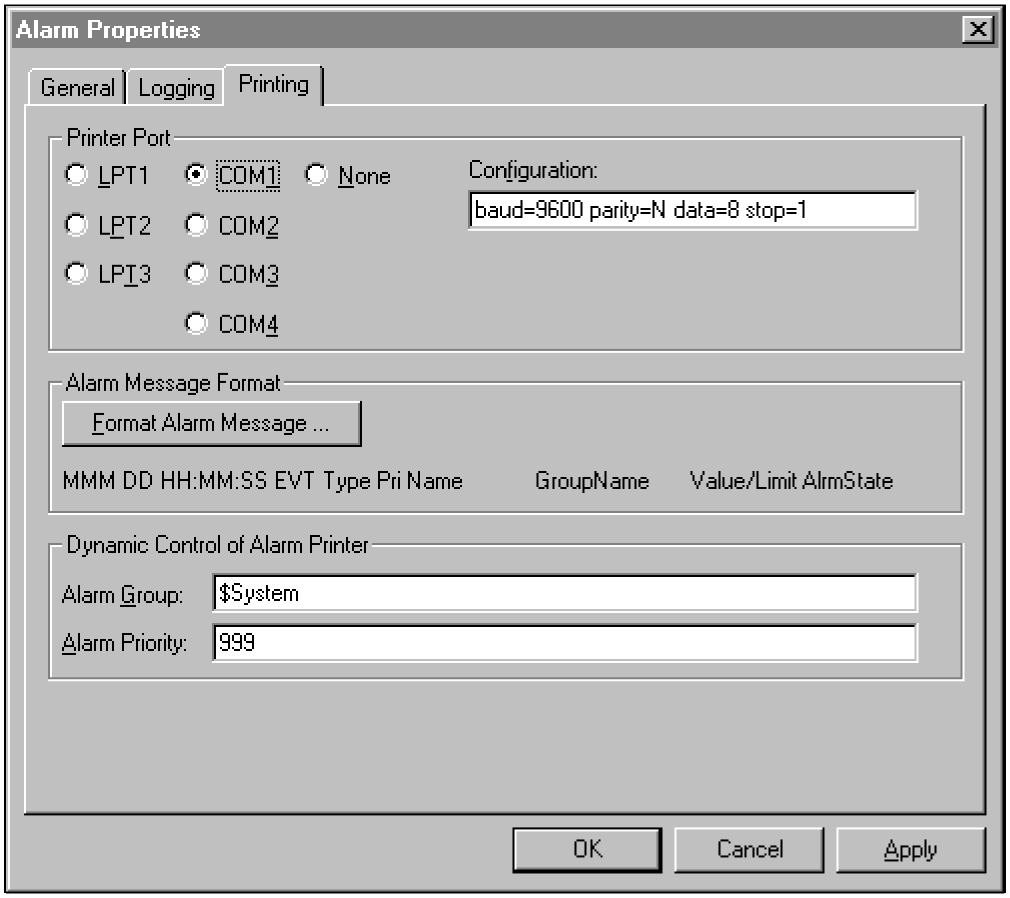
Alarm Printing Properties

In addition to displaying and logging, InTouch also allows you to print alarms. You configure various parameters for alarm printing.

When you are printing alarms, InTouch takes complete control of the port. Therefore, a dedicated printer is required. You cannot perform any other printing functions until you stop alarm printing. Because your alarms printouts are limited in configuration (fonts, point size, an so on), a dot-matrix type printer should be sufficient for alarms printing.

➢ **To configure alarms/events printing:**

1. On the **Special** menu, point at **Configure**, and then click **Alarms**, or in the Application Explorer under **Configure**, double-click **Alarms**. The **Alarm Properties** dialog box will appear.
2. Click the **Printing** tab to activate the **Printing** property sheet:



Printer Port: ● COM1

Configuration: baud=9600 parity=N data=8 stop=1

MMM DD HH:MM:SS EVT Type Pri Name GroupName Value/Limit AlrmState

Alarm Group: $System

Alarm Priority: 999

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

3. Select the **Print to** port being used by the printer.
4. If your printer is using a serial port (COM1-COM4), in the **Configuration** box, type the values for the baud rate, parity, data bits and stop bits. For example:

   baud= 9600 parity=N data=8 stop=1

   The valid values for each entry in the string:

baud	110, 150, 300, 600, 1200, 4800, 9600, 19200
parity	O (odd), E (even), N (none)
data	7 or 8
stop	1 or 2

5. If you want to configure the contents of the alarm message that is written to the log file, click **Format Alarm Message**. The **Format Alarm Message** dialog box will appear.

Note Displayed, printed and logged alarm messages are all formatted in the same manner.

For more information on configuring the alarm message format, see "Standard Alarm/Event Message Format."

6. If you want to specify a specific group of alarms that you want to be logged, in the **Alarm Group** box, type an Alarm Group name or the name of a Group Variable. If you want to log all Alarm Groups, type **$System**.

Note A Group Variable is a tagname that is defined as a **Group Var** type with the name of an Alarm Group assigned to it.

If you want to control the choice of alarms logged at runtime, create a Group Variable type tagname, for example, **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 **Alarm Priority** box, type the lowest priority level that you want to be logged. Valid entries are 1 to 999 with 1 being the highest priority. To log all priorities at all times, enter 999 (lowest priority). This will cause all alarms with a priority of less than or equal to 999 to be written to the log file.

You can type an analog tagname or an expression if you want a tagname's value to determine the priority level to be logged. You can control the alarm priority level being logged by assigning a value to this tagname through an analog input link or QuickScript.

8. Click **OK** to save your settings and close the dialog box.

Using the Standard Alarm System for Remote Alarming

The standard alarm system is primarily intended for single-node alarm monitoring. It can, however be configured to allow for the remote display and acknowledgment of alarms from identical InTouch applications. These applications can be configured so that a master or Alarm Server node can share its alarms with one or more remote nodes. These alarms are displayed in real-time on the remote nodes as they occur on the master node. Also, the alarms can be acknowledged remotely by tagname or by Alarm Group. The only requirement is that each node has Wonderware NetDDE running and each node must run identical InTouch Tagname Dictionarys.

Assuming that you have already defined the tagnames and that alarm states exist for some of these tags, you can configure your application for remote network alarming.

➢ **To set up an application for remote alarming:**

1. Paste a standard alarm display object to your window. (Click the wizard tool, select the **Alarm Displays** category. Double-click the **Standard Alarm Display** wizard, and then click in the window to paste it.)
2. Configure the alarm display to retrieve its alarms from the Alarm Server node as described in the following section.
3. Configure InTouch so it recognizes the Alarm Server node.
4. If any remote acknowledgment occurs, define a I/O type tagname for each tagname or Alarm Group to be acknowledged (procedure follows).
5. Configure NetDDE on each remote node sharing the alarms.

📖 For more information on configuring NetDDE, see your *NetDDE for Windows User's Guide*.

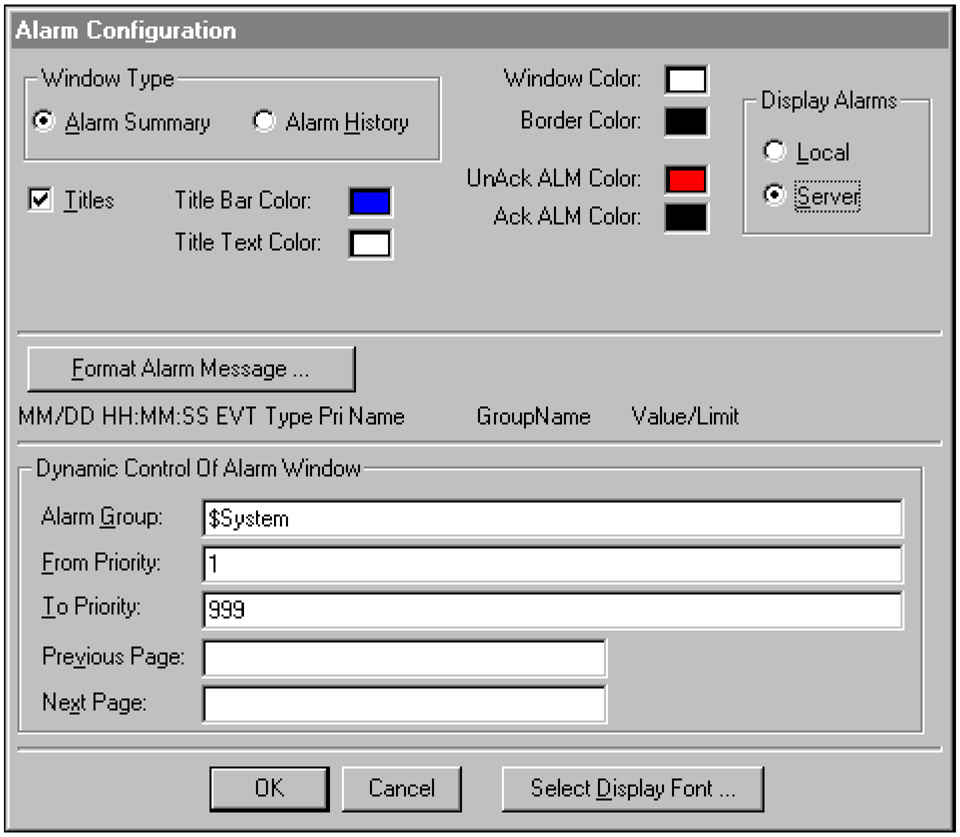
🖱 As described above, you can use the standard alarm system to do remote alarming, however, we recommend that you use the distributed alarm system instead.

👓 For more information on configuring distributed alarming, see "The Distributed Alarm System."

➢ **To configure a standard alarm display for "Alarm Server" Mode:**

1. Double-click the alarm display or, with the alarm display selected, on the **Special** menu, click **Animation Links**. The **Alarm Configuration** dialog box appears:

   To quickly access the dialog box, right-click the alarm display object, and then click **Properties**.


Alarm Configuration

Window Type: [✓ Alarm Summary] Alarm History

[✓ Titles]

Title Bar Color: 🟦

Title Text Color: ⬜

Window Color: ⬜

Border Color: ⬛

UnAck ALM Color: 🟥

Ack ALM Color: ⬛

Display Alarms: Local [✓ Server]

Format Alarm Message ...

MM/DD HH:MM:SS EVT Type Pri Name GroupName Value/Limit

Dynamic Control Of Alarm Window

Alarm Group: $System

From Priority: 1

To Priority: 999

Previous Page:

Next Page:

   If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

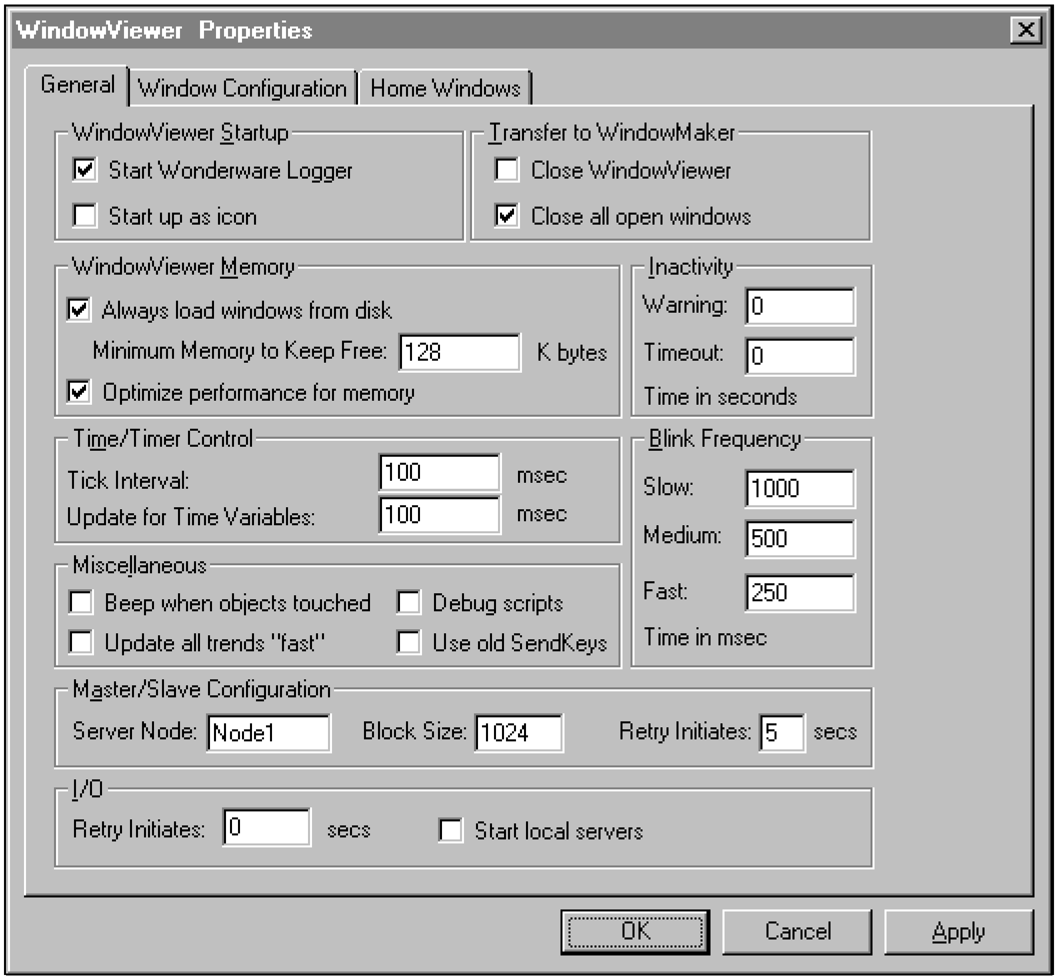
2. In the **Display Alarms** group, select **Server** to display the alarms/events collected by the server node.

3. Click **OK**.

➢ **To configure the alarm server node:**

1. On the **Special** menu, point to **Configure**, and then click **WindowViewer**. The **WindowViewer Properties** dialog box appears with the **General** properties sheet active:

   To quickly access the **WindowViewer Properties**, in the Application Explorer, under **Configure**, double-click **WindowViewer**.


WindowViewer Properties

General | Window Configuration | Home Windows

WindowViewer Startup
- [✓] Start Wonderware Logger
- [ ] Start up as icon

Transfer to WindowMaker
- [ ] Close WindowViewer
- [✓] Close all open windows

WindowViewer Memory
- [✓] Always load windows from disk
- Minimum Memory to Keep Free: 128 K bytes
- [✓] Optimize performance for memory

Inactivity
- Warning: 0
- Timeout: 0
- Time in seconds

Time/Timer Control
- Tick Interval: 100 msec
- Update for Time Variables: 100 msec

Blink Frequency
- Slow: 1000
- Medium: 500
- Fast: 250
- Time in msec

Miscellaneous
- [ ] Beep when objects touched
- [ ] Debug scripts
- [ ] Update all trends "fast"
- [ ] Use old SendKeys

Master/Slave Configuration
- Server Node: Node1
- Block Size: 1024
- Retry Initiates: 5 secs

I/O
- Retry Initiates: 0 secs
- [ ] Start local servers

   If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Master/Slave Configuration** group, in the **Serve Node** box, type the name of the alarm server node that you want remote alarm nodes to retrieve their alarm information. (You cannot enter the local node name here.)

   The node name used here is not used as part of the distributed alarm system.

3. Click **OK**.

➢ **To configure remote alarm acknowledgment:**

1. Define an I/O type tagname for each tagname or Alarm Group to be acknowledged. The I/O item for this tagname is the .Ack field of the tagname or Alarm Group.

   For example, let's assume you are alarming 10 tagnames: Temp1 through Temp10. Temp1 through Temp5 uses the Alarm Group Group1. Temp6 through Temp10 uses the Alarm Group Group2. The options for acknowledging these alarms are:

   - Acknowledge alarms individually by tagname.
   - Acknowledge alarms by Subgroup (that is, Group1 or Group2).
   - Acknowledge all alarms at once by using the **$System** Alarm Group.

   ➢ **To setup this alarm acknowledgment:**

   a) Define an "I/O Discrete" type tagname. The tagnames you can use include Temp1_Ack, Ack_Temp1, Group1_Ack, or System_Ack.

   b) Define and use an Access Name with these properties:

Access Name:	***Server***
Node Name:	***Node1***
Application Name:	***\\ServerNodeName\View***
Topic:	***Tagname***
Protocol:	***DDE***

   c) Use one of the following for the tagname's **Item**, depending on what you want to acknowledge:

   - Temp1.ack through Temp10.ack - To acknowledge individual tagnames.
   - Group1.ack or Group2.ack - To acknowledge Subgroups.
   - **$System.ack** - To acknowledge all alarms.

2. To acknowledge the alarms in any of the three ways above, create a button for each. Double-click and assign it to the **Discrete Value Touch Pushbutton** animation link. For **Action**, choose **SET**, and for **Tagname**, use one of the following:

   Temp1_Ack or **Group1_Ack** or **System_Ack** ,etc.

   When an alarm occurs for a tagname or an Alarm Group, InTouch resets the corresponding .Ack field to zero and sets the .Alarm field to 1. Setting the .Ack field to 1 acknowledges the alarm or group of alarms. You do not need to reset it. It will be reset when another alarm occurs.

   You can also see if a specific tagname or group is in an alarm state by using the **.Alarm** .field for tagnames or Alarm Groups.

Alarm .Fields

InTouch provides various alarm "**.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, for example, Excel, or a remote View application (described later in this chapter).

For example, if you create an analog alarmed tagname called **Analog_Tagname**, it will have "attributes" associated with it such as its name, its **HiHi** setpoint, an so on. Some of these "attributes" are accessible through logic scripts, expressions and user inputs and are known as **.fields** (dot fields).

The syntax required to access these data fields associated with a tagname is **Tagname.field**. For example, if you want to allow runtime changes to the **HiHi** alarm limit on **Analog_Tagname**, you could create an **Analog - User Input** touch link could be applied to a button and **Analog_Tagname.HiHiLimit** would be entered as the expression in the link's dialog box. During runtime, the operator would simply click on the button and type in a new value for the **HiHi** alarm limit being used for **Analog_Tagname**.

The following briefly describes each examples of how to use the alarm **.fields**, see your *InTouch Reference Guide*.

.Field	Description
.Ack	Monitors/controls the alarm acknowledgment status.
.Alarm	Signals that an alarm condition exists.
.AlarmDevDeadband	Monitors/controls the deviation percentage deadband for both minor and major deviation alarms.
.AlarmEnable	Disables/enables events and alarms.
.AlarmValDeadband	Monitors/controls the value of an alarm's deadband.
.DevTarget	Monitors/controls the target for minor and major deviation alarms.
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	Read/write analog tagname .fields that monitors/controls 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 determines 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 monitors and/or controls 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.

Acknowledging Local Alarms

You can acknowledge local alarms by using the **.Ack** (**.field**) in an action or key script.

➢ **To create a local 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 or select it, and then on the **Special** menu, click **Animation Links**.
3. In the **Touch Pushbuttons** section of the animation link selection dialog box, click **Action**. The QuickScript editor will appear.
4. Type 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.

or,

$System.Ack=l;	Acknowledges all local alarms in the system.
***Group Name*.Ack=l;**	Acknowledges all local alarms in a specific Alarm Group.
***Group Var*.Ack=l;**	Acknowledges all local alarms in a group indicated by the value of the Group Variable, an indirect Alarm Group tagname type.
***Tagname*.Ack=l;**	Acknowledges a specific tagname's alarms.

5. Click **OK**.

For more information on the QuickScript editor and its features, see Chapter 6 - Creating InTouch QuickScripts.

The Distributed Alarm System

InTouch provides two alarm systems: standard and distributed. Both provide services to display, log, print, and acknowledge process alarms and system events. The standard system is used to display and acknowledge events and alarms generated by the local InTouch application. The distributed system expands this scope to allow the display and acknowledgment of alarms generated by the local alarm systems of other InTouch applications.

Both the standard and distributed systems can be used in a distributed application. The major difference is that the standard system is limited to only those alarms generated by an identical InTouch application, while the distributed system has no such limitation.

The distributed alarm system features include:

- The ability to display and acknowledge alarms from any InTouch node on a network.
- A new alarm display that has built-in scroll bars, sizable display columns, multiple alarm selections, an update status bar, dynamic display types, and display colors based on alarm priority.
- QuickScript functions that provide dynamic control over the alarm display and alarm acknowledgment.
- A grouping mechanism that allows multiple Alarm Groups across different applications to be called via a single name.
- The capability of adding comments to alarms when acknowledged.

The distributed alarm system can be thought of as an extension of the standard alarm system. The standard alarm system provides local alarm display, printing, logging, and acknowledgment of alarms. The distributed alarm system expands the scope of the display and acknowledgment features to include alarms generated by remote applications (alarm providers).

Since the distributed alarm system is an extension of the standard alarm system, it shares many of the same configurations, all presented previously. The following sections outline just those configurations that are specific to the distributed alarm system.

📖 To use the distributed alarm system, you must configure certain node-level settings. For more information, see your *FactorySuite System Administrator's Guide*.

Distributing the Application

The application may be distributed either manually or by using the NAD system. When the application is distributed, the alarm group list file is distributed automatically as it is part of the application.

👓 For more information on NAD, see Chapter 3 - Building a Distributed Application.

Distributed Alarm Group Lists

The distributed alarm system uses the same Alarm Group mechanism as the standard alarm system. This mechanism groups alarms into a local hierarchical tree structure that both the standard and the distributed alarm displays can use to filter alarms for display. However, the distributed alarm system allows you to view these groups from multiple nodes on a network. To provide a grouping for these node and Alarm Groups, the distributed alarm system uses an **Alarm Group List**.

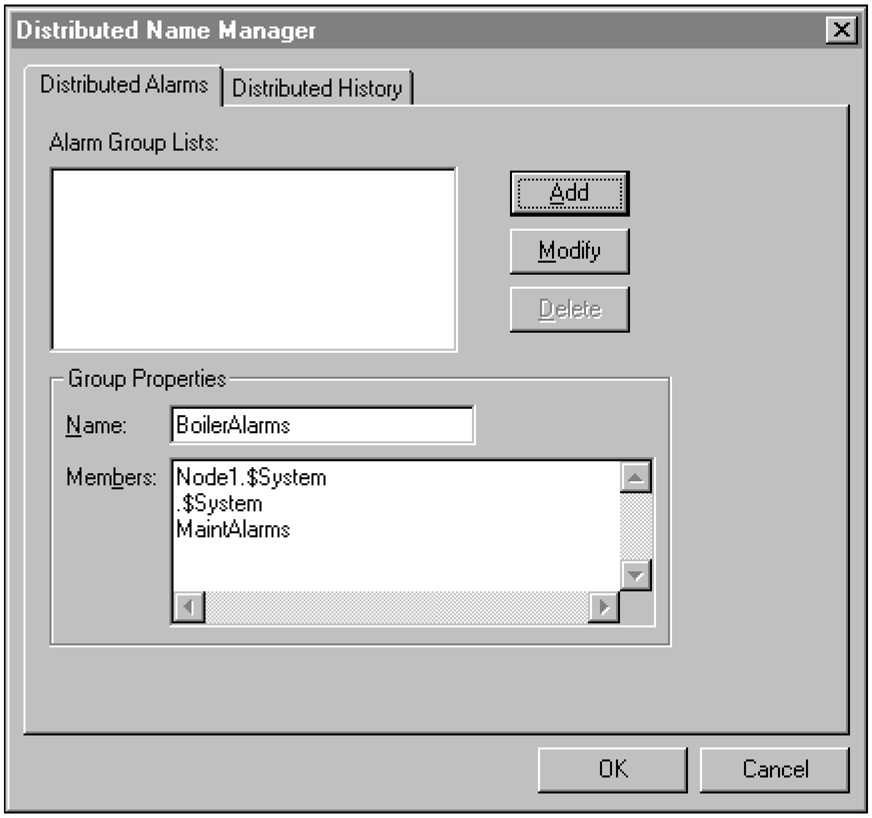
The **Alarm Group List** is a named list consisting of InTouch nodes and the Alarm Groups defined on each of those nodes. It can also contain other Alarm Group List Names and local Alarm Groups. This list is used by the distributed alarm display to query for alarms.

For example, if you were interested in all of the boiler alarms across several InTouch nodes, you could build a query called "BoilerAlarms." The list attached to that query would contain all the Alarm Groups on all the nodes you were interested in that correspond to boiler alarms.

➢ **To create an Alarm Group list:**

1. On the **Special** menu, point to **Configure**, and then click **Distributed Name Manager**. The **Distributed Name Manager** dialog box appears with the **Distributed Alarms** property sheet active:

   🖰 To quickly access the dialog box, in the Application Explorer under **Configure**, double-click **Distributed Name Manager**.



Name: BoilerAlarms

Members:
Node1.$System
.$System
MaintAlarms

   🖰 If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Group Properties** section, in the **Name** box, type the name of the query.

3. In the **Members** box, type the list of InTouch nodes and Alarm Groups that you want to include in your query. The valid syntax for these lists include:

Standard Group Entries

\\Node\InTouch!Group	Fully qualified path to Alarm Group on a remote node
\InTouch!Group	Same as above, but Node assumed to be local
GroupList	Another Group List

Shortcut Group Entries

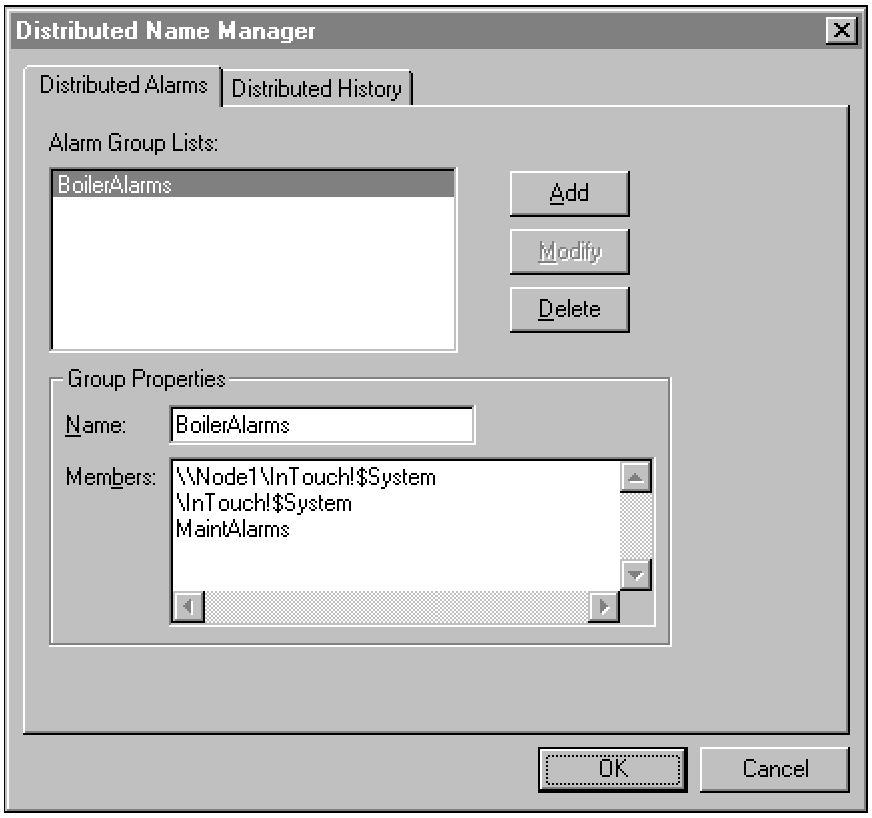
Node.Group	Shortcut that equates to **\\Node\InTouch!Group**
.Group	Shortcut that equates to **\InTouch!Group**

Node identifies the name of the InTouch remote node and **.Group** identifies the Alarm Group on that node. If the Alarm Group is local, you can enter just the Alarm Group name with a period. For example, **.AlarmGroup**.

The **Shortcut Group Entries** provide you with an easy way to enter node and Alarm Group information into the dialog box. It's important to note that this information is translated into the **Standard Group Entry** format when you save the Alarm Group List.

Note The **Node.Group** and **.Group** syntax can only be used in this configuration dialog box. It is not valid in the alarm display configuration or any alarm QuickScript function.

4. Click **Add** to add this list to your Alarm Group file. The syntax of the **Members** will automatically be converted. For example, **.$System** will be converted to **\InTouch!$System**, as shown below:



BoilerAlarms

Name: BoilerAlarms

Members:
\\Node1\InTouch!$System
\InTouch!$System
MaintAlarms

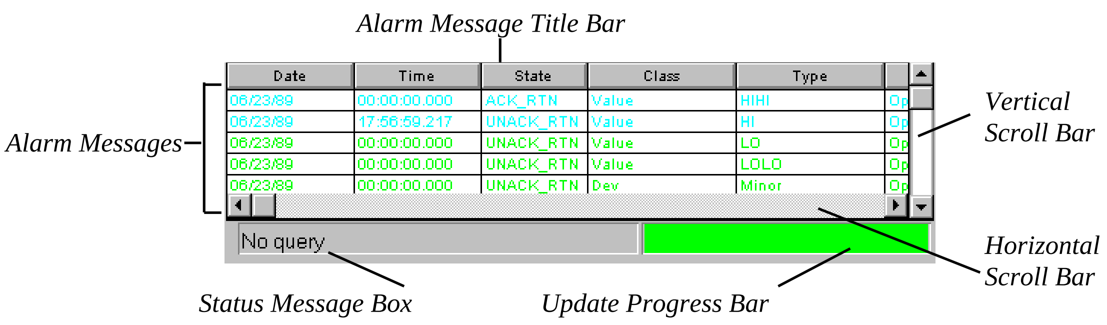
If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

5. Click **OK**.

The Distributed Alarm Display

The distributed alarm system has a unique display object to show both locally and remotely generated alarms. This display object's features include: built in scroll bars, sizable display columns, multiple selection of alarms, update status bar, and alarm display colors based on alarm priority.

InTouch allows you to modify the appearance of the alarm display (including the information that is displayed), the colors used for various alarm conditions, and the Alarm Group and alarm priority levels displayed.



Date	Time	State	Class	Type	
06/23/89	00:00:00.000	ACK_RTN	Value	HIHI	Op
06/23/89	17:56:59.217	UNACK_RTN	Value	HI	Op
06/23/89	00:00:00.000	UNACK_RTN	Value	LO	Op
06/23/89	00:00:00.000	UNACK_RTN	Value	LOLO	Op
06/23/89	00:00:00.000	UNACK_RTN	Dev	Minor	Op

No query

Scroll Bars

The distributed alarm display has built-in horizontal and vertical scroll bars that allow you to move through listed alarms. You can configure the display of these scroll bars.

Next/Prev Page Controls

The **.NextPage** and **.PrevPage** alarm display control properties are also supported for the distributed alarm display.

For more information see, "Alarm Display Control Properties."

Sizable Display Columns

The distributed alarm display uses a grid to hold the alarm messages. This grid allows for dynamic sizing of the column widths simply by selecting a column and dragging it to set the column width. This functionality is available only during runtime. You can configure whether or not the grid can be used to size the columns.

Grid column changes are not saved; therefore, if you make grid column changes and close the window containing the alarm display, the grid columns will again be at their default width upon re-opening that window.

Multiple Selection

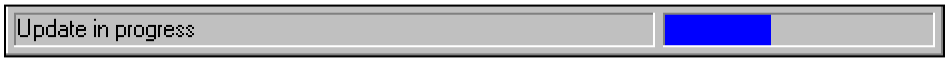
The grid allows you to select a single or multiple alarms in a list box. The selected alarms can be acknowledged by using the **almAckSelect()** QuickScript function described later in this chapter. When you configure the distributed alarm display, you can also define the selection behavior to allow either toggle selection (item by item), or multiple selection (holding down CTRL or SHIFT in conjunction with a mouse click to select multiple alarms). You can turn off runtime selection.

Alarm Message Colors

You can also configure up to eight different colors for each displayed alarm message based on the priority of the alarm and whether it is acknowledged or not.

Update Status Bar

The distributed alarm display includes a status bar that contains two indicators: A status message and a progress bar. These indicators provide an overview of the current state of the display query. You can turn off the display of the status bar in runtime.



Feature	Description
Status Message	The status message at the left end of the status bar provides a more detailed description of the current query status.
Progress Bar	The update progress bar at the right end of the status bar provides a visual indication of the current query progress.

State/Indicator	Status Message	Progress Bar
No Query	None	None
Query Incomplete	Update Incomplete	By Formula
Query Complete	Update Successful	Solid Blue

Distributed Alarm Display Guidelines

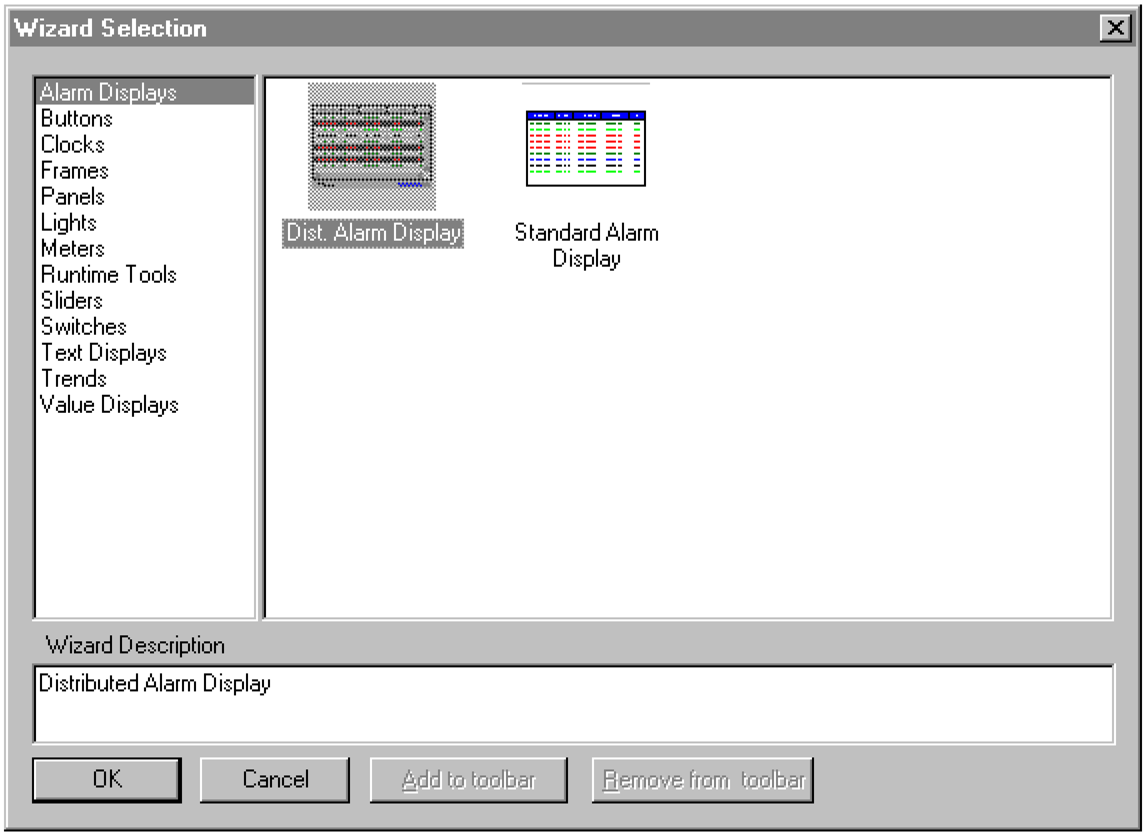
The distributed alarm display is a complicated wizard. While it may appear like the standard display wizard, it uses the same mechanism as the Window Control wizards. This mechanism requires that certain guidelines be observed when using objects such as the distributed alarm display. These guidelines are as follows:

- Each display must have an identifier so the associated QuickScript functions know which display to modify. This identifier, entered as **Display Name** in the alarm display configuration dialogs, must be unique for each display.
- Displays should not overlap other InTouch objects such as windows controls or graphic objects. You can easily verify this by clicking on the distributed alarm display in WindowMaker, and checking the display's "handles." The handles should not touch another graphic object on the screen.
- Displays should be used sparingly. Placing numerous displays on one screen can result in reduced system performance. When possible, limit the number of displays on your screen and call further screens (dialog boxes) with additional displays if necessary.

Creating a Distributed Alarm Display

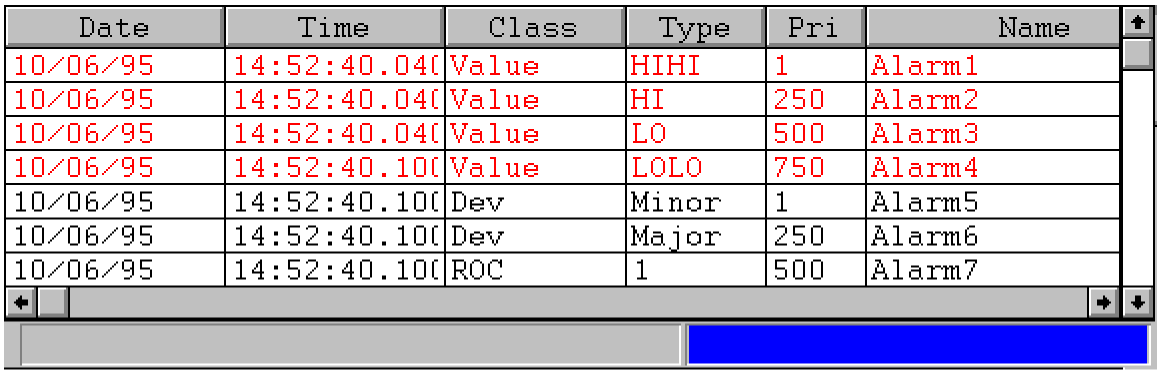
➤ **To create a distributed alarm display:**

1. Click the wizard tool in the **Wizard/ActiveX Toolbar**. The **Wizard Selection** dialog box appears:



Distributed Alarm Display

2. Select **Alarm Displays** in the list of wizards to display both alarm wizards.
3. Double-click the **Dist. 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.
4. Click in the window to paste the alarm display wizard:



Date	Time	Class	Type	Pri	Name
10/06/95	14:52:40.04(	Value	HIHI	1	Alarm1
10/06/95	14:52:40.04(	Value	HI	250	Alarm2
10/06/95	14:52:40.04(	Value	LO	500	Alarm3
10/06/95	14:52:40.10(	Value	LOLO	750	Alarm4
10/06/95	14:52:40.10(	Dev	Minor	1	Alarm5
10/06/95	14:52:40.10(	Dev	Major	250	Alarm6
10/06/95	14:52:40.10(	ROC	1	500	Alarm7

To size the wizard, point to one of its selection handles then drag it until the desired size is reached.

5. You are now ready to configure the display as described in the next section.

Configuring a Distributed Alarm Display

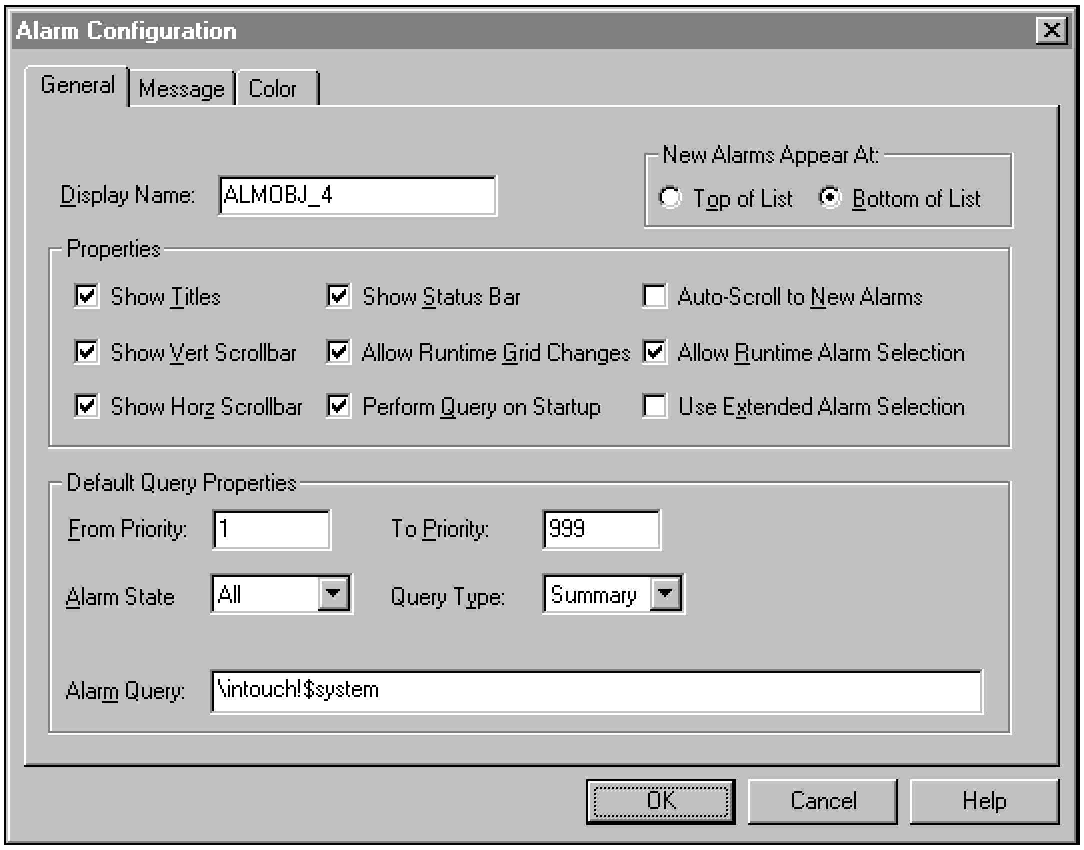
The **Alarm Configuration** dialog box has three property sheets that contain the options for **General**, **Message**, and **Color** configuration.

Note The configuration dialog box behaves like any standard Windows property sheet in that no settings are recorded until you click **OK**. The options are verified for proper entries, however, when you change from one property sheet (tab) to another, if an entry verification fails, the property sheet containing the failed entry is brought back into focus, and a message box appears indicating the error. If you click **Cancel**, all input is ignored and the dialog box closes.

Distributed Alarm Display General Properties

➢ **To configure a distributed alarm display:**

1. Double-click on the distributed alarm display or right-click it, and then click **Properties**. The **Alarm Configuration** dialog box appears with the **General** property sheet active:



Display Name: ALMOBJ_4

New Alarms Appear At: Top of List [✓ Bottom of List]

Properties
- [✓] Show Titles
- [✓] Show Status Bar
- [ ] Auto-Scroll to New Alarms
- [✓] Show Vert Scrollbar
- [✓] Allow Runtime Grid Changes
- [✓] Allow Runtime Alarm Selection
- [✓] Show Horz Scrollbar
- [✓] Perform Query on Startup
- [ ] Use Extended Alarm Selection

Default Query Properties

From Priority: 1 To Priority: 999

Alarm State: All Query Type: Summary

Alarm Query: \intouch!$system

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Display Name** box, type the name for the alarm display. This name must be unique for each alarm display used.

   The name you type here will be used throughout the system for referring to this object for execution of tasks such as alarm acknowledgment and queries.

3. Select the **New Alarms Appear At** option for where you want new alarms to appear in the object:

   Top of List - Displays the most recent alarm at the top of the list.

   Bottom of List - Displays most recent alarm at the bottom of the list.

4. Select the **Properties** as described below:

Property	Description
Show Titles	Displays alarm message title bar.
Show Vert. Scroll Bar	Displays vertical scroll bar.
Show Horz. Scroll Bar	Displays horizontal scroll bar.
Show Status Bar	Displays status bar.
Allow Runtime Grid Changes	Allows the user to change column settings in runtime.
Perform Query on Startup	Automatically begins updating the display using default query properties, if selected. If not selected, you need to perform an **almDefQuery** or **almQuery** before the display will update.
Auto-Scroll to New Alarms	If the user scrolls the list from the beginning, this automatically jumps to the new alarm. (New alarms are defined as those that are not currently displayed within the display object.)
Allow Runtime Alarm Selection	Allows user to select alarms at runtime.
Use Extended Alarm Selection	Allows multiple alarms to be selected by holding down Ctrl or Shift in conjunction with a mouse. The default is to toggle selection of alarms by simply clicking on them (visible only if **Allow Runtime Alarm Selection** check box is selected).

5. Select the **Default Query Properties** options as described below:

   The **Default Query Properties** are used if you select the **Perform Query on Startup** option or, if the **almDefQuery** QuickScript function is executed.

Property	Description
From Priority	Default minimum alarm priority.
To Priority	Default maximum alarm priority. For more information on alarm priorities, see the "Alarm Priorities" section of this chapter.
Alarm State	Default alarm state to query (All, UnAck, Ack).
Query Type	Sets display type as either Summary or Historical.
Alarm Query	Sets the initial Alarm query. This field accepts text only; it does not accept tags. The valid syntax for these lists include:

\\Node\InTouch!Group	Full path to Alarm Group
\InTouch!Group	Full path to local Alarm Group
GroupList	Another Group List

To perform multiple queries, separate each query with a space. For example:

.$System \\Master\InTouch!\MyGroup LocalGroupList Node1.GroupA

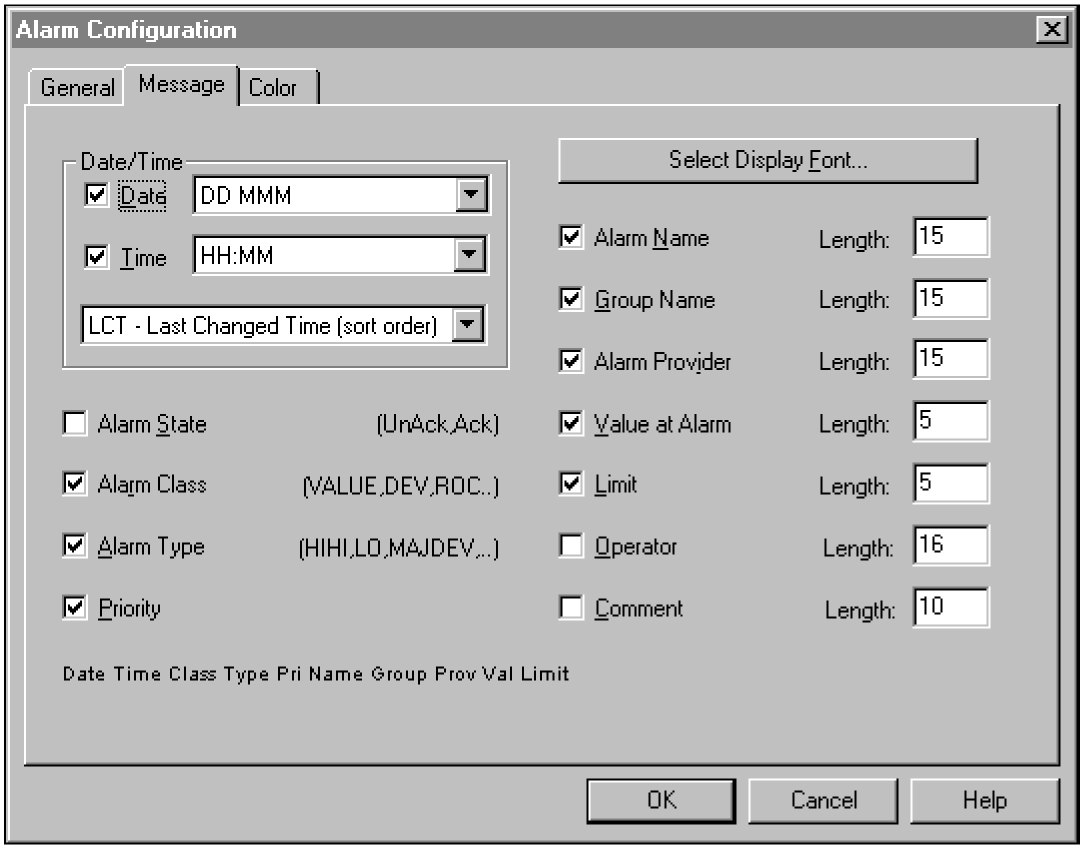
Distributed Alarm Message Format

The information shown in a distributed alarm display object, logged to disk or printed is configurable. For example, information you want displayed and in some cases, how many characters you want displayed for an item.

The order of appearance of the items in the message is fixed and cannot be altered.

➢ **To configure the alarm display message format:**

1. Double-click the distributed alarm display object or right-click it, and then click **Properties**. The **Alarm Configuration** dialog box will appear.
2. Click the **Message** tab to activate the **Message** property sheet:


Alarm Configuration

General | Message | Color

Date/Time
- [✓] Date: DD MMM
- [✓] Time: HH:MM
- LCT - Last Changed Time (sort order)

- [ ] Alarm State (UnAck,Ack)
- [✓] Alarm Class (VALUE,DEV,ROC..)
- [✓] Alarm Type (HIHI,LO,MAJDEV,...)
- [✓] Priority

Select Display Font...

- [✓] Alarm Name Length: 15
- [✓] Group Name Length: 15
- [✓] Alarm Provider Length: 15
- [✓] Value at Alarm Length: 5
- [✓] Limit Length: 5
- [ ] Operator Length: 16
- [ ] Comment Length: 10

Date Time Class Type Pri Name Group Prov Val Limit

OK | Cancel | Help

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

Note The preview area (at the bottom of the dialog) displays an example of the alarm message as currently configured. This example will show the message using the font selected, but not color.

3. Select **Date** if you want to display the date in the alarm message, and then select click the arrow to select the format for the date. The available formats are:

Selection	Display	Selection	Display
DD MMM	28 Feb	**MM/DD**	02/28
DD MM YYYY	28 Feb 1997	**MM/DD/YY**	02/28/97
DD/MM	28/02	**MMM DD**	Feb 28
DD/MM/YY	28/02/97	**MMM DD YYYY**	Feb 28 1997

4. Select **Time** if you want to display the time in the alarm message, and then select the arrow to select format for the time. The values in this field are used as a template to specify the format of the time. For example, to specify the time as **10:24:30 AM**, use **HH:MM:SS AP**. The template characters are as follows:

AP	Selects the AM/PM format. For example, three o'clock in the afternoon is displayed as **3:00 PM**. A time without this designation defaults to 24 hour military time format. For example, three o'clock in the afternoon is displayed as **15:00**.
HH	Displays the hour the alarm/event occurred.
MM	Displays the minute the alarm/event occurred.
SS	Displays the second the alarm/event occurred.
SSS	Displays the millisecond the alarm/event occurred.

5. In the sort order box below **Time**, select the order in which you want the alarms to be sorted in the alarm object. There are three choices:

   LCT - Last Changed Time (sort order)
   LCT - But OAT on ACK -
   OAT - Original Alarm Time

6. Select **Alarm State (UnAck,Ack)** if you want to display the state of the alarm.
7. Select **Alarm Class (VALUE.DEV, ROC..)** if you want to display the category of the alarm.
8. Select **Alarm Type (HIHI,LO,MAJDEV,…)** if you want to display the alarm type.

   For more information on available alarm types, see "Alarm Types."

9. Select **Operator** if you want to display the logged-on operator's ID associated with the alarm condition. Enter a value in the Length field to control the number of characters displayed (16 characters maximum).
10. Select **Priority** if you want to display the alarm priority.
11. Select **Comment** if you want to display the tagname's comments. These comments were typed in the **Comments** box when the tagname was defined in the database. In the **Length** box, type the number of characters that you want displayed (50 characters maximum).
12. Click **Select Display Font** to access the **Font** dialog box to change the font, style and size used in the alarm display.
13. Select **Alarm Name** if you want to display the alarm/tagname. In the **Length** box, type the number of characters that you want displayed (32 characters maximum).
14. Select **Group Name** if you want to display the Alarm Group name. In the **Length** box, type the number of characters that you want displayed (32 characters maximum).
15. Select **Alarm Provider** if you want to display the name of the alarm provider. . In the **Length** box, type the number of characters that you want displayed (32 characters maximum).
16. Select **Value at Alarm** if you want to display the value of the tagname when the alarm occurred. In the **Length** box, type the number of characters that you want displayed.

    The value should be large enough to provide the desired level of precision (15 characters maximum).

17. Select **Limit** if you want to display the alarm limit value of the tagname. In the **Length** box, type the number of characters that you want displayed.

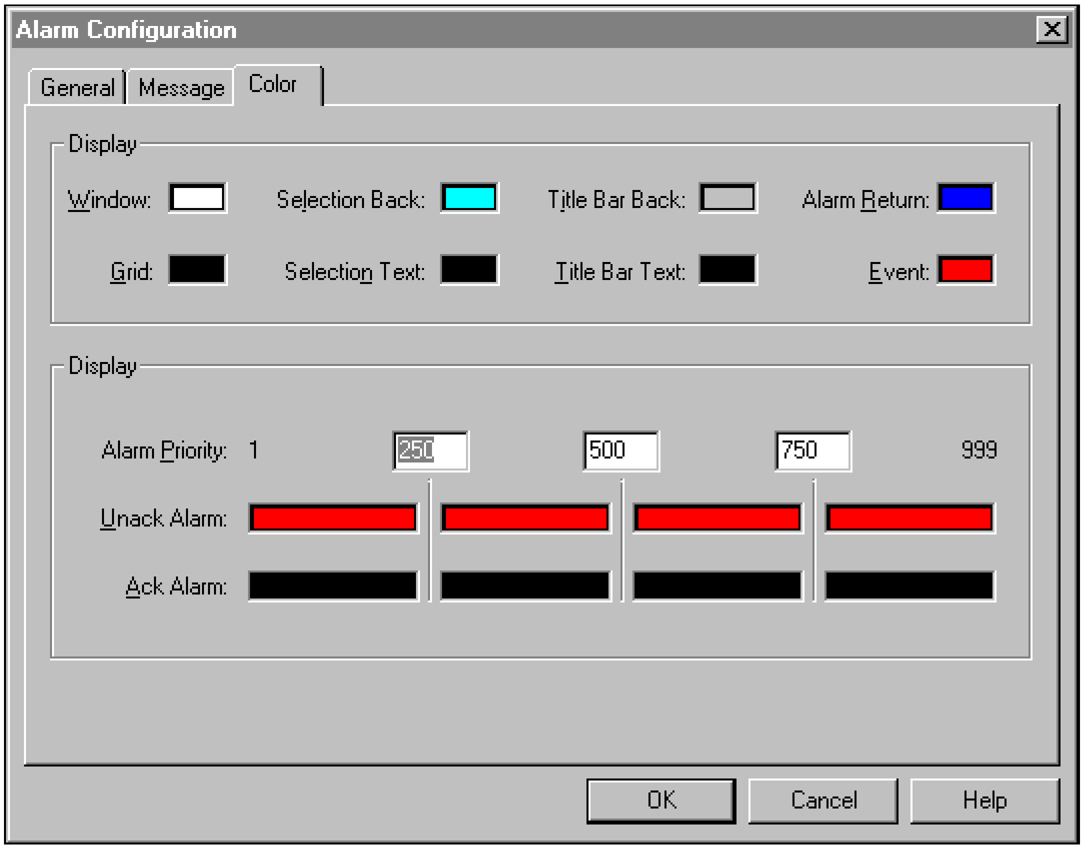
The size of this field should be large enough to provide the desired level of precision (32 characters maximum).

18. Click **OK**.

Distributed Alarm Display Color Properties

➢ **To configure the alarm display colors:**

1. Double-click the distributed alarm display object or right-click it, and then click **Properties**. The **Alarm Configuration** dialog box will appear.
2. Click the **Color** tab to activate the **Color** property sheet:



If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

3. In the top **Display** group, click each color box to open the InTouch Palette. Click the color that you want to use in the palette for each of the following:

Option	Description
Window	Sets display background color.
Grid	Sets display grid color.
Selection Back	Sets highlighted text background color.
Selection Text	Sets highlighted text color.
Title Bar Back	Sets title bar background color (visible only if Show Titles option is on).
Title Bar Text	Sets title bar text color (visible only if Show Titles option is on).
Alarm Return	Sets color of returned alarms (alarms that have returned to normal without being acknowledged).
Event	Sets color of Event alarms.

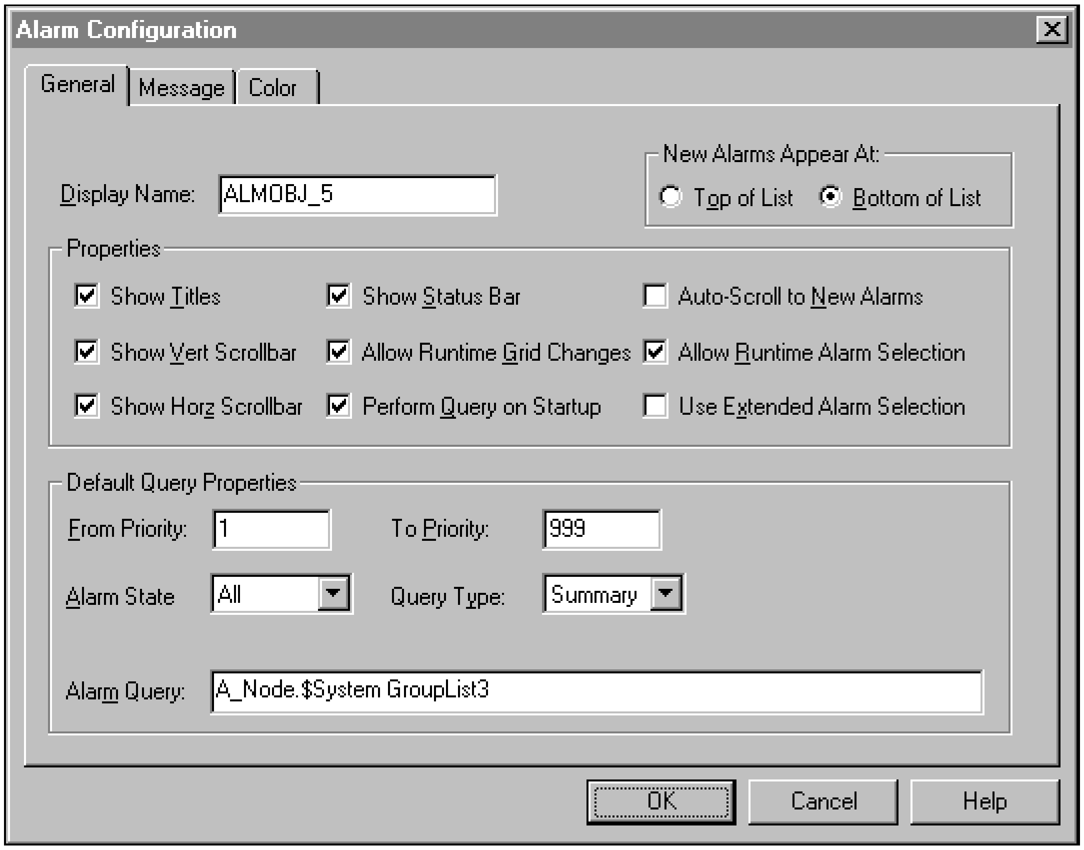
4. In the **Alarm Priority** boxes, type the breakpoint values for the alarm display.
5. Click the **UnAck Alarm** and **Ack Alarm** color boxes to open the InTouch palette. Click the color in the palette that you want to use.
6. Click **OK**.

Dynamically Controlling the Display Type

The distributed alarm display can show summaries of active alarms or listings of historical alarms. Unlike the standard alarm display, which you configured to either view summaries or historical alarms, when you configure it, the distributed alarm display can show either, dynamically.

➢ **To change the distributed alarm object query default:**

1. Double-click the distributed alarm object or right-click it, and then click **Properties**. The **Alarm Configuration** dialog box appears with the **General** property sheet active:



Display Name: ALMOBJ_5

New Alarms Appear At: ○ Top of List [✓ Bottom of List]

Properties

✓ Show Titles	✓ Show Status Bar	☐ Auto-Scroll to New Alarms
✓ Show Vert Scrollbar	✓ Allow Runtime Grid Changes	✓ Allow Runtime Alarm Selection
✓ Show Horz Scrollbar	✓ Perform Query on Startup	☐ Use Extended Alarm Selection

Default Query Properties

From Priority: 1 To Priority: 999

Alarm State: All Query Type: Summary

Alarm Query: A_Node.$System GroupList3

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Click the **Query Type** arrow and select the type of alarm display that you want to use for the runtime default.

   For example, in runtime, the display type is determined by the query QuickScript function that you use with it. If you run an *almQuery( )* QuickScript against display *AlarmObj_2* with the *Type* parameter set to "Summ," then the display will show summaries of current alarms. Conversely, if the same display has an *almQuery( )* run against it with the *Type* parameter set to "Hist," it will show historical alarms. The property, *QueryType* reflects the current state of the alarm display.

3. Click **OK**.

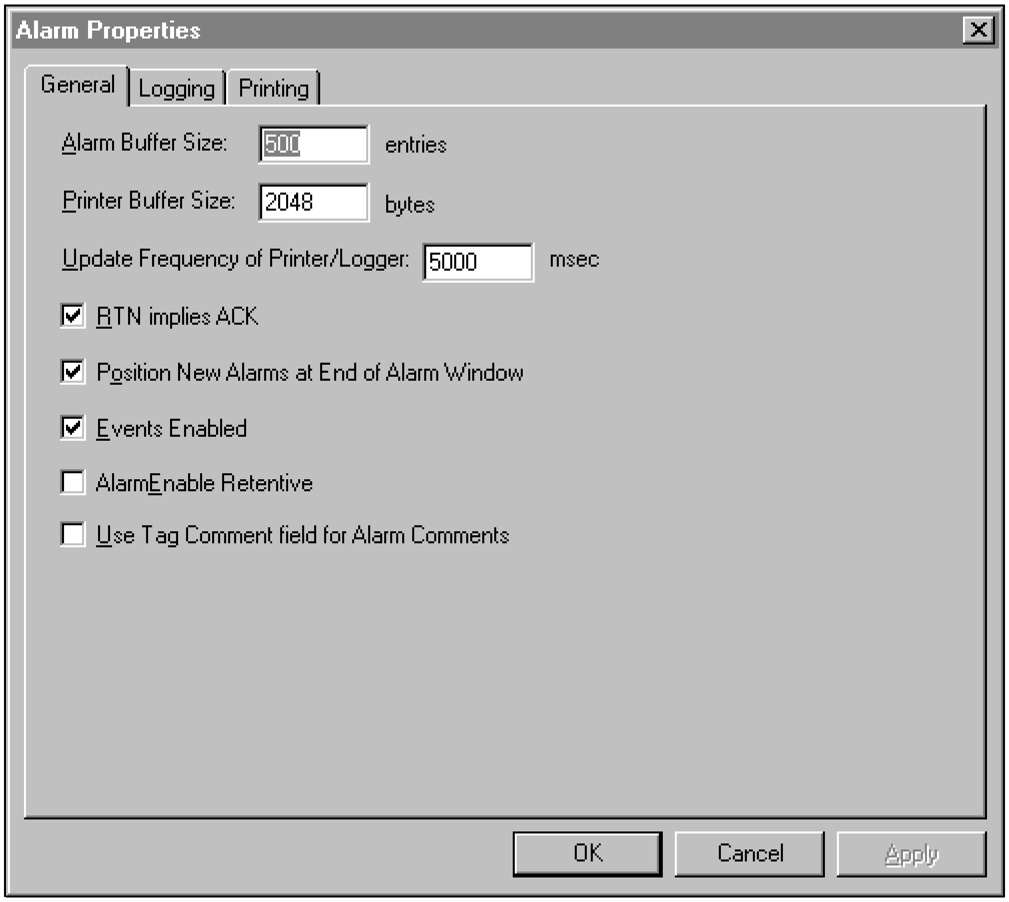
Attaching Comments to an Alarm Ack Function

Each alarm acknowledgment function can have a comment attached to it. The comment can be used by the operator acknowledging the alarm to attach information about the alarm. InTouch can store these alarms in log files and print them. To allow for this support, the distributed alarm system uses the comment box of the alarmed tagname's definition in the database to pass these comments to the logging and printing subsystems.

You can choose whether to allow the alarm system to use the tagname comment box for alarm comments. If you use the tagname comment box to hold alarm comments, any comments in that field will be overwritten during runtime (this will not affect the development database, however).

➢ **To use the tagname comment field for alarm acknowledge comments:**

1. Double-click the distributed alarm display object or right-click it, and then click **Properties**. The **Alarm Properties** appears with the **General** properties sheet active:



Alarm Buffer Size: 500 entries

Printer Buffer Size: 2048 bytes

Update Frequency of Printer/Logger: 5000 msec

- [✓] RTN implies ACK
- [✓] Position New Alarms at End of Alarm Window
- [✓] Events Enabled
- [ ] AlarmEnable Retentive
- [ ] Use Tag Comment field for Alarm Comments

If you right-click a text box in any alarm configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Select **Use Tagname Comment Field for Alarm Comments**.
3. Click **OK**.

Using the Distributed Display to Monitor Local Alarms

The distributed alarm display can be used to display and acknowledge both local and remote alarms.

➢ **To set up a display to monitor just local alarms:**

1. Paste a distributed alarm display object to your window. (Click the wizard tool, select the **Alarm Displays** category. Double-click the **Dist. Alarm Display** wizard, and then click in the window to paste it.)
2. Double-click on the display or right-click it, and then click **Properties**. The **Alarm Configuration** dialog box will appear with the **General** property sheet active.
3. In the **Alarm Group** box, type "**\InTouch!$System** or **.$System**.

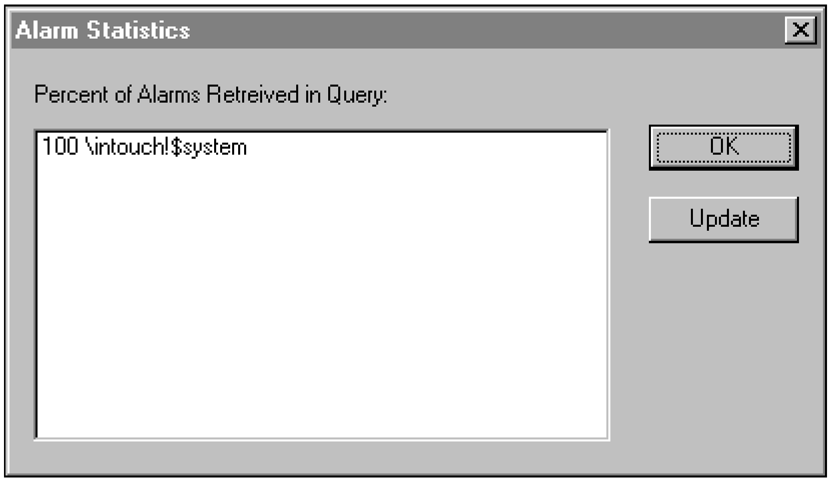
   You may substitute any valid Alarm Group for **$System**. You can also define an Alarm Group List containing just **\InTouch!$System**, and then use this Group List in step 3 instead of a direct reference.
4. Configure the other parameters of the **Default Query Properties** for the type of display and any filtering your application requires.
5. Configure the node's application program to be an Alarm Provider.

   For more information on configuring the node, see "The Distributed Alarm System."
6. Switch to WindowViewer to run the application.

Displaying Alarm Statistics

The Distributed Alarm System provides a built-in alarm statistics dialog box. The application developer can design the application to call up the **Alarm Statistics** dialog box to list the status of the current query for a particular alarm display.

The **Alarm Statistics** dialog box provides you with an overview of the current alarm query for a particular alarm display. It lists the actual alarm providers requests and the results of each. It's important to note that even though you may have requested a single Alarm Group List name, that name may equate to several individual Alarm Provider queries. For example:



100 \intouch!$system

Each row in the dialog box lists a number and a query. The number represents the percentage of that query that has been returned. The dialog box provides a static display of the query results.

➢ **To update the Percent of Alarms Retrieved in Query list:**

1. Click **Update**.
2. Click **OK** to close the dialog box.

Distributed Alarm Properties and Functions

The distributed alarm system includes multiple tagname **.fields** and QuickScript functions. The following section briefly describes how you can use the QuickScript functions.

🕮 For more information and examples of how to use the QuickScript functions, see your *InTouch Reference Guide*.

Alarm Display Monitoring Properties

The alarm display has several QuickScript-exposed properties that can be used to monitor the status of the display at runtime. These properties are accessible through the *GetPropertyX( )* function, where X is the data type (D for Discrete, I for Integer, and M for Message).

Query Properties	Description
.AlarmGroup	Message property containing the current query list.
.PriFrom	Integer property containing the current query priority low filter value.
.PriTo	Integer property containing the current query priority high filter value.
.QueryType	Integer property containing the current query type (1 = History; 2 = Summary).

For more information, see "Dynamically Controlling the Display Type."

.QueryState	Integer property containing the current query filter (0 = All, 1 = Unack, 2 = Ack).

Query Status Properties	Description
.Successful	Discrete property containing the current query status (0 = Error, 1 = OK).
.ProvidersReq	Integer property containing the number of alarm providers in the current query.
.ProvidersRet	Integer property containing the number of alarm providers that have successfully returned their query results.

Display Properties	Description
.NumAlarms	Integer property containing the number of alarms in the current query.
.PageNum	Integer property containing the current page number displayed in the alarm display.
.TotalPages	Integer property containing the total number of pages in the alarm display.

Alarm Display Control Properties

The alarm display also has two QuickScript-exposed properties that may be used to control the movement of the display's screen in **Runtime**. These properties are controllable through the *SetPropertyD* function.

Control Properties	Description
.NextPage	Scrolls the alarm display one page down when this property transitions from 0 to 1.
.PrevPage	Scrolls the alarm display one page up when this property transitions from 0 to 1.

Whenever the value of this discrete variable transitions from On (1, True) to Off (0, False), the alarm display object will display the page that corresponds to that QuickScript (Next or Prev.). Once that page is displayed, the discrete variable will automatically reset On (1, True).

Note These functions are provided to ease the conversion of a standard display to the distributed display. Their functionality has been replaced with the scroll bars and the *almMoveWindow* QuickScript function.

Alarm Query QuickScript Functions

The distributed display retrieves alarm information by submitting an alarm query. The parameters of this query and the query type are specified in one of two QuickScript functions: **almDefQuery** and **almQuery**. The specific syntax of these functions is detailed in the *InTouch Reference Guide*.

Function	Description
almDefQuery	Performs a query using the configuration dialog default properties. These properties include: From Priority, To Priority, Alarm List, and Display Type. The default properties can only be changed at development time and are not overwritten by other alarm queries.
almQuery	Performs a query for either summary or historical alarm information. All query properties are provided in this function.

Alarm Acknowledgment QuickScript Functions

The distributed alarm system is capable of acknowledging any alarms that it can query (summary display only). To provide this capability, the distributed alarm system includes four alarm acknowledgment QuickScript functions: **almAckAll**, **almAckDisplay**, **almAckSelect**, and **almAckRecent**. These functions supplement the **.Ack** dot field which the standard alarm system uses to acknowledge local alarms, Alarm Groups, and group vars. The specific syntax of these functions is addressed in the *InTouch Reference Guide*.

Function	Description
almAckAll	Acknowledges all the alarms in the current alarm query. Since the alarm display has only a limited display area, the **almAckAll** function may acknowledge alarms that are not visible in the display.
almAckDisplay	Acknowledges only those alarms that are currently visible in the alarm display.

almAckSelect	The distributed alarm display allows alarms to be selected by clicking on them with the mouse at runtime. The **almAckSelect** function can be used to acknowledge those alarms.
almAckRecent	Acknowledges only the most recent alarm that has occurred in the current alarm query.

Alarm Display Manipulation QuickScript Functions

The distributed alarm system provides several QuickScript functions to manipulate the display object. These functions allow movement of the display window, selection of alarms within the display, and display of the statistics window.

& The specific syntax of these functions is addressed in the online *InTouch Reference Guide*.

Function	Description
almMoveWindow	Provides commands to manipulate the display window. These commands include: Page Up, Page Down, Scroll Right, Scroll Left, Line Up, Line Down, Top, End, and more.
almSelectAll	Toggles the selection of all the alarms in a display. Since the alarm display has only a limited display area, the **almSelectAll** function may select alarms that are not visible in the display.
almSelectItem	Toggles the selection of the item that is highlighted in an alarm display.
almShowStats	Displays the alarm statistics dialog box.

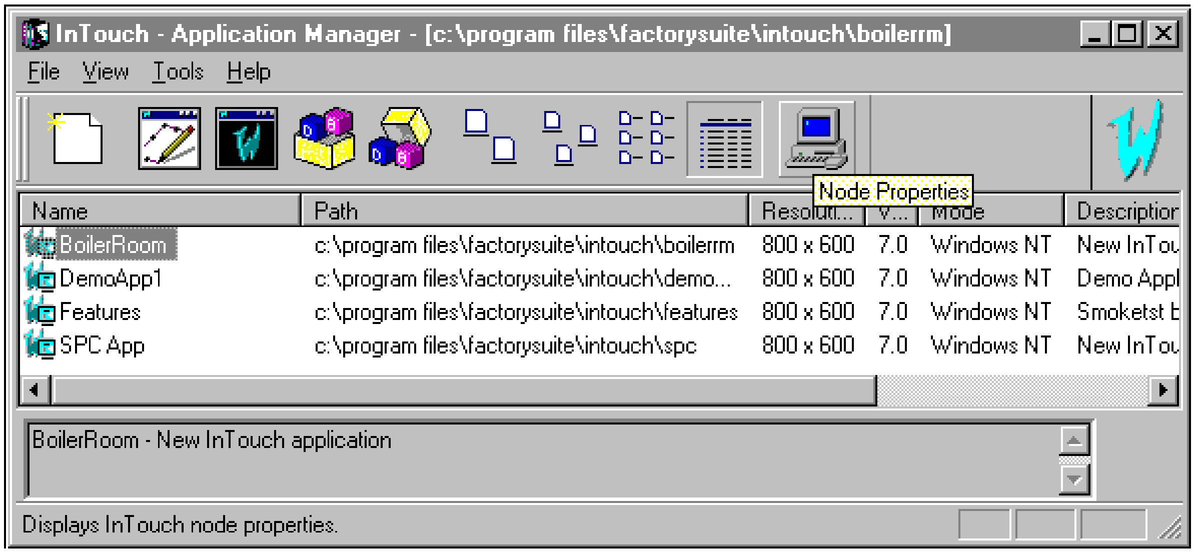
Configuring a Node for Distributed Alarms

Most configurations for InTouch applications are defined in WindowMaker. These configuration settings reside in the application and are copied to wherever the application is copied.

However, in a distributed environment, certain settings may be unique to each View node that runs an application. These settings are, therefore, configured at the View node instead of in the application that is common to all nodes. The distributed alarm system, provides two such settings: "Alarm Server" and "Alarm Provider." Both of these settings are specific only to the behavior of a the View node and are not a part of the InTouch application it is running.

➢ **To configure a node as an alarm server or alarm provider:**

1. Start the InTouch program (INTOUCH.EXE). The **InTouch - Application Manager** dialog box appears:


InTouch - Application Manager - [c:\program files\factorysuite\intouch\boilerrm]
File View Tools Help
Node Properties

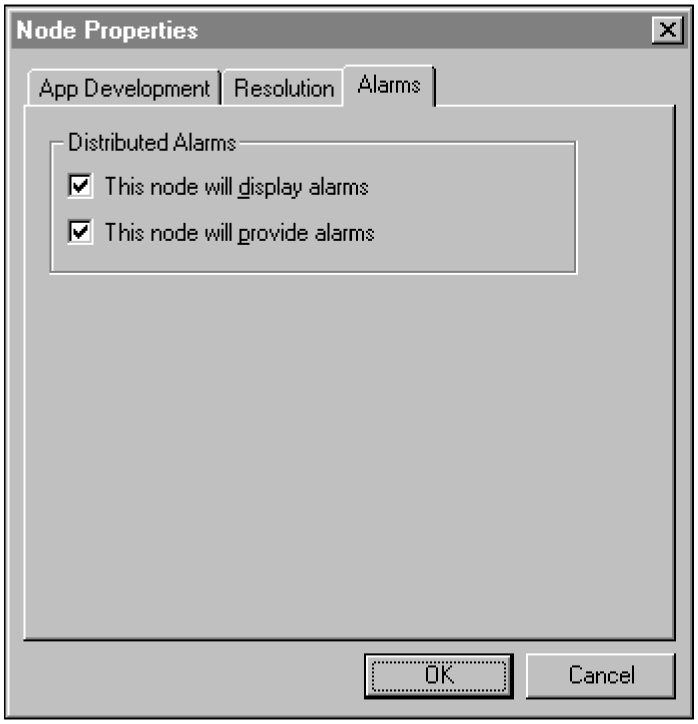
Name	Path	Resolu...	V...	Mode	Descriptior
BoilerRoom	c:\program files\factorysuite\intouch\boilerrm	800 x 600	7.0	Windows NT	New InTou
DemoApp1	c:\program files\factorysuite\intouch\demo...	800 x 600	7.0	Windows NT	Demo Appl
Features	c:\program files\factorysuite\intouch\features	800 x 600	7.0	Windows NT	Smoketst b
SPC App	c:\program files\factorysuite\intouch\spc	800 x 600	7.0	Windows NT	New InTou

BoilerRoom - New InTouch application

Displays InTouch node properties.

2. Click the **Node Properties** tool. The **Node Configuration** dialog box appears with the **App Development** property sheet active.

3. Click the **Alarms** tab to activate **Alarms** property sheet:



Distributed Alarms

- [✓] This node will display alarms
- [✓] This node will provide alarms

4. In the **Distributed Alarms** group select the options that you want to use as follows:

This node will display alarms	Sets the local node to display distributed alarms. When you select this option, the node will start a background task called Alarm Manager. This task will allow the node to connect to the distributed alarm system. This setting must be set for the distributed alarm display to show any alarms.
This node will provide alarms	Sets the local node to act as an alarm provider and serve alarms to other nodes. distributed alarms. When you select this option, the node will start two background tasks called Alarm Manager and Alarm. These tasks will allow the node to connect to the distributed alarm system and provide alarms. This setting must be set for the distributed alarm display to show local alarms.

5. Click **OK**.

Using Both Alarm Systems in an Application

The distributed alarm system is an extension of the standard alarm system. Both are fully compatible, and can be used together in the same application. The table summarizes the major differences between the two systems from an application developer's perspective:

Feature	**Standard Alarm**	**Distributed Alarm**
Acknowledging alarms	Use .*Ack* and the Ack function	Use *ackAlmX* QuickScript functions
Alarm display colors	Two colors (one for acknowledged alarms and one for unacknowledged ones)	Eight colors (based on priority range and acknowledged status)
Alternating an alarm display between Summary and History type displays	Requires two displays, one set up as Summary and one set up as History	Needs only one display which can perform multiple queries and switched dynamically in runtime
Categorizing Alarms	Use Alarm Group	Use Alarm Group and Group Lists
Changing displays	Use GroupVar to change Alarm Group	Use *almQuery* QuickScript function
Paging the alarm display	PgUp, PgDn	Scroll bars and a*lmMoveWindow* QuickScript
Resizing columns	Development only	Runtime or Development
Selecting alarms for acknowledgment	Not available	Alarms can be selected via mouse action or by *almSelectX* QuickScript functions

Changing a Standard System to a Distributed System

Modifying a standard alarm system to include the capabilities of a distributed system is a fairly easy task. The following list provides you with the main items that you need to implement this change:

1. In WindowMaker, paste a distributed alarm display object to your window. (Click the wizard tool, select the **Alarm Displays** category. Double-click the **Dist. Alarm Display** wizard, and then click in the window to paste it.)
2. Size the alarm display to the same approximate size as the old display. Remember to account for the scroll bars which may limit the number of alarm display lines.
3. Double-click on the object to open its configuration dialog box. Modify the settings as applicable for your application.
4. If your existing window used Page Up and Page Down buttons for the alarm display, you can either use those to activate the new alarm display by replacing the existing button scripts with *almMoveWindow(ObjectName, Option, Repeat)* where *Option* = *'PageUp'* or '*PageDn'*, or you can eliminate the buttons and use the built-in scroll bars.
5. If your existing window used an Alarm Acknowledge function, you can continue to use the current button script to acknowledge local alarms, or you can add the appropriate QuickScript functions to acknowledge alarms in the display query.

   For more information, see "Alarm Acknowledgment QuickScript Functions."
6. Your standard alarm display more than likely used a **Group Var** type tagname for redirecting Alarm Groups. While **Group Vars** can be used to refer to local Alarm Groups, they cannot be used to reference Alarm Group Lists or remote Alarm Groups. The distributed alarm display and QuickScript functions do not **support Group Var**s. To replace **Group Var**s in your application, substitute each **Group Var** type tagname for a Message tagname. Whenever the **Group Var** is set to a value, instead set the Message tagname to the name of the Alarm Group. Finally, set up a Data Change script based on that message tagname that will submit a new query to the alarm display using the *almSummQuery()* function.

C H A P T E R 8

Real-time and Historical Trending

InTouch provides you with two types of trend display objects: "Real-time" and "Historical. You can configure both trend objects to display graphical representations of multiple tagnames over time. Real-time trends allow you to chart up to four pens (data values), while Historical trends allow you to chart up to eight pens. Both types of trends are created using special tools in WindowMaker. InTouch also provides you with complete control over the configuration of your trends. 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.

📖 The FactorySuite Productivity Pack includes a Pen Configuration Grid that allows you to chart 16 pens. For more information, see your *Productivity Pack User's Guide.*

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.

In addition to its trending capabilities, InTouch, includes two utilities, HDMerge and HistData that are designed to work with InTouch historical log files. The HistData utility converts encrypted historical log files (.LGH) to comma separated variable (.CSV) files for use in spreadsheet or text editing environments such as Microsoft Excel. The HDMerge utility merges .CSV log file into historical log files.

Contents

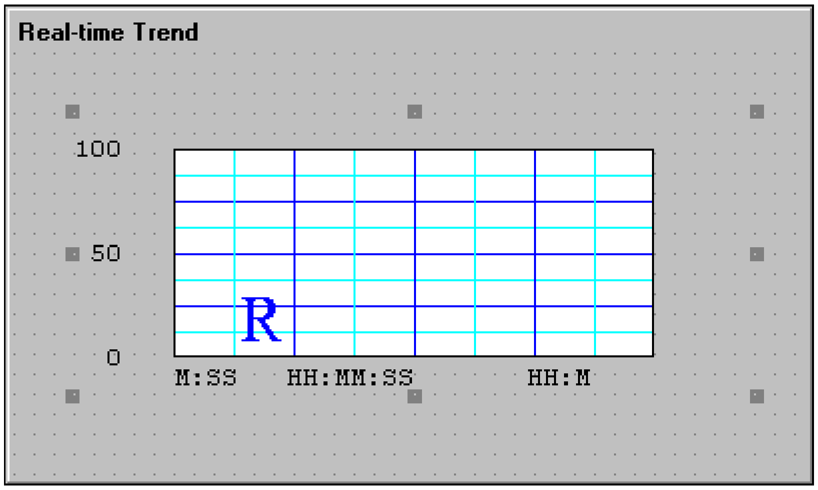
Real-time Trends

Real-time trends are dynamic. They are updated continuously during runtime. They plot the changes of up to four local tagnames or expressions as they occur.

Creating a Real-time Trend

➢ **To create a real-time trend:**

1. Select the real-time trend tool in the **Wizard Toolbar**.
2. Click in the window, then drag the mouse diagonally to draw a rectangle the size that you want your trend to be. (You can draw the trend chart any size you choose, and there is no limit to the number of charts you can place on a screen.)
3. Release the mouse. The real-time trend object appears in the window:



In runtime, the data is written in the trend from the right to the left.

4. Double-click the trend to open its configuration dialog box.

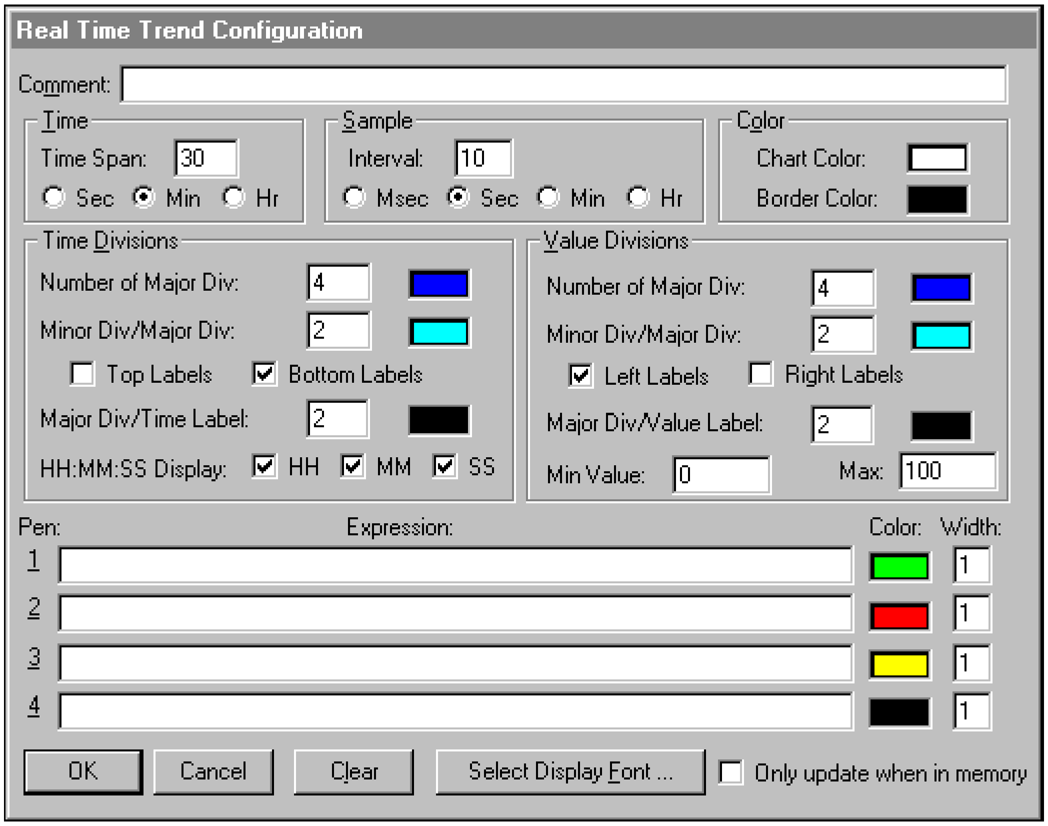
A trend object is like any other object drawn in WindowMaker. It can be moved by grabbing it with the mouse or it can be resized by grabbing one of the object "handles." You can place multiple trends in a window.

Configuring a Real-time Trend

The first time you paste a real-time trend object, the system default configuration settings are used. Once you have configured a real-time trend, the next one you create will, by default, be configured with the same settings.

➢ **To configure a real-time trend:**

1. Double-click the trend or, with the trend selected, on the **Special** menu, click **Animation Links**. The **Real Time Trend Configuration** dialog box appears:



Real Time Trend Configuration

Comment:

Time — Time Span: 30 — Sec [✓ Min] Hr

Sample — Interval: 10 — Msec [✓ Sec] Min Hr

Color — Chart Color: ⬜ — Border Color: ⬛

Time Divisions — Number of Major Div: 4 🟦 — Minor Div/Major Div: 2 🟦 — ☐ Top Labels ☑ Bottom Labels — Major Div/Time Label: 2 ⬛ — HH:MM:SS Display: ☑ HH ☑ MM ☑ SS

Value Divisions — Number of Major Div: 4 🟦 — Minor Div/Major Div: 2 🟦 — ☑ Left Labels ☐ Right Labels — Major Div/Value Label: 2 ⬛ — Min Value: 0 — Max: 100

Pen:	Expression:	Color:	Width:
1		🟩	1
2		🟥	1
3		🟨	1
4		⬛	1

OK — Cancel — Clear — Select Display Font ... — ☐ Only update when in memory

If you right-click a text box in the real-time trend configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

Note All entries made in the **Real Time Trend Configuration** dialog box are independent of the size of the trend and are not modifiable at runtime.

2. In the **Time Span** box, type length of time you want to display horizontally (x-axis) on the trend then select time increment option for the length of time.

   For example, it you enter 30 for the **Time Span** then select **Min**, the horizontal time span of the chart will be 30 minutes long.

3. In the **Sample Interval** box, type 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** box to open the InTouch color palette. Click the color in the palette that you want to use for the trend's background.

5. In the **Color** group, click the **Border Color** box to open the InTouch color palette. Click the color in the palette that you want to use for the trend's border.

   Repeat this process for all color selections.

6. In the **Time Divisions** group, in the **Number of Major Div** box, type the number of major time divisions you want in the trend, and then select the color you want to use 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 **Time Divisions** group, in the **Minor Div/Major Div** box, type the number of minor time divisions that you want to be visible within each major time division, and then select the color you want to use for the division lines.

8. In the **Time Divisions** group, select **Top Labels** if you want time labels displayed at the top of the trend.

9. In the **Time Divisions** group, select **Bottom Labels** if you want time labels displayed at the bottom of the trend.

   Your trend can have both top and bottom labels or no labels at all.

10. If you are using time labels, in the **Time Divisions** group, in the **Major Div/Time Label** box, type the number of time labels per major time division line that you want for the trend.

11. In the **Time Divisions** group, select the color you want to use for the major time division lines.

12. The settings in **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 to do so. For example, 0.00 to 100.00.

13. In the **Expression** box, type the local tagname or expression that you want each **Pen** to 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 that you want each pen to use to plot each tagname in the trend.

15. In the **Width** box, type the number of pixels wide you want each pen 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 to select the font, style and size that you want to use when you print the trend.

17. Select **Only update when in memory** if you want your trend to update only when it is displayed in the active window.

    If you do not select this option, the trend will always be updated, even if it is not in an open window. This may result in slightly slower system performance of the overall system.

18. Click **OK**.

➢ **To Increase 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 you set the Time Span to 30 minutes and the Sample Interval to 2 seconds, the number of samples taken during the 30 minutes will be calculated as:

   30* 60/2 = 900

   If you set the Time Span to 30 minutes and the Sample Interval to 5 seconds, the number of samples taken during the 30 minutes will be calculated as:

   30* 60/5 = 360

Historical Trends

Historical trends provide you with a "snapshot" 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, for example, 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. You can create "scooters" 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 to you.

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. You can create custom scales and link them 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.

Note You must select the **Log Data** option for each tagname in the Tagname Dictionary in order for it to be trended.

For more information on logging tagnames, see "Logging Tagnames."

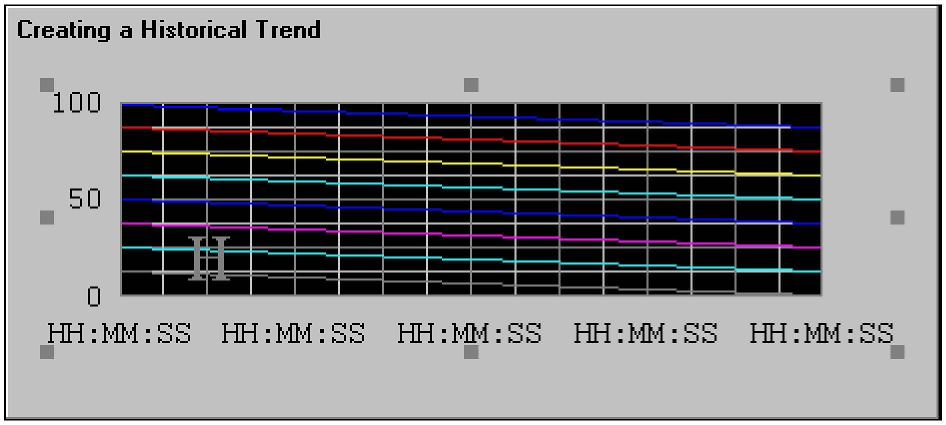
Also see, "Configuring Historical Logging Properties."

Creating a Historical Trend

➤ **To create a historical trend:**

1. Select the historical trend tool in the **Wizard Toolbar**.Historical trend tool used to draw historical trend objects.
2. Click in the window, then drag the mouse diagonally to draw a rectangle the size that you want your trend to be.

   You can draw the trend chart any size you choose. You can also place multiple trends in your window.
3. Release the mouse. The historical trend appears in the window:



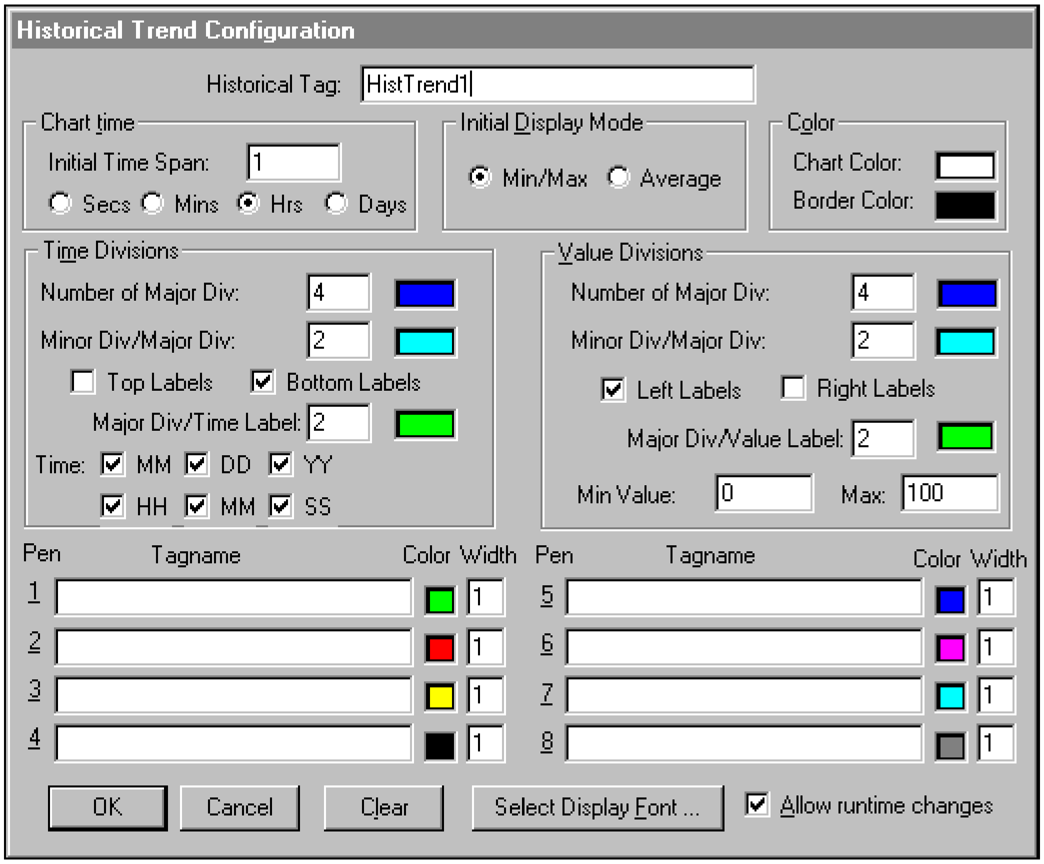
A trend object is like any other object drawn in WindowMaker. It can be moved by grabbing it with the mouse or it can be sized by grabbing any its "handles."

Configuring a Historical Trend

The first time you paste a historical trend object, the system default configuration settings are used. Once you have configured a trend, the next one you create will, by default, be configured with the same settings.

➢ **To configure a Historical trend:**

1. Double-click the trend or, with the trend selected, on the **Special** menu, click **Animation Links**. The **Historical Trend Configuration** dialog box appears:



Historical Trend Configuration

Historical Tag: HistTrend1

Chart time
Initial Time Span: 1
Secs Mins [✓ Hrs] Days

Initial Display Mode
[✓ Min/Max] Average

Color
Chart Color: ⬜
Border Color: ⬛

Time Divisions
Number of Major Div: 4 🟦
Minor Div/Major Div: 2 🟦
☐ Top Labels ☑ Bottom Labels
Major Div/Time Label: 2 🟩
Time: ☑ MM ☑ DD ☑ YY
☑ HH ☑ MM ☑ SS

Value Divisions
Number of Major Div: 4 🟦
Minor Div/Major Div: 2 🟦
☑ Left Labels ☐ Right Labels
Major Div/Value Label: 2 🟩
Min Value: 0 Max: 100

Pen	Tagname	Color	Width	Pen	Tagname	Color	Width
1		🟩	1	5		🟦	1
2		🟥	1	6		🟪	1
3		🟨	1	7		🟦	1
4		⬛	1	8		⬛	1

☑ Allow runtime changes

If you right-click a text box in any historical trend configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. In the **Historical Tag** box, type the tagname that you want to use for the trend.

If the tagname you type is not currently defined in the Tagname Dictionary, you will be asked if you want 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, type length of time you want to display horizontally (x-axis) on the trend then select time increment option for the length of time.

   Example: If you enter 30 for the **Initial Time Span** then select **Min**, the horizontal time span of the chart will be 30 minutes long.

4. Select **Initial Display Mode** that you want to use for the trend as follows:

   Min/Max - Each pixel on the chart will display the minimum to maximum range the point covered in the time represented by that pixel.

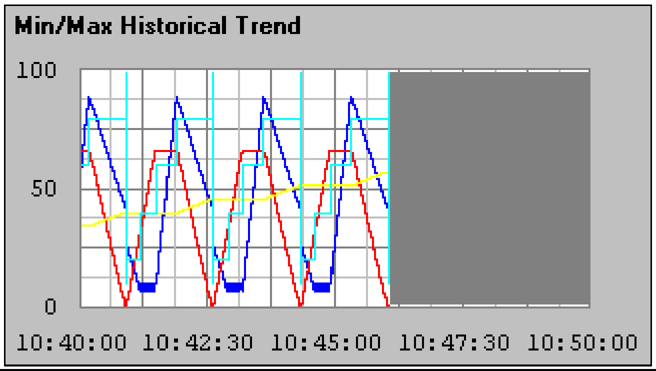
   Average - Displays the average value for each pixel, for example, time segment.

5. In the **Color** group, click the **Chart Color** box to open the InTouch color palette. Click the color in the palette that you want to use for the trend's background.

6. In the **Color** group, click the **Border Color** box to open the InTouch color palette. Click the color in the palette that you want to use for the trend's border.

   Repeat this process for all color selections.

Note The blank area on the right side indicates that no data was collected during that time period either because WindowViewer was not running or, historical logging was turned off.



7. In the **Time Divisions** group, in the **Number of Major Div** box, type the number of major time divisions you want in the trend, and then select the color you want to use for the division lines.

   The number of major time divisions must be an even multiple of the number of **Minor Div/Major Div**.

8. In the **Time Divisions** group, in the **Minor Div/Major Div** box, type the number of minor time divisions that you want to be visible within each major time division, and then select the color you want to use for the division lines.

9. In the **Time Divisions** group, select **Top Labels** if you want time labels displayed at the top of the trend.

10. In the **Time Divisions** group, select **Bottom Labels** if you want time labels displayed at the bottom of the trend.

    Your trend can have both top and bottom labels or no labels at all.

11. If you are using time labels, in the **Time Divisions** group, in the **Major Div/Time Label** box, type the number of time labels per major time division line that you want for the trend.

12. In the **Time Divisions** group, select the color you want to use for the major time division lines.

13. The settings in **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.

To display decimal points for the minor and major value divisions at runtime, they must be formatted here to do so. For example, 0.00 to 100.00.

14. In the **Expression** box, type the local tagname or expression that you want each **Pen** to trend.

    Up to eight pens can be visible in a trend. (Message type tags cannot be logged or trended.)

    The FactorySuite Productivity Pack includes a Pen Configuration Grid that allows you to chart 16 pens. For more information, see your *Productivity Pack User's Guide.*

15. Click the color box to select the color that you want each pen to use to plot the tagname in the trend.

16. In the **Width** box, type the number of pixels wide you want each pen to be.

    Selecting a pen width greater than 1 significantly reduces performance in screen updating and printing.

17. Select **Allow runtime changes** if you want the operator to be able to make changes to the trend's configuration in runtime. These changes include changing pen assignments, start date, time and so on.

    If you select this option, when the operator clicks on the trend (or touches it when using a touch screen) in runtime, the **Runtime Setup** dialog box will appear and he will be able to make changes to the trend.

    For more information, see "Updating a Historical Trend During Runtime."

18. Click **Select Display Font** to access the **Font** dialog box to select the font, style and size that you want to use for the trend display.

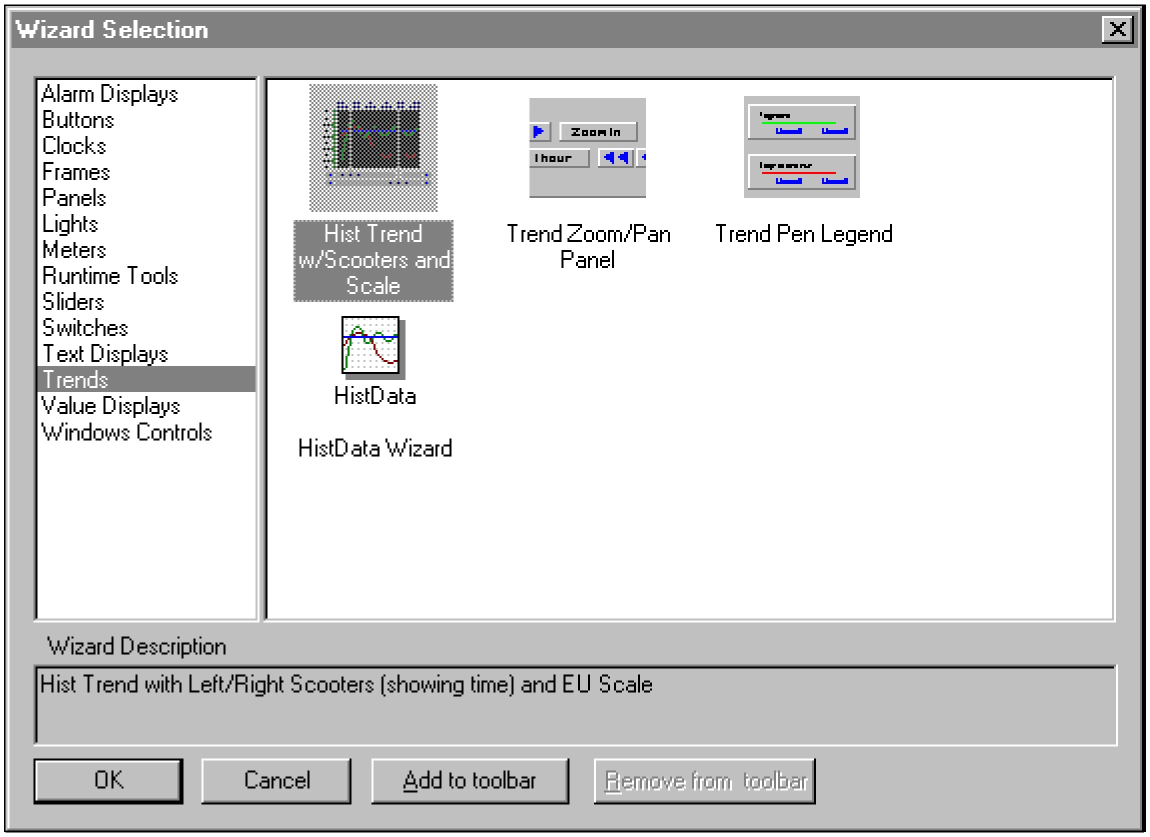
19. Click **OK**.

Using Historical Trend Wizards

InTouch provides you with a quick and easy method to create a historical trend: the trend wizard. The trend wizard allows you to configure a full-featured historical trend with scooters, zooming, and so on., with just a few mouse clicks.

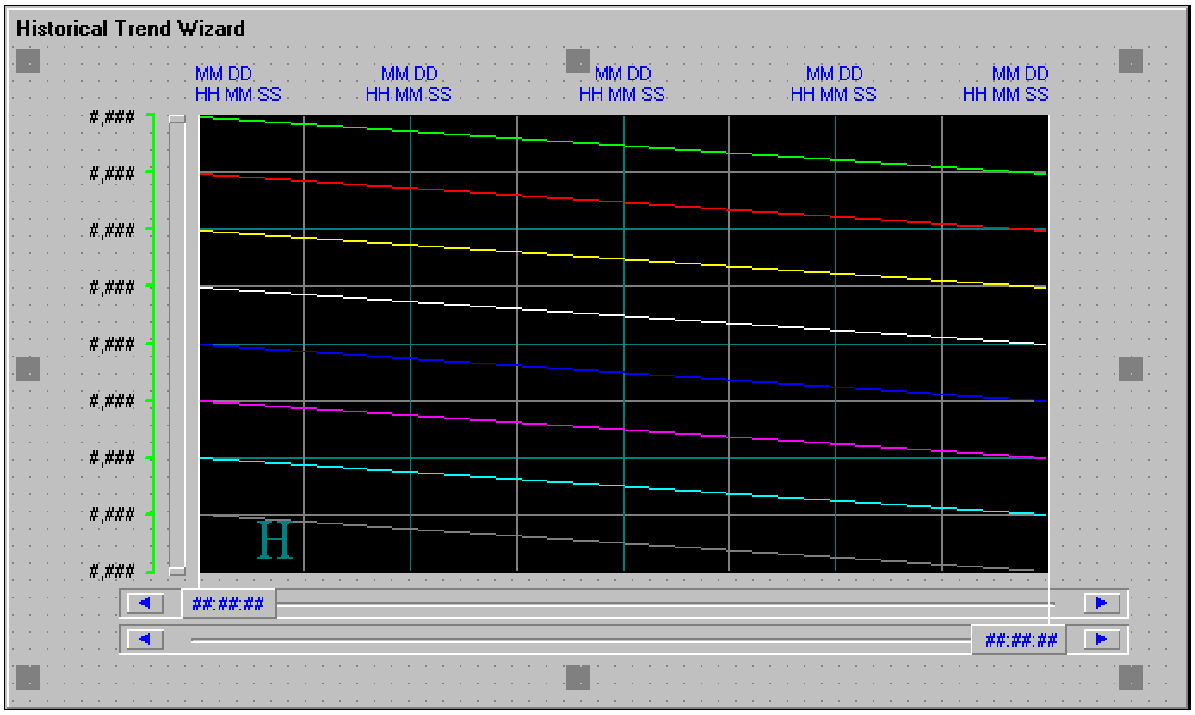
➢ **To use a historical trend using the wizard:**

1. Click the wizard tool in the **Wizard Toolbar**. The **Wizard Selection** dialog box appears:



Hist Trend with Left/Right Scooters (showing time) and EU Scale

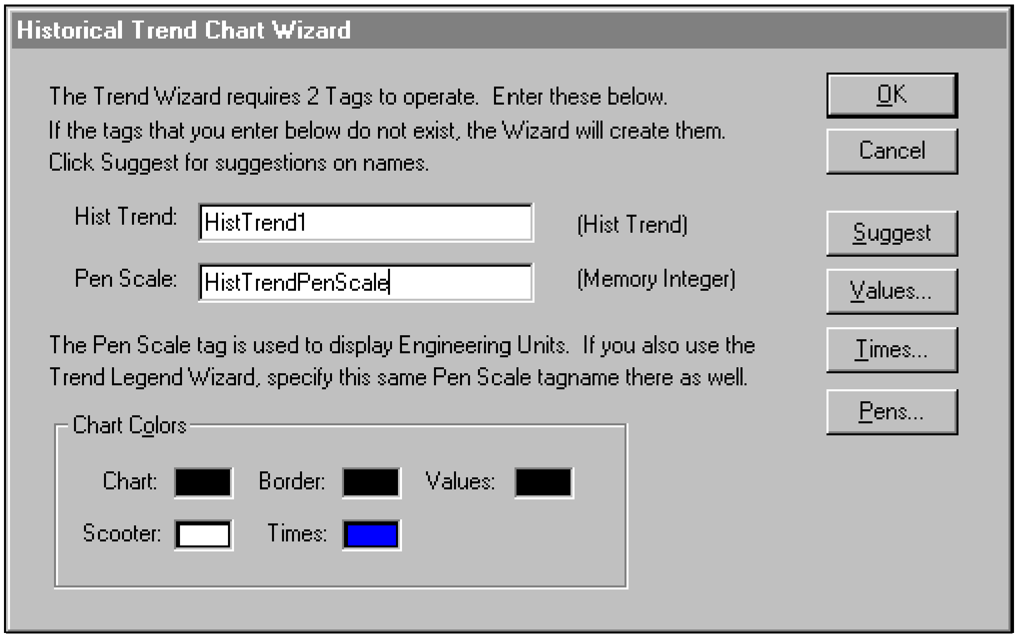
2. Select **Trends** in the list of wizards to display the available trend wizards.
3. Select the **Hist Trend with Scooters** wizard, then click **OK**. The dialog box closes and your window reappears with the cursor in the "paste" mode.
4. Click in the window to paste the trend wizard:



A trend object is like any other object drawn in WindowMaker. It can be moved by grabbing it with the mouse or it can be resized by grabbing one of the object "handles." You can place multiple trends in a window.

5. You are now ready to configure the trend wizard.

6. Double-click on the trend wizard to open the **Historical Trend Chart Wizard** configuration dialog box:



Historical Trend Chart Wizard

The Trend Wizard requires 2 Tags to operate. Enter these below.
If the tags that you enter below do not exist, the Wizard will create them.
Click Suggest for suggestions on names.

Hist Trend: HistTrend1 (Hist Trend)

Pen Scale: HistTrendPenScale (Memory Integer)

The Pen Scale tag is used to display Engineering Units. If you also use the Trend Legend Wizard, specify this same Pen Scale tagname there as well.

If you right-click a text box in the wizard configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

7. Enter the required information to configure the trend, then click **OK**.

Click **Suggest** if you want the wizard to automatically fill in the configuration settings. The settings you configure for a historical trend wizard are the same as those you configure when you create the historical trend object drawn using WindowMaker's trend tool in the **Draw Object Toolbar**.

For more information, see "Configuring a Historical Trend."

8. 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 all work together, they must use the **Hist Trend** tagname.

   Like all InTouch wizards, this wizard can be broken into its individual components.

➢ **To break the wizard:**

1. Select the historical trend wizard.
2. On the **Arrange** menu, click **Break Cell** or, click the break cell tool in the **Arrange Toolbar**.
3. You can then customize it to your needs.

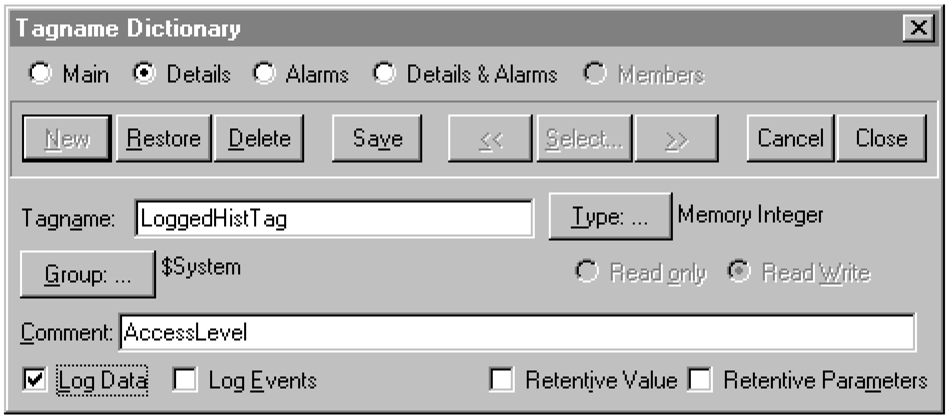
Logging Tagnames

In WindowViewer, the value 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.

➢ **To configure a tagname for logging:**

1. On the **Special** menu, click **Tagname Dictionary** or, in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box appears:



Details selected. Tagname: LoggedHistTag; Type: Memory Integer; Group: $System; Comment: AccessLevel; Log Data [✓].

2. Open the desired tagname's definition, then select **Log Data**.

   In order for your tagnames to actually be logged, you enable logging as described in the next section.

   If you change a tagname 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.

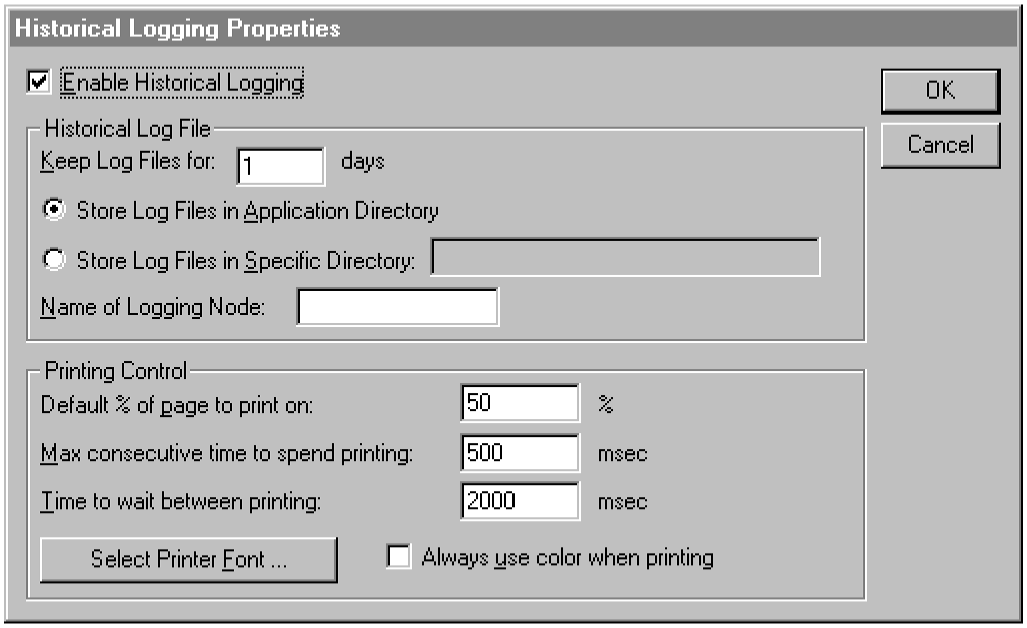
Note The Min/Max Engineering Units are very important for displaying historical trend data. The historical trend displays from 0-100% of EU scale.

Configuring Historical Logging Properties

In order for the tagnames that you have configured with the **Log Data** option to be written to the historical log file, the global logging function must be enabled.

➤ **To configure historical logging:**

1. On the **Special** menu, point to **Configure**, and then click **Historical Logging**. The **Historical Logging Properties** dialog box appears:
2. To quickly access the dialog box, in the Application Explorer under **Configure**, double-click **Historical Logging**.



Historical Logging Properties

[✓] Enable Historical Logging

Historical Log File
Keep Log Files for: 1 days
(●) Store Log Files in Application Directory
( ) Store Log Files in Specific Directory:
Name of Logging Node:

Printing Control
Default % of page to print on: 50 %
Max consecutive time to spend printing: 500 msec
Time to wait between printing: 2000 msec
[ ] Always use color when printing

If you right-click a text box in any historical trend configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

3. Select **Enable Historical Logging** to turn on global tagname logging.
4. In the **Keep Log Files for** box, type the number of days' (prior to today) log files that you want to keep saved to disk.

Note InTouch will create and save two historical log files each day (24 hours). Therefore, disk space must be considered when you set this value. If your hard disk does not have enough free space to save a historical log file, logging will stop and you must free disk space then restart logging. You can start and stop historical logging in runtime by linking the **$HistoricalLogging** internal tagname to a button or QuickScript or by using the **Restart Historical Logging** Command in WindowViewer.

For example, if you type 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 save on disk. The file for the 1st will automatically be deleted. If you type a zero, the log files are kept indefinitely.

5. Select **Store Log Files in Application Directory** if you want the historical log file to be saved in your application directory. Or, select **Store Log Files in specific Directory** and type the complete path to the directory that you want to use.

This entry must be either a DOS path such as C:\HISTLOG or, if you are performing distributed history, it must be a Universal Naming Convention (UNC) path such as \\NODE\SHARE\DIRECTORY.

For more information, on distributed history, see "The Distributed History System."

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

Note This version of InTouch supports the newer log files with extensions of **.LGH** and **.IDX**. Earlier versions of InTouch used the extension **.LOG** for log files.

6. In the **Name of Logging Node** box, type the NetDDE node name (not the computer name) for the node that will be logging to the history log file.
7. In the **Default % of page to print on** box, type the percentage ratio for the page size to trend size.

   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.

   There are many factors that affect the performance of printing historical trends. The primary performance factor is the size of the chart on the printed page. You can improve performance by reducing the percentage of the page that is used.
8. In the **Max consecutive time to spend printing** type 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, type 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.

    For more information on fonts, see Chapter 2 - Working with Text Objects.

Configuring a Historical Trend in Runtime

If you selected the **Allow runtime changes** option 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, and so on.

➢ **To update a historical trend in runtime:**

1. Click the trend in WindowViewer, the **Historical Trend Setup** dialog box appears:

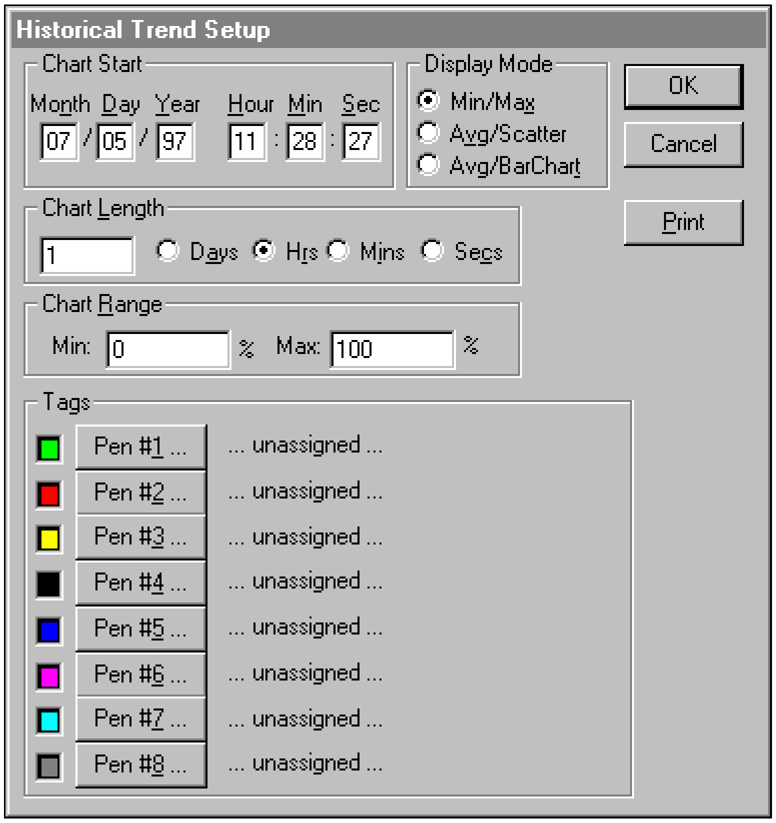

Historical Trend Setup

Chart Start

Month Day Year: 07 / 05 / 97

Hour Min Sec: 11 : 28 : 27

Display Mode: [✓ Min/Max] Avg/Scatter Avg/BarChart

Chart Length: 1 Days [✓ Hrs] Mins Secs

Chart Range: Min: 0 % Max: 100 %

Tags

🟩	Pen #1 ...	... unassigned ...
🟥	Pen #2 ...	... unassigned ...
🟨	Pen #3 ...	... unassigned ...
⬛	Pen #4 ...	... unassigned ...
🟦	Pen #5 ...	... unassigned ...
🟪	Pen #6 ...	... unassigned ...
▨	Pen #7 ...	... unassigned ...
▨	Pen #8 ...	... unassigned ...

2. In the **Chart Start** group, type the starting date and time for the chart.

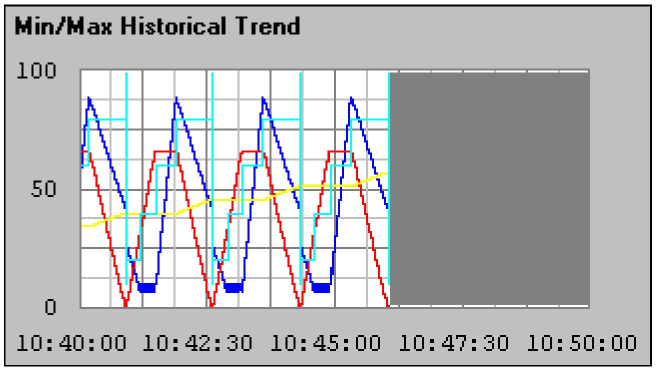
3. Select the **Display Mode** for your chart. There are three modes as illustrated and described in the examples below:

Note The display mode of the trend affects performance. The primary is the length of the lines being drawn to generate the trend. The longer the lines, the longer it takes to generate the trend. Line widths are also a performance factor; wide lines take significantly longer to draw. **Min/Max** or **Average/Scatter** trends are generally much faster to generate than **Average/Bar Chart**.

There are three modes as illustrated and described in the examples below:

Min/Max Historical Trend Example

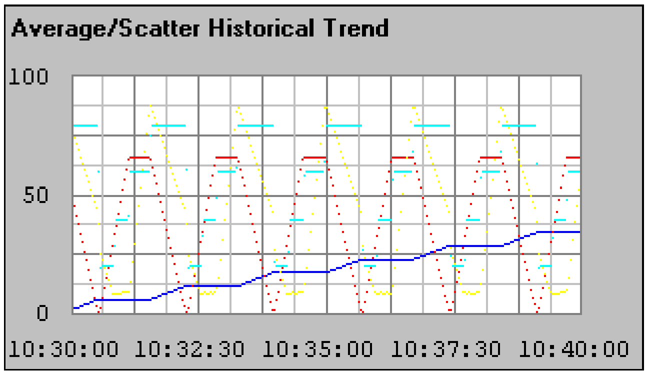
This mode displays the trends or changes in the percentage of Engineering Units scale as a vertical line over the time span with emphasis on time flow and rate-of-change, rather than amount of change.



Note The blank area on the right side indicates that no data was collected during that time period either because WindowViewer was not running or historical logging was turned off.

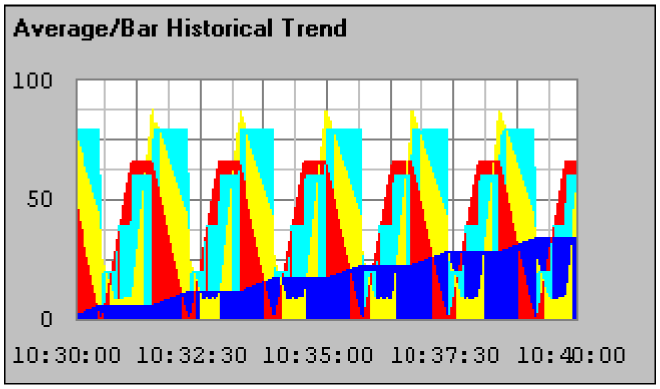
Average/Scatter Historical Trend Example

This mode shows the average value of the point during the time intervals.



Average/Bar Chart Historical Trend Example

This mode shows the average value of the point during the time intervals in bar form.



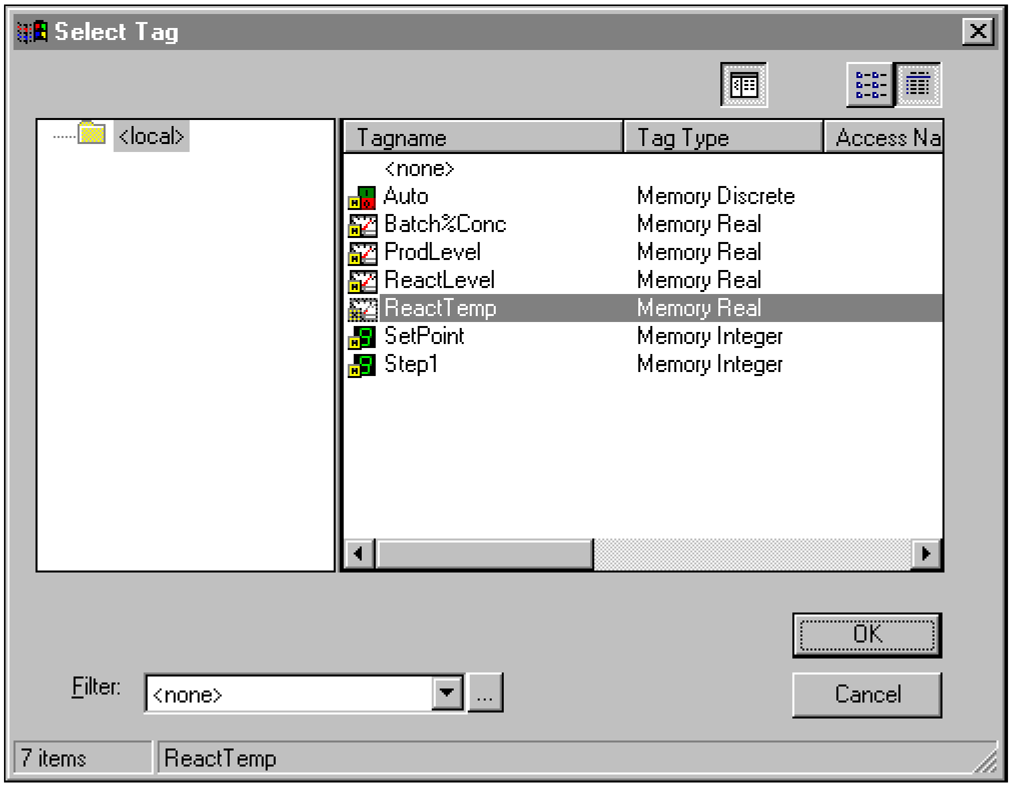
4. In the **Chart Length** box, type the horizontal (x-axis) length of time to be displayed on the trend, and then select the time increment for the length.

   If you type a 1 and select **Hrs**, your trend will be 1 hour long.

5. In the **Chart Range** boxes, type 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. For example, if you want to trend the variance of the selected tags from 40 to 45 percent of scale, enter 40 and 45 in the **Min** and **Max %** range boxes respectively.

6. Click each **Pen#** to select the tagname that you want the pen to trend. The Tag Browser appears in the filtered selection mode:



Tagname	Tag Type	Access Na
<none>		
Auto	Memory Discrete	
Batch%Conc	Memory Real	
ProdLevel	Memory Real	
ReactLevel	Memory Real	
ReactTemp	Memory Real	
SetPoint	Memory Integer	
Step1	Memory Integer	

   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 that you want the selected pen to plot on the trend, or select the tagname, and then click **OK**. The **Historical Trend Setup** dialog box will reappear showing 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 you can use 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.

   When you use a filter or, click the **Filter** ... button and create a new filter, the Tag Browser will be repopulated with all tagnames defined with the **Log Data** option that meeting the criteria specified in the filter for the selected tag source.

   For more information on the Tag Browser and filters, see Chapter 4 - Tagname Dictionary.

8. Click **Print** to print the historical trend.

   The printing operation takes place "in the background" while WindowViewer continues to process all other inputs. WindowViewer will add two items to its menu during printing: **CancelPrint** and ***X* % Done**. Clicking on **CancelPrint** will cancel the current print job.

   After selecting **Print**, do not change the trend until the **CancelPrint** and **X % Done** items disappear in the WindowViewer menu bar. During this time, WindowViewer is saving the trend information in memory for printing. Once these two items disappear in the menu bar, the trend can be changed without affecting the print that is in progress.

   You can create a button to print the historical trend by linking it to an action QuickScript that executes **PrintHT** QuickScript function.

   ```
   PrintHT(HistTrendTagname);
   ```

   Note The printing operation uses the current historical trend as a basis for printing. Therefore, if any field in the **Historical Trend Setup** dialog box is changed, the **Print** button will not be active. Changes made in the setup cannot be printed until you click **OK** in the **Historical Trend Setup** dialog box, and then access it again and click **Print**.

   For more information on printing historical trends, see "Configuring Historical Trend Printing."

Updating a Historical Trend in Runtime

In WindowViewer when a historical trend is first shown, 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** in order for it to update itself after the initial data display. 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) so the operator can manually change the trend's time and/or date to force the update.

2. Use the following in a QuickScript or on a button to allow the operator to update the chart:

   ```
   Hist_TrendTag.UpdateTrend = 1
   ```

3. Use any of the following in a QuickScript or on a button:

   ```
   HTUpdateToCurrentTime(Hist_Tag);
   HTScrollLeft(Hist_Tag,Percent);
   HTScrollRight(Hist_Tag,Percent);
   HTZoomIn(Hist_Tag,LockString);
   HTZoomOut(Hist_Tag,LockString);
   HTSetPenName(Hist_Tag,PenNum,Tagname);
   ```

4. Change any of the following trend tagname **.fields**:

   .ChartStart
   .ChartLength
   .MaxRange
   .MinRange
   .Pen1-.Pen8

& For more information on using QuickScript functions and .fields , see the online *InTouch Reference Guide.*

Configuring Historical Trend Printing

There are many factors that affect the performance of printing historical trends. The primary performance factor is the size of the trend on the printed page. The display mode of the trend also affects printing performance. **Min/Max** or **Average/Scatter** printouts are usually generated much faster than **Average/Bar Chart** trends. The longer and wider the lines on the trend are, the longer it takes to print.

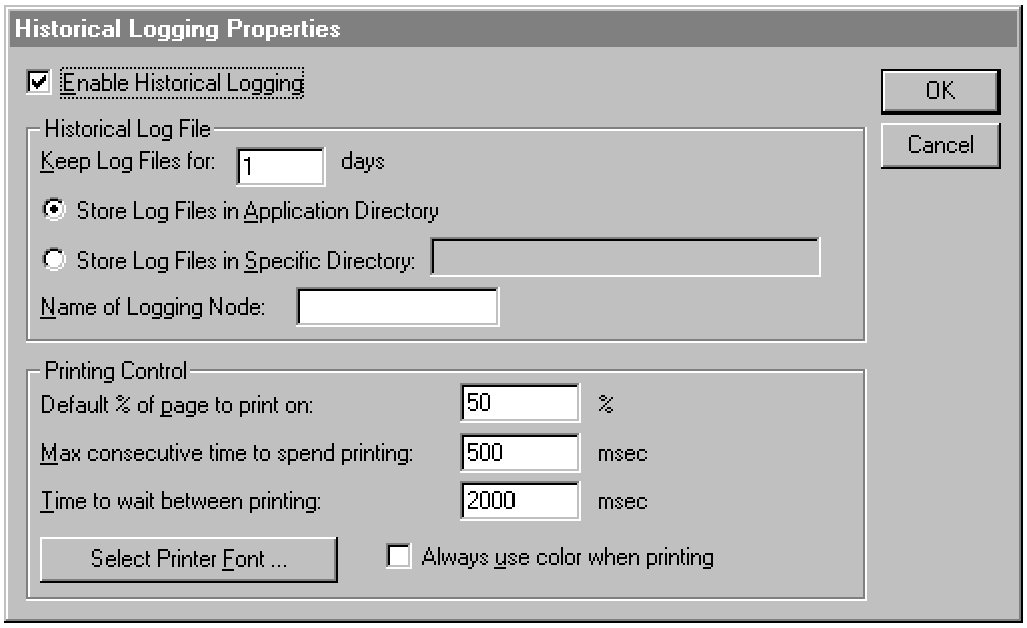
Since the printing operation takes place "in the background", InTouch devotes a certain amount of time to print processing and a certain amount of time to other processing. The times in this equation are controlled by the values set in the **Max consecutive time to spend printing** and **Time to wait between printing** boxes when you configure historical logging.

In other words, InTouch spends the number of milliseconds that you specified in the **Max consecutive time to spend printing** box processing printing and then spends the number of milliseconds that you specified in the **Time to wait between printing** field processing other requests. To raise the priority of printing, increase the value for **Max consecutive time to spend printing** and decrease the value for **Time to wait between printing**. To lower the priority of printing, do the opposite.

➢ **To configure historical trend printing:**

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

   To quickly access the dialog box, in the Application Explorer under **Configure**, double-click **Historical Logging**.



Historical Logging Properties

☑ Enable Historical Logging

Historical Log File

Keep Log Files for: 1 days

● Store Log Files in Application Directory

○ Store Log Files in Specific Directory:

Name of Logging Node:

Printing Control

Default % of page to print on: 50 %

Max consecutive time to spend printing: 500 msec

Time to wait between printing: 2000 msec

   If you right-click a text box in any historical trend configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Specify the percentage of the page to be used for printing the trend in the **Default % of page to print on** box.

   If you type 50 in this box, WindowViewer to use half of the page (vertically and horizontally). A printout this size would take one quarter of the time to prepare as a full page printout.

   As a printing alternative, you may want to investigate using the **PrintWindow** QuickScript function.

3. In the **Max consecutive time to spend printing** box, type the process time slice (milliseconds) the print module will consecutively spend printing.

4. In the **Time to wait between printing** box, type the time(milliseconds) the print module will wait before taking another processor time slice.

   Another factor that affects printing performance is the background color that you select for the trend. In most cases, a white background will print much faster. The best test is to experiment with white versus a color and see if there is a significant difference.

5. Click **Select Printer Font** to access the **Font** dialog box to select the font, style and size that you want to be used for the printout.

6. Click **OK**.

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 briefly described below

📖 For more information on using **.fields**, see your *InTouch Reference Guide.*

.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 only be equated to another TagID tagname. It cannot 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 to monitor and/or control the tagname being trended by a pen.

.Field	Description
.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 you can use to specify the tagname to be trended by each pen, display the value at a scooter location, scroll the trend by a percentage, etc.

📖 For complete examples of how to use these functions and their valid arguments, see your *InTouch Reference Guide*.

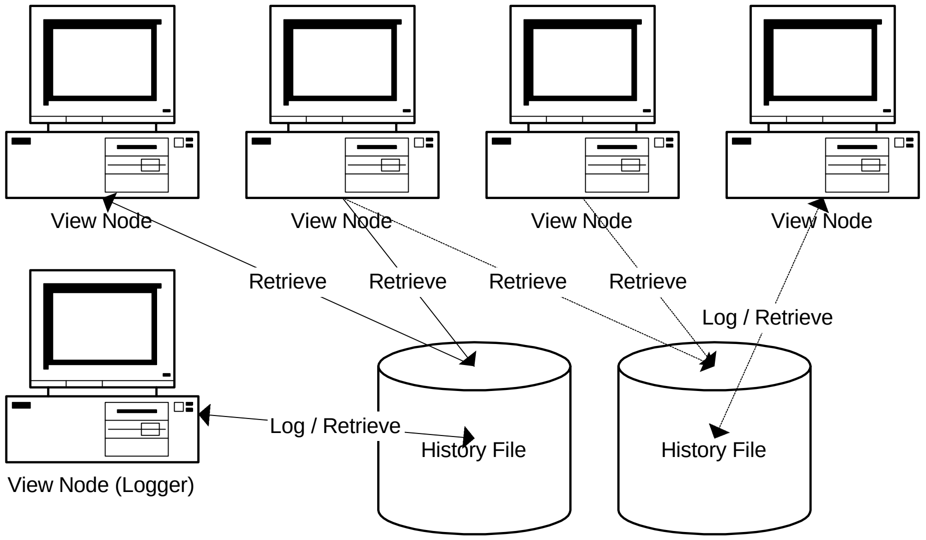
Function	Description
HTGetLastError	Determines if there was an error during the last retrieval of a specified pen.
HTGetPenName	Returns the tagname of the tagname currently used for the specified pen # of the specified trend.
HTGetTimeAtScooter	Returns the time in seconds since 00:00:00 hours GMT, January 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.

Function	Description
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, then 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*.

The Distributed History System

InTouch provides a distributed history system that allows retrieval of historical data from any InTouch application, even those across a network. This system extends the capabilities of the standard InTouch history by allowing remote retrieval of data from multiple historical databases simultaneously. These databases are referred to as history providers. Up to eight history providers can be displayed simultaneously, one for each historical trend chart pen.

Using the capabilities of the distributed history system, you can easily configure a networked system that provides access to multiple history providers:



Each distributed history file is limited to one node writing (logging) to the file. However, there is no restriction on the number or type of InTouch nodes that can view that file.

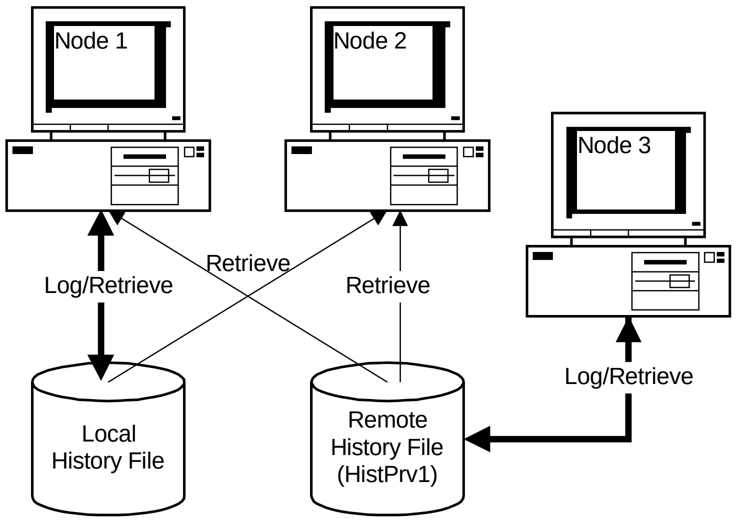
Note Only applications developed in InTouch Version 5.6 or later can be history providers. To remotely view history files from an earlier version, you must first convert that application to the Version 5.6 or later.

A remote node retrieving data from a history file may not see data for the last hour of data (based on the logger node's time). Remote trends can only view data that has been written to the logging node's disk. The logging node will write data to disk after 22 samples have been collected.

Using the Distributed History System

The following diagram illustrates how you should setup your distributed history system. This system is a typical distributed application using Network Application Development (NAD) to distribute the application.

☞ For more information on NAD, see Chapter 3 - Building a Distributed Application.



Nodes 1 and 2 contain copies of the same InTouch application; however, the application is configured to allow only Node 1 to log **to** a local history file, whereas either node can retrieve **from** the local history file or the remote history file. Node 3 is also logging to and retrieving from the same remote history. This provider is assigned the name **HistPrv1**. Node 1 is both a development and runtime station, while Node 2 is just a runtime station.

The major steps you need to perform to create this application include:

1. Create a history provider list.
2. Create and configure a historical trend object.
3. Configure the application for distributed logging.
4. Distribute the application.

All of these steps are described in this chapter.

Distributing Your Application

You can distribute your application manually or by using the NAD system. When you distribute your application, the historical provider list file is distributed as part of the application.

☞ For more information on using NAD, see Chapter 3 - Building a Distributed Application.

After you have distributed your application, you can run the View nodes and retrieve both local tagnames and tagnames from a remote history provider. While the application will run on all the View nodes, only the logging node will log to the historical log file; other nodes will only be able to read from it.

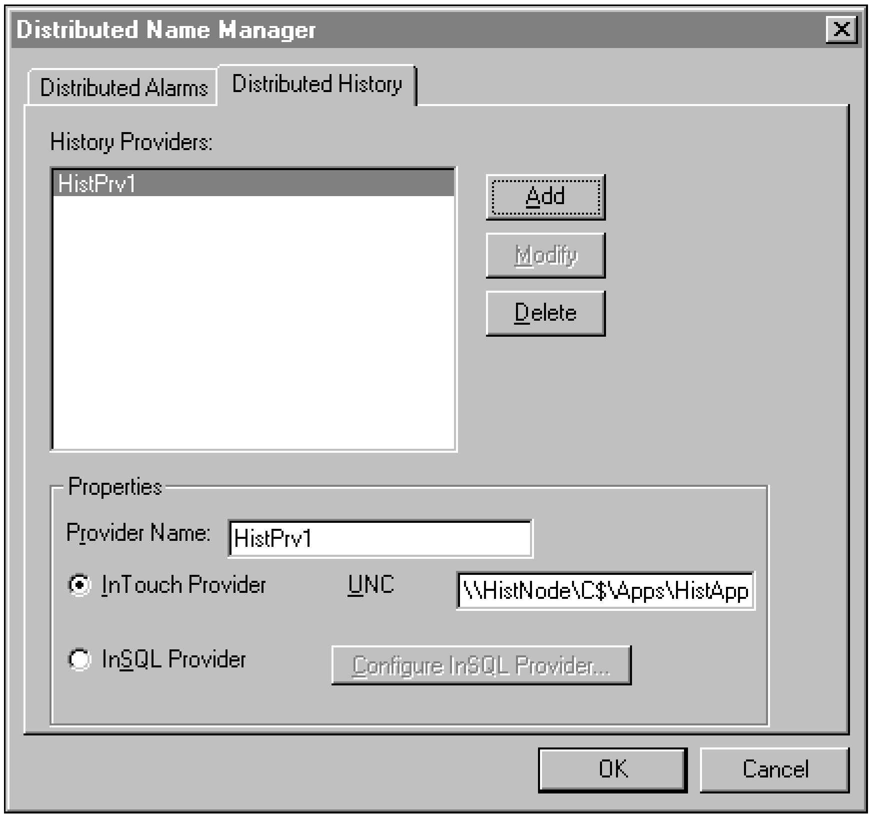
Configuring the Distributed History Provider List

Each remote history provider you intend to retrieve historical data from must be registered in the InTouch history provider list. This list allows you to specify a name and network location for each history provider. These names will be used whenever you refer to a history provider in InTouch.

➢ **To configure the historical provider list:**

1. On the **Special** menu, point to **Configure**, and then click **Name Manager**. The **Distributed Name Manager** dialog box appears:

   To quickly access the dialog box, in the Application Explorer, under **Configure**, double-click **Distributed Name Manager**.



History Providers: HistPrv1

Properties

Provider Name: HistPrv1

(●) InTouch Provider UNC \\HistNode\C$\Apps\HistApp

( ) InSQL Provider

   If you right-click a text box in any historical trend configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Click the **Distributed History** tab to activate the distributed history providers property sheet.
3. In the **Provider Name** box, type the name you want to use for the new historical provider.
4. If you want to access an InTouch application's log file, select **InTouch Provider** and type the UNC (Universal Naming Convention) path for the InTouch application directory in the **UNC** box. The valid format is:

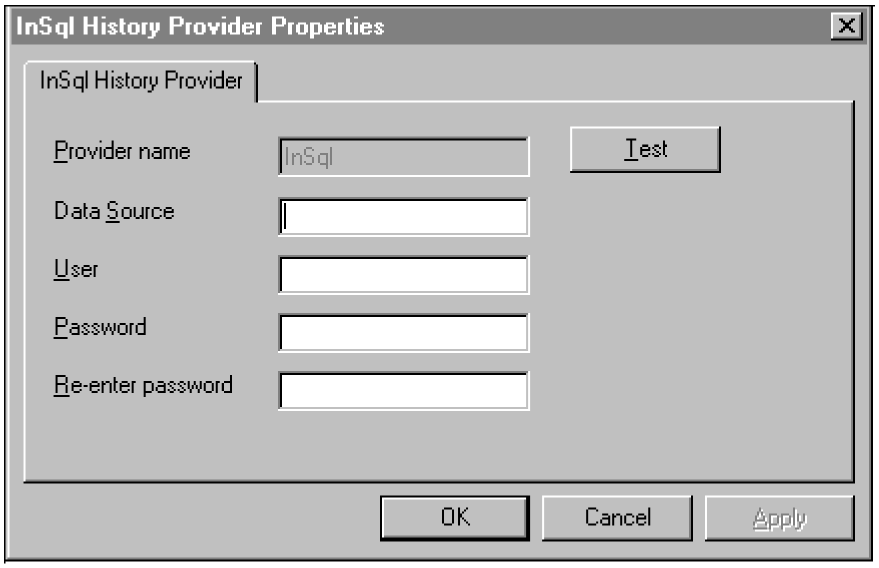
```
\\Node\Share\ApplicationPath
```

In the above example, the path to **HistPrv1** includes the node name "HistNode," the Share "C$," and the application path of "\Apps\HistApp."

If the UNC location is password-protected, you must first establish a connection using the Windows Explorer in Windows 95 or Windows NT.

For more information of UNC paths, see Chapter 3 - Building a Distributed Application.

5. Select **InSQL Provider** to access data in an IndustrialSQL Server runtime database, and then click **Configure InSQL Provider**. The **InSql History Provider Properties** dialog box appears:



By default, the logon parameters from the last successful logon will be displayed. If necessary, modify the logon parameters to connect to the selected IndustrialSQL Server.

5a. In the **Provider name** box, type the IndustrialSQL Server to which you want to connect.

5b. In the **Data Source** box, type the name of the database to which you want to connect.

5c. In the **User** box, type the name for your log on account.

5d. In the **Password** box, type the password for the log on account.

5e. In the **Re-enter password** box, type the password again to verify it.

Note A user account is comprised of the user name and password. A user account must be associated with the right to retrieve data, or else the log on will fail. For more information on user accounts, contact your system administrator.

5f. Click **Test** to validate the connection to the InSQL Server. A message box will appear informing you of the success or failure of the connection. Click **OK** to close the message box.

5g. Click **OK**.

Important Notes

1) When data is queried in the InSQL database for the InTouch trend object, 1000 evenly-spaced rows are retrieved for the given time period (a row count) and plotted on the historical trend. The minimum and maximum values shown for a tagname in the historical trend may not be the actual minimum and maximum values for the tagname.

2) Do not use the InTouch HistData utility if IndustrialSQL is selected as the InTouch history provider.

For more information on history providers, see "Configuring the Distributed History Provider List."

6. Click **Add**.

When WindowViewer is presented with a history provider name, the history system will look up that name in the provider list. If the name exists in the list, it reads the history log file from that provider. If the name does not exist, the reference is ignored and an error message is written to the Wonderware Logger. While the local InTouch application is considered a history provider, it does not need to be configured in this file.

For more information on the Wonderware Logger, see your *FactorySuite Administrator's Guide*.

Configuring Remote History Providers

The historical trend supports the display of tagnames from both local and remote history providers.

➤ **To display a tagname from a remote history provider:**

1. Double-click the historical trend to access the **Historical Trend Configuration** dialog box.
2. In each pen's **Tagname** box, type the tagname in the format:

   HistPrv1.tagname

   Each pen can reference a different remote history provider. For example, when you configure the historical trend, if you want Pen1 to plot the tagname **Boiler1** in the remote history provider defined as **HistPrv1**, you would type **HistPrv1.Boiler1** in Pen1's **Tagname** box.

Note The **.TagID** tagname .field cannot be used in remote history provider tagnames references.

Using the Tag Browser to Access Remote History Providers

The following procedure demonstrates how the application developer can use the Tag Browser to select a remote tagname reference.

➢ **To define a remote history provider as a tag source:**

1. Create an Access Name that specifies the **Node Name** where the history provider resides.

   The **Node Name** you specify in the Access Name does not have to be the actual name of the node where the tagname resides. But, you must create an Access Name or you won't be able to define the remote history provider as a tag source.

2. Double-click the historical trend. The **Historical Trend Configuration** dialog box will appear.
3. Double-click a pen's **Tagname** input box. The Tag Browser will appear.
4. Click the Define Tag Sources button ... to define the remote history provider as a tag source.
5. Click the **Tag Source** arrow and select the new remote history provider tag source in the list, or click the Tree View button and select the tag source in the tree view pane. The Tag Browser will then be repopulated with the selected remote history provider's tagnames.
6. Double-click the tagname that you want to assign to the historical pen, or, select it, and then click **OK**.
7. The **Historical Trend Configuration** dialog box reappears with the selected tagname displayed in the pen's **Tagname** box in the format: **AccessName:Item**.
8. Replace the **AccessName:** portion with the history provider name that you defined in the Distributed Name Manager. For example, **HistPrv1.Tagname**.

   This process may seem cumbersome, but once you have defined the history provider as a tag source in the Tag Browser, each time you double-click another tagname input box, you simply double-click the tagname in the Tag Browser, and then replace the **AccessName:** portion with the history provider name. By using this process you will reduce your chance of error when you specify a remote history provider tagname.

Note In WindowViewer, if runtime changes are allowed for the historical trend, when the user clicks a pen button to change the tagname, the Tag Browser will appear but, only the local application's tagnames will be accessible.

For more information on using the Tag Browser, see Chapter 4 - Tagname Dictionary.

Dynamically Configuring Remote History Providers

In runtime, you can also dynamically configure a historical trend's remote history provider by creating a QuickScript that specifies the remote history provider tagname reference in the **HTSetPenName** function. For example:

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

Where, 1 specifies the pen that will plot the specified remote history provider tagname.

Note The runtime **Historical Trend Setup** dialog box and **.Pen** are not supported for remote history providers.

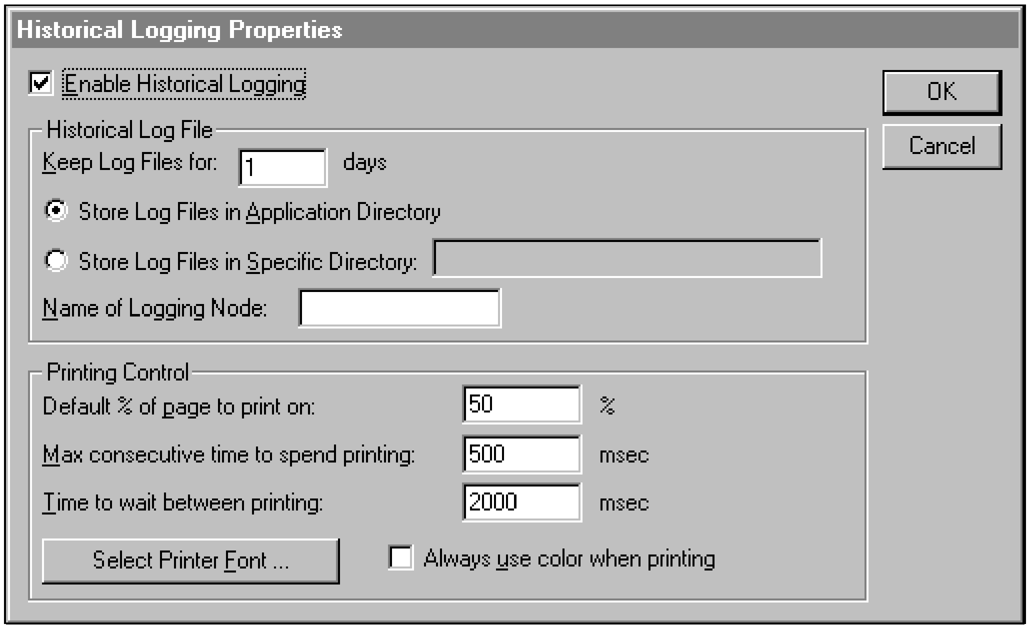
For more information on the InTouch QuickScript functions, see the online *InTouch Reference Manual*.

Configuring Distributed Historical Logging

➢ **To configure distributed historical logging:**

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

   To quickly access the dialog box, in the Application Explorer under **Configure**, double-click **Historical Logging**.



[✓] Enable Historical Logging

Keep Log Files for: 1 days

[● Store Log Files in Application Directory]

○ Store Log Files in Specific Directory:

Name of Logging Node:

Default % of page to print on: 50 %

Max consecutive time to spend printing: 500 msec

Time to wait between printing: 2000 msec

☐ Always use color when printing

   If you right-click a text box in any historical trend configuration dialog box, a menu will appear displaying the commands that you can apply to the selected text.

2. Select **Enable Historical Logging** to turn on global tagname logging.
3. Select **Store Log Files in Specific Directory**, and then in the input box, type the path of the location where the log files will be stored. You must type a valid Universal Naming Convention (UNC) path. For example, "\\Node\Share\Path."

   If NAD is being used, make sure that the path points to a directory other than the application directory.

4. In the **Name of Logging Node** box, type the name of the node that will be logging to the history log file.

   This setting will only allow the node named here to log to the file.

5. Click **OK**.

Note When an application with the **Enable Historical Logging** option selected is distributed to a WindowViewer node, that node checks this option to determine if it should log or not. If **Enable Historical Logging** is selected, the possible settings are:

Field equals name of Node - Logging enabled

Field does not equal name of Node - Logging disabled

Creating Historical Trend Scooters

Scooters are positional indicators along a time scale that can be changed in order to retrieve specific pieces of data for precise points of time. By tying slider objects to a scooter **.field**, you can slide over a historical trend display and access a desired section of data. QuickScript functions are provided to access the average, minimum, and maximum values at a specified scooter position. A left and right scooter can be created and using additional InTouch QuickScript functions, can return values based on an analysis performed on the data between the scooters or at the scooter location. Analysis types include Average, Min, Max, Min/Max Value, Min/Max EU and Standard Deviation. Zooming in can also be performed between the two scooters.

For more information on the scooter **.fields**, see "Historical Trend .Fields."

You can also add the ability to display data based on a known location on the chart. Zooming in and out of a trend is also beneficial. The following pages describe the links and expressions that can be used to incorporate this functionality into historical trends.

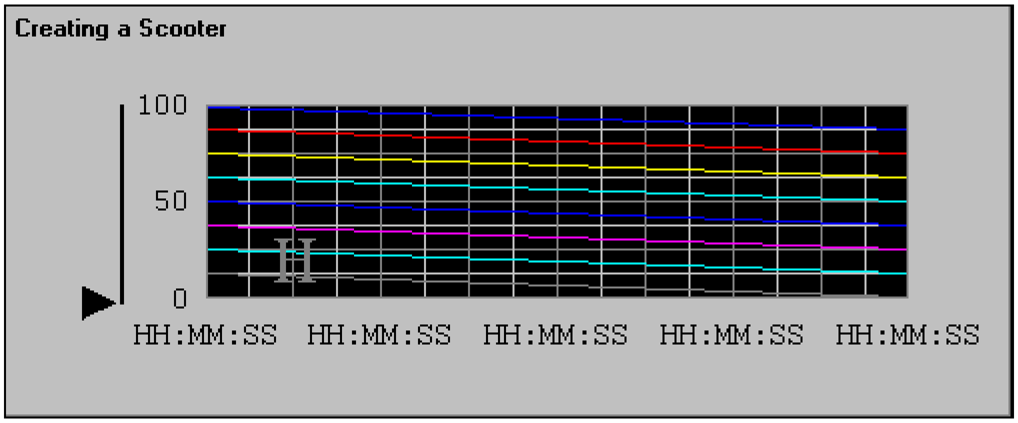
These features are already configured in use the Historical Trend Wizards.

For more information, see "Using Historical Trend Wizards."

You can use the **Horizontal Slider** link to create a scooter that moves over the top of your historical trend. (Each trend supports two scooters, left and right.)

➢ **To create a scooter:**

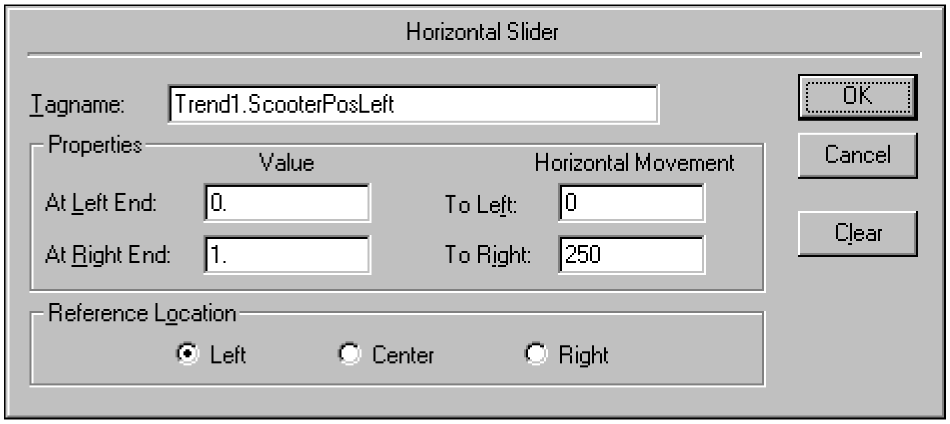
1. Create the object to be used as your scooter. In the example below, we will use a polygon and vertical line symbol:



2. To properly define the scooter's horizontal slider link, you will need to know how wide your chart is. To determine the chart's width, draw a horizontal line from one end of it to the other and note the values displayed in WindowMaker's status bar. Look at the third value in from the left -- this is the width of your chart. Write this number down. Delete the "measuring" line.

3. Double-click on the scooter object. The animation link selection dialog box will appear.

4. In the **Touch Links - Sliders** section, click **Horizontal**. The **Horizontal Slider** dialog box appears:


Horizontal Slider

Tagname: Trend1.ScooterPosLeft

Properties

	Value		Horizontal Movement
At Left End:	0.	To Left:	0
At Right End:	1.	To Right:	250

Reference Location: [✓ Left] Center Right

5. In the **Tagname** box, type the name of the historical trend plus the **.field**, **.ScooterPosLeft**.

   For example, if the trend has the tagname **Trend1**, then the tagname is **Trend1.ScooterPosLeft**. The **At Left End** value is 0.0 and the **At Right End v**alue is 1.0. The **Horizontal Movement To Left** is 0 and **To Right** is the pixel value you found when the horizontal line was drawn from one end of the trend to the other. (In the example above, the chart was 250 pixels wide.)

6. Click **OK**.

Position the left scooter at the very left edge of your trend. The left scooter is now complete. Duplicate the process for the right scooter using **.ScooterPosRight**. The value fields remain the same.

Displaying Values at Scooter Positions

To display values based on the current location of the scooter, create a numeric text object, for example, #.00 and assign it to a **Value Display - Analog** link with an expression. For example:

```
HTGetValueAtScooter( "Trend1", Trend1.UpdateCount, 1,
Trend1.ScooterPosLeft, 1, "PenValue" )
```

This example will retrieve the value for Pen1 at scooter position left on Trend1. If the slider is moved in Runtime, the **Value Display - Analog** link will automatically be updated with the new value at the new scooter location.

Retrieving Values Between Zones

To receive data between the scooters current location in the form of either, max value, min value, average value or standard deviation, create a numeric text object, for example, #.00, and assign an **Value Display - Analog** link with an expression. For example:

```
HTGetValueAtZone( "Trend1", Trend1.UpdateCount,
Trend1.ScooterPosLeft, Trend1.ScooterPosRight, 1, "PenMaxValue" )
```

This example will retrieve the Max value for Pen1 on Trend1 between scooter position left and scooter position right.

Zooming In and Out

To zoom in on the trend, create a button and assign it to a **Touch Pushbutton - Action** link. Choose the **On Button Down** condition and enter the following QuickScript:

```
HTZoomIn( "Trend1", "Center" );
```

This example will "anchor" the center time of the trend and the new chart width will equal the old chart width divided by two if the scooters are at the far left and right positions. If the scooters are moved, the new chart width is the time between scooter left and scooter right and the ***LockString*** ("Center") is not used.

For more information using historical QuickScript functions, see "Historical QuickScript Functions."

Historical Trending and Daylight Savings Time

The InTouch history system can automatically account for daylight savings time. To use this feature, you must have your computer properly configured to note the difference between UCT (Universal Coordinated Time), also known as Greenwich Mean Time or GMT, and your local time zone. The configuration instructions are listed below:

➢ **To set your time zone when using the Windows 3.x, Windows for Workgroups, or the Windows 95 operating system:**

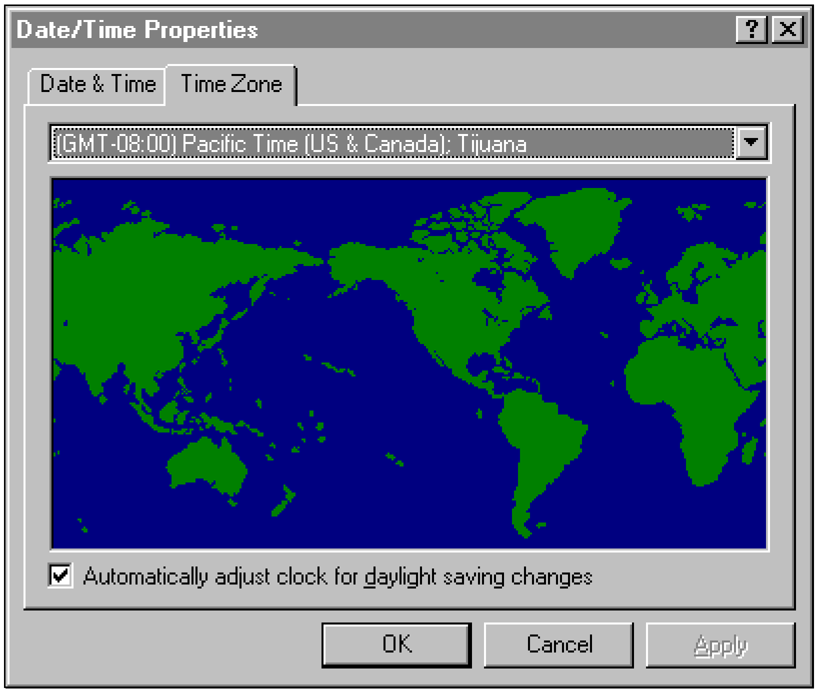
The setting, **set TZ=GMT[+ | -]*X***, must be included in the AUTOEXEC.BAT of the computer.

Where: *X* is the offset from GMT for the time zone the computer resides in.)

For example, to set the **TZ** environment variable to correspond to the current time zone in California, you could use either **set TZ=GMT8** or **set TZ=GMT+8**.

➢ **To set your time zone on the Windows NT operating system:**

1. Open the Windows Control Panel.
2. Double-click the **Date/Time** icon or, double-click the clock in the Windows **Taskbar**. The **Date/Time Properties** dialog box appears:



(GMT-08:00) Pacific Time (US & Canada); Tijuana

☑ Automatically adjust clock for daylight saving changes

3. Click the **Time Zone** tab, and then click the arrow to open the list of time zones.
4. Select your time zone in the list.
5. Click **OK**.

Note The Windows 95 and Windows NT operating systems may be set to automatically adjust the clock for daylight savings time. It is recommended that this feature be disabled. To disable this feature, use the Date/Time utility in the Windows Control Panel or, double-click the clock on the Windows **Taskbar**.

For more information on system time, see Chapter 3 - Building a Distributed Application.

Automatically Changing the System Time

Windows 95 and Windows NT will attempt to automatically adjust the clock for daylight savings time. You should turn this feature off by using the Date/Time utility in the Windows Control Panel and use InTouch QuickScripts to automatically set the clock at these times.

➢ **To set the clock forward in the spring:**

Create the following Condition QuickScript:

```
$Year == yyyy and $Month == 04 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 0 ;
```

where: yyyy = the year (i.e., 1993, 1994, or 1995 ...)
dd = the date of the time change
DaylightSavingsTime = a user-defined memory discrete tagname to indicate daylight savings time

ON TRUE:

```
DaylightSavingsTime = 1;
StartApp "c:\windows\control.exe" ;
SendKeys "%(st)" ;
SendKeys "%(t)" ;
SendKeys "03" ;
SendKeys "~" ;
SendKeys "%({F4})" ;
```

➢ **To set the clock back in the fall:**

Create the following Condition QuickScript:

```
$Year == yyyy and $Month == 10 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 1 ;
```

where: yyyy = the year (i.e., 1993, 1994, or 1995 ...)
dd = the day of the time change for that year
DaylightSavingsTime = a user-defined memory discrete tagname to indicate daylight savings time

Note Whenever a systems clock is set back, the historical logging engine could overwrite existing data in the historical log file. To prevent this loss of data, we recommend that you first back up the log files before changing the clock back.

ON TRUE:

```
DaylightSavingsTime = 0;
StartApp "c:\windows\control.exe" ;
SendKeys "%(st)" ;
SendKeys "%(t)" ;
SendKeys "01" ;
SendKeys "~" ;
SendKeys "%({F4})" ;
```

Historical Data Merge Utility Program

The Historical Data Merge utility program (HDMERGE.EXE) provides a mechanism for merging a .CSV (comma separated variable) data file into an existing InTouch application's historical log file. HDMerge determines the historical log file that the data will be merged into by the date specified for the data values in the data file. (Historical log files are saved by date stamp. For example, September 30, 1995 would be saved as **95093000.LGH**). HDMerge can also be used to create historical log files. If a historical log file does not exist in the target application for the date specified in the .CSV file, HDMerge will create one for that date. If the data file contains multiple days of data, each historical log file will be updated with the data for its day. The time specification for each data sample can have up to 1 millisecond resolution.

For all records to be retrieved after merge, you must add one dummy record to the end of the file. This record must be different from the last "real" record in the file. This dummy record will not display in the history trend.

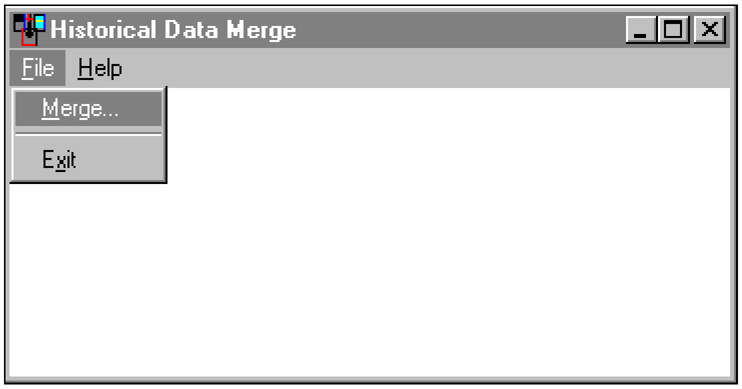
Note Keep in mind that the HDMerge program was designed to work with "historical" data (data created in the past).

Legacy InTouch history systems used an extension of .LOG for log files. This version of HDMerge supports the newer log files with an extension of .LGH. To merge .CSV files into .LOG files, you must use an older version of HDMerge.

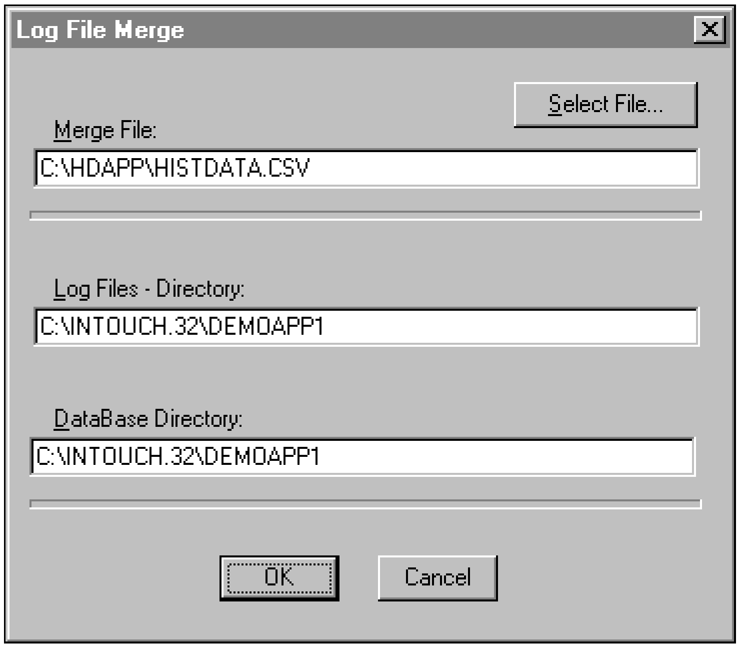
Running the HDMerge Program

➢ **To run the HDMerge program:**

1. On the Windows Taskbar, click **Start**. Point to **Programs**, and then click **HDMerge**. The **Historical Data Merge** program appears:



2. On the **File** menu, click **Merge**. The **Log File Merge** dialog box appears:



Merge File: C:\HDAPP\HISTDATA.CSV

Log Files - Directory: C:\INTOUCH.32\DEMOAPP1

DataBase Directory: C:\INTOUCH.32\DEMOAPP1

3. In the **Merge File** box, type the name of the .CSV file that you want to merge into the historical log file, or click **Select File** to locate and select the file.

   If you click **Select File**, when the dialog box reappears, the full path to the .CSV file you selected will automatically be displayed in the **Merge File** field.

4. In the **Log Files - Directory** box, type the full path to the directory containing the historical log file into which you want to merge the .CSV file.

5. In the **DataBase Directory** box, type the full path to the directory containing the InTouch application's Tagname Dictionary files (tagnames.x and tagname.ndx) associated with the target historical log file.

6. Click **OK** to execute the merge operation.

7. When the merge operation is complete, a message box will appear displaying the statistics regarding your merge operation.
8. Click **OK**.

Note If you run HDMerge from the command line, the message box will not appear.

Creating a HDMerge Data File

The data file must be created and saved as a .CSV file (comma separated variable format). You can create the data file by using a text editor or any other program that supports the .CSV file format, for example, Windows Notepad or Microsoft Excel. Once you create the data file, you can merge it by using the HDMerge utility program.

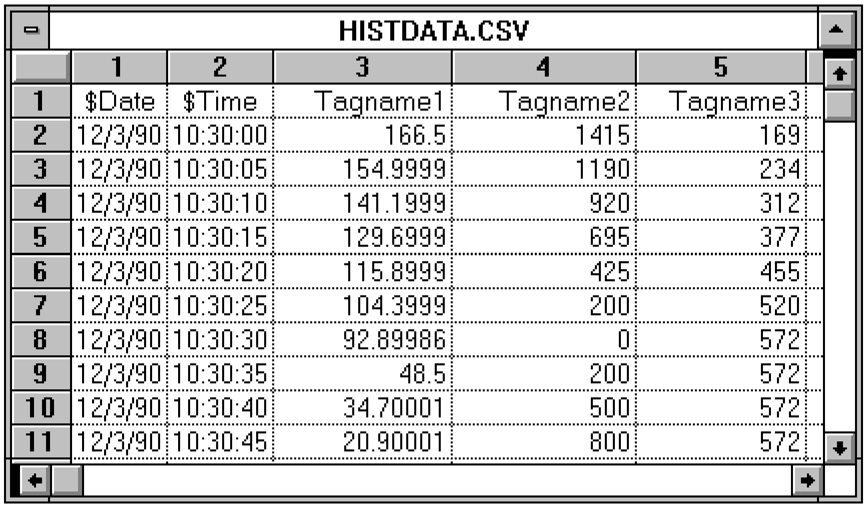
When the HistData utility program is used, creation of a properly formatted .CSV file becomes simple. By using HistData the user can specify the data sample to be extracted from the InTouch historical log file and automatically write that data sample to a .CSV file. HDMerge can then merge the data in that .CSV file into any other InTouch application's historical log file.

For more information on the HistData utility program, see the "HistData Utility Program."

Example .CSV Files

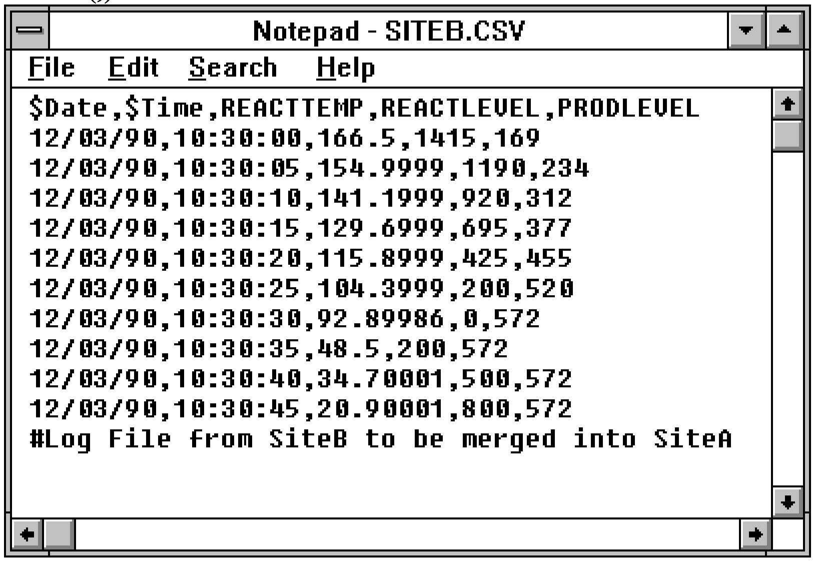
The following examples illustrate properly formatted .CSV files in both Excel and Notepad.

When a .CSV file is opened in Microsoft Excel, all data values are automatically separated into individual columns.


HISTDATA.CSV

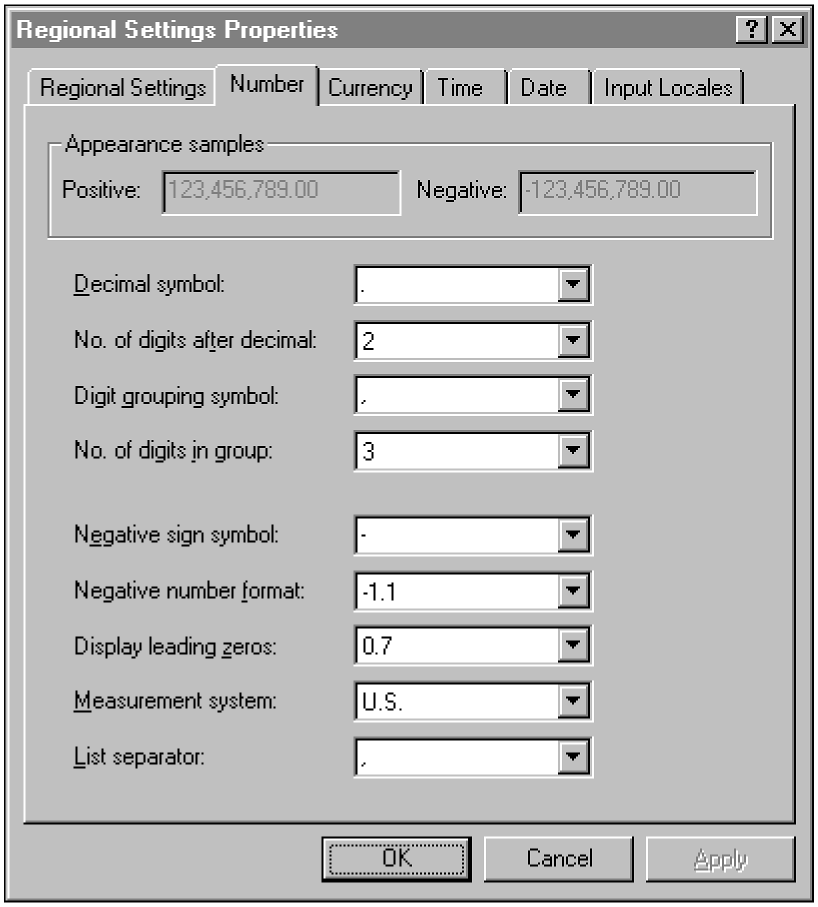
	1	2	3	4	5
1	$Date	$Time	Tagname1	Tagname2	Tagname3
2	12/3/90	10:30:00	166.5	1415	169
3	12/3/90	10:30:05	154.9999	1190	234
4	12/3/90	10:30:10	141.1999	920	312
5	12/3/90	10:30:15	129.6999	695	377
6	12/3/90	10:30:20	115.8999	425	455
7	12/3/90	10:30:25	104.3999	200	520
8	12/3/90	10:30:30	92.89986	0	572
9	12/3/90	10:30:35	48.5	200	572
10	12/3/90	10:30:40	34.70001	500	572
11	12/3/90	10:30:45	20.90001	800	572

When a .CSV file is created or opened in Notepad, each data value is separated by a comma (,).



```
$Date,$Time,REACTTEMP,REACTLEVEL,PRODLEVEL
12/03/90,10:30:00,166.5,1415,169
12/03/90,10:30:05,154.9999,1190,234
12/03/90,10:30:10,141.1999,920,312
12/03/90,10:30:15,129.6999,695,377
12/03/90,10:30:20,115.8999,425,455
12/03/90,10:30:25,104.3999,200,520
12/03/90,10:30:30,92.89986,0,572
12/03/90,10:30:35,48.5,200,572
12/03/90,10:30:40,34.70001,500,572
12/03/90,10:30:45,20.90001,800,572
#Log File from SiteB to be merged into SiteA
```

Note If you are an international user, the delimiter character may be different. You can specify the character to be used for the delimiter by selecting **Regional Settings** in the Windows Control Panel, and then clicking the **Number** tab:



Positive: 123,456,789.00 Negative: -123,456,789.00

Decimal symbol:	.
No. of digits after decimal:	2
Digit grouping symbol:	,
No. of digits in group:	3
Negative sign symbol:	-
Negative number format:	-1.1
Display leading zeros:	0.7
Measurement system:	U.S.
List separator:	,

In the **List separator** box, type the delimiter character that you need to use.

You can also specify the delimiter in the command line in your .CSV file.

For more information on the delimiter, see "Merge Data File Commands."

There are a few things you need to remember when creating a .CSV data file:

- The data must be in chronological order.
- The row in the .CSV file preceding the actual data to be merged must begin with **$Date** and **$Time** followed by the tagnames. For example:

  $Date,$Time,Tagname1,Tagname2,Tagname3,....,

- If using Excel, **$Date** must be in the first column and **$Time** in the second column followed by each tagname in subsequent columns, all in the same row.
- Each row of data must start with the date and the time. For example:

  12/03/96, 10:30:05, **...**

- The time specification format is "HH:MM:SS.MMM" . The level of resolution is optional (for example, "HH", "HH:MM", "HH:MM:SS", or "HH:MM:SS.MMM"). If milliseconds is specified, three digits must be used. For example:

  ..., 10:30:05.100, **...**

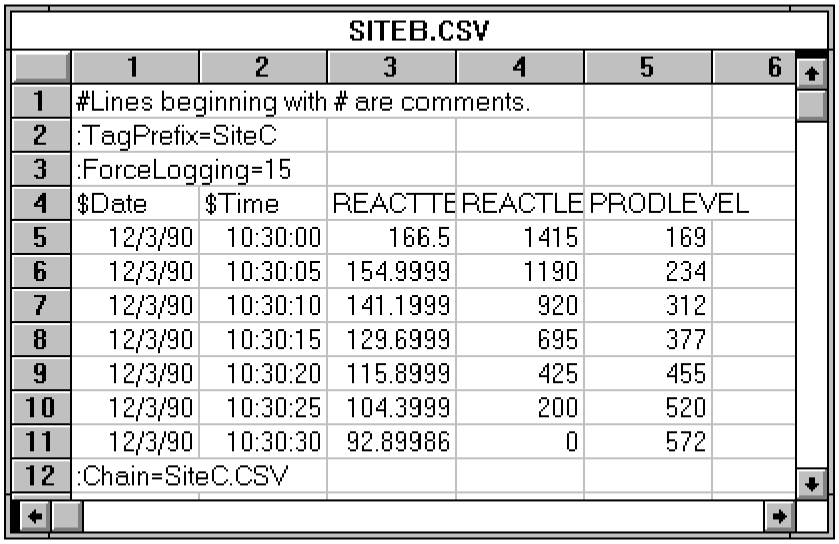
  ..., 10:30:05.200, **...**

- All the tagnames in the "tagname line" must be defined in the target application's database.
- Begin "comments" (lines that are not processed) with a "**#**" character.
- Begin all "command" lines with a colon (**:**). (For more information on using commands, see the following section, "Merge Data File Commands".)

Note Use caution when merging files because the data contained in the .CSV file will overwrite any data for tagnames currently existing in the log files from the mergers start time through the end of the affected day's log file. Therefore, it is recommended that you make a back-up copy of the target log file before you perform the merge function.

Merge Data File Commands

There are several commands that can be used to automate various actions when merging a data file. In most cases, these commands are inserted into the .CSV file before the tagname line. Each command line must begin with a colon (**:**). For example:



SITEB.CSV

	1	2	3	4	5	6
1	#Lines beginning with # are comments.					
2	:TagPrefix=SiteC					
3	:ForceLogging=15					
4	$Date	$Time	REACTTE	REACTLE	PRODLEVEL	
5	12/3/90	10:30:00	166.5	1415	169	
6	12/3/90	10:30:05	154.9999	1190	234	
7	12/3/90	10:30:10	141.1999	920	312	
8	12/3/90	10:30:15	129.6999	695	377	
9	12/3/90	10:30:20	115.8999	425	455	
10	12/3/90	10:30:25	104.3999	200	520	
11	12/3/90	10:30:30	92.89986	0	572	
12	:Chain=SiteC.CSV					

Note The "#" character is used at the beginning of a line to designate it as a "comment".

The following describes each log input file command and its valid format:

TagPrefix

This command is used on any line preceding the tagname line of the .CSV data file to automatically add a prefix to all the tagnames in the .CSV file when it is merged. Valid format:

:TagPrefix=*name*

Let's assume that we have applications at Site A and Site B that are logging the same tagnames. We now want to merge the data values for the tagnames in Site B's log file into Site A's log file without overwriting Site A's data values. To do so, we can use the HistData utility program to extract the data from Site B's log file and create a SITEB.CSV file. We can now change the tagnames in the file by using the prefix command as the first line. For example:

:TagPrefix=SiteB

By using this command line, all the tagnames in the SITEB.CSV file will be prefixed with "**SiteB**" when they are merged into Site A's log file.

Note You must define the prefixed tagnames in the target application's database (Site A in this example) prior to performing the merge.

TagSuffix

This command is used on any line preceding the tagname line of the .CSV data file to automatically add a suffix to all the tagnames in the file when it is loaded/merged. (See previous **TagPrefix** deQuickScription.) Valid format:

:TagSuffix=*name*

Note You must define the suffixed tagnames in the target application's database prior to performing the merge.

Chain

This command is used as the last line of the .CSV log input file to cause multiple .CSV files to be consecutively merged into the target application's log file. Valid format:

:Chain=*filename*.*CSV*

Let's assume we have three .CSV log input files (SITEB.CSV, SITEC.CSV and SITED.CSV**)** that we want to consecutively load into Site A's log file with one merge execution. We want to merge SITEB..CSV first, SITED.CSV second and SITEC.CSV last. To do so, we would add the following as the last line in the SITEB..CSV file:

:Chain= SITED.CSV

Then we would add the following as the last line in the SITED.CSV file:

:Chain= SITEC.CSV

By adding these command lines, HDMerge will automatically merge the SITEB..CSV file, then the SITED.CSV file and lastly, the SITEC.CSV file.

Output

This command is used on any line preceding the tagname line of the .CSV data file to specify the format of the output log files for either Windows or the Windows NT operating system. The command can be set to 32 (to specify 32-bit format), 16 (to specify 16-bit format),or 0 (to specify 16-bit format for Windows or 32-bit format for the Windows NT operating system). File formats can be converted in addition to merging data. If the output command is not used, the merge process will use the file format corresponding to the operating system being used (16-bit for Windows and 32-bit for the Windows NT operating system) for all new log files.

If the output command is not used and there are existing log files, the merge process will use the file format of the preceding files (regardless of the operating system being used).

Valid format:

:Output=#

where:
# represents the format the output log files should be: 16, 32, or 0.

CSVcharacter

This command is used on any line preceding the tagname line of the .CSV data file to specify the character being used as the delimiter (replaces comma) in the data file. Valid format:

:CSVcharacter=.

DecimalCharacter

This command is used as the **first line** of the .CSV log input file to specify the character being used for a decimal point in the file. Valid format:

:DecimalCharacter=,

Using the HDMerge Command Line

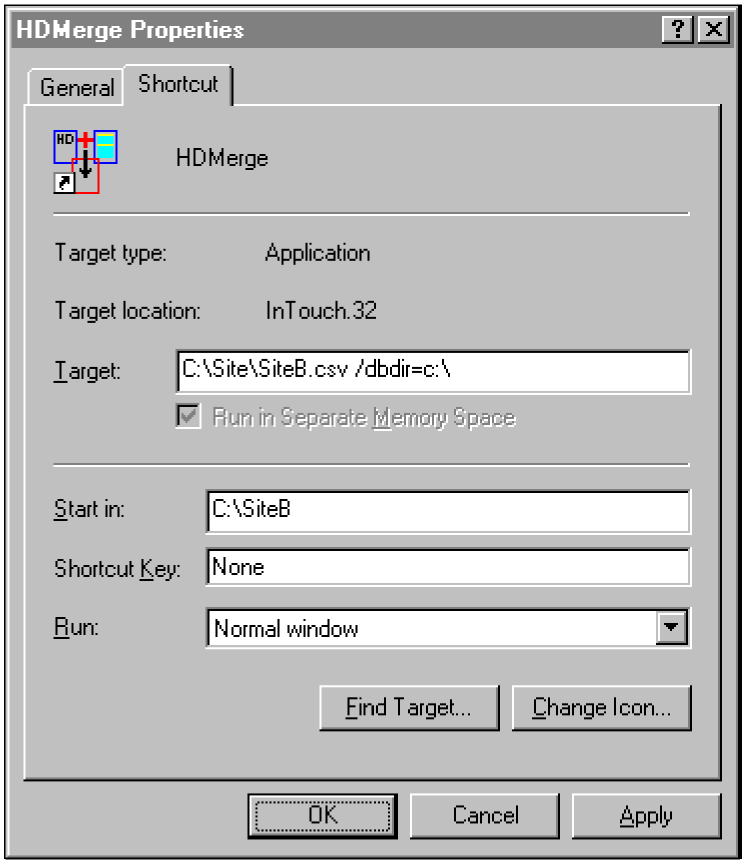
Once you have created a valid .CSV file, the HDMerge program can be used to load/merge the data contained in that file into the target InTouch application's historical log file. Interaction with the merge application can be accomplished either by using the HDMerge program's menu or by using the following fully automatic command line:

HDMerge.exe *filename.CSV* [/dbdir="*path*"] [/logdir="*path*"][/journal=0];

Where:

filename.CSV	Is the name of the .CSV data file to be merged (include the full "path").
[/dbdir="*path*"]	Specifies the full path to the directory where the target InTouch application's database resides.
[/logdir="*path*"]	Specifies the full path to the directory where the target InTouch application historical log file resides.
[/journal=0]	Setting this parameter equal to 0 will stop the journal report from being generated. By default, journal reporting is turned on.

There are a number of ways that the command line can be used to automate the merge operation. For example, you can link it to an action pushbutton or use it in a QuickScript in InTouch. HDMerge short cut icons can be created that have different command lines, for example:



HDMerge

Target type: Application

Target location: InTouch.32

Target: C:\Site\SiteB.csv /dbdir=c:\

Run in Separate Memory Space

Start in: C:\SiteB

Shortcut Key: None

Run: Normal window

For example, let's assume that we are using the following command line for a HDMerge icon:

HDMerge.EXE C:\SiteB\SiteB.CSV /dbdir=c:\SiteA /logdir=c:\SiteA

When we run HDMerge from this icon, the SITEB.CSV data file (in the SITEB directory) will be merged into the SITEA InTouch application's historical log file.

HDMerge Journal Reports

By default, HDMerge will produce a basic "journal" report each time a merge function is executed. The basic journal report will contain information relating to the merge activity and any errors that may have been encountered during its execution.

The journal report is automatically saved in the directory where the merged .CSV file is located. The journal report will be saved with the same filename as the .CSV file but, with the extension .JNL.

The WIN.INI File

Parameters can be set in the **[HDMERGE]** section of your WIN.INI file to control journal reporting. To produce a detailed (verbose) journal report, the following parameter can be set:

[HDMERGE]
JournalVerbose=1

To disable journal reporting completely, the following parameter can be set:

NoJournal=1

HistData Utility Program

The HistData utility program provides DDE (Dynamic Data Exchange) 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.

Note The HistData program cannot be used with remote tagname references, for example, DBS.TAG.

The HistData Database

The HistData program contains its own internal database. The items in the internal database are used to specify start period, duration and sampling interval, and so on, 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, for example, C:\InTouch\App.
DBDIR	Message	Pathname of the directory containing the InTouch Tagname Dictionary, for example, C:\InTouch\App.
STARTDATE	Message	Data sample start date in the format MM/DD/YY
STARTTIME	Message	Data sample start time in the format HH:MM:SS using the 24-hour clock.

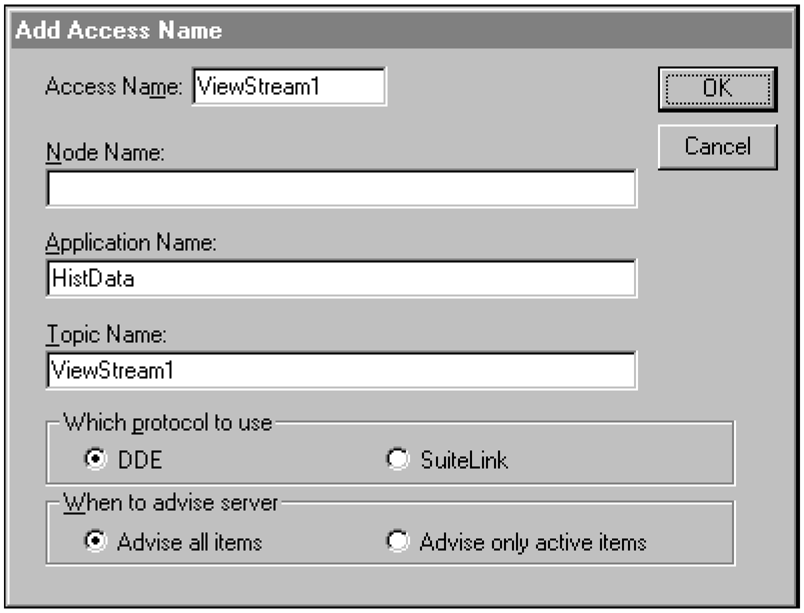
Item	Type	Description
DURATION	Message	The length of time for which data is to be returned. **DURATION** can be expressed in weeks, days, hours, minutes and seconds. The following are the valid characters: **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). Fractional values are also permitted, for example, .5s for 500 milliseconds. To request a single sample, set **DURATION** to 0 (zero).
INTERVAL	Message	The length of time in between samples. **INTERVAL** can be expressed in weeks, days, hours, minutes and seconds, for example, 1w represents 1 week. Fractional values are also allowed, for example, .25d represents 6 hours. (The valid characters are the same as those for **DURATION**.)
		Note The maximum length of time allowed for **DURATION** and **INTERVAL** is 6 weeks. This applies to all request types, days, seconds, etc. For example, if using days, 42 is the maximum (7 days x 6 weeks = 42).
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: "$Date,TagA,TagB" or, "$Time,TagA,TagB" or, "$Date,$Time,TagA,TagB"
TAGS1, TAGS2,....	Message	The **TAGS** string can be 131 characters in WindowViewer and 255 characters in Excel. The string can be appended for longer requests by adding tagname items named "Tags1," "Tags2" and so on. 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"
		Note Duplicate tagnames are not allowed and the maximum length of each tags string is 512 bytes.

Item	Type	Description
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	Message	This item is used to hold the requested data in the HistData program in comma separated variable format. It is used by other applications which want 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, **SENDDATA** is automatically reset to 0 (zero). **Note** To you receive an error message telling you that you have requested too much data when you use **SENDDATA**, shorten the **DURATION** or reduce the number of tagnames that you are requesting.
FILENAME	Message	Complete pathname of the file to write the requested data, for example, C:\INTOUCH\ HDFILE.CSV.
WRITEFILE	Integer	When set to 1, HistData will write the requested data to the 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. A 1 indicates success and a 0 (zero) indicates an error occurred.
ERROR	Message	A string containing a description of the last error. It will be "None" when **STATUS** is 1, and will contain an error message string when **STATUS** is 0 (zero).

Using HistData with InTouch

In this section, we have created a sample window to show you one method that you can use to request and display historical data in InTouch.

In order for InTouch to request data from the HistData program, the following Access Name was defined:



Access Name: ViewStream1

Node Name:

Application Name: HistData

Topic Name: ViewStream1

Which protocol to use: [✓ DDE] SuiteLink

When to advise server: [✓ Advise all items] Advise only active items

Note The **Access Name** can be any arbitrary name up to 32 characters. The **Topic Name** can also be any arbitrary name up to 32 characters. It is recommended that the same name be used for both of these items. The **Application Name** must be the program name, **HistData** (less the .EXE).

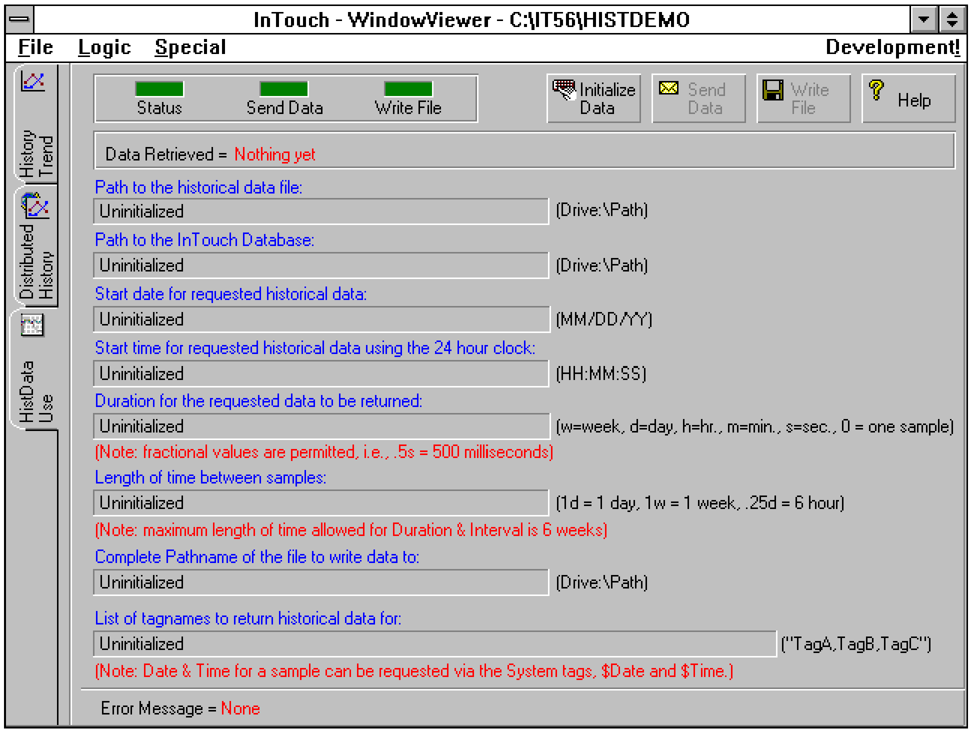
It is also recommended that the **Advise all items** option be enabled whenever HistData is being used.

For more information on Access Names, see Chapter 9 - I/O Communications.

After the Access Name was defined, the following I/O type tagnames were created for each HistData internal database item:

DATA	Message	**SENDDATA**	Integer
DATADIR	Message	**STARTDATE**	Message
DBDIR	Message	**STARTTIME**	Message
DURATION	Message	**STATUS**	Discrete
ERROR	Message	**TAGS**	Message
FILENAME	Message	**WRITEFILE**	Integer
INTERVAL	Message	**PRINTTAGNAMES**	Discrete

In order to specify the historical data to be accessed, input fields were created and linked to the various I/O Message tagnames in the window. All of the input fields were assigned the initial value of **Uninitialized** as shown:



Data Retrieved = Nothing yet

Path to the historical data file:
Uninitialized (Drive:\Path)

Path to the InTouch Database:
Uninitialized (Drive:\Path)

Start date for requested historical data:
Uninitialized (MM/DD/YY)

Start time for requested historical data using the 24 hour clock:
Uninitialized (HH:MM:SS)

Duration for the requested data to be returned:
Uninitialized (w=week, d=day, h=hr., m=min., s=sec., 0 = one sample)
(Note: fractional values are permitted, i.e., .5s = 500 milliseconds)

Length of time between samples:
Uninitialized (1d = 1 day, 1w = 1 week, .25d = 6 hour)
(Note: maximum length of time allowed for Duration & Interval is 6 weeks)

Complete Pathname of the file to write data to:
Uninitialized (Drive:\Path)

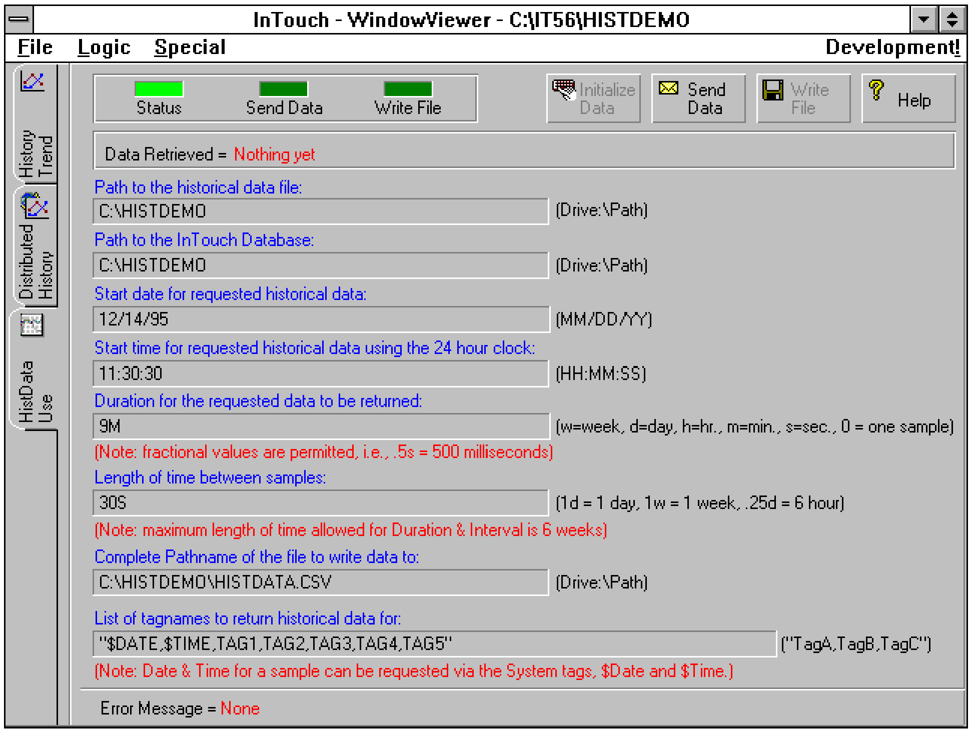
List of tagnames to return historical data for:
Uninitialized ("TagA,TagB,TagC")
(Note: Date & Time for a sample can be requested via the System tags, $Date and $Time.)

Error Message = None

An **Initialize Data** button was created and linked to the following **Touch Pushbutton Action** QuickScript:

```
HDDataDir = InfoInTouchAppDir();
HDDbDir = InfoInTouchAppDir();
HDStartDate = "12/14/95";
HDStartTime = "11:30:30";
HDDuration = "9M";
HDInterval = "30S";
HDTags = "$Date,$Time,Tag1,Tag2,Tag3,Tag4,Tag5";
HDFileName = InfoInTouchAppDir() + "HISTDATA.csv";
```

When the application is first started, press the Help button for more information. Once completed, click the **Initialize Data** button and the historical data (specified in the action QuickScript) will be automatically entered into the corresponding input fields:


InTouch - WindowViewer - C:\IT56\HISTDEMO
File Logic Special Development!
History Trend
Distributed History
HistData Use
Status Send Data Write File
Initialize Data
Send Data
Write File
Help

Data Retrieved = Nothing yet

Path to the historical data file:
C:\HISTDEMO (Drive:\Path)

Path to the InTouch Database:
C:\HISTDEMO (Drive:\Path)

Start date for requested historical data:
12/14/95 (MM/DD/YY)

Start time for requested historical data using the 24 hour clock:
11:30:30 (HH:MM:SS)

Duration for the requested data to be returned:
9M (w=week, d=day, h=hr., m=min., s=sec., 0 = one sample)
(Note: fractional values are permitted, i.e., .5s = 500 milliseconds)

Length of time between samples:
30S (1d = 1 day, 1w = 1 week, .25d = 6 hour)
(Note: maximum length of time allowed for Duration & Interval is 6 weeks)

Complete Pathname of the file to write data to:
C:\HISTDEMO\HISTDATA.CSV (Drive:\Path)

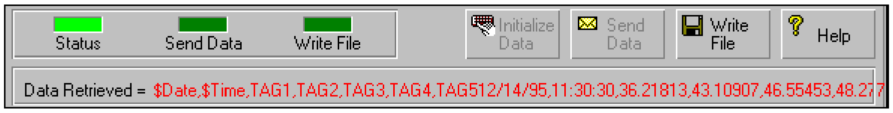
List of tagnames to return historical data for:
"$DATE,$TIME,TAG1,TAG2,TAG3,TAG4,TAG5" ("TagA,TagB,TagC")
(Note: Date & Time for a sample can be requested via the System tags, $Date and $Time.)

Error Message = None

The requested historical data can be changed by clicking on the respective input box, typing in the new data, and then pressing the ENTER key.

Once the historical data has been specified, click **Send Data**, the HistData program will update its internal **DATA** database item with the requested data. (The **DATA** database item is used to hold the requested historical data in comma separated variable format. It is used by other applications that want to **ADVISE** or **REQUEST** data.)

The historical data sent to the **DATA** item will also be displayed in the string output field at the top of the screen labeled **Data Retrieved**.

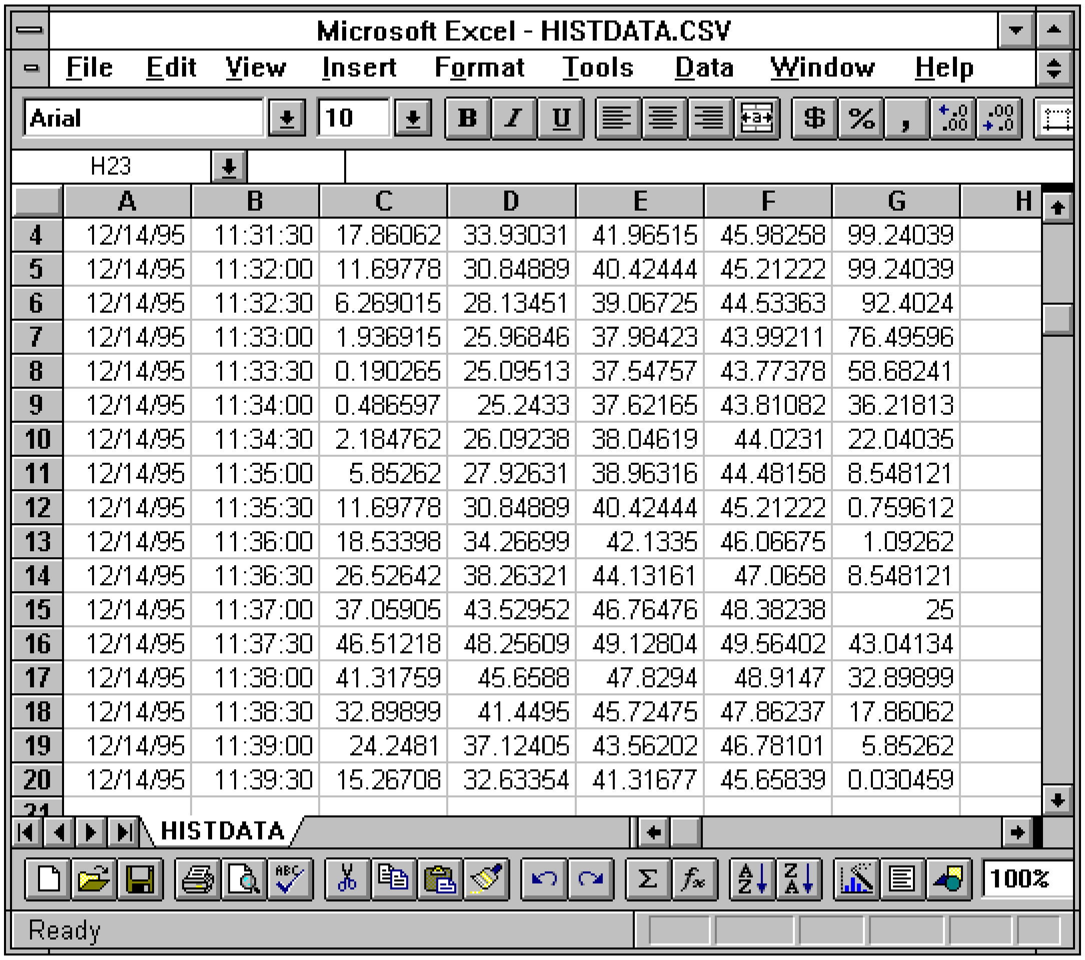


Data Retrieved = $Date,$Time,TAG1,TAG2,TAG3,TAG4,TAG512/14/95,11:30:30,36.21813,43.10907,46.55453,48.27

The HistData program has the ability to write the requested historical data to the file specified in the file name field. Example: C:\HISTDEMO\HISTDATA.CSV. The file created will be in .CSV (comma separated variable) format. To create the .CSV file, click **Write File**.

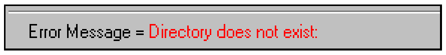
Note If the complete path is not specified, the file will automatically be created and saved in the current Windows default directory.

When a .CSV file is opened in Microsoft Excel, the data is automatically separated into columns. It is recommended that the file be created with the .CSV extension. For example:


Microsoft Excel - HISTDATA.CSV
File Edit View Insert Format Tools Data Window Help
Arial 10
H23
HISTDATA
100%
Ready

	A	B	C	D	E	F	G	H
4	12/14/95	11:31:30	17.86062	33.93031	41.96515	45.98258	99.24039	
5	12/14/95	11:32:00	11.69778	30.84889	40.42444	45.21222	99.24039	
6	12/14/95	11:32:30	6.269015	28.13451	39.06725	44.53363	92.4024	
7	12/14/95	11:33:00	1.936915	25.96846	37.98423	43.99211	76.49596	
8	12/14/95	11:33:30	0.190265	25.09513	37.54757	43.77378	58.68241	
9	12/14/95	11:34:00	0.486597	25.2433	37.62165	43.81082	36.21813	
10	12/14/95	11:34:30	2.184762	26.09238	38.04619	44.0231	22.04035	
11	12/14/95	11:35:00	5.85262	27.92631	38.96316	44.48158	8.548121	
12	12/14/95	11:35:30	11.69778	30.84889	40.42444	45.21222	0.759612	
13	12/14/95	11:36:00	18.53398	34.26699	42.1335	46.06675	1.09262	
14	12/14/95	11:36:30	26.52642	38.26321	44.13161	47.0658	8.548121	
15	12/14/95	11:37:00	37.05905	43.52952	46.76476	48.38238	25	
16	12/14/95	11:37:30	46.51218	48.25609	49.12804	49.56402	43.04134	
17	12/14/95	11:38:00	41.31759	45.6588	47.8294	48.9147	32.89899	
18	12/14/95	11:38:30	32.89899	41.4495	45.72475	47.86237	17.86062	
19	12/14/95	11:39:00	24.2481	37.12405	43.56202	46.78101	5.85262	
20	12/14/95	11:39:30	15.26708	32.63354	41.31677	45.65839	0.030459	

The HistData program has built-in error messages. Therefore, if an error occurs, a corresponding message will appear in the **Error Message** string output field in the lower left corner. For example:



Error Message = Directory does not exist.

Note When an error occurs, the **DATA** item is not updated.

Using HistData with Excel

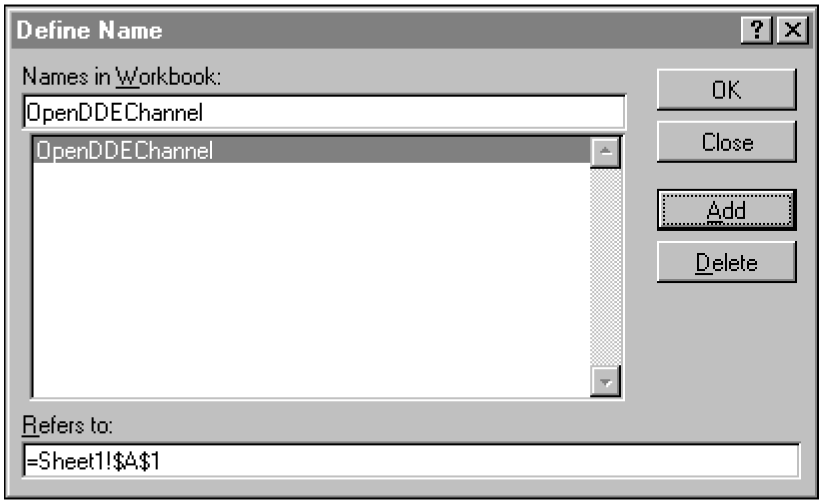
The HistData program responds to the **INITIATE**, **POKE**, and **TERMINATE** functions of products such as Microsoft Excel. The **POKE** function with a **keyword** (an internal database item) is used to set up the parameters that define a query. Once the query is properly set up, the macro is run to request the selected historical data.

Excel can be used to create macro sheets containing various macros. The following examples demonstrate how to open and close a DDE channel from Excel to HistData, how to poke values to the various HistData internal database items and how to initiate the **SendData** and **WriteFile** functions of the HistData program.

Note Excel examples in this User's Guide generally illustrate Excel version 4.0. The following Excel menus and commands may differ based on the installed version.

In order for Excel to recognize a macro, the name entered for it must be defined as a macro as follows:

1. On the **Insert** menu, point to **Name** then click **Define**. The **Define Name** dialog box appears:

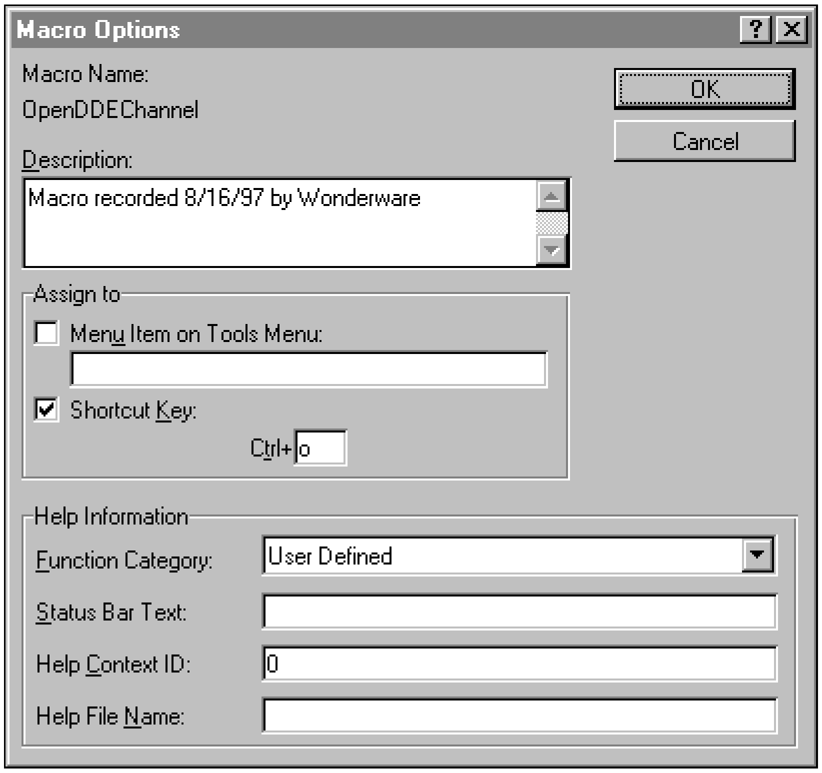


Names in Workbook: OpenDDEChannel

OpenDDEChannel

Refers to: =Sheet1!A1

2. In the **Names in Workbook** box, type the name of the macro, and then click **Add**.
3. Click **Options**, to assign a short cut key to execute the macro. The **Macro Options** dialog box appears:



Macro Name:
OpenDDEChannel

Description:
Macro recorded 8/16/97 by Wonderware

- [ ] Menu Item on Tools Menu:
- [x] Shortcut Key: Ctrl+ o

Function Category: User Defined
Status Bar Text:
Help Context ID: 0
Help File Name:

4. Select **Shortcut Key**, and then in the **Ctrl+** box, type the key you want to use to execute the macro.
5. Click **OK**. The **Macro** dialog box reappears.
6. Click **Run** to execute the macro.

Note If a macro has a command key assigned to it, the key is displayed in front of the macro name in the **Run** dialog box (accessed by executing the Macro/Run command). When you record a macro, you're creating a Visual Basic Sub procedure.

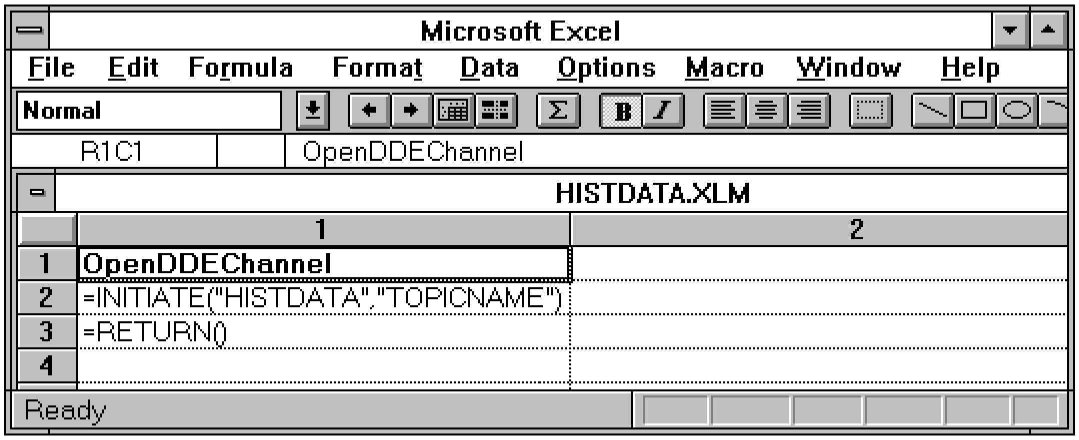
1 On the Tools menu, click **Macro**.

2 In the **Macro Name/Reference** dialog box, enter a macro name.

3 Click **Run**.

OpenDDEChannel Macro

The following macro initiates a DDE channel to the HistData program from Excel:



	1	2
1	OpenDDEChannel	
2	=INITIATE("HISTDATA","TOPICNAME")	
3	=RETURN()	
4		

Where:

=INITIATE("HistData","TOPICNAME") opens a DDE channel from Excel to the HistData program. (HistData is always used for the *application name* and **TOPICNAME** is any arbitrary topic name (up to 32 characters long).

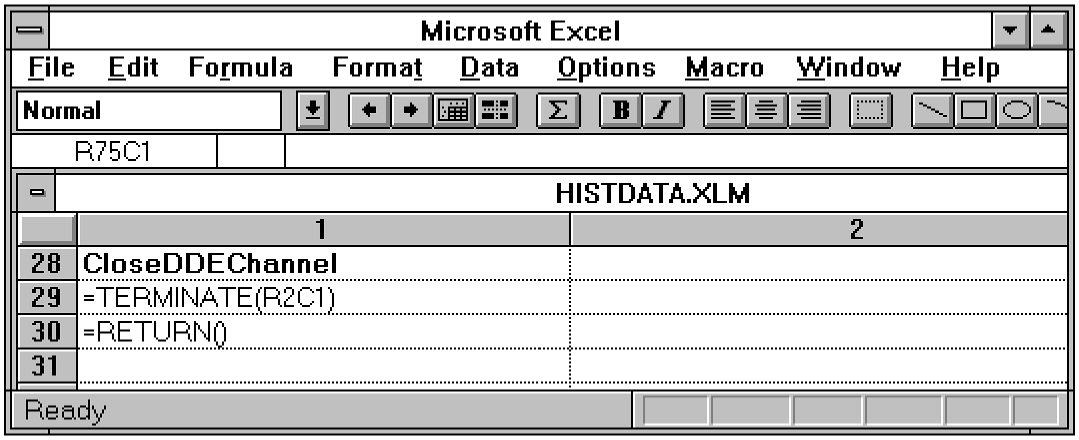
Note In Excel, when text is used in a formula, it must be enclosed in quotation marks (" ").

=RETURN() marks the end of this macro

Note In this example the **INITIATE** function is in cell R2C1. Note that all the **POKE** commands in the following macros will refer to this cell ID as the DDE channel.

CloseDDEChannel Macro

In some cases, if a **TERMINATE** statement is included in a macro which has **POKE** commands, the terminate command may execute before all poke commands are completed. To avoid this problem, a **CloseDDEChannel** macro is run:



	1	2
28	CloseDDEChannel	
29	=TERMINATE(R2C1)	
30	=RETURN()	
31		

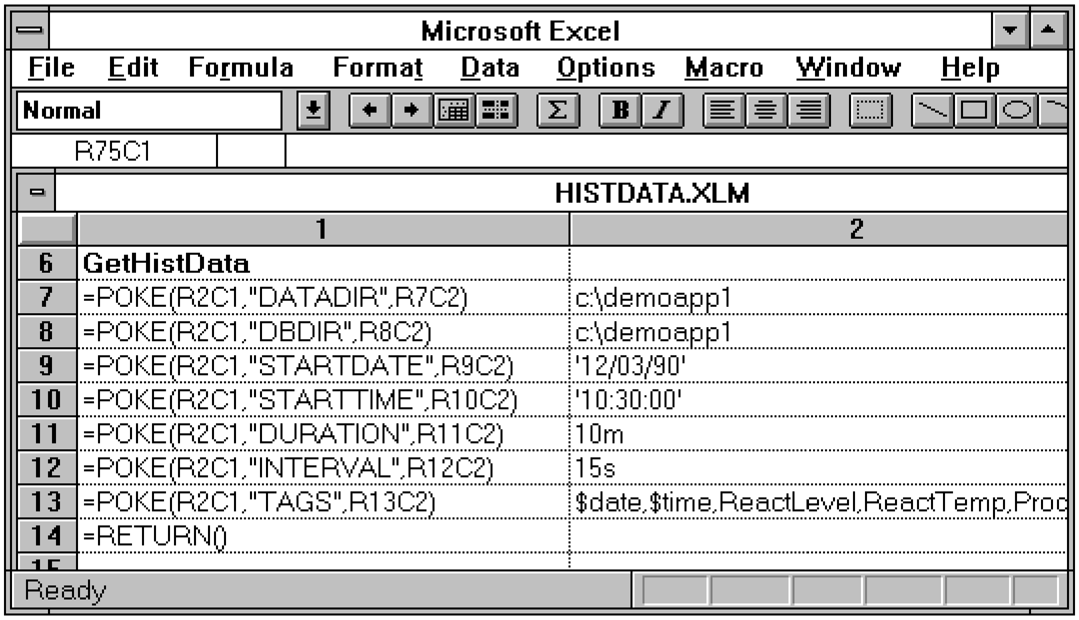
Where:

=TERMINATE(R2C1) closes the DDE channel initiated in cell R2C1.

=RETURN() marks the end of this macro.

GetHistData Macro

Once the DDE channel to the HistData program has been initiated (by first running the **OpenDDEChannel** macro) the following macro is run by pressing the assigned command keys (**Ctrl+g**). This macro will **POKE** values from Excel to the various HistData program's database items:



	1	2
6	GetHistData	
7	=POKE(R2C1,"DATADIR",R7C2)	c:\demoapp1
8	=POKE(R2C1,"DBDIR",R8C2)	c:\demoapp1
9	=POKE(R2C1,"STARTDATE",R9C2)	'12/03/90'
10	=POKE(R2C1,"STARTTIME",R10C2)	'10:30:00'
11	=POKE(R2C1,"DURATION",R11C2)	10m
12	=POKE(R2C1,"INTERVAL",R12C2)	15s
13	=POKE(R2C1,"TAGS",R13C2)	$date,$time,ReactLevel,ReactTemp,Proc
14	=RETURN()	

Where:

=POKE(R2C1, "DATADIR", R7C2) POKEs the pathname of the directory (in cell R7C2) containing the historical data files to the HistData program's internal variable "**DATADIR**".

=POKE(R2C1, "DBDIR", R8C2) POKEs the pathname of the directory (in cell R8C2) containing the InTouch Tagname Dictionary to the HistData program's internal variable "**DBDIR**".

=POKE(R2C1, "STARTDATE", R9C2) POKEs the value contained in cell R9C2 to the HistData program's internal variable "**STARTDATE**".

=POKE(R2C1, "STARTTIME", R10C2) POKEs the value contained in cell R10C2 to the HistData program's internal variable "**STARTTIME**".

=POKE(R2C1, "DURATION", R11C2) POKEs the value contained in cell R11C2 to the HistData program's internal variable "**DURATION**".

=POKE(R2C1, "INTERVAL", 12C2) POKEs the value contained in cell R12C2 to the HistData program's internal variable "**INTERVAL**".

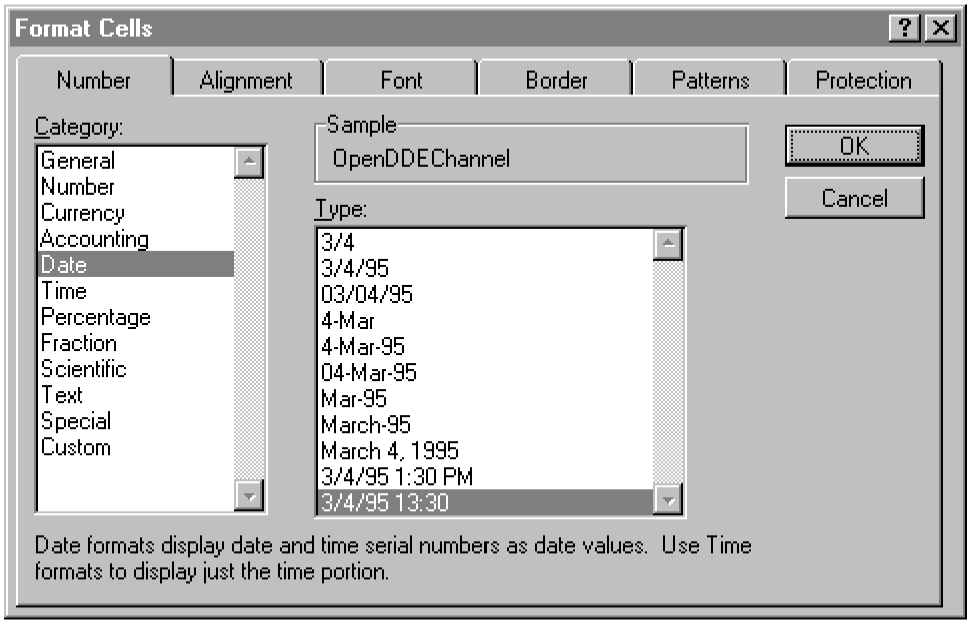
=POKE(R2C1, "TAGS" ,R13C2) POKEs the tagnames contained in cell R13C2 to the HistData program's internal variable "**TAGS**".

=RETURN() marks the end of this macro.

Note In Excel Version 7.0 or later, when a date and time are used in a formula, they must be enclosed in single quotes (') and formatted properly.

➢ **To format the date and time**

1. On the **Format** menu, click **Cells**. The **Format Cells** dialog box will appears:



Category:
General
Number
Currency
Accounting
Date
Time
Percentage
Fraction
Scientific
Text
Special
Custom

Sample
OpenDDEChannel

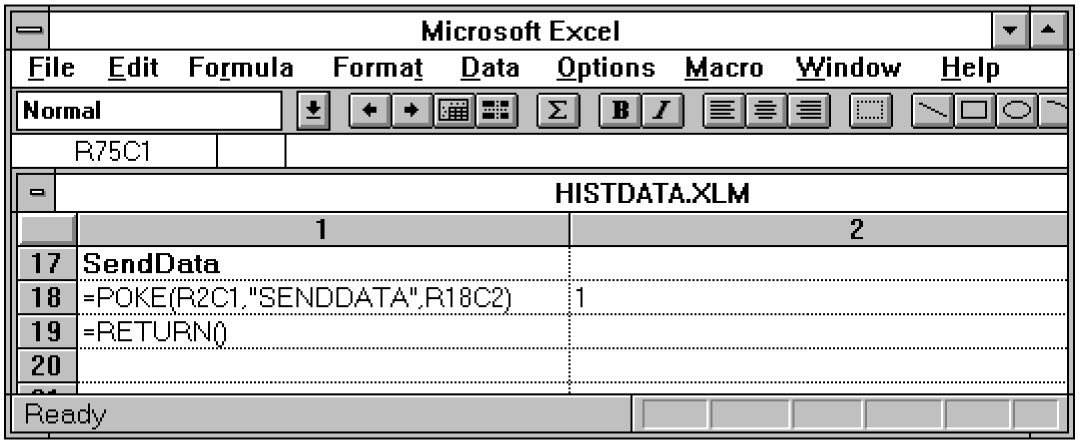
Type:
3/4
3/4/95
03/04/95
4-Mar
4-Mar-95
04-Mar-95
Mar-95
March-95
March 4, 1995
3/4/95 1:30 PM
3/4/95 13:30

Date formats display date and time serial numbers as date values. Use Time formats to display just the time portion.

2. Click the **Number** tab to activate the Number property sheet.
3. In the **Category**, select **Date** or **Time**.
4. In the Type, select the format that you want to use for the date and time.
5. Click **OK**.

SendData Macro

Once a DDE channel has been initiated (by first running the **OpenDDEChannel** macro) and the **GetHistData** macro has been run, the following macro is run by pressing the assigned command keys (**Ctrl+s**). This macro causes the **SendData** function of the HistData program to update the HistData database item **Data** with the historical data in the **GetHistData** macro:



	1	2
17	SendData	
18	=POKE(R2C1,"SENDDATA",R18C2)	1
19	=RETURN()	
20		

Where:

=POKE(R2C1, "SENDDATA", R18C2) POKEs the value of 1 (from cell R18C2) to the HistData internal discrete variable "**SENDDATA**" to cause the internal database item **DATA** to be updated with the requested historical data. (**DATA** is the item which holds the data in comma separated variable format.)

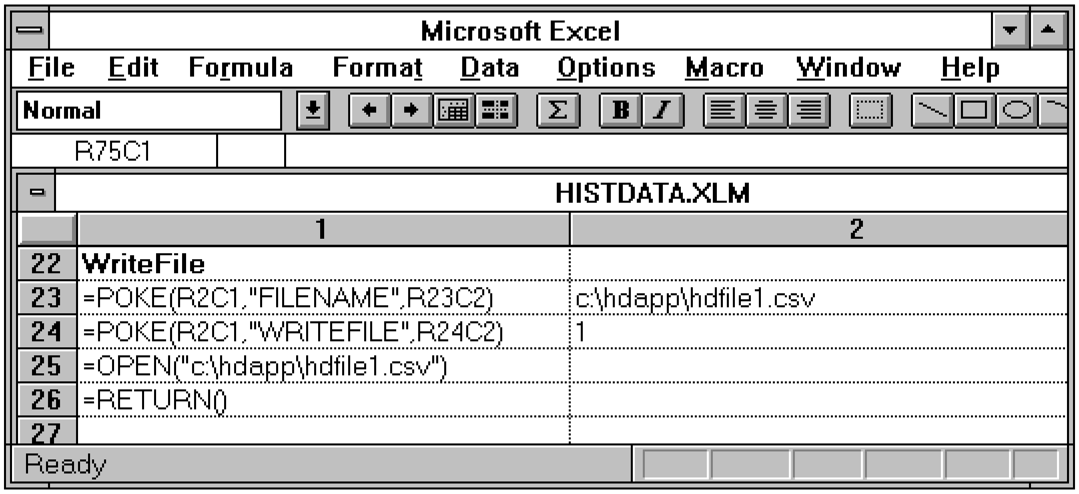
Note The **SENDDATA** item will be reset to 0 (zero) after the update.

=RETURN() marks the end of this macro.

Note Once this macro has been run, if no other activity is required, run the **CloseDDEChannel** macro (Ctrl+c) to terminate the DDE channel.

WriteFile Macro

Once a DDE channel has been initiated (by running the **OpenDDEChannel** macro) and the **GetHistData** macro has been run, the following macro is run by pressing the assigned command keys (Ctrl+w). This macro causes the **WRITEFILE** function of the HistData program to write the requested historical data to the specified file:



	1	2
22	WriteFile	
23	=POKE(R2C1,"FILENAME",R23C2)	c:\hdapp\hdfile1.csv
24	=POKE(R2C1,"WRITEFILE",R24C2)	1
25	=OPEN("c:\hdapp\hdfile1.csv")	
26	=RETURN()	
27		

Where:

=POKE(R2C1, "FILENAME", R23C2) POKEs the complete pathname of the file contained in cell R23C2 the HistData internal variable "**FILENAME**". (This is the name of the file to which the requested data will be written.)

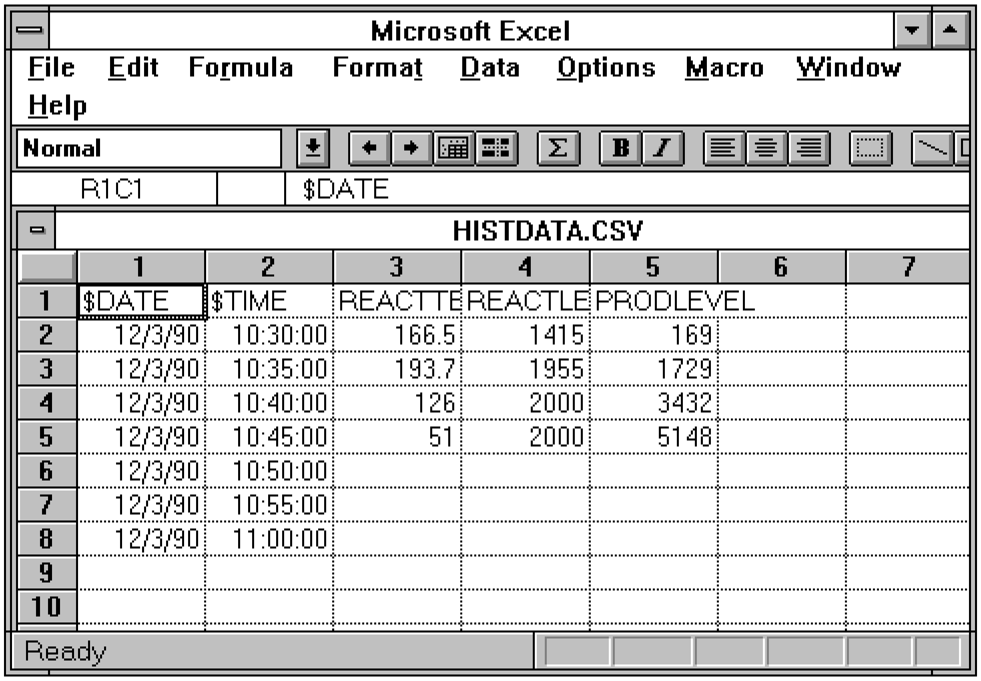
=POKE(R2C1, "WRITEFILE", R24C2) POKEs the value of 1 (from cell R24C2) to the HistData internal discrete variable "**WRITEFILE**" which causes the HistData program to create the specified file and write the requested historical data to the file. (The **WRITEFILE** item will automatically be reset to 0 (zero) after the file is written.)

=OPEN("c:\InTouch\hdfile1.csv") automatically opens the specified file as an Excel spreadsheet once the macro is completed. (If the file is created with the extension .CSV when it opens in Excel the data will automatically be separated into columns.) (See sample on next page.)

=RETURN() marks the end of this macro.

Note Once this macro has been run, if no other activity is required, run the **CloseDDEChannel** macro (Ctrl+c) to terminate the DDE channel.

When the **WriteFile** macro is run, the specified file will automatically be opened as a spreadsheet in Excel:



	1	2	3	4	5	6	7
1	$DATE	$TIME	REACTTE	REACTLE	PRODLEVEL		
2	12/3/90	10:30:00	166.5	1415	169		
3	12/3/90	10:35:00	193.7	1955	1729		
4	12/3/90	10:40:00	126	2000	3432		
5	12/3/90	10:45:00	51	2000	5148		
6	12/3/90	10:50:00					
7	12/3/90	10:55:00					
8	12/3/90	11:00:00					
9							
10							

Note The data is automatically separated into columns since the file was created in the comma separated variable format (.CSV). If no data is available for the tags for the specified start date and time, their respective columns will be blank.

If using Excel version 7.0 or later, select the entire spreadsheet, then on the **Format** menu, point to **Column**, and then click **AutoFit Selection** to automatically adjust the size of the columns in relation to the data.

The previous sample macros could be further enhanced to report errors by incorporating the HistData database item **STATUS**. Once a **POKE** command is performed a **REQUEST** can be made on the state of the **STATUS** item. If **STATUS=0**, a request can be made to display the HistData built-in "Error" message to provide debugging information.

Note The demo applications supplied with InTouch include the Hist Demo. The Hist Demo is a generic InTouch application specifically designed to extract data from a log file by using the HistData utility and will work without any modifications.

Using Excel 5.0 VBA Macros with HistData

Excel 5.0 VBA can be used to write macros to interface with InTouch HistData (or any other DDE Server). The following is a simple macro to do this:

```
Sub  GetHistData ()
     Dim rangeToPoke
     Dim channelNumber

     channelNumber = Application.DDEInitiate("histdata", "topic")
     Set rangeToPoke = Worksheets("Sheet1") .Cells(1, 1)
     Application.DDEPoke channelNumber, "startdate", rangeToPoke
     Application.DDETerminate channelNumber

End  Sub
```

Note that the data (which is **POKEd**) comes from a cell in the Excel spreadsheet "Sheet1." This is similar to the method using Excel 4.0 in that the data has to reference a spreadsheet cell. Thus, the following will not work:

```
Sub  GetHistData ()
     Dim startDate
     Dim channelNumber

     channelNumber = Application.DDEInitiate("histdata", "topic")
     startDate = "12/03/90"
     Application.DDEPoke channelNumber, "startdate", startDate
     Application.DDETerminate channelNumber

End  Sub
```

In Excel 4.0, the macro resided in a spreadsheet that you could use for cell references. Excel 5.0 VBA, macros to not reside in a spreadsheet. They reside on a Worksheet in your Excel workbook.

C H A P T E R   9

I/O Communications

InTouch uses the Microsoft Dynamic Data Exchange (DDE), FastDDE, NetDDE and Wonderware SuiteLink protocols to communicate with other Windows programs, Wonderware I/O Servers and third-party I/O Server programs that are communicating with the real world.

Contents

Supported Communication Protocols

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 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:

- 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.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft’ s standard Winsock interface.

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.

➢ **To use the SuiteLink communication protocol:**

1. You must have Microsoft TCP/IP configured and working properly.

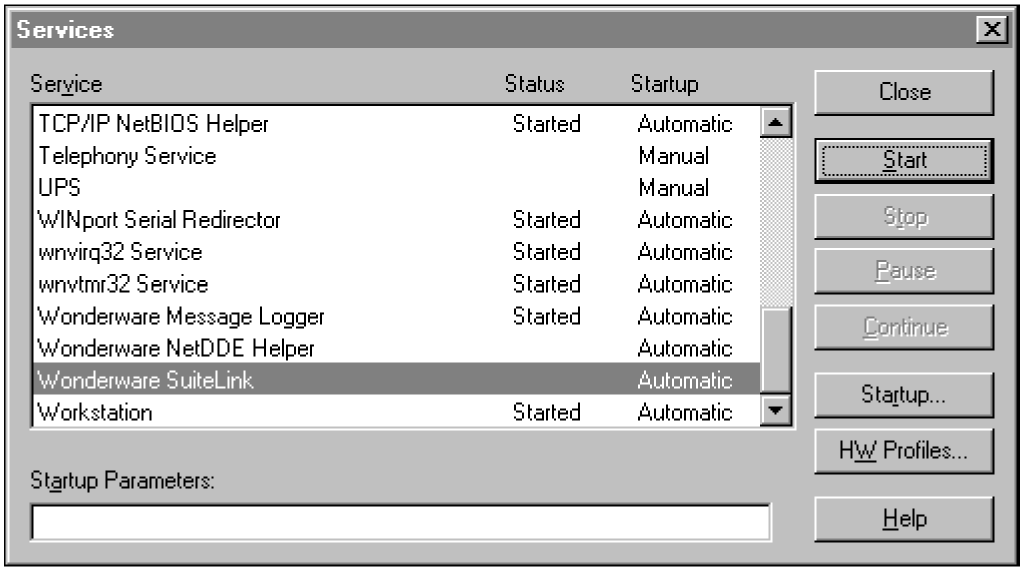
   Refer to your Windows NT operating system's Help for complete details on installing and configuring Microsoft TCP/IP.

2. Wonderware SuiteLink must be running as a service. If for some reason SuiteLink has been stopped, you will need to start it again.

   SuiteLink is automatically installed when you install InTouch and by default, it is configured to startup automatically as an NT Service.

   a) To start SuiteLink as an NT Service, open the Windows Control Panel.

   b) Double-click **Services**. The **Services** dialog box appears:



Service	Status	Startup
TCP/IP NetBIOS Helper	Started	Automatic
Telephony Service		Manual
UPS		Manual
WINport Serial Redirector	Started	Automatic
wnvirq32 Service	Started	Automatic
wnvtmr32 Service	Started	Automatic
Wonderware Message Logger	Started	Automatic
Wonderware NetDDE Helper		Automatic
Wonderware SuiteLink		Automatic
Workstation	Started	Automatic

   c) Select **Wonderware SuiteLink**, and then click **Start**.

   d) Click **Close**.

The InTouch I/O Address Convention

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.

The InTouch I/O Address

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 always 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'

Note If you are networking using Wonderware NetDDE, the *application name* portion of the I/O address must be prefixed with the remote node's name preceded by two backslashes and followed by one backslash. For example:

\\NodeName\VIEW|TAGNAME!'*ActualTagname'*

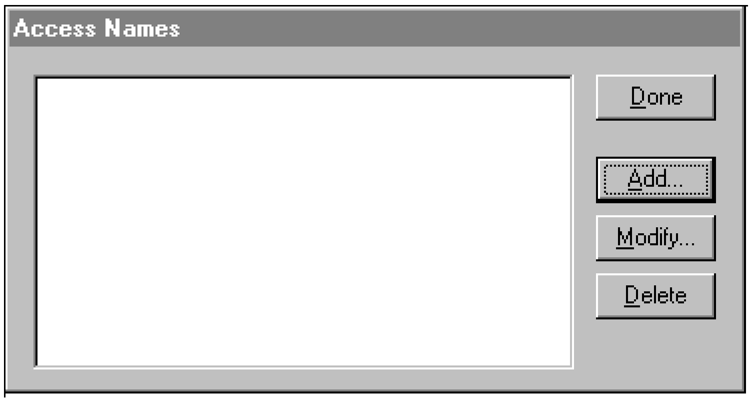
InTouch Access Names

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.

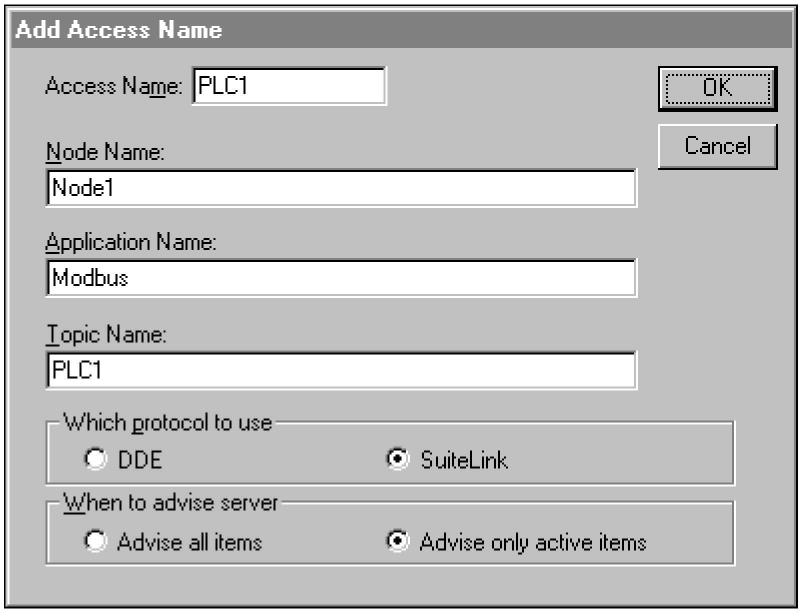
➢ **To create an Access Name:**

1. On the **Special** menu, click **Access Names**, or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears:

   In the Application Explorer, you can right-click **Access Names**, and then click **Open**. You can also create Access Names while you are defining an I/O type tagname in the Tagname dictionary.



2. Click **Add**. The **Add Access Name** dialog box appears:



Access Name: PLC1

Node Name: Node1

Application Name: Modbus

Topic Name: PLC1

Which protocol to use: ( ) DDE [✓ SuiteLink]

When to advise server: ( ) Advise all items [✓ Advise only active items]

3. In the **Access Name** box type the name you want InTouch to use to this Access Name. (For simplicity, use the same name that you will use for the *topic name* here.)

   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, in the **Node Name** box, type the remote node's name.

5. In the **Application Name** box, type 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.

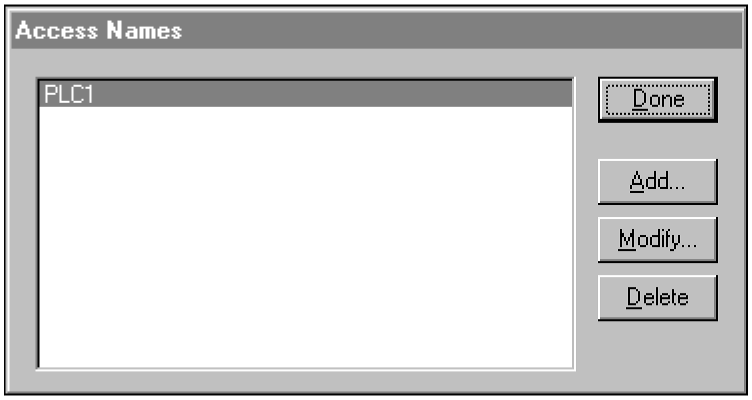
6. In the **Topic Name** box, type the *topic name* you want to access.

   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 that you are using.

8. Select **Advise all items** if you want the server program 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, therefore its use is not recommended.
9. Select **Advise only active items** if you want the server program 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 will reappear displaying the new Access Name selected in the list:

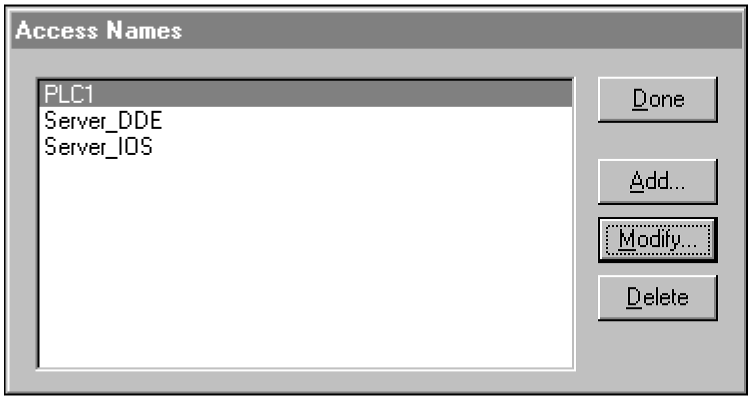


11. Click **Done** to close the dialog box and return to your tagname definition.

➢ **To modify or delete an Access Name:**

12. On the **Special** menu, click **Access Names**, or in the Application Explorer under **Configure**, double-click **Access Names**. The **Access Names** dialog box appears:

    In the Application Explorer, you can right-click **Access Names**, and then click **Open**.



13. To change an Access Name's definition, select it in the list, and then click **Modify**. The **Modify Access Name** dialog box will appear. Make your required changes, and then click **OK**. The **Access Names** dialog box reappears. Click **Done** or repeat this procedure if you need to modify other defined Access Names.
14. 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 of the selected Access Name. Click **Yes** to delete it or click **No** to cancel the deletion. Click **Done** or repeat this procedure if you need to delete other defined Access Names.

Note Access Names that are used in tagnames cannot be deleted.

Defining an I/O Item in InTouch

InTouch can receive data from other local or remote Windows applications when you define I/O type tagnames in the Tagname Dictionary. Each I/O type tagname references a valid *item* in the I/O Server program.

For more information on distributed applications see, Chapter 3 - Creating a Distributed Application.

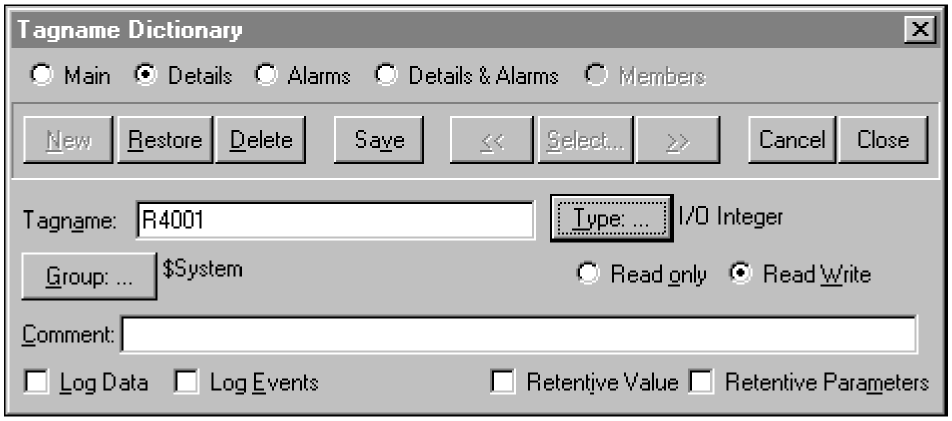
➤ **To define an I/O type tagname:**

1. On the **Special** menu, click **Tagname Dictionary**, or in the Application Explorer double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box appears.

   For more information on defining I/O tagnames, see Chapter 4 - Tagname Dictionary.

2. Click **New**. The **Tagname** box clears.

   If you right-click any of the text entry boxes in any of the Tagname Dictionary dialog boxes, a menu will appear displaying the commands that you can apply to the selected text.



Tagname: R4001
Type: I/O Integer
Group: $System
Read only [✓ Read Write]
Comment:

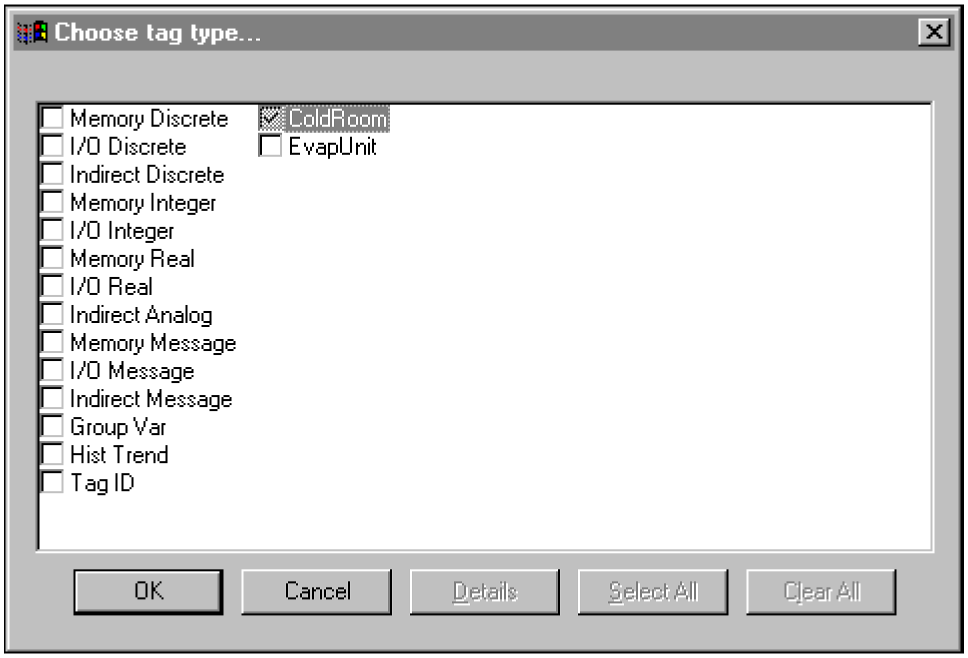
   The first time you access the Tagname Dictionary, the definition for the internal system tagname **$AccessLevel** is displayed. Once you define tagnames in the Tagname Dictionary, when you access it again, the last edited tagname's definition is displayed.

3. In the **Tagname** box, type the name you want to use for the 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 an 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 and so on.

4. Click **Type**. The **Choose tagname type** dialog box appears.

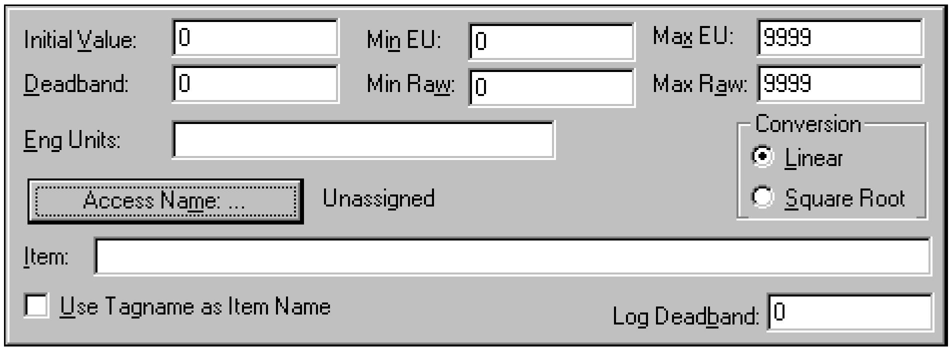

Choose tag type...

- [ ] Memory Discrete
- [ ] I/O Discrete
- [ ] Indirect Discrete
- [ ] Memory Integer
- [ ] I/O Integer
- [ ] Memory Real
- [ ] I/O Real
- [ ] Indirect Analog
- [ ] Memory Message
- [ ] I/O Message
- [ ] Indirect Message
- [ ] Group Var
- [ ] Hist Trend
- [ ] Tag ID
- [✓] ColdRoom
- [ ] EvapUnit

5. Select the I/O type for the tagname as follows:

I/O Discrete	input/output value of either True (1) or False (0)
I/O Integer	(whole number) input/output value
I/O Real	floating (decimal) point input/output value
I/O Message	(string) input/output value

6. Once you select the I/O type, and then click **OK**. The respective "details" dialog box for the selected I/O type will appear. For example, if you select I/O Integer, the following dialog box appears:

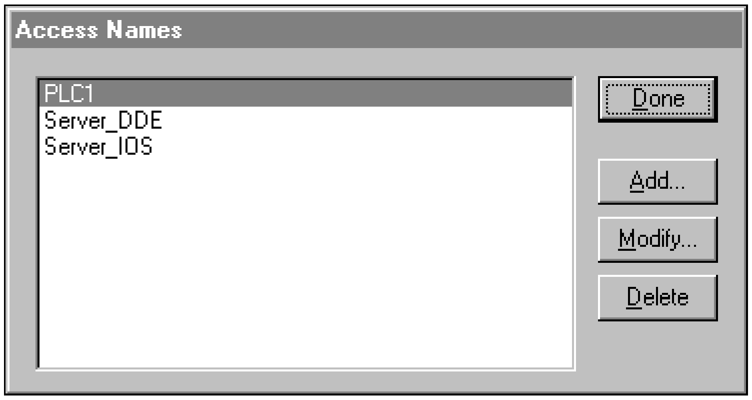


Initial Value: 0
Min EU: 0
Max EU: 9999
Deadband: 0
Min Raw: 0
Max Raw: 9999
Eng Units:
Conversion: (● Linear) (○ Square Root)
Unassigned
Item:
☐ Use Tagname as Item Name
Log Deadband: 0

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:

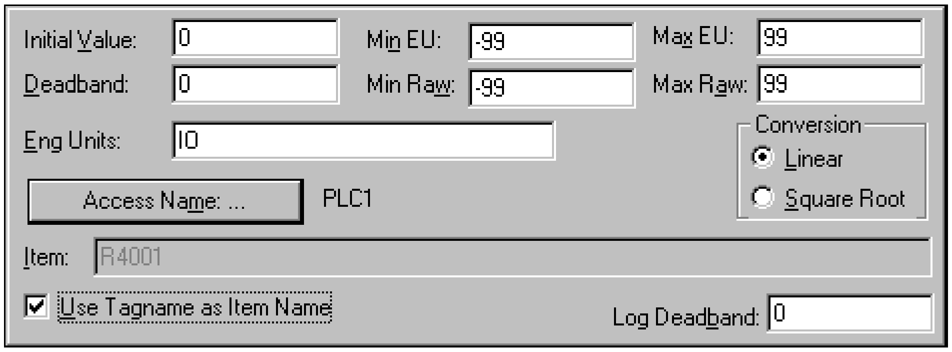


Access Names

PLC1
Server_DDE
Server_IOS

9. Double-click the Access Name that you want to use in the list or select it, and then click **Done**.

10. The Access Name that you selected (now associated with this tagname definition) appears adjacent to the **Access Name** button in the details dialog box. For example:



Initial Value: 0
Min EU: -99
Max EU: 99
Deadband: 0
Min Raw: -99
Max Raw: 99
Eng Units: IO
Conversion: [✓ Linear] Square Root
Access Name: ... PLC1
Item: R4001
[✓] Use Tagname as Item Name
Log Deadband: 0

11. In the **Item** box, type the *item name* for the data value in the I/O Server program.

Note It is important to understand that the "tagname" is the name used within InTouch to refer to a data value. The **Item** is the name used by a remote Windows application to refer to the same value. These names do not have to be the same but, it is recommended when applicable to use the same names. Also, if the **Item** is a cell in Excel, it must be specified either by its defined name in Excel, or by its row/column identification. For example, R1C1.

12. Click **Done**.

Monitoring the Status of an I/O Conversation

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

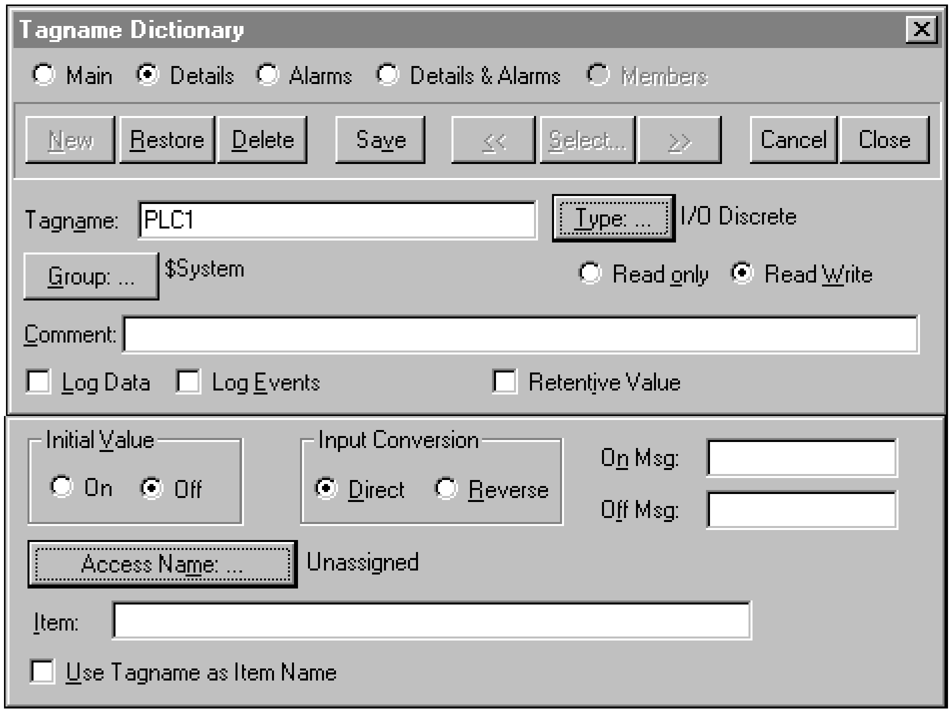
Let's assume that WindowViewer (View) is communicating with the Wonderware Simulate I/O Server to a PLC that has been defined in the I/O Server with **PLC1** for its *topic name*.

(Simulate is a generic Wonderware I/O Server that is intended to be used as a training tool. Simulate is included with FactorySuite.)

➢ **To monitor the status of I/O communications:**

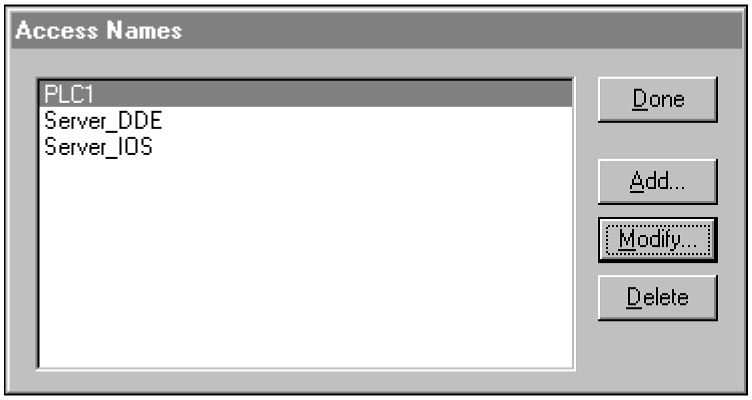
1. On the **Special** menu, click **Tagname Dictionary**, or in the Application Explorer, double-click **Tagname Dictionary**. The **Tagname Dictionary** dialog box appears.
2. Create an **I/O Discrete** type tagname. (In this example, for simplicity, we will make our tagname the same as the *topic name* that we want to monitor):

   When you are monitoring a I/O conversation using **IOStatus**, you must define at least one I/O type tagname to the Access Name being monitored.

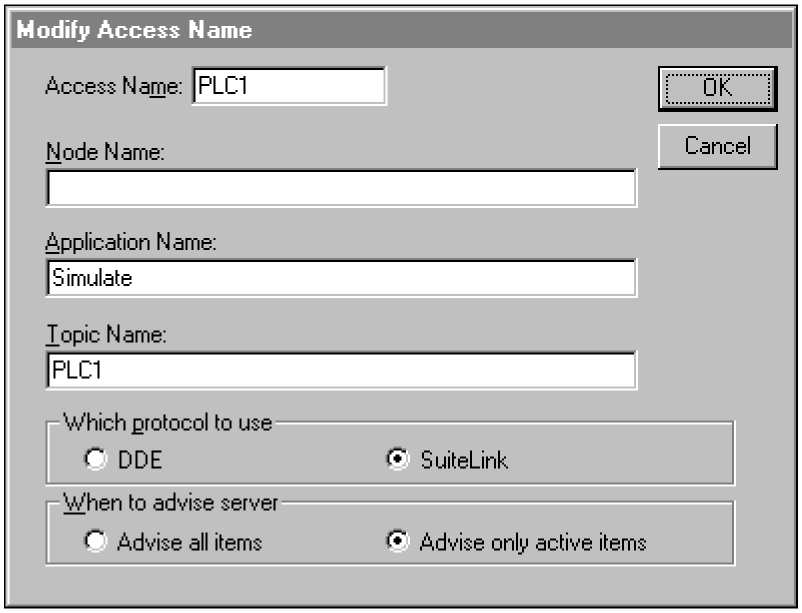


Details selected
Tagname: PLC1
Type: I/O Discrete
Group: $System
Read only [✓ Read Write]
Comment:
Log Data ☐ Log Events ☐ Retentive Value ☐
Initial Value: On [✓ Off]
Input Conversion: [✓ Direct] Reverse
On Msg:
Off Msg:
Access Name: Unassigned
Item:
☐ Use Tagname as Item Name

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 want 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:

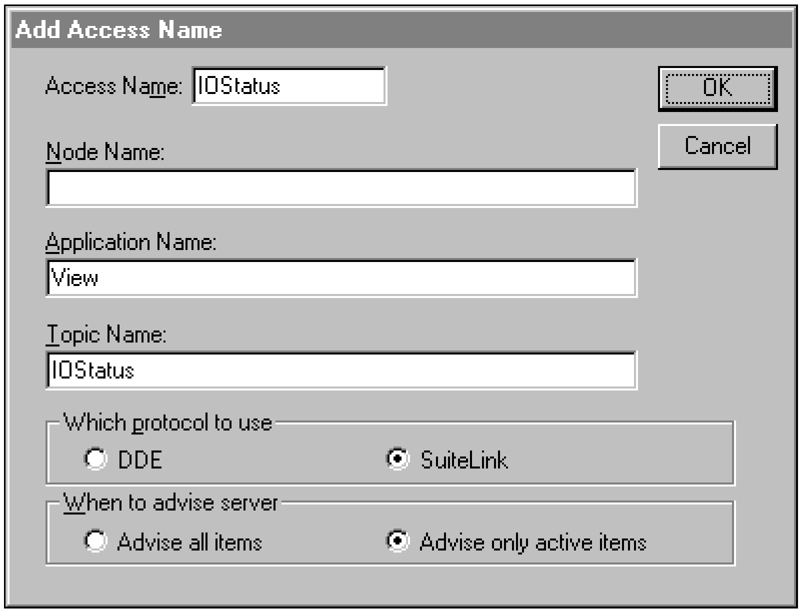


Access Name: PLC1
Node Name:
Application Name: Simulate
Topic Name: PLC1
Which protocol to use: DDE [✓ SuiteLink]
When to advise server: Advise all items [✓ Advise only active items]

Finding the Access Name containing the right *topic name* in this example was 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 will appear:



Access Name: IOStatus

Node Name:

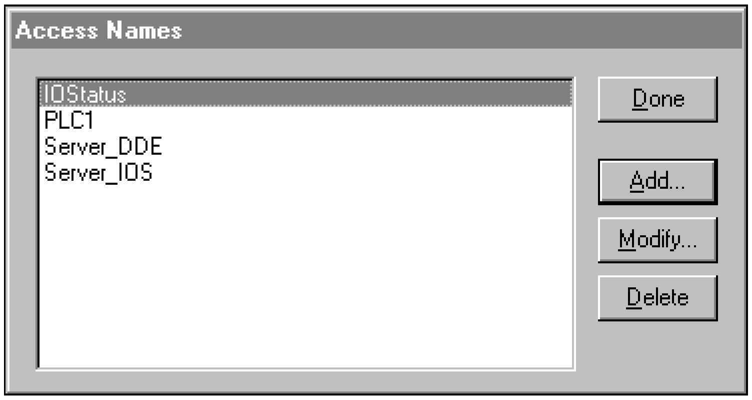
Application Name: View

Topic Name: IOStatus

Which protocol to use: DDE [✓ SuiteLink]

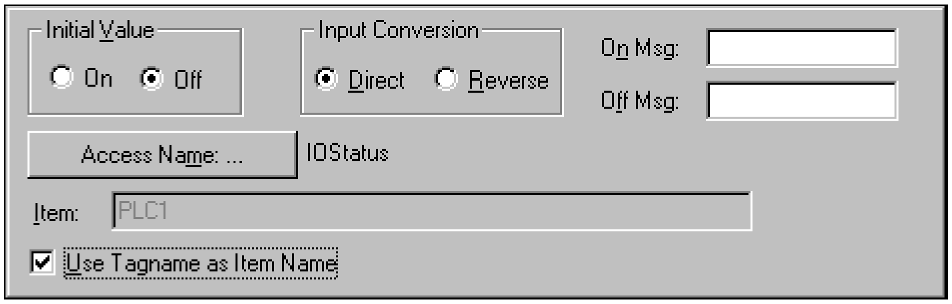
When to advise server: Advise all items [✓ Advise only active items]

6. In the **Access Name** box, type **IOStatus**.
7. Since you are going to monitor the status in WindowViewer, in the **Application Name** box, type "View."
8. In the **Topic Name** box, type 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:



IOStatus
PLC1
Server_DDE
Server_IOS

11. Click **Done** to close the dialog box and associate the new **Access Name** with your **I/O Discrete** tagname:



IOStatus

Item: PLC1

12. In the **Item** box, type the actual **Topic Name** that you want to monitor. In this case, **PLC1**.

    Since your tagname is the same as the **Topic Name**, you can select **Use Tagname as Item Name** and automatically enter it for the **Item**.

Note When using the built-in topic **IOStatus** (**DDEStatus** prior to InTouch Version 7.0) to monitor an I/O conversation, the name you type in the **Access Name** box is always also used 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:

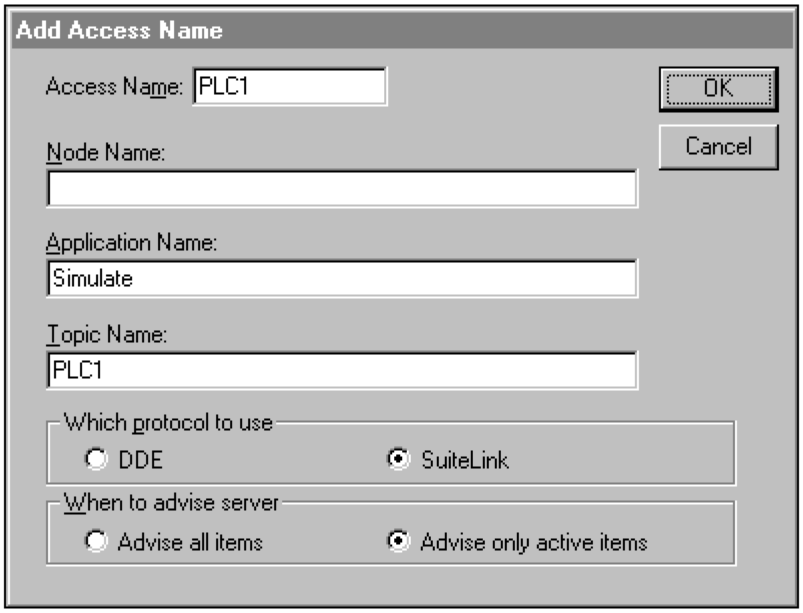
=view|IOStatus!'PLC1'

Monitoring I/O Server Communications Status

For each *topic name* being used, there is a built-in discrete *item*, **Status**, that you can use to monitor the state of your communications with the I/O Servers program. **Status** is set to "0" when communications with the device fails (cable disconnected, PLC is powered down, and so on.) and set to "1" when communications is successful.

Note When you monitor the status of a topic using the **Status** item, there must at least one I/O point active to the topic being monitored.

From InTouch, you can read the state of the server communications by defining a tagname and associating it with the *topic* configured for the device by using the word **Status** as the *item name*. For example, if WindowViewer is communicating with a PLC using the Wonderware Simulate I/O Server, the Access Name definition would be:



Access Name: PLC1

Node Name:

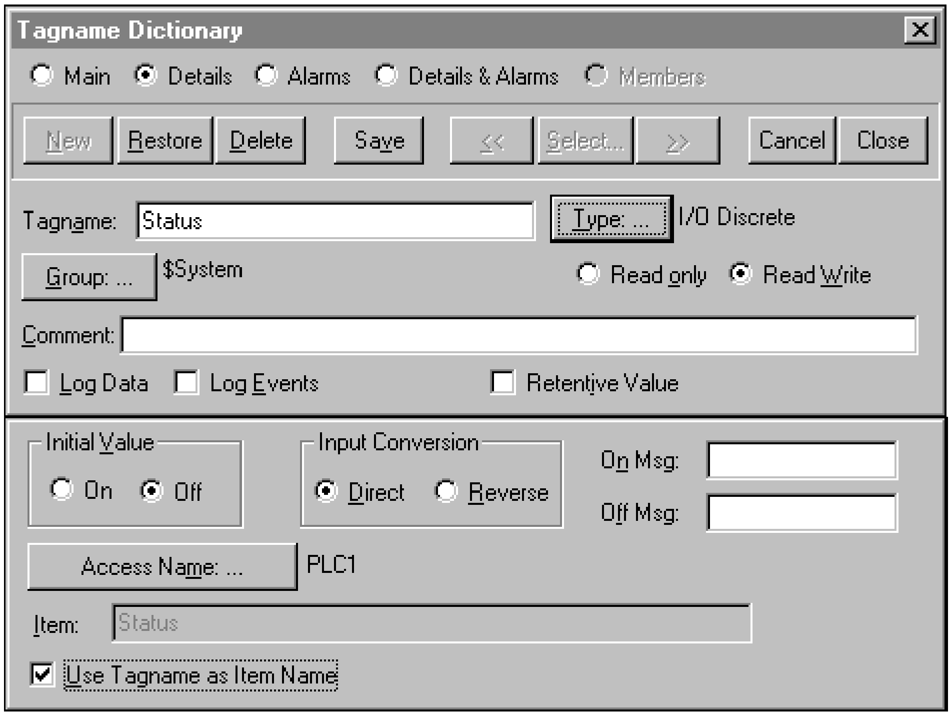
Application Name: Simulate

Topic Name: PLC1

Which protocol to use: DDE [✓ SuiteLink]

When to advise server: Advise all items [✓ Advise only active items]

To monitor the status of all communication to the *topic*, PLC1, you would create the following tagname definition:



Main [✓ Details] Alarms Details & Alarms Members

Tagname: Status

Type: I/O Discrete

Group: $System

Read only [✓ Read Write]

Comment:

☐ Log Data ☐ Log Events ☐ Retentive Value

Initial Value: On [✓ Off]

Input Conversion: [✓ Direct] Reverse

On Msg:

Off Msg:

Access Name: PLC1

Item: Status

☑ Use Tagname as Item Name

For more information on troubleshooting I/O communications see your Wonderware I/O Server's User Guide.

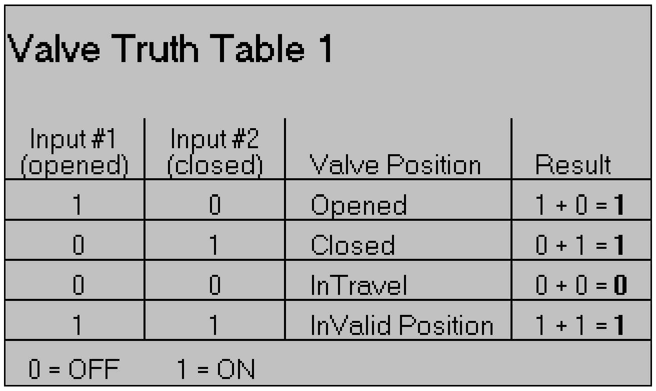
From Excel, you can read the status of the PLC communications by entering the following formula in a cell:

=SIMULATE|PLC1!'STATUS'

Monitoring Multiple Input Device Status

This section describes how you can display the status of an object using multiple inputs.

Example 1
In this example, the status of a spring-return motorized valve using two inputs is being viewed. The two inputs represent a pair of limit switches installed on the valve. One input is only on when the valve is in the open position and off when the valve is in travel or closed. The other input is only on when the valve is in the closed position and off when the valve is in the travel or open position. A truth table of the inputs would appear as follows:

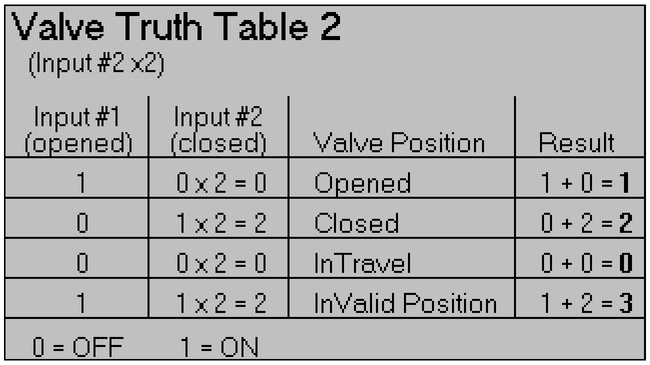


Valve Truth Table 1

Input #1 (opened)	Input #2 (closed)	Valve Position	Result
1	0	Opened	1 + 0 = **1**
0	1	Closed	0 + 1 = **1**
0	0	InTravel	0 + 0 = **0**
1	1	InValid Position	1 + 1 = **1**

0 = OFF 1 = ON

The inputs can be weighed by multiplying the closed input by 2. The results of the valve positions then change to the following:



Valve Truth Table 2
(Input #2 x2)

Input #1 (opened)	Input #2 (closed)	Valve Position	Result
1	0 x 2 = 0	Opened	1 + 0 = **1**
0	1 x 2 = 2	Closed	0 + 2 = **2**
0	0 x 2 = 0	InTravel	0 + 0 = **0**
1	1 x 2 = 2	InValid Position	1 + 2 = **3**

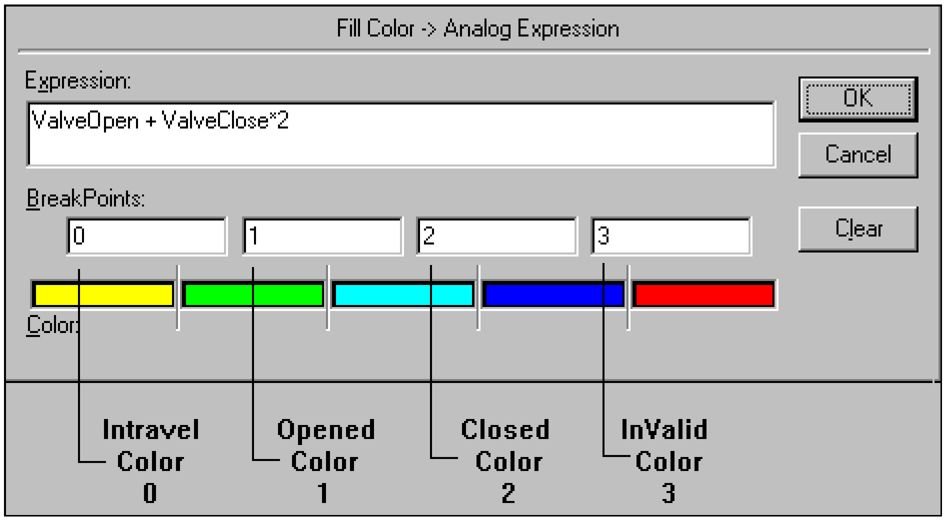
0 = OFF 1 = ON

Note The invalid position can be used to show a defective limit switch.

Now that there is a significant numerical difference between the valve positions, a **Fill Color - Analog** animation link can be used to display the valve status.

For more information on creating animation links, see Chapter 5 - Creating Animation Links.

Two I/O Discrete tagnames are created. One for the valve open input and one for the valve closed input. For example, **ValveOpen** and **ValveClosed**. An object is created to display the valve status. This object is assigned to a **Fill Color - Analog** animation link with the following properties:

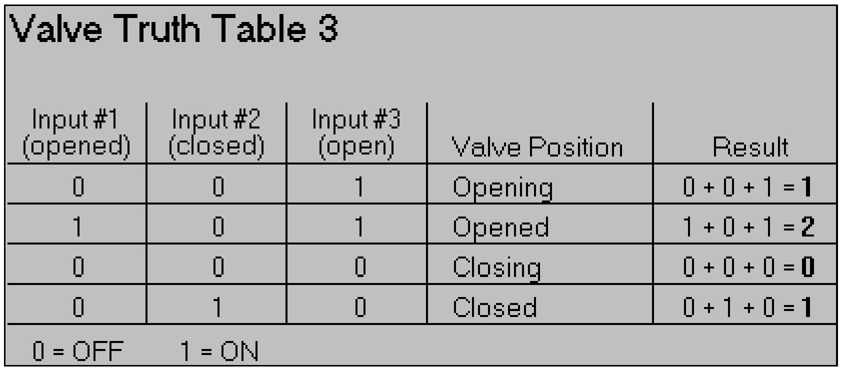


Fill Color -> Analog Expression

Expression: ValveOpen + ValveClose*2

BreakPoints: 0, 1, 2, 3

Example 2
In this example, one more input has been added to the existing two. This new input is the actual output to the valve that causes it to open. The new input will be on when the valve is opening or open, and off when the valve is closing or closed. The new truth table appears as follows:

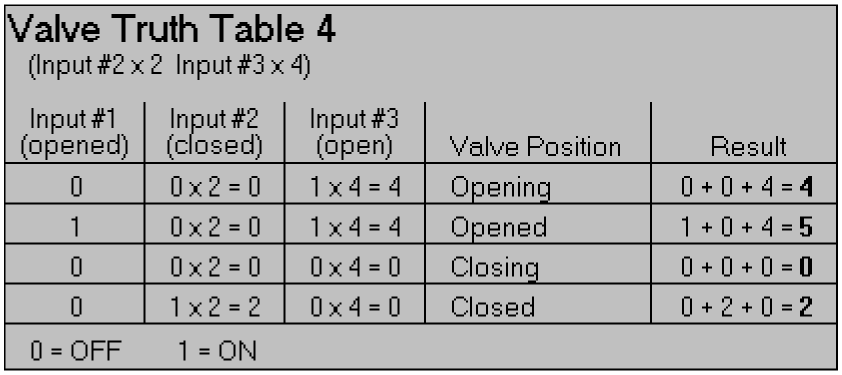


Valve Truth Table 3

Input #1 (opened)	Input #2 (closed)	Input #3 (open)	Valve Position	Result
0	0	1	Opening	0 + 0 + 1 = **1**
1	0	1	Opened	1 + 0 + 1 = **2**
0	0	0	Closing	0 + 0 + 0 = **0**
0	1	0	Closed	0 + 1 + 0 = **1**

0 = OFF   1 = ON

Once again the inputs are weighed. As previously explained, the closed input will be multiplied by 2 and the new input will be multiplied by 4 for the following results:

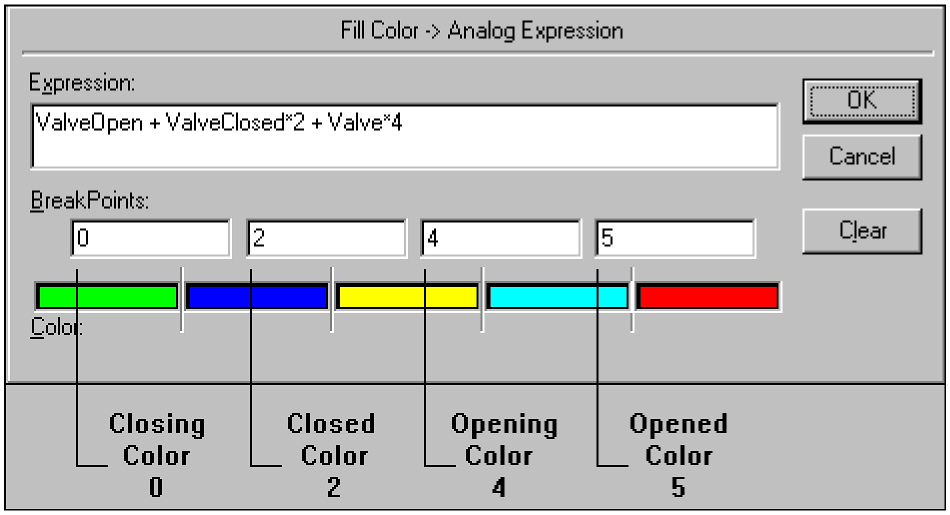


Valve Truth Table 4
(Input #2 x 2 Input #3 x 4)

Input #1 (opened)	Input #2 (closed)	Input #3 (open)	Valve Position	Result
0	0 x 2 = 0	1 x 4 = 4	Opening	0 + 0 + 4 = **4**
1	0 x 2 = 0	1 x 4 = 4	Opened	1 + 0 + 4 = **5**
0	0 x 2 = 0	0 x 4 = 0	Closing	0 + 0 + 0 = **0**
0	1 x 2 = 2	0 x 4 = 0	Closed	0 + 2 + 0 = **2**

0 = OFF 1 = ON

Another I/O Discrete tagname (**Valve**) is created for the new open input and assigned to a **Fill Color - Analog** animation link with the following properties:



Fill Color -> Analog Expression

Expression: ValveOpen + ValveClosed*2 + Valve*4

BreakPoints: 0, 2, 4, 5

Using this method, additional inputs can be used. The fourth input would be multiplied by 8, the fifth by 16, and so on.

Glossary of Terms

Accelerators	Accelerators are used by the application in creating a keyboard interface. They are normally offered as alternatives to using the menu for indicating choices. An accelerator is a keystroke that has special meaning to the application and that can be used to generate a command message.
Access	The obtaining of data. Locating desired data.
Active Application	The application that created the window that currently has the keyboard focus. Applications do not need to be the active application in order to receive and process messages. Applications are notified by message whenever they are gaining or losing the status of "the active application." The user normally determines the active application, but applications can override this decision.
ActiveX Control/Container	ActiveX controls, originally known as OLE controls or OCXs, are standalone software components that perform specific functions in a standard way. They define standard interfaces for reusable components. ActiveX controls are not separate applications. Instead, they are servers that are placed into a control container. To use ActiveX controls, they must to be placed in an ActiveX container. InTouch is an ActiveX container. VisualBasic and internet browsers are also ActiveX containers.
Alarm	A warning signal that is displayed or activated whenever a critical deviation from normal conditions occur.
Algorithm	A sequence of instructions which are mechanically carried out to perform a procedure.
Analog	Referring to the representation of numerical quantities by the measurement of continuous physical variables.

Application	A program or group of programs used for a particular kind of work, such as InTouch.
Argument	A variable to which either a logical or a numerical value may be assigned. Up to 16 arguments can be specified for an InTouch Quick Function. See **QuickFunctions**.
Assignment Operator	An operator used in an assignment statement that causes the value on the right to be placed into the variable on the left of the operator.
Assignment Statement	A programming language statement that gives a value to a variable, such as in $x = x + 1$ *or* $y = 6$.
Asterisk	A symbol (*) used to represent a multiplication operator in many programming languages.
Asynchronous	Pertaining to a mode of data communications that provides a variable time interval between characters during transmission. See **Synchronous transmission**.
b	An abbreviation for byte or baud. Use bits when referring to storage, or baud rate when referring to communications. Kb = 1000 bytes or baud (technically 1K = 1024 bytes). See Baud or Byte.
Background	In multiprogramming, the environment in which low priority programs are executed. Also, the part of a display screen not occupied with displayed characters or graphics (foreground).
Backing Up	The creation of a backup copy of a specified file or files, transferring them from either a floppy disk or a hard disk to another removable or fixed disk.
Baud Rate	A unit for measuring data transmission speed. One baud is 1 bit per second. Since a single character requires approximately 8 bits to represent it, divide the baud rate by 8 to calculate the characters per second (cps) to be transmitted. For example, 300 baud equals 37.5 cps, 1200 baud equals 150 cps, 2400 baud equals 300 cps.

Beta Testing	Pretesting of hardware and software products with selected "typical" users, to discover bugs before the product is released to the general public.
Binary	Pertaining to the number system with a radix of 2, or to a characteristic or property involving a choice or condition in which there are exactly two possibilities.
Binary Code	A coding system in which the encoding of any data is done through the use of bits--that is, 0 or 1.
Binary Coded Decimal (BCD)	A computer coding system in which each decimal digit is represented by a group of four binary **1**s and **0**s.
BIOS	An acronym for **B**asic **I**nput/**O**utput **S**ystem. In some operating systems, the part of the program that customizes it to a specific computer.
Bit	A binary digit; a digit (1 or 0) in the representation of a number in binary notation. The smallest unit of information recognized by a computer and its associated equipment. Several bits make up a byte, or a computed word.
Bitmap	A memory image of a portion of a display device surface. In Windows, a bitmap is actually a data structure containing a pointer to this memory image, plus information about the display device. The amount of memory required for a bitmap is device-specific, being dependent upon the color capabilities and pixel resolution of the device in question.
Boot	To start or restart a computer system by reading instructions from a storage device into the computer's memory. It involves loading part of the operating system into the computer's main memory. If the computer is already turned on, it's a "warm boot;" if not, it's a "cold boot."
Border	The line surrounding the current active window. A window can be resized by dragging on the border when the two-header arrow is present.

Buffer	An area of storage used to temporarily hold data being transferred from one device to another. Used to compensate for the different rates at which hardware devices process data; for example, a buffer would be used to hold data waiting to print, in order to free the CPU for other tasks, since it processes data at a much faster rate.
Bus	A channel or path for transferring data.
Button	Large rounded-rectangular or small round buttons which appear in dialog boxes. Click with the cursor arrow to select the button's option or command.
Byte	A grouping of adjacent binary digits operated on by the computer as a unit. The most common size byte contains 8 binary digits.
Check Box	A small square box which appears in a dialog box that can be turned on or off. Check boxes are generally associated with multiple options which can be set. To set a check box option, move to it and click the mouse button. When an **X** appears it is selected. When it is blank it is not.
Clipboard	A storage area for holding data (text, bitmap, graphic object, etc.) which is being copied or moved to another application or window.
Close	To remove an application's window and icon from the screen and free the memory used by the application. To close an application, choose the *Control/Close* command. Once an application is closed, it must be run to use it again.
Command	A word or phrase, usually found in a menu, that carries out an action.
Command Button	A round-cornered rectangle with a label on it that describes an action, such as **OK**, **Cancel** or **Close**. When chosen, the command button carries out the action.

Command Key	Any keyboard key used to perform separate functions.
Command Line	The string of arguments that follow any MS-DOS command, including the command to initiate an application program. The arguments in the command line are passed to the MS-DOS function or the program at startup time.
Computer Graphics	A general term meaning the appearance of pictures or diagrams, as distinct from letters and numbers, on the display screen or hard-copy output device.
Concatenate	To link together or join two or more character strings into a single character string, or to join one line of a display with the succeeding link.
CONFIG.SYS	An ASCII text file that MS-DOS processes when the system is turned on or restarted. It allows the user to configure certain aspects of the operating system, such as the number of internal disk buffers allocated, the number of files that can be open at one time, etc.
Control Name	A
Crop	In computer graphics, to cut off some part of an image.
CSV	Comma Separated Variable is the format used by the Clipboard for transfer of columns of text and numerical data between applications. A CSV data item is like text with each variable separated by commas. Although Microsoft Excel is probably the principle creator of CSV clipboard data, many DOS applications support this format.
Current File	The file that is running in the application.
Database	A collection of logically related records or files. A database consolidates many records into a common pool of data records which serves as a single central file.

Default	An option, command, or device that is automatically selected or chosen by the system. For example, one of the command buttons in a dialog box is already selected when the dialog box is opened. This indicates that it is the default value and will be chosen automatically if the **<Enter>** key is pressed. Default values are overridden by selecting another appropriate option, command, or device.
Device Driver	A program that controls how the computer interacts with a devices such as a printer, monitor, or mouse. A device driver enables the use of devices with the computer.
Dialog Box	A window that appears when Windows needs further information before it can carry out a command. For example, if the **Save** command on a **File** menu is selected a dialog box will appear asking for a name for the file to be saved under.
Directory	A structure for organizing files into convenient groups. A directory is like an address showing where files are located. A directory can contain files, or sub directories of files.
Discrete Value	A variable which only has two states: '1' (True, On) or '0' (False, Off).
Disk Operating System	(DOS) An operating system in which the operating system programs are stored on magnetic disks. Typically, it keeps track of files, saves and retrieves files, allocates storage space, and manages other control functions associated with disk storage.
Display	The physical representation of data on the screen.
Dithered	Intermingled dots of various colors which produce what appears to be a new color.
Document	A unit of printer output that must be printed contiguously; that is, no other output may be interspersed within a document. A document, then, is analogous to a report. The application must specify the start and end of each document.

DRA	**D**ynamic **R**esolution
DRC	**D**ynamic **R**esolution **C**onversion enables each View node to scale an application to a number of user-defined options, including a custom resolution. This scaling takes place while WindowViewer compiles the application, and does not require WindowMaker.
Drive	A letter in the range A-Z, followed by a colon (:), indicating a logical disk drive.
Dynamic Data Exchange	DDE is the passage of data between applications, accomplished without user involvement or monitoring. In the Windows environment, DDE is achieved through a set of message types, recommended procedures (protocols) for processing these message types, and some newly defined data types. By following the protocols, applications that were written independently of each other can pass data between themselves without involvement on the part of the user. For example, InTouch and Excel.
ENTER Key	The key on the keyboard which executes a statement or command. Same as RETURN key on some keyboards.
Events	Events are associated with ActiveX controls and occur through the ActiveX container. You can execute ActiveX control events in runtime (WindowViewer) by designing a particular action and associating it to the event by creating ActiveX Event Scripts. For example, **Control.click (shift)**. **FileViewer.DoubleClick (name)**. See **Properties** and **Methods**.
Expression	A general term for numerals, numerals with signs of operation, variables and combinations of these: 6, 3+6, n+10 are all expressions.
Extend	To select more than one item in a window. To extend a selection, hold down the SHIFT key until everything is selected.

Extension..	The period and three letters at the end of a filename. An extension identifies what kind of information a file contains. For example, an extension may be .EXE, .BAT indicate that a file contains an application.
FactorySuite..................................	Wonderware's software package that includes InTouch (and all its add-on programs and utilities), InControl and its I/O Server's, IndustrialSQL Server, numerous other I/O Server programs, Productivity Pack, NetDDE for Windows and NetDDE Extensions for Windows NT.
FactorySuite Plus	Enhanced FactorySuite software package that includes Wonderware's InTrack and InBatch products.
File..	A mechanism for holding and storing information on a hard disk or diskette for later use. File also may refer to any document or database created by the user, such as a word processing document, spreadsheet, etc. Each file appears in its own window and in most cases, the name of the file will appear in the title bar at the top of the window.
Filename...	Filenames consist of a base name containing no more than eight characters and a three-character extension. For example, INTOUCH.EXE
Format..	To prepare a disk so it can hold information. Formatting a disk erases any previously stored data. Format is the term used for an object rendition. In most Windows applications, available formats include Text, Bit map, etc.
Graphics Object	A visually oriented object, such as a scroll bar, bit map or icon that is used in the presentation of the visual interface. Graphic objects can be created by either the application or by Windows for use by the application.

Help..	Online instructions that explain how to use a Windows application. The Help menu displays specific Help topics.
Highlight..	Indicates that the object is selected and will be affected by the next action or command. A highlighted object appears in reverse video. A selected icon is outlined in white and displays the application's name.
Inactive	A window or icon that is not selected. See **Select**.
Insertion Point	The place where text will be inserted when the user types. The insertion point usually appears as a flashing vertical line (the cursor) and can appear in the workspace or within a dialog box. The text typed appears to the left of the insertion point, which is pushed to the right as text is entered.
Integer..	Any member of the set consisting of the positive and negative whole numbers and zero. Examples: -59, -3, 0.
I/O	An abbreviation for Input/Output.
Key Accelerator	A special keyboard sequence that executes menu commands. For example, Ctrl + A. See **Accelerators**.
List Box..	A box within a dialog box listing all available choices for a command. For example, a list of filenames on a disk. Usually an item is selected from the list box, then "OK" is chosen. If there are more choices than can fit in the list box, it will have vertical scroll bars. Selecting the down arrow next to the first item in the list will display the rest of the list box.
Local Variable	You can declare local variables within a script to store temporary results and create complex calculations with intermediate scripting values without impacting or decreasing your licensed tagname count and increase performance. Local variables or tagnames can be used interchangeably within the same script.

Macro	A single, symbolic programming-language statement that when translated results in a series of machine-language statements.
Maximize	To make a window or icon fill the entire screen. To maximize a window, choose the *Control/Maximize* command, or click on the Maximize box in the upper right corner of the window. See also **Minimize** and **Restore**.
MB	An abbreviation for megabyte. One million bytes. 1000KB.
Megabyte	1,048,576 bytes or 1024 kilobytes, actually; or roughly one million bytes or one thousand kilobytes.
Menu	Menus are group listings of available Windows and application commands. Menu titles appear in the menu bar at the top of the window. A command is chosen by displaying the menu, then choosing the desired command.
Menu Bar	The horizontal bar that lists the names of an application's menus. The menu bar appears below the title bar of a window Each Window's application has a menu bar that is distinct for that application, although some menus (and commands) are common to many of these applications.
Message Box	A special dialog box through which an application displays error messages or other important information. Message boxes alert the user when an error occurs or when the application needs information to complete an action or command.
Method	Methods are associated with ActiveX controls. They are similar to script function calls that you can call from the ActiveX container. For example, **Browser.Navigate("URLPageName")**, **Engine.start()**. See **Properties** and **Events**.
Millisecond	One thousandth of a second, abbreviated ms or msec.

Minimize..	To turn a window into an icon. To minimize a window, choose the *Control/Minimize* command, or click on the Minimize box in the upper right corner of the window. See **Maximize** and **Restore**.
Mirroring	The display or creation of graphics that portray an image in exactly the reverse of its original orientation. For example, flipping the graphic on its x- or y-axis.
Mode	A method or condition of operation.
Modulo...	A mathematical function that yields the remainder of division. A number *x* evaluated modulo *n* gives the integer remainder of *x/n*. For example, 200 modulo 47 equals the remainder of 200/47, or 12.
MS/DOS...	An abbreviation for **M**ICRO**S**OFT **D**ISK **O**PERATING **S**YSTEM, the standard operating system used by the IBM Personal Computer and compatible computers. Developed by Microsoft, Inc.
Multitasking	The ability of a computer to perform two or more functions (tasks) simultaneously.
NAD	Network Application Development or NAD is an architecture that combines the best of the client-based and server-based architectures. NAD provides automatic notification of application changes and automatic distribution of the updated applications to View nodes. NAD can even be used to automatically distribute master/slave applications.
Object	A set of data. Objects come in several formats; bitmap images, text, Real-time and Historical trend graphs, etc.
Off-line...	Pertaining to equipment or devices not in direct communication with the central processing unit of a computer. Equipment not connected to the computer.
Operand...	A quantity or data item that is operated upon.

Operating System	Software that controls the execution of computer programs and that may provide scheduling, debugging, input/output control, storage assignment, etc. Abbreviated OS.
Operator	In the description of a process, that which indicates the action to be performed on operands.
Option Button	A small round button appearing in a dialog box An option button is selected to set the option, but within a group of related option buttons, only one can be selected. An option button has a black dot when it is selected and is blank when it is not selected.
Option Button Group	A group of related options in a dialog box. Only one option button in a group can be selected at any one time.
Page	A page is a block of information that is selected and stored in a file. For example, a paragraph of text from Microsoft Word may be a page and a chart from Microsoft Excel may be another. Pages may be held in a variety of formats in the same file. Pages are numbered as they are placed into the file.
Palette	The set of available colors in a computer graphics system.
Parity	An extra bit added to a byte, character, or word to ensure that there is always either an even number or an odd number of bits, according to the logic of the system. If, through a hardware failure, a bit should be lost in transmission, its loss can be detected by checking the parity. The same bit pattern remains as long as the contents of the byte, character or work remain unchanged.
Paste	To insert something into a document or file from the Clipboard. Some applications (including InTouch) may have a Paste command that performs this task. If using some other standard application that runs in a window, Windows adds the Paste command to the Control menu.

Path	The hierarchy of files through which control passes to find a particular file. Designates one or more disk drives and/or directory paths to be searched sequentially for a program or batch file if the file cannot be found in the current or specified drive and directory. The drives and/or directory paths are searched in the order they appear in the Path .
Pathname	A description of the location of a directory or file within the system. The pathname consists of the drive letter, a colon (:), followed by directory and subdirectory names, followed by a filename. Each name is separated from the previous one by a backslash (\). If not specified, a default drive and directory are used.
Pixel	A picture cell. Shortened version of "picture element." The visual display screen is divided into rows and columns of tiny dots, squares or cells, each of which is a pixel. The smallest unit on the display screen grid that can be stored or displayed. A computed picture is typically composed of a rectangular array of pixels. The resolution of a picture is expressed by the number of pixels in the display. For example, a picture with 560 x 720 pixels is much sharper than a picture with 275 x 400 pixels.
Poke	An instruction used to place a value (poke) into a specific location in the computer's storage.
Polling	A communications control method used by some computer/terminal systems whereby a computer asks many devices attached to a common transmission medium, in turn, whether they have information to send.
Port	That portion of a computer through which a peripheral device may communicate. A connection between the CPU and a peripheral device.
Precedence	Rules that state which operators should be executed first in an expression.

Process Control	The use of the computer to control industrial processes such as oil refining and steel production.
Process Control Computer	A computer used in a process control system, generally limited in instruction capacity, word length and accuracy. Designed for continuous operation in non-air-conditioned facilities.
Processing	The application that currently has control of the processor. An application is given control of the processor upon receipt of a message. It retains control of the processor until the message is processed.
Properties	Properties are associated with ActiveX controls and can be associated with InTouch tagnames. The properties that you can configure for a particular ActiveX control are determined by the ActiveX control designer. Some properties are one-directional, meaning either the property sets the tagname's value, or the tagname's value sets the property. While other properties are bi-directional, meaning the value can be set from either the tagname or the property.
Protocol	Set of rules or conventions governing the exchange of information between computer systems or applications.
Queue	A group of items waiting to be acted upon by the computer. The arrangement of items determines the processing priority. For example, documents waiting to be printed.
QuickFunction	QuickFunctions are scripts that you can write and call from other scripts or expressions. You can use up to 16 arguments per QuickFunction. They are stored in the application in which they are created. QuickFunctions can be defined as asynchronous that, when executed, run in the background of the main WindowViewer (runtime) process.

QuickScript	Script created in InTouch. 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, and so on.
Register	A high-speed device used in a central processing unit for temporary storage of small amounts of data or intermittent results during processing.
Remote Tagname	A tagname that resides in a remote tag source but is referenced in a local InTouch application. Client applications can be designed without using any tagnames in the local Tagname Dictionary by using remote tagname references.
Restore	Icons can be restored to full-sized windows by double-clicking on them. To restore a window, choose the Restore command from the Control menu, or click on the Restore box in the upper right corner of the window. See **Maximize** and **Minimize**.
Run	To start an application. The Run command lets you specify parameters for the application. An application can also be run by double-clicking on its name or icon.
Running	An application that is "running" is an application that is in the system as a task, can receive messages, and is (normally) known to the user. From initiation to termination, an application is always running, but it is not always *processing*. See **Processing**.
Runtime	The time during which data is fetched by the control unit and actual processing is performed in the arithmetic-logic unit. Also, the time during which a program is executing.
Save	To store a file or changes to a file on a disk.
Scaling	The process of changing the size of an image.

Scroll	To move data or text up and down, or left and right to view parts of the file that cannot fit on the screen.
Scroll Bars	The bars that appear at the right side or bottom of a window. Use the scroll bars to move through a window that contains more information than can be shown on one screen. The scroll bar at the right side of a window scrolls vertically. The scroll bar at the bottom of a window scrolls horizontally.
Scroll Box	The small white box in the scroll bar. The scroll box reflects the position of the information within the window in relation to the total contents of the file. For example, if the scroll box is in the middle of the scroll bar, then the text or data in the window is in the middle of the file. The mouse can be used to scroll by dragging the scroll box in the scroll bar. See **Scroll Bars**.
Serial Port	An input/output port in a computer through which data is transmitted and received one bit at a time. In most cases, in personal computers, serial data is passed through an RS232C serial interface port.
Service	Special kind of program that is "privileged" and operates at a very low level within the system. Services run automatically in the background, and they do not require a user to log on. Because the Windows NT operating system is a secure operating system, normal programs are not allowed to access hardware directly, such as a hard disk, or other system objects, such as the Event Lot. Service programs can access the hardware and system objects for other normal programs. For example, the Wonderware Logger, WindowViewer can both be run as NT services.
Spreadsheet	A program that arranges data and formulas in a matrix of cells. For example, Microsoft Excel.
Statement	An expression of instruction in a computer language.

Standalone	A single, self-contained computer system, as opposed to a computer that is connected to and dependent upon a remote computer system. A standalone computer will operate by itself, requiring no other equipment.
String	A connected sequence of characters or bits treated as a single data item.
Subdirectory	Subdirectories are located within Directories. They are a structure for organizing files into convenient groups. A subdirectory is like an address showing where files are located.
SuperTag	InTouch supports a template structure that allows you to define composite tagname types called SuperTags. SuperTag templates can contain up to 64 member tagnames and 2 nesting levels. See **TemplateMaker**.
Synchronous transmission	Data transmission in which the bits are transmitted at a fixed rate. The transmitter and the receiver both use the same clock signals for synchronization. See **Asychronous**.
Syntax	The rules governing the structure of a language and its expressions.
Task	A task is an executing application. Task is a synonym for "process".
Tagname	The name assigned to a variable defined in the Tagname Dictionary (InTouch database).
TemplateMaker	InTouch utility that allows you to create SuperTag templates. See **SuperTag**.
Text Box	A box where information needed to carry out a command is typed. A text box usually appears in a dialog box.
Tiled Window	A tiled window is a window whose size, shape, and location on the screen is determined by Windows. Tiled windows are the only style of window that cannot overlap other windows, can be placed into the icon area and can have menus. Each application

	normally creates just one tiled window. All additional windows created by an application are normally cascading or popup windows.
Time slice	A unit of time.
Title Bar	The bar across the top of each window that contains the name of the application and the document or file being used by that application. (In **InTouch** an option exists to eliminate the Title bar.) Title bars are also used to move a window on the screen by grabbing it while dragging the mouse.
Toggle	Pertaining to any device having two stable states. Synonymous with Flip-Flop.
Touch-sensitive screen	A display screen on which the user can enter commands by pressing designated areas with a finger or other object.
Viewing Area	The viewing area (also called "Workspace") in Windows applications displays one page of a file. See **Workspace**.
Window	A rectangular area on the screen in which an application is viewed and worked. Multiple windows can be open on the screen at one time which can be sized and moved independently.
Windows	An operating environment developed by Microsoft.
Windows Application	An application that is designed especially for the Microsoft Windows operating environment and that uses all Windows features such as menus, scroll bars, and icons.
Workspace	The area of an application window that displays the application itself and all other open windows.
x-axis	On a coordinate plane, the horizontal axis.
y-axis	On a coordinate plane, the vertical axis.

Index