

InTouch® HMI Concepts and Capabilities Guide

Invensys Systems, Inc.

Revision A
Last Revision: 6/14/07



Copyright

© 2007 Invensys Systems, Inc. All Rights Reserved.

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

<http://www.wonderware.com>

For comments or suggestions about the product documentation, send an e-mail message to productdocs@wonderware.com.

Trademarks

All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DT Analyst, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A², InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, MaintenanceSuite, MuniSuite, QI Analyst, SCADAAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

Contents

	Welcome.....	7
	Documentation Conventions.....	7
	Technical Support	8
Chapter 1	InTouch HMI: Visualization for your Manufacturing Enterprise	9
	ArchestrA Technology	10
	Building Applications.....	12
	Running Applications.....	13
	Managing Applications	14
Chapter 2	Application Management.....	15
	Types of InTouch Applications	16
	Stand-Alone Applications.....	17
	Managed Applications.....	18
	Published Applications.....	19
	InTouchView Applications	20
	InTouch Application Creation	21
	Creating an Application with Application Manager	22
	Creating an Application with the ArchestrA IDE	23
	Application Server Architecture	24
	Network Application Development (NAD).....	25

Security	26
InTouch Security	27
Operating System-Based Security	28
ArchestrA-Based Security.....	29
Application Security	30
Importing and Exporting InTouch Applications	31
InTouch Application Conversion	32
Converting Older InTouch Applications	32
Converting InTouch Applications to Another Screen Resolution.....	33
Creating a Managed Application from a Stand-alone Application	34
Licensing.....	35
Chapter 3 Visualization: Windows and Graphics	37
InTouch Windows.....	38
InTouch Basic Objects.....	39
InTouch Symbols	40
ArchestrA Symbols.....	41
Pre-Defined InTouch Complex Objects	42
ActiveX Controls.....	42
Wizards	43
Historical Trends.....	44
Animation Links.....	45
Interacting with Animation Links	46
Exporting and Importing InTouch Windows	47
Importing and Exporting ArchestrA Symbols	48
Re-using SmartSymbols.....	49
Chapter 4 Data Management	51
About Tags and Their Properties	52
Tag Types.....	53
Data Types.....	54
User-Defined Tags and System Tags	55
Indirect Tags.....	56
SuperTags	57
Tag Properties	58
Tag Data Limits	59
Tag Deadbands	60
Tag Retention	61
Tag Dotfields.....	62

Remote Tags	63
Access Names	64
Managing Tags in an Application.....	65
Exporting and Importing InTouch Tags	66
I/O Failover.....	67
Historical Data	68
InTouch and the Wonderware Historian	69
Chapter 5 Alarms and Events.....	71
Types of Alarms	72
Discrete Alarms	72
Value Alarms	72
Deviation Alarms.....	72
Rate-of-Change Alarms.....	73
InTouch Distributed Alarm System	74
Alarm Queries	75
Saving Historical Alarm Data	76
Viewing Historical Alarms.....	77
Printing Historical Alarms	78
Managing Current Alarm Data.....	79
Viewing Current Alarms.....	80
Providing Alarm Redundancy.....	81
About InTouch Events.....	82
Chapter 6 Scripting and Logic.....	83
InTouch Script Language.....	85
Script Types	86
Predefined Script Functions	87
Custom Script Functions	88
Importing Scripts	89
Associating Scripts with ArcestrA Symbols.....	90
Using Methods in ArcestrA Symbol Scripts	91
Chapter 7 Applications at Run Time	93
Issuing Commands During Run Time.....	94
Enforcing Security in WindowViewer	95
Switching the Language of an Application	96
Running InTouch Applications at a Different Screen Resolution	97
Running InTouch Applications on Multiple Monitors.....	98

Running InTouch in a Terminal Services Environment .99
 Running a Managed InTouch Application with
 Terminal Services 100
Running InTouch on a Tablet PC..... 101
Running InTouch Windows over the Internet 102

Chapter 8 Supplementary Components 103

Recipe Manager 104
SQL Access Manager..... 105
SPCPro..... 106
16-Pen Trend 107
Symbol Factory..... 108

Index 109

Welcome

This documentation provides a conceptual overview of InTouch. It describes the principal components of InTouch and how you use them to build applications. Also, each section includes a cross reference to other InTouch books for more information about a topic.

This documentation assumes you know how to use Microsoft Windows, including navigating menus, moving from application to application, and moving objects on the screen. If you need help with these tasks, see the Microsoft online help.

You can view this document online or you can print it, in part or whole, by using the print feature in Adobe Reader.

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
Bold	Menus, commands, dialog box names, and dialog box options.
Monospace	Code samples and display text.

Technical Support

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

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

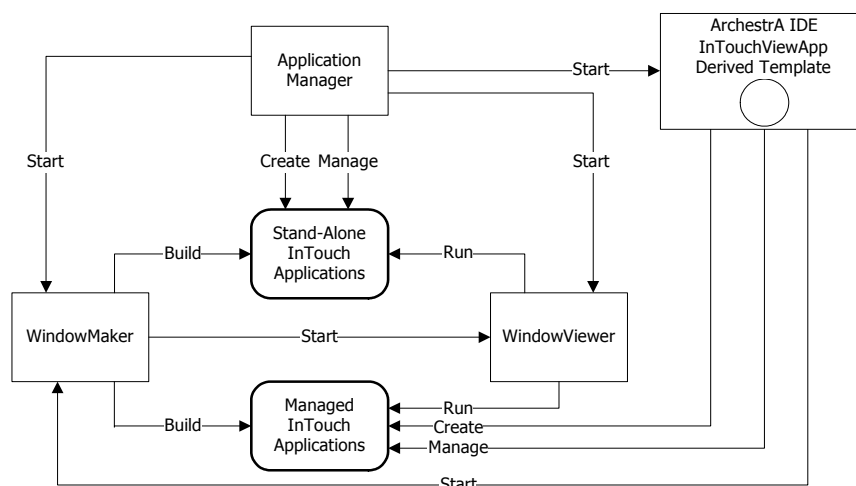
- The type and version of the operating system you are using.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

InTouch HMI: Visualization for your Manufacturing Enterprise

A Human Machine Interface (HMI) software application shows a graphical representation of a manufacturing environment. The tools, materials, and processes used to create a product appear as visual elements in an HMI application's windows. Plant operators interact with an application's graphical interface to monitor and administer the manufacturing processes.

The following figure shows the core components of the InTouch HMI that you use to build and run applications.



As its name suggests, you use Application Manager to create and manage InTouch applications. The application development environment, called WindowMaker, includes a set of graphic and other development tools to build your applications. You run your applications using WindowViewer.

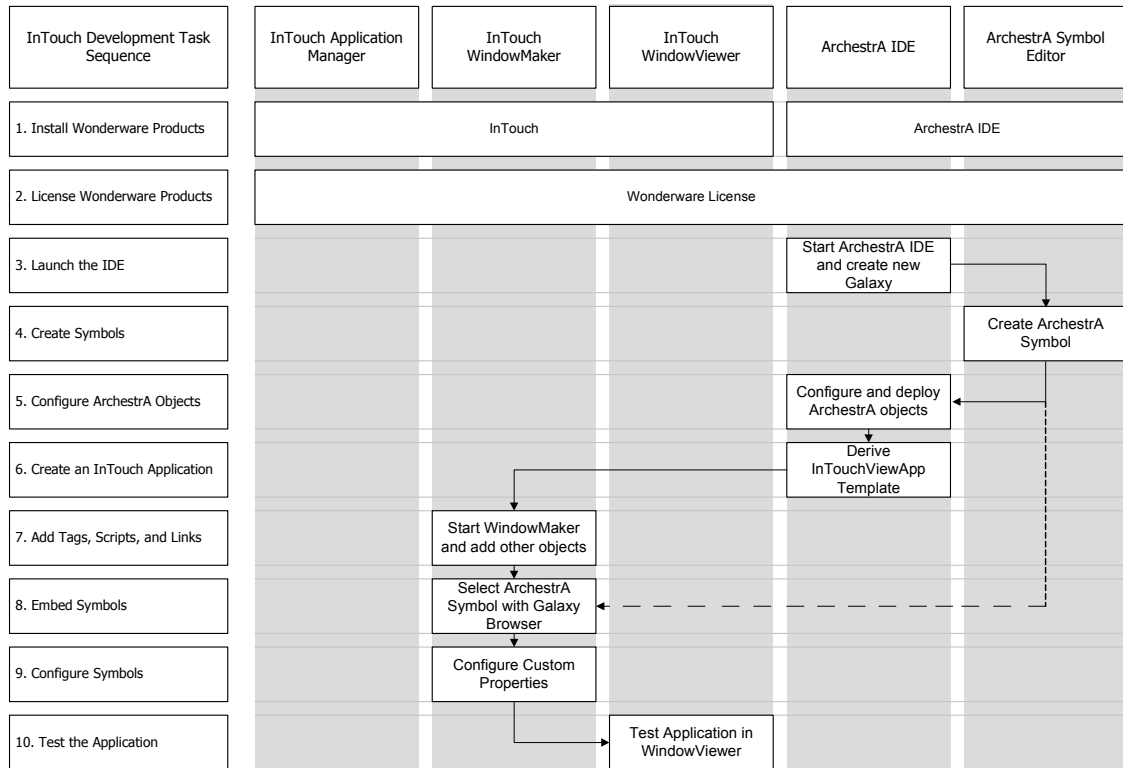
ArchestrA Technology

The InTouch HMI is tightly coupled with the ArchestrA Integrated Development Environment (IDE). When the ArchestrA IDE is installed along with the InTouch HMI, you can build applications that incorporate advanced ArchestrA graphics. If the Wonderware System Platform is also installed, you can integrate ArchestrA graphics into automation objects.

Instead of creating an InTouch application entirely with WindowMaker visualization elements, you can:

- Create an InTouch application with the ArchestrA InTouchViewApp object that is identified as *managed* in Application Manager.
- Develop visualization objects with the ArchestrA Symbol Editor, which is part of the ArchestrA IDE.
- Embed ArchestrA Symbols into InTouch application windows.

The following diagram shows the development workflow through the core InTouch components when you use the ArchestrA IDE and Symbol Editor to build an application.

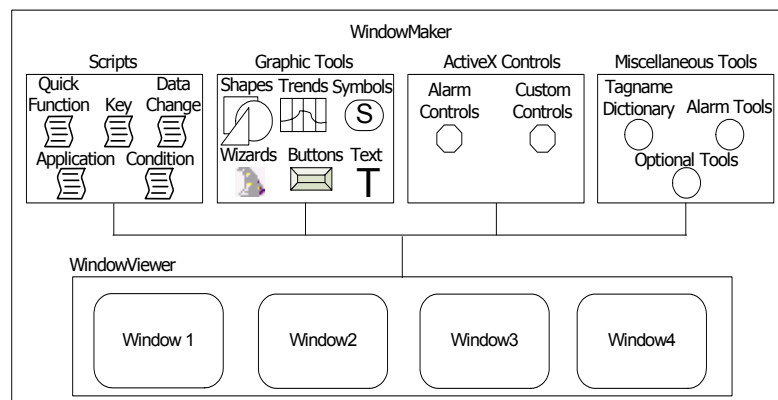


Building Applications

You use WindowMaker to create the visual interface for your HMI applications. WindowMaker provides graphic tools, a scripting language, and tag management utilities that define the behavior of objects that appear in your application's windows.

Using WindowMaker, you can create tags that represent data points associated with window objects. Data from a manufacturing process is ultimately associated as a tag value. This tag data can be used in your application for alarm monitoring, creating trends, and determining how the application behaves during run time.

The following figure shows some of WindowMaker's tools to create HMI applications.



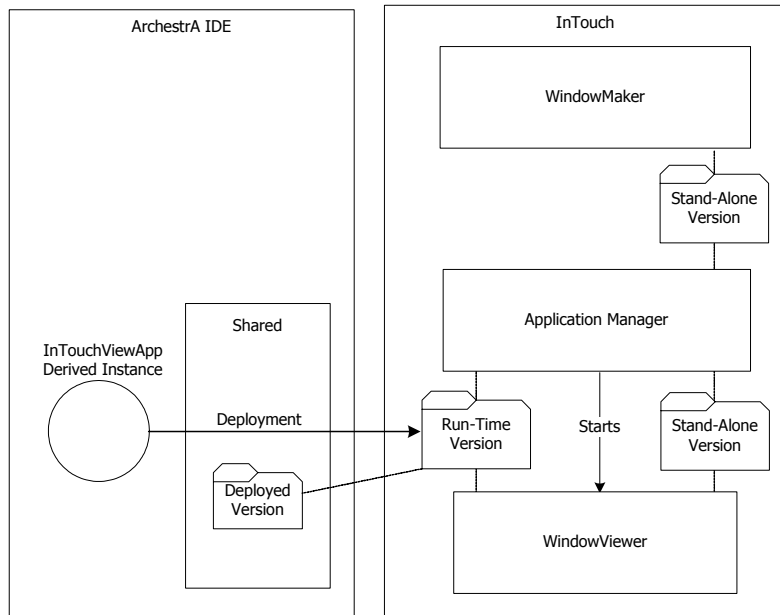
Key Points

- You can use a wide variety of graphic tools that range from simple shapes that can be combined to create more complex objects to standard symbols with predefined properties.
- You can create different types of scripts based upon their triggering mechanism. You can also insert predefined InTouch functions into your scripts.
- You can define a variety of tag value thresholds with the Tagname Dictionary that determines when a tag is in a normal or an alarmed state.
- After creating a managed application with the Archestra IDE, you can edit the application with WindowMaker by opening the InTouchViewApp editor.

For more information, see Chapter 3, Visualization: Windows and Graphics.

Running Applications

You use WindowViewer to run all types of InTouch applications. The following figure shows how WindowViewer loads managed and stand-alone applications into memory before running them.



After you deploy a managed application from the Arcestra IDE, you open it in WindowViewer from Application Manager.

Key Points

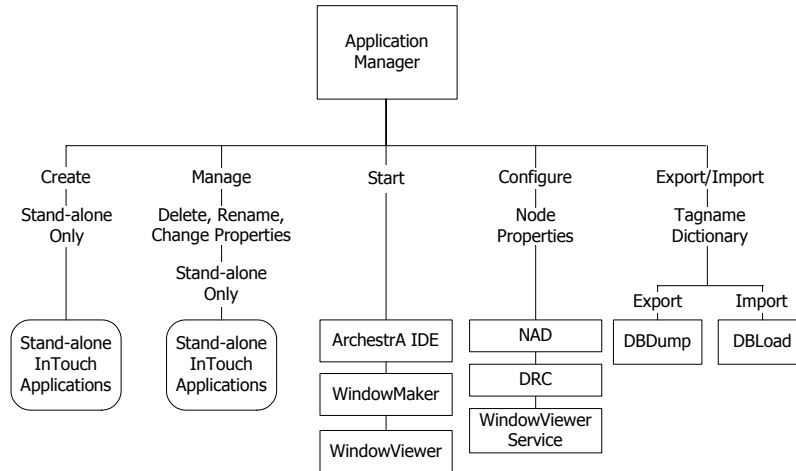
- You can use a wide variety of run-time triggers to start scripts while an application is running.
- You can configure WindowViewer to store application data and alarms in files or SQL Server databases.
- You can enforce security by requiring operators to log on to WindowViewer before using the InTouch application.
- You can configure the application to prevent operators from making any changes to the computer running WindowViewer.
- Operators can start and stop an application's historical logging by selecting WindowViewer menu commands.
- You can configure the computer that runs WindowViewer to act as a client or a server based upon how tag data is stored and distributed.

For more information, see Chapter 7, Applications at Run Time.

Managing Applications

You use Application Manager to manage your InTouch applications. After starting Application Manager, you see a list of existing InTouch applications. You select an application from the list and then use Application Manager commands or utilities to make your changes.

The following figure shows the utilities and commands available from Application Manager.



You can use Application Manager to create new applications or open existing applications in WindowMaker, WindowViewer, or the ArchestrA IDE.

Key Points

- You can create *stand-alone* applications with Application Manager that do not use ArchestrA symbols.
- You can delete, rename, or change the properties of stand-alone applications from Application Manager. You must use the ArchestrA IDE to delete or rename managed applications.
- You start WindowViewer from Application Manager to run both managed and stand-alone applications.
- You use Application Manager to configure general InTouch properties that apply to all applications.
- You can export the contents of an application's Tagname Dictionary with Application Manager's DBDump utility. You use the corresponding DBLoad utility to import tag definitions to an application's Tagname Dictionary.

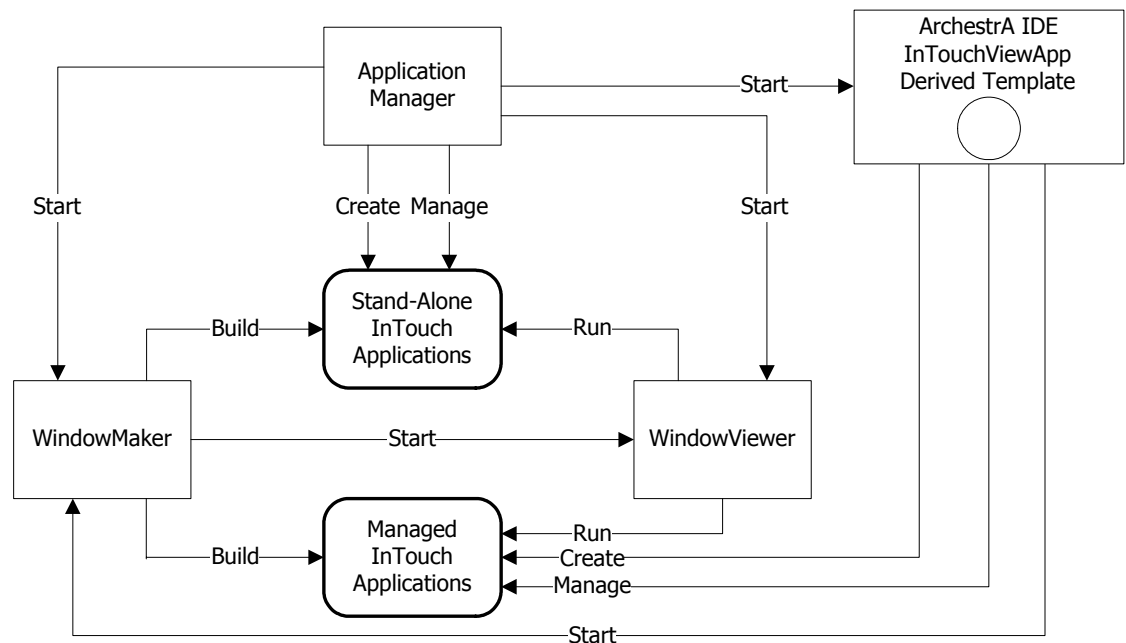
For more information, see Application Management on page 15.

Chapter 2

Application Management

After you create an application, you need to periodically manage your InTouch environment. You manage applications with Application Manager or the ArchestrA IDE, based upon how the applications were created.

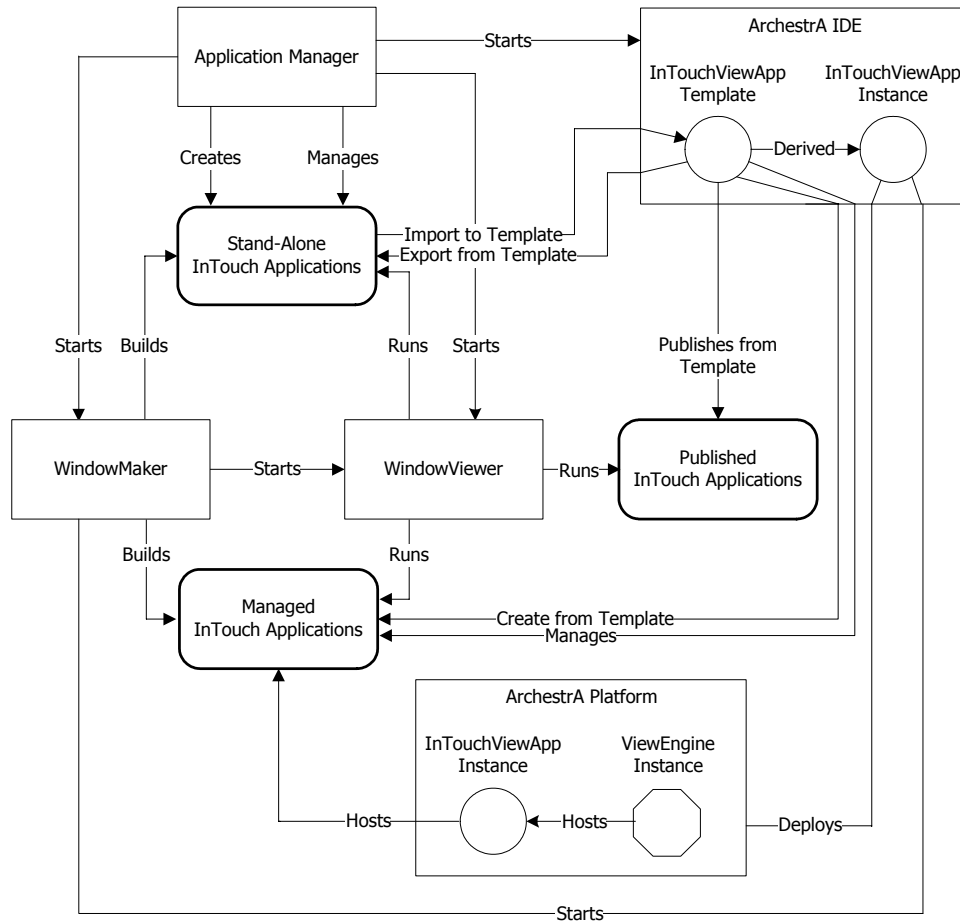
The following figure shows how InTouch applications are created and then administered with either Application Manager or the ArchestrA IDE.



Stand-alone applications are created and managed by Application Manager. Managed applications are created and managed by the ArchestrA IDE.

Types of InTouch Applications

The following figure shows a detailed view of how various types of InTouch applications are created and managed by both the ArchestrA IDE and Application Manager.

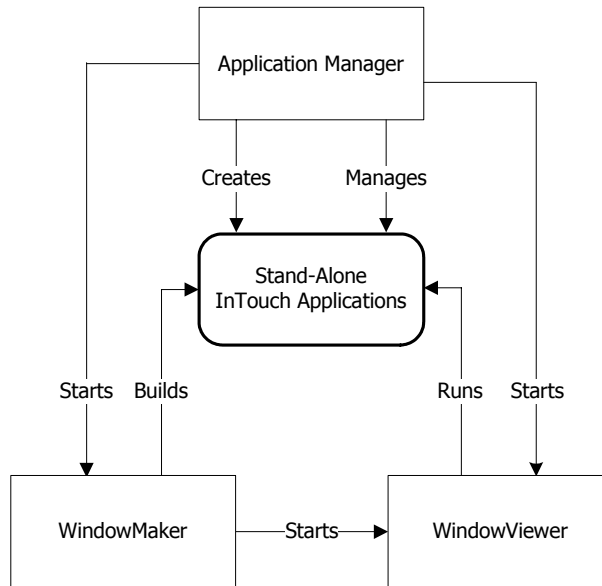


- A stand-alone application is created with the same InTouch components used in versions earlier than 10.0. For more information, see [Stand-Alone Applications](#) on page 17.
- A managed application is created and managed with the ArchestrA IDE. For more information, see [Managed Applications](#) on page 18.
- You can create a published application by exporting a managed application from the derived InTouchViewApp template. For more information, see [Published Applications](#) on page 19.

Not shown in the figure above is an InTouchView application, which can be created either from Application Manager or the ArchestrA IDE. For more information, see [InTouchView Applications](#) on page 20.

Stand-Alone Applications

A stand-alone application uses the same InTouch components as InTouch versions earlier than 10.0. The following figure shows the InTouch components used to create, build, and run a stand-alone application.



Key Points

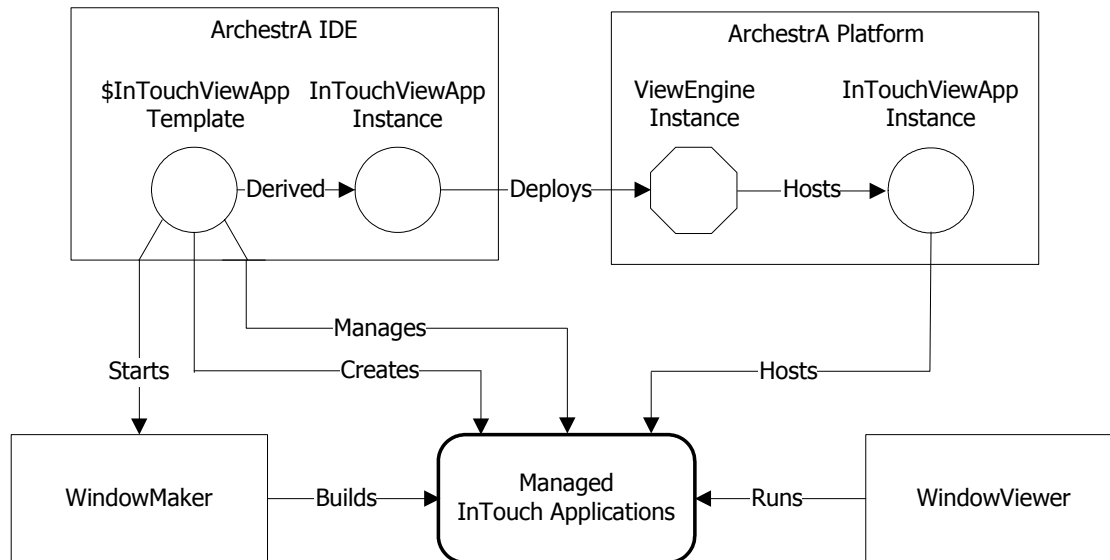
- A stand-alone application is created and managed by Application Manager. A stand-alone application consists of a set of files maintained by the InTouch HMI in the directory file system.
- A stand-alone application is built entirely with WindowMaker and run with WindowViewer.
- A stand-alone application does not interact with the ArchestrA IDE, nor does it contain any ArchestrA graphics.
- A stand-alone application can be deployed across multiple network nodes and is not restricted to a single node.
- A stand-alone application can be imported to the ArchestrA IDE and converted to a managed application.

For more information about creating and managing stand-alone applications, see Chapter 1, Managing InTouch Applications, in the *InTouch® HMI Application Management and Extension Guide*.

For more information about building stand-alone applications, see Chapter 1, WindowMaker: Your Development Environment, in the *InTouch® HMI Visualization Guide*.

Managed Applications

In addition to the same InTouch components used to create and manage a stand-alone application, a managed application uses the ArcestrA IDE to provide advanced graphics capability. The following figure shows the InTouch components used to create, build, and run an application.



Key Points

- Each managed application is associated with an ArcestrA InTouchViewApp object, which is derived from a base template.
- The ArcestrA IDE includes the Symbol Editor, which you can use to create symbols that represent production processes in your InTouch application.
- A managed application appears in Application Manager as **Managed** and can be edited only by starting WindowMaker from the ArcestrA IDE.
- From Application Manager, you can only start a managed application in WindowViewer. When WindowViewer starts, it copies a managed application's files to a folder during run time.
- You can fast switch between WindowMaker and WindowViewer to test a managed application.

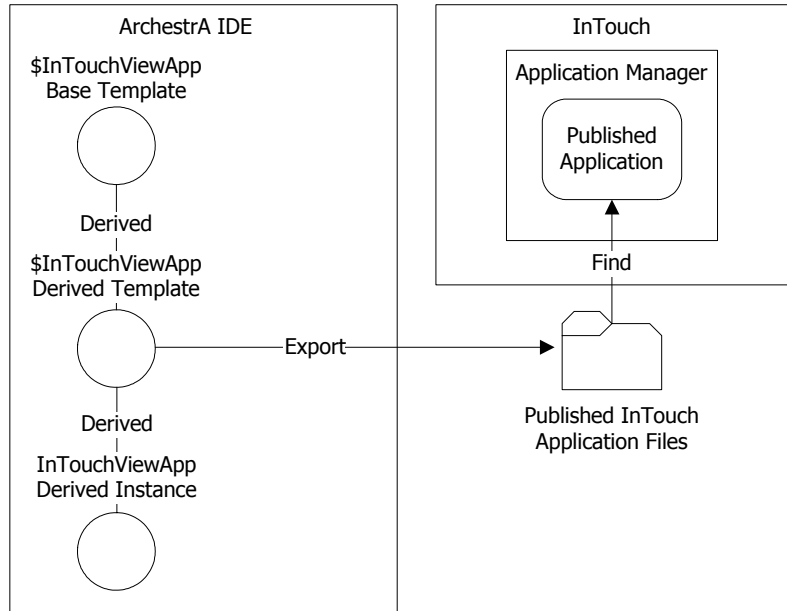
For more information about creating managed applications, see Chapter 1, About InTouch and ArcestrA Integration, in the *InTouch® HMI and ArcestrA® Integration Guide*.

For more information about building managed applications, see the Application Server IDE documentation.

Published Applications

You can publish a managed application from the derived InTouchViewApp template.

When you publish a managed application, a user-defined folder is created containing InTouch application files and any ArchestrA symbols embedded in the application. You use Application Manager's **Find** utility to locate the folder. Thereafter, the converted application appears in Application Manager as a published application.



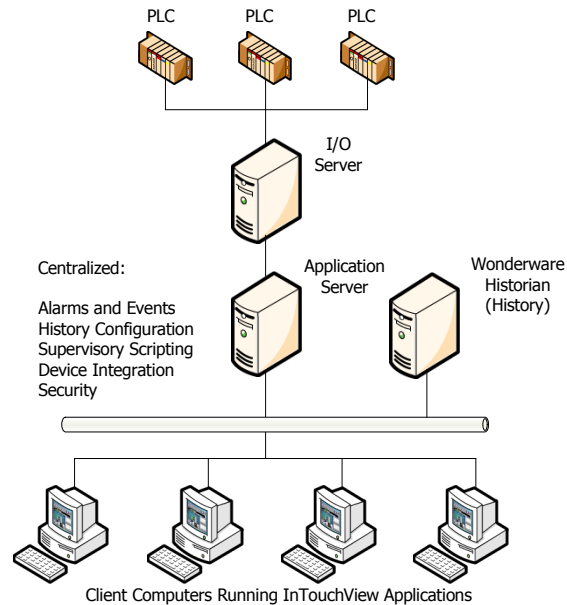
Key Points

- After you publish a managed application, you cannot use the ArchestrA Symbol Editor to edit the ArchestrA symbols embedded in the published application.
- After you publish a managed application, you can still use embedded ArchestrA symbols to write data to a Galaxy or visualize data.
- After you publish a managed application, you can use WindowMaker's graphic tools to modify some basic characteristics of an embedded ArchestrA symbol, such as size, orientation, and screen position.
- A published application cannot be imported again into a Galaxy.

For more information, see Chapter 2, Managing InTouch Applications with IDE, in the *InTouch® HMI and ArchestrA® Integration Guide*.

InTouchView Applications

You can use an InTouchView application as the visual interface for applications designed specifically for a Wonderware Application Server environment. InTouchView applications run in WindowViewer with Application Server providing most of the HMI functionality.



Key Points

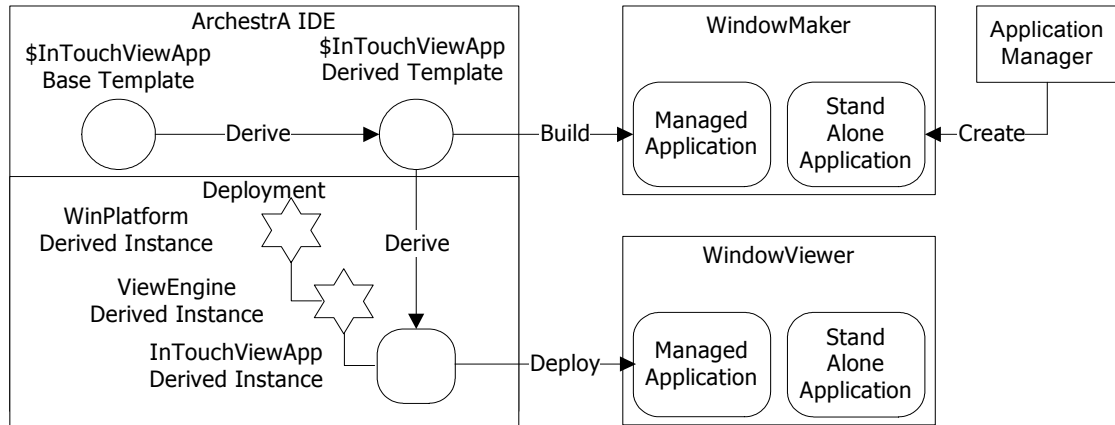
- You create an InTouchView application by selecting an option from Application Manager or the ArchestrA IDE.
- You build an InTouchView application with WindowMaker.
- An InTouchView application offers only some of the standard functions that are available with a full-featured InTouch application. An InTouchView application:
 - Cannot connect to I/O sources other than the ArchestrA Application Server Galaxy.
 - Cannot reference tags within embedded ArchestrA symbols.
 - Can be secured only with ArchestrA security.
 - Cannot generate alarms.
 - Does not log application data or events.
- You can convert a standard InTouch application to an InTouchView application. Likewise, you can convert an InTouchView application to an InTouch application.

For more information, see Chapter 7, Viewing Applications at Run Time, in the *InTouch® HMI Application Management and Extension Guide*.

InTouch Application Creation

If you only have the InTouch HMI installed, you create a new application with the Application Manager. Starting with InTouch 10.0, you can also create an application with the ArchestrA IDE.

The following figure shows how InTouch applications are created with both the Application Manager and the ArchestrA IDE.



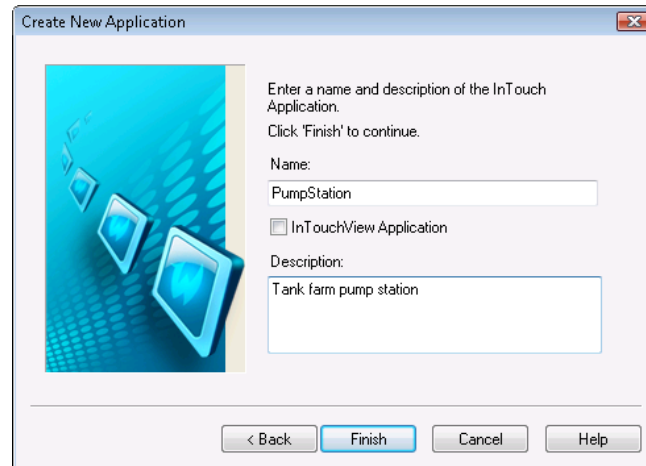
Key Points

- The major steps to creating a managed application are creating a derived template from the \$InTouchViewApp base template and then deriving an instance from the derived template.
- At a minimum, you must create instances of the ArchestrA WinPlatform and ViewEngine Galaxy objects to support your managed application.
- After you create a managed application, its files are placed in a shared folder on the computer hosting the ArchestrA IDE.
- After you deploy an instance of InTouchViewApp, the managed application is listed from Application Manager with the timestamp of its last deployment.
- You have the option of creating an InTouchView application with either the Application Manager or the ArchestrA IDE.

For more information, see Chapter 2, Managing InTouch Applications with IDE, in the *InTouch® HMI and ArchestrA® Integration Guide*.

Creating an Application with Application Manager

You create a stand-alone application with Application Manager. After selecting the **New** command from Application Manager's **File** menu, you provide information about your new stand-alone application.



You must specify the following information to create a stand-alone application:

- Folder path to save the application's folder
- Folder name to place InTouch application files
- Name of the application and an optional comment

Key Points

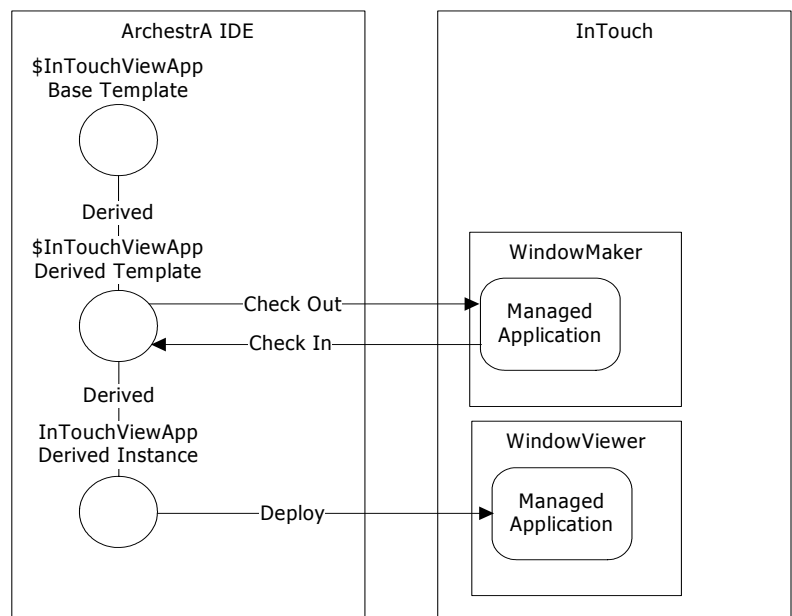
- After creating a stand-alone application, you can start the application in WindowMaker or WindowViewer from Application Manager.
- You can select the InTouchView **Application** option from Application Manager to create an InTouchView application. The ArchestrA IDE includes a similar InTouchView **Application** option.
- The screen resolution of the stand-alone application is defined by values set from the **Node Properties** dialog box of Application Manager.
- You can specify NAD support for a stand-alone application by setting options from the **Node Properties** dialog box of Application Manager.

For more information, see Chapter 1, Managing InTouch Applications, in the *InTouch® HMI Application Management and Extension Guide*.

Creating an Application with the ArcestrA IDE

ArcestrA manages your InTouch applications with an ArcestrA object called the InTouchViewApp object. The InTouchViewApp object does not contain application files. Instead, the InTouchViewApp object only contains a reference to the managed InTouch application folder and other behavior-specific information of the managed InTouch application.

Application files are stored in separate folders in the ArcestrA file repository. One folder contains the most recent checked-in version of the InTouch application and the other contains the most recent checked-out version.



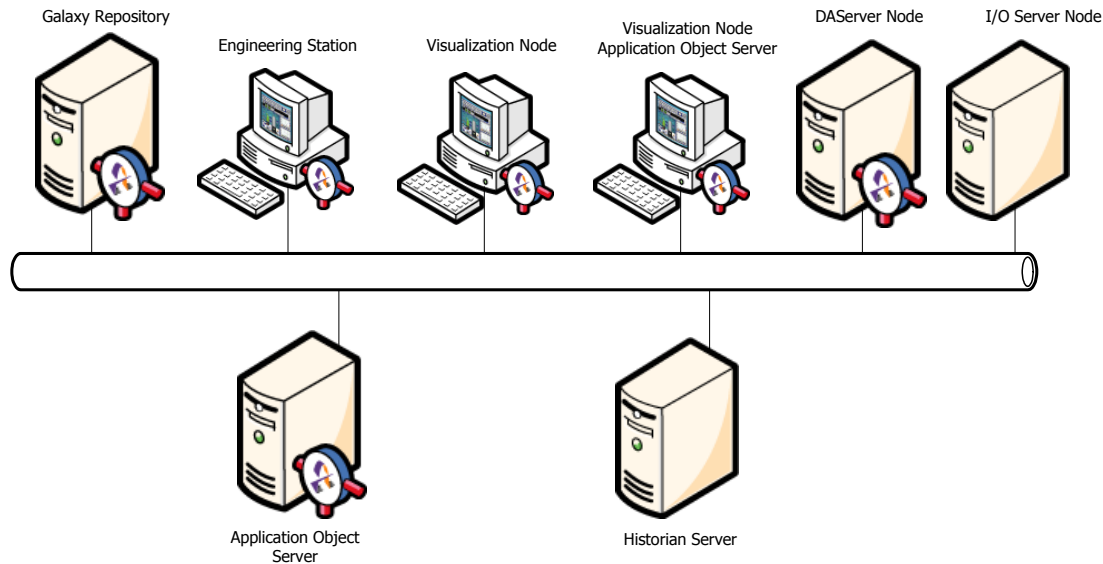
Key Points

- You create a derived InTouchViewApp template for each managed application.
- You build a managed application by checking out the derived InTouchViewApp template from the ArcestrA IDE to edit with WindowMaker. You check in the template to save your work.
- When you deploy the InTouchViewApp instance to a target node to run in WindowViewer, application files are stored in a folder on the ArcestrA IDE node and a folder on the computer where the InTouch HMI is installed.

For more information, see Chapter 1, About InTouch and ArcestrA Integration, and Chapter 2, Managing InTouch Applications with IDE, in the *InTouch® HMI and ArcestrA® Integration Guide*.

Application Server Architecture

Application Server relies on ArcestrA technology to provide services for distributed InTouch applications. Application Server services are distributed across a set of nodes with the InTouch HMI installed on the Visualization Node to provide the application interface.



Key Points

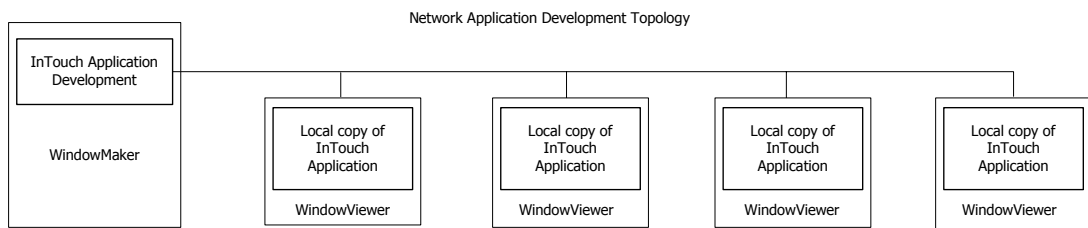
- An Application Server Galaxy represents the system's configuration based on a single logical namespace that incorporates a variety of components that are distributed across multiple nodes.
- Application Server can store production data similar to InTouch tags, but it supports more data types and arrays.
- Application Server can provide alarm capabilities that are compatible with the InTouch alarm system.
- Application Server provides an InTouch HMI compatible scripting language with significant enhancements for .NET function support.
- Application Server objects can also contain ArcestrA graphical symbols, which can be used in InTouch applications.
- To be able to read and write Galaxy data, you need to install the Application Server Bootstrap component on the InTouch node. To be able to browse the Galaxy, you need to install the Application Server IDE component.

For more information, see the Wonderware Application Server documentation.

Network Application Development (NAD)

You can use NAD to build and distribute stand-alone applications. NAD allows many client stations to maintain a copy of a single application without restricting the development of that application. Client stations can be notified when an InTouch application changes.

In a typical NAD network topology, you maintain a master copy of an application on a central network server, which is usually the development node running WindowMaker. Each View node copies the application to a user-defined location and runs it with WindowViewer. Using an individual copy of the application provides Viewer redundancy.



Key Points

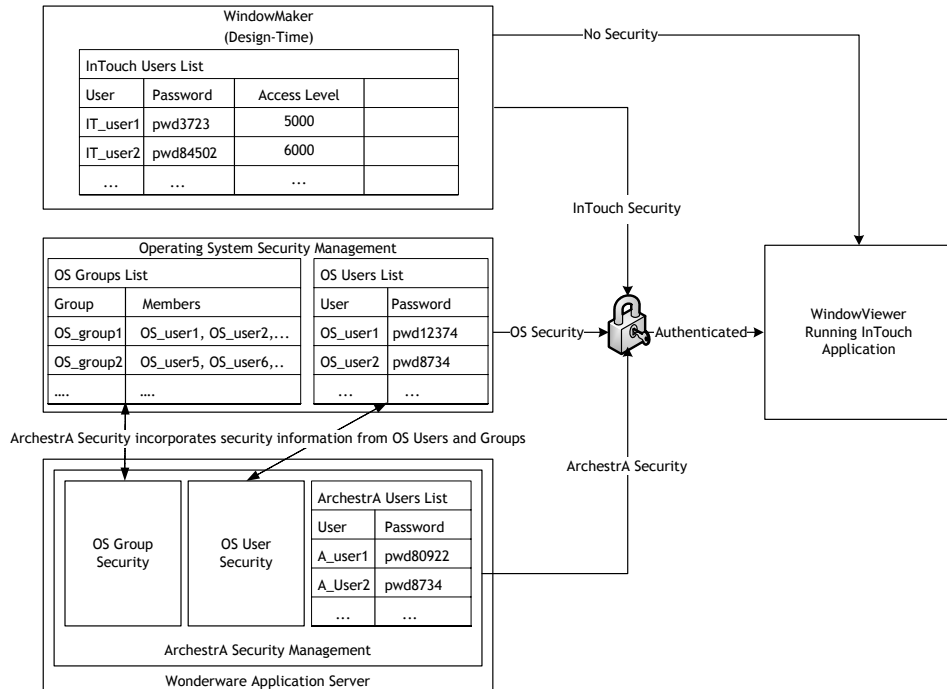
- When a View node copies and runs an InTouch application, it automatically monitors the status of the master copy on the development node. A flag in the master application folder indicates when the master copy has been updated. You can set this flag manually from WindowMaker by issuing the **Notify Clients** command.
- You can configure how you want application changes handled by the View nodes from Application Manager. After you enable NAD, you can set these change modes:
 - Ignore application updates.
 - Automatically restart WindowViewer and load the updated application.
 - Prompt the user to restart WindowViewer and load the updated application.
 - Load application changes into WindowViewer.
 - Prompt the user to load application changes into WindowViewer.
- You can use the \$ApplicationChanged system tag in a data change script to build a node update notification script. You can also include the RestartWindowViewer() function in a script that initiates the shutdown process.

For more information, see Chapter 2, Distributing Applications, in the *InTouch® HMI Application Management and Extension Guide*.

Security

You can manage access to WindowMaker and WindowViewer by using security, security-related tags, and security-related QuickScripts.

InTouch supports three types of security:



- **InTouch security:** InTouch defines and manages the users that are allowed to access InTouch.
- **Operating system-based security:** The users that are allowed to access InTouch are defined and managed in the User Manager of the Windows operating system.
- **ArchestrA-based security:** The users that are allowed to access InTouch are defined and managed in the Application Server IDE.

All types of InTouch security authenticate users during the logon process with a user name and password combination. Each type of security provides a different mechanism to verify the user name and password during the authentication process.

Applying security to your application is optional. By default, an InTouch application is not secured. However, you can restrict which functions an operator is allowed to perform by linking those functions to internal tags. In addition, when you establish security on your application, audit trails can be created that associate alarms and events to the operator logged on to the InTouch HMI.

InTouch Security

InTouch security is a two-step process of first determining if the person attempting to use an application is recognized as a valid user. The second step determines what InTouch privileges have been granted to an authenticated user.

Key Points

- Security is based on operators authenticating themselves by entering a user name and password to log on to an InTouch application.
- You must assign an account name, password, and access level (0-9999) to each operator who will interact with an InTouch application.
- After an operator logs on to the application, access to any protected function is granted upon verification of the operator's password and access level against the value specified for the internal security tag linked to the function.
- You can also link a User Input - Discrete button to the \$ConfigureUsers tag to allow an authorized operator with an access level of equal to or greater than 9000 to access the **Configure Users** dialog box to edit the security user name list.
- The \$AccessLevel and \$Operator system tags contain information of the currently logged-on user. If you want to restrict access to objects at run time, you can use these system tags in an animation link to prevent access.

For more information, see Chapter 5, Securing InTouch, in the *InTouch® HMI Application Management and Extension Guide*.

Operating System-Based Security

Operating system-based security authenticates InTouch users based upon their personal Windows user accounts. Operating system-based security authorizes user access to InTouch functions based upon membership in one or more authorized Windows user groups. You create Windows user groups either on the local computer or an Active Directory server. You must add your anticipated InTouch users as members to these groups.

An operating system-based authentication method inherits enforcement of some account policies from the Windows operating system. For example, maximum password age and minimum password length are enforced by the Windows operating system. Other security policies are still enforced from the InTouch HMI.

Key Points

- Each Windows user account has an associated access level that determines what InTouch functions are accessible to users.
- The Windows domain must be set up with the desired account policies to enforce authorization to InTouch functions.
- You assign InTouch access levels to the Windows groups using the `AddPermission()` function in a script. The `AddPermission()` function is typically called on application start-up so WindowViewer recognizes all the authorized user groups when an operator is ready to log on.
- The operating system manages passwords internally. The InTouch HMI does not store passwords on the node hosting the application.
- 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 tags.

For more information, see Chapter 5, *Securing InTouch*, in the *InTouch® HMI Application Management and Extension Guide*.

ArchestrA-Based Security

The ArchestrA security system is a global function that applies to every object in the Galaxy database. It is a relationship-based security system between users and the objects and functions of the Galaxy.

ArchestrA security is designed to allow system administrators to easily define users and assign the operations they are allowed to perform. The security permissions are defined in terms of the operations the users can perform using automation objects.

Key Points

- ArchestrA-based security is based on security roles (configuration, system administration, and run-time permissions) and security groups, which determine a particular security role's run-time permissions on an object-level basis.
- When you configure a node to use ArchestrA security, the InTouch HMI uses methods and dialog boxes from Application Server for logon and logoff operations. Users are configured on the Application Server IDE.
- ArchestrA-based security includes advanced security mechanisms that also affect InTouch, such as:
 - Secured Write: writing data to a Galaxy attribute from InTouch requires operators to re-enter their passwords to complete the writeback.
 - Verified Write: writing data to a Galaxy attribute from InTouch requires operators to re-enter their passwords and also authorization of a second operator to complete the writeback.
- You cannot use ArchestrA security if the InTouch application only uses the Galaxy as a provider of ArchestrA symbols. To use ArchestrA security, you must deploy a WinPlatform to the node on which the deployed managed application is running.

For more information, see Chapter 5, Securing InTouch, in the *InTouch® HMI Application Management and Extension Guide*.

Application Security

You can prevent operators from closing an InTouch application in WindowViewer or switching to another application. To protect your InTouch application while it is running, you can:

- Set an inactivity time-out period.
- Disable keys on the computer running WindowViewer.
- Hide WindowViewer menus and commands.

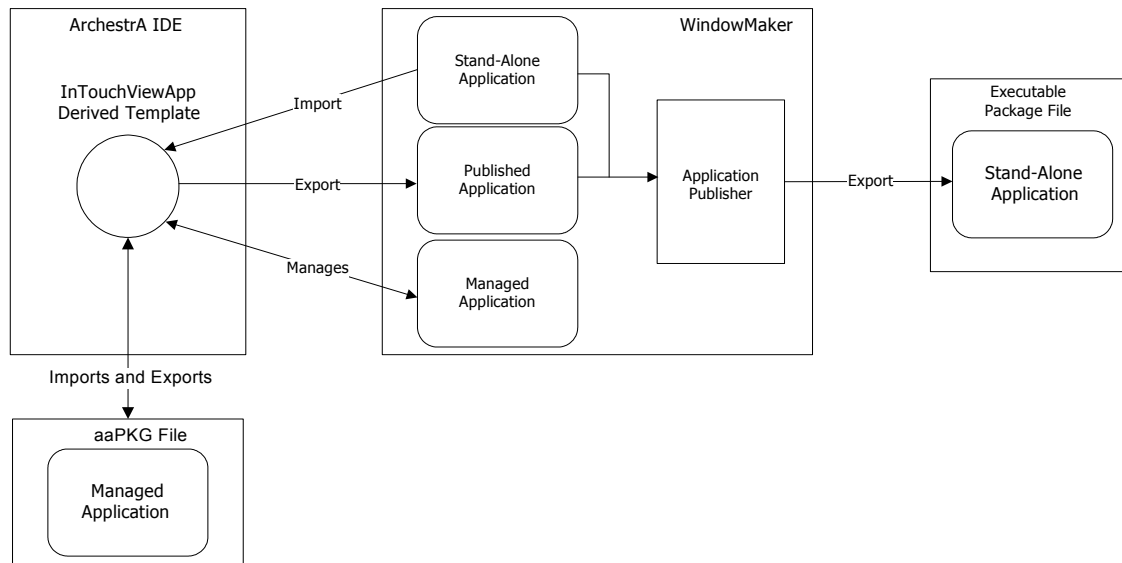
Key Points

- You can use scripting to log off an inactive operator from an InTouch application. This prevents unauthorized access to your InTouch application when operators leave their workstations unattended.
- You can restrict operator access to standard Windows functions by disabling system keys on the computer running an InTouch application. Disabling system keys prevents operators from switching from the InTouch HMI to another application using standard Windows key combinations.
- You can write a script that enables or disables system keys based on the access level of the person logging on to WindowViewer. Use the `EnableDisableKeys()` function in a script to selectively enable or disable Windows function keys.
- You can restrict operator access to WindowViewer menus and commands by hiding them while an InTouch application is running. The **WindowViewer Properties** dialog box includes options to hide WindowViewer menus, commands, and window controls.

For more information, see Chapter 5, *Securing InTouch*, in the *InTouch® HMI Application Management and Extension Guide*.

Importing and Exporting InTouch Applications

You can export and import entire InTouch applications from both the ArchestrA IDE and WindowMaker.



Key Points

- You can use Application Publisher from WindowMaker, to create a compressed, self-extracting package file that contains all relevant files and procedures to install an InTouch application on another computer.
- You can import a stand-alone application to use as a managed application by creating a derived InTouchViewApp template and then associating it with the application you want to import.
- You can publish a managed application by exporting the application associated with the InTouchViewApp object. A folder is created containing the application and any ArchestrA symbols.
- You can export the InTouchViewApp object, which creates a package file containing the managed application associated with the InTouchViewApp object. You can then import the application into another Galaxy.

For information about publishing applications, see Chapter 1, *Managing InTouch Applications*, in the *InTouch® HMI Application Management and Extension Guide*.

Also, see Chapter 2, *Managing InTouch Applications with the IDE*, in the *InTouch® HMI and ArchestrA® Integration Guide*.

InTouch Application Conversion

After you upgrade to the current version of the InTouch HMI, you typically need to complete several tasks from Application Manager to migrate your existing applications:

- You must convert your existing applications that were developed with an earlier version of the InTouch HMI to the current version.

You may need to change the screen resolution selected to run your applications if it is different than the screen resolution of the computer running WindowViewer.

Converting Older InTouch Applications

You can still run your older applications by first converting them to the current version of the InTouch HMI. After an older application is converted, you can modify it with WindowMaker or run it in WindowViewer.

Key Points

- You can convert applications developed with InTouch version 7.11 and later to the current version.
- You cannot open an older application in WindowViewer until you have first converted it to the current version using WindowMaker.
- You cannot open newer InTouch applications with an older version of WindowMaker or WindowViewer.
- You cannot convert a newer InTouch application to an older version.
- You can create a backup copy of your original application when you convert it to the current version of the InTouch HMI.

For more information, see Chapter 1, *Managing InTouch Applications*, in the *InTouch® HMI Application Management and Extension Guide*.

Converting InTouch Applications to Another Screen Resolution

You can set the Application Manager's Dynamic Resolution option that determines whether an application's screen resolution can be changed when it runs in WindowViewer.

Key Points

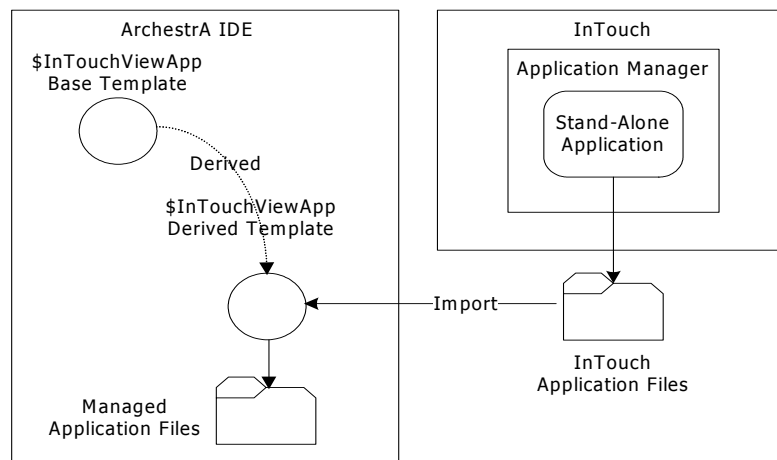
- If you do not use Dynamic Resolution, then you must first convert your InTouch applications to the resolution of the computer running WindowViewer.
- If you select Dynamic Resolution, you can select from several options to run your InTouch applications at different screen resolutions. You can select options to:
 - Run the application with its original screen resolution.
 - Convert the application's resolution to the resolution of the computer that hosts WindowViewer.
 - Customize the application's screen resolution by setting the horizontal and vertical dimensions of the screen in pixels.

For more information, see Chapter 2, Distributing Applications, in the *InTouch® HMI Application Management and Extension Guide*.

Creating a Managed Application from a Stand-alone Application

You can convert a stand-alone application to a managed application in a two-step process. First, you create a derived InTouchViewApp template to associate with the imported stand-alone application. Then, you open the derived template and import the stand-alone application.

You can also import and export an InTouchViewApp derived template. A derived template contains all information for hosting a managed application. You can move a managed application across Galaxies by exporting and then importing the derived InTouchViewApp template associated with the managed application.



Key Points

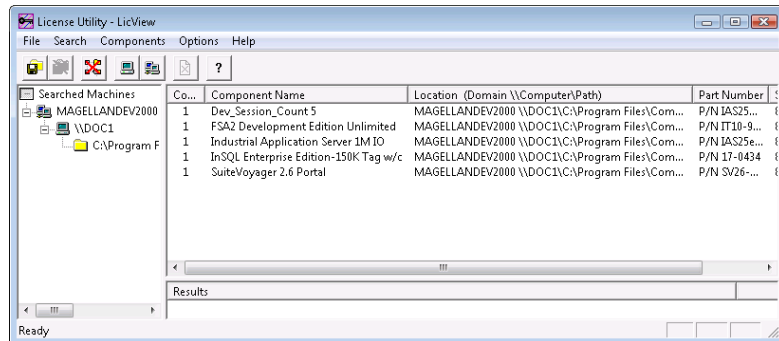
- The Archestra IDE creates a copy of the stand-alone application in a folder managed by the Archestra IDE. The original stand-alone application files are unchanged.
- After you deploy the InTouchViewApp instance, the imported application appears as a managed application in Application Manager.
- The original stand-alone application is unchanged and still appears in Application Manager after you create a managed application from it.

For more information, see Chapter 2, *Managing InTouch Applications with IDE*, in the *InTouch® HMI and Archestra® Integration Guide*.

Licensing

An InTouch license is managed by a licensing system that manages all Wonderware product licenses. A Wonderware license is a file that lists the product features you are authorized to use. The license file also lists the expiration date of each feature.

The licensing system includes the License Utility. You use the License Utility to install, view, and delete licenses for InTouch and other Wonderware products. The License Utility program manages licenses for the local computer and network computers.



Key Points

- The single wwsuite.lic file stores license information for all installed Wonderware products and features.
- You can edit the license file to remove products and features that you no longer require or that have expired.
- When you use the License Utility to import a license file, it reads the contents of the file and stores it in a shared folder, which is usually C:\Program Files\Common Files\Archestra\License.
- If a wwsuite.lic file already exists in the shared folder, you can use the License Utility to either replace the existing license or append new and existing license information into the file.

For more information, see the Wonderware License Utility documentation.

Chapter 3

Visualization: Windows and Graphics

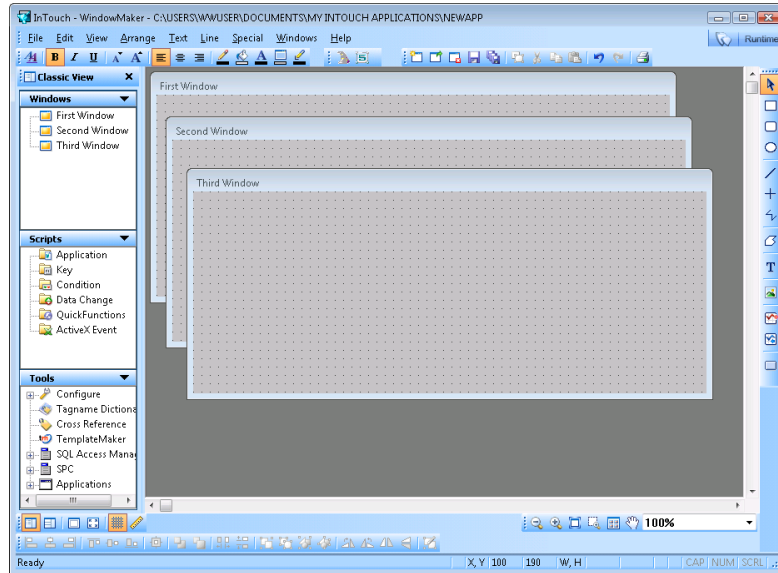
You use WindowMaker to create the visual interface used by operators to view and manage your manufacturing processes. An InTouch interface shows data from and writes data back to the production environment.

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

- Windows are panels containing visual elements for plant operators to manage a production process.
- Basic objects are simple graphical elements, such as rectangles, circles, lines, and text.
- User-defined complex objects consist of one or more basic objects that represent elements in your production environment, such as valves and tanks.
- Pre-defined complex objects perform specific functions, such as alarming and historical trending.
- Animation links are properties of simple and complex objects to animate their appearance and transmit user input to tasks and production data changes.
- Wizards are pre-defined complex objects that perform specific functions or have specific appearance, such as sliders and meters.
- ActiveX controls are controls that you can place in InTouch windows to perform specific functions such as to show current alarms.

InTouch Windows

An application window is a container for one or more graphics that model your production processes. InTouch windows are visual panels that contain the user interface of your InTouch application.



Key Points

- There are three types of InTouch windows:
 - A replacement window automatically closes any windows it intersects with when it appears on the screen.
 - A popup window stays on top of all other windows. Popup windows usually require a response from the user to be closed.
 - An overlay window appears on top of currently open windows. Overlay windows are useful for creating applications consisting of multiple panels with conditional or optional variations.
- You can define the following characteristics of a window when you create it:
 - Horizontal and vertical size in pixels
 - Type of frame
 - Background color
 - Presence of standard controls for windows

For more information, see Chapter 2, Application Windows, in the *InTouch® HMI Visualization Guide*.

InTouch Basic Objects

Both WindowMaker and the Tools Panel of the ArcestrA Symbol Editor include a set of graphic tools to create basic objects in InTouch windows. As you build your applications, you combine these simple objects to create more complex objects.

WindowMaker Basic Objects



ArcestrA Tools Panel Graphic Primitives



Key Points

- After selecting a basic object, you use your mouse to place the basic object in a window or on the ArcestrA canvas.
- You right-click on a basic object to show an action menu with commands to modify the object.
- After placing basic objects in a window or on the ArcestrA canvas, you can select them by:
 - Clicking on them with the mouse.
 - Dragging the lasso around them with the mouse.
 - Selecting them with a menu option or with a shortcut key.
 - Selecting them from the Elements List (Tools Panel only).
- After you draw elements, you have the same cut, copy and paste options available to you as in any other Windows application.
- You can move basic elements or groups by dragging them to the new location or you can open the properties for the element or group and change the X and Y properties.

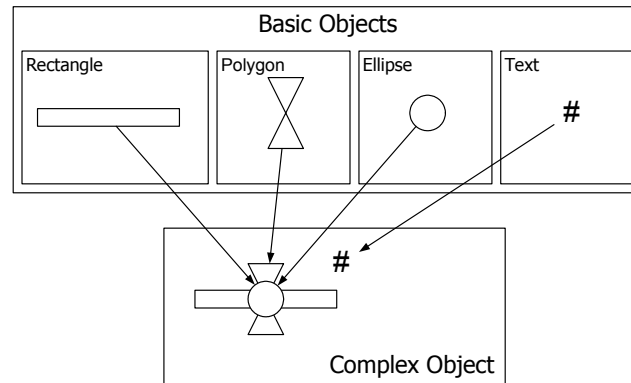
For more information about InTouch basic objects, see Chapter 3, WindowMaker Objects, in the *InTouch® HMI Visualization Guide*.

Also, see the Application Server IDE documentation.

InTouch Symbols

You can create complex objects by grouping basic objects together to represent a component of your production process or to create complex objects by combining several.

For example, you can create a complex object that represents a valve by combining simple rectangle, line, and circle objects.

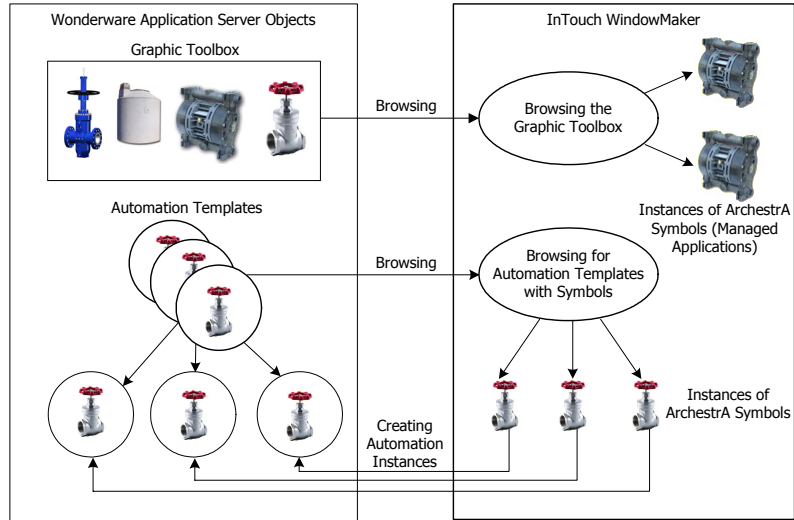


Key Points

- The individual elements of a complex object are usually grouped together to:
 - Prevent the complex object from breaking apart during further editing.
 - Duplicate the entire object.
 - Assign a common set of properties to its individual elements.
- You can use four different user-defined complex objects:
 - Symbols represent a collection of basic objects to create a more complex object. A symbol has unique properties that apply to all contained elements.
 - Cells consist of a set of basic objects to create a more complex object. But, unlike a symbol, a cell has no unique properties. A cell maintains the properties of the individual elements within the cell.
 - SmartSymbols are created from cells and allow for enhanced administration and integration with Application Server.
 - ArcestrA symbols are created in the Symbol Editor of the ArcestrA IDE and embedded into an InTouch window. The properties that are set for the symbol in the IDE are inherited by the InTouch HMI.

ArchestrA Symbols

The ArchestrA IDE includes the Symbol Editor that you can use to create symbols that represent production processes. These symbols can be placed directly into managed InTouch applications.



You can create ArchestrA symbols that are assigned to the Graphic Toolbox. These symbols are not associated with a specific automation object template or object instance. Or, you can create symbols that belong to a specific Automation template or instance.

Key Points

- You can place an ArchestrA symbol from the Graphic Toolbox, an Automation template, or an Automation instance into a managed application.
- If you insert an ArchestrA symbol from a template, the InTouch HMI can create an instance of that object and the symbol instance references the new instance.
- All changes to the source ArchestrA symbol are propagated to all embedded ArchestrA symbols derived from it. This affects the ArchestrA symbols in WindowMaker as well as the ArchestrA symbols hosted by derived automation objects.
- You can only modify external links of an inserted ArchestrA symbol in WindowMaker. Using standard WindowMaker tools, you are restricted to modifying the size, location, and visibility of an inserted ArchestrA symbol.

For more information, see the Application Server IDE documentation.

Pre-Defined InTouch Complex Objects

You can use pre-defined complex objects to represent components of your plant process, such as:

- Wizards
- ActiveX Controls

A typical ActiveX Control included with the InTouch HMI is the Alarm Viewer control that shows an application's current alarms.

Time	State	Class	Type	Priority	Name	Group	Provider
04/24/2006 03:04:07 PM	UNACK	VALUE	HIHI	1	tanklevel	\$System	\intouch
04/24/2006 03:04:09 PM	UNACK	DEV	MAJDEV	1	product	\$System	\intouch
04/24/2006 03:04:21 PM	UNACK	DSC	DSC	1	valve	\$System	\intouch

Displaying 1 to 3 of 3 alarms. Default Query 100 % Complete

ActiveX Controls

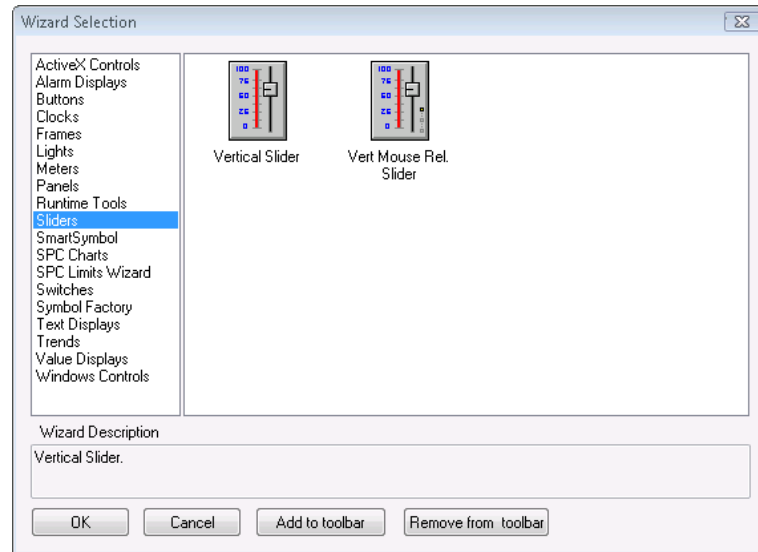
ActiveX controls are compiled third-party objects that you can place in your InTouch application. ActiveX controls have properties, methods, and events that can be modified while an application is running to change the behavior of the control.

The InTouch HMI comes with pre-defined ActiveX controls that you can use in your InTouch applications. Some of these ActiveX controls are:

- Alarm Viewer control: lets you see current alarms generated by the InTouch HMI or any other alarm source in a table format. You can also acknowledge alarms from the Alarm Viewer control.
- Alarm Tree Viewer control: lets you see and select current alarm sources and alarm areas in a tree view.
- Alarm DB View control: lets you see historized alarms in an alarm database in a table format.
- Alarm Pareto control: lets you see historized alarms in an alarm database in a Pareto distribution view.

Wizards

Wizards are pre-designed, pre-built, and pre-programmed objects you only need to select, place, and configure for your application.



Using wizards, you do not spend time drawing the individual components for the object, entering the value ranges for the object, or animating the object.

Key Points

- You select a wizard from a wizard category in WindowMaker and then use your mouse to place the instance of the wizard in a window.
- Trend objects are wizards that chart the values of tags over time. There are three main types of trend objects:
 - Real-time trends chart up to four tags in real time.
 - Historical trends chart up to eight tags over a past time period.
- The Windows Controls Wizards are user interface objects such as drop-down lists, combo boxes, option selections or check boxes.
- Each wizard includes a configuration dialog box. You determine the behavior of your wizard instance by setting its properties the dialog box.

For more information, see Chapter 5, Wizards, in the *InTouch® HMI Visualization Guide*.

Historical Trends

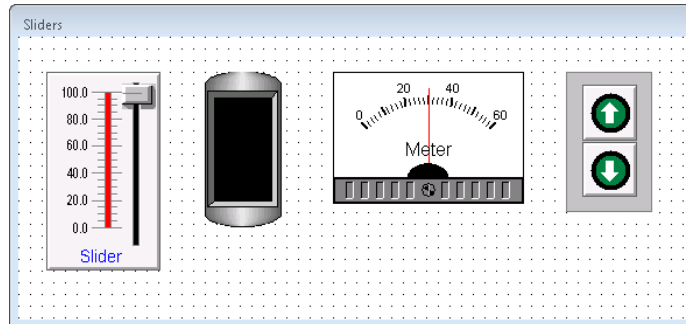
Trends are a special type of wizard. You can use either the:

- Real Time Trend from the Drawing toolbar to show the real-time values of up to four tags in a graph.
- 16-Pen Trend from the Wizards panel to show historical or real-time values of up to 16 tags in a graph.
- Simple Historical Trend from the Drawing toolbar to show the historical values of up to eight tags in a graph.
- Advanced Historical Trend from the Wizards panel to show the historical values of up to eight tags in a graph.

Animation Links

Animation links are properties of objects that appear in InTouch windows. You can set the behavior and appearance of objects by configuring their animation links.

For example, you can create a slider or tank symbol that shows the liquid level in the tank. An animation link with the meter symbol shows a range of tank levels. You can also use animation links with touch screen symbols to enable the operator to control the level of the tank.



Key Points

- Visualization animations modify the visual characteristics of an object like color, fill, line style, text fonts, or blinking.
- Interaction animations specify how users can interact with a visual element like a push button or a slider. As the user interacts with an object, the animation link alters the value of the tag linked to the object.
- In ArchestrA symbols, element-specific animations are used for Status element and Windows Common Control elements.
- Each element in your ArchestrA symbol can have one or more animations.
- You connect the element behavior and appearance with an ArchestrA attribute or a custom property. The ArchestrA attribute provides values at run time that control the behavior and appearance of the element.
- You can also cut, copy, and paste animations between elements, regardless of their type. Only animations that are supported by the target element are pasted.

For more information about InTouch animation links, see Chapter 4, Animating Objects, in the *InTouch® HMI Visualization Guide*.

For more information about ArchestrA animation links, see the Application Server IDE documentation.

Interacting with Animation Links

There are two basic types of animation link interactions:

- Display links show information to operators. Examples of display links are window elements changing colors, changing fill levels, moving vertically or horizontally, and blinking.
- Interaction links allow operators to interact with a window element to monitor or manage an application. Examples of interaction links are sliders or push buttons that respond to operator input.

Key Points

- You can set an animation link for a text object to show the current value of a tag. This type of animation shows information to an operator like tank fill levels, machine on/off status, or alarm messages.
- You can create animations to make objects move at run time by using location links. You can make an object move horizontally, vertically, or both, as the value of an analog tag or expression changes that is linked to the object
- You can make a window appear during run time in a precise location relative to a touch sensitive object. For example, an operator can select an object to see the status, name, or other data linked to that object. When the operator selects the object, either by a click or mouse-over, the window appears in the location you specify.
- You can design operator input touch links to change the value of discrete tags from one state to another. For example you can create a discrete animation link for a switch symbol that turns a pump on or off.
- You can create objects that users can drag back and forth through the use of slider touch links. As the user moves the object, it alters the value of the tag linked to it. You can link an object to a horizontal or a vertical slider.

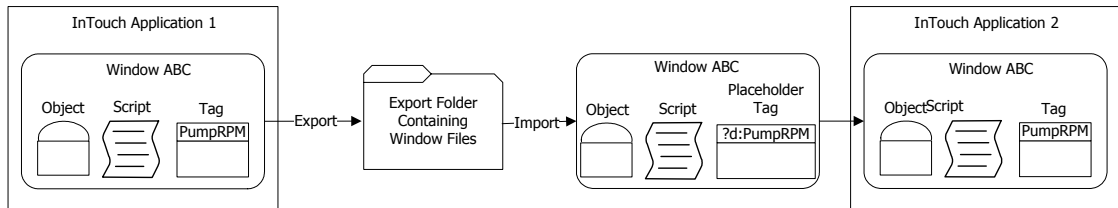
For more information about interacting with animation links, see Chapter 4, Animating Objects, in the *InTouch® HMI Visualization Guide*.

Also, see the Application Server IDE documentation.

Exporting and Importing InTouch Windows

Exporting and importing windows between InTouch applications reduces your development time by allowing you to reuse windows, objects, and scripts.

You can import InTouch windows into the same or a different application using WindowMaker either from the directory you exported to, or from an InTouch application folder directly.



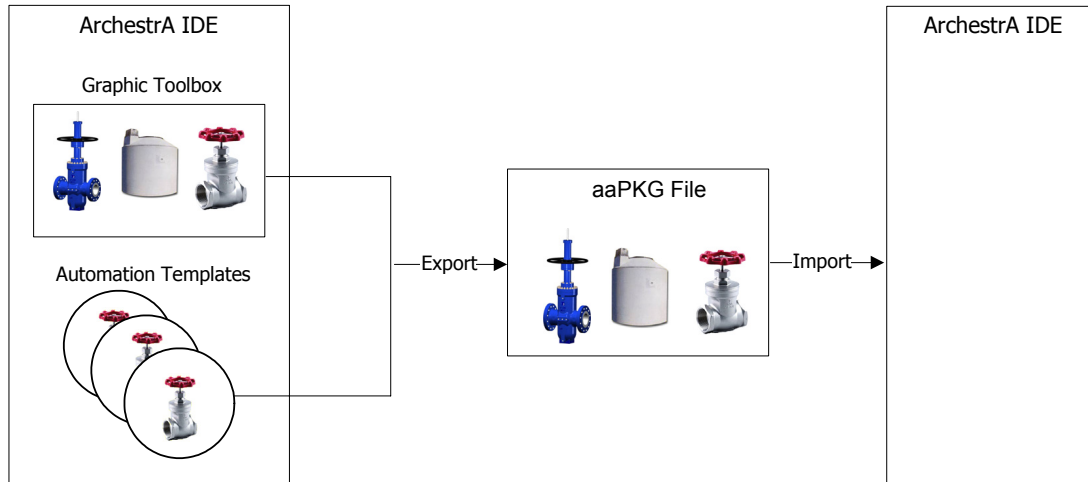
Key Points

- When you export a window, all objects, scripts, and animation links associated with the window are also exported.
- The tags associated with exported windows are converted into placeholder tags to prevent them from overwriting existing tags in the application that imports the windows.
- Placeholder tags are identified by a three-character prefix. Up to a maximum of 4096 placeholder tags can be defined for imported windows.
- To use an imported placeholder tag in your application, you must do one of the following:
 - Convert the placeholder tag to a local tag and define it in the Tagname Dictionary.
 - Convert the placeholder tag to a remote tag reference.
- When you import a window containing SuperTags, only the SuperTag instances are imported into the new application. The original SuperTag template is not exported with the window.
- If you attempt to import a window that already exists in the application, you can replace the existing window with the imported window, import the window with a different name, do not import the window, or cancel the import operation.

For more information, see Chapter 4, Exporting and Importing Tag Definitions, Windows, and Scripts, in the *InTouch® HMI Application Management and Extension Guide*.

Importing and Exporting ArcestrA Symbols

You can import and export symbols as ArcestrA automation object files. When you export or import templates or instances that contain symbols, the symbols are exported or imported with the template or instance.



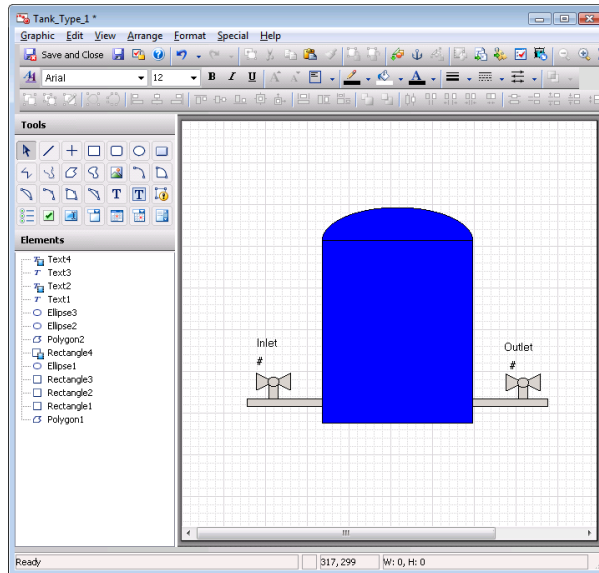
Key Points

- You export symbols and their toolset folders to a symbol .aaPKG file. You import symbols and graphic toolsets from a symbol .aaPKG file.
- When you export or import all automation objects, the contained symbols and the symbols in the graphic toolsets are also exported or imported.
- When you export an automation object that contains symbols, if these symbols contain embedded:
 - Symbols from the Graphic Toolbox, these symbols are exported along with the symbols associated with the automation object.
 - Graphics from other automation objects, these symbols (including any embedded symbols from the Graphic Toolbox contained in them) and their parent automation object are not exported with the automated object. They remain as references, and upon import, the system tries to reestablish the connection with those exact references in the target Galaxy.

For more information, see the Application Server IDE documentation.

Re-using SmartSymbols

You can re-use InTouch SmartSymbols in ArchestrA symbols by importing them. After the SmartSymbol is imported, it appears in the ArchestrA Symbol Editor.



Key Points

- You can import any InTouch SmartSymbol into an ArchestrA symbol. The SmartSymbol may contain objects that cannot be imported, or can be imported but have limited functionality.
- When you re-use an InTouch SmartSymbol, all data configured in InTouch animations is migrated to ArchestrA animations. InTouch animations and ArchestrA animations often have a different name, but are functionally similar.
- When you migrate a SmartSymbol, all action scripts associated with objects in SmartSymbol are migrated as well. An action script in a SmartSymbol becomes an action script animation in an ArchestrA symbol.
- When you migrate a SmartSymbol, the InTouch keyword is appended to local tags, SuperTags, and I/O references as a prefix in the form: *InTouch:tag_name* or *InTouch:I/O_Reference*

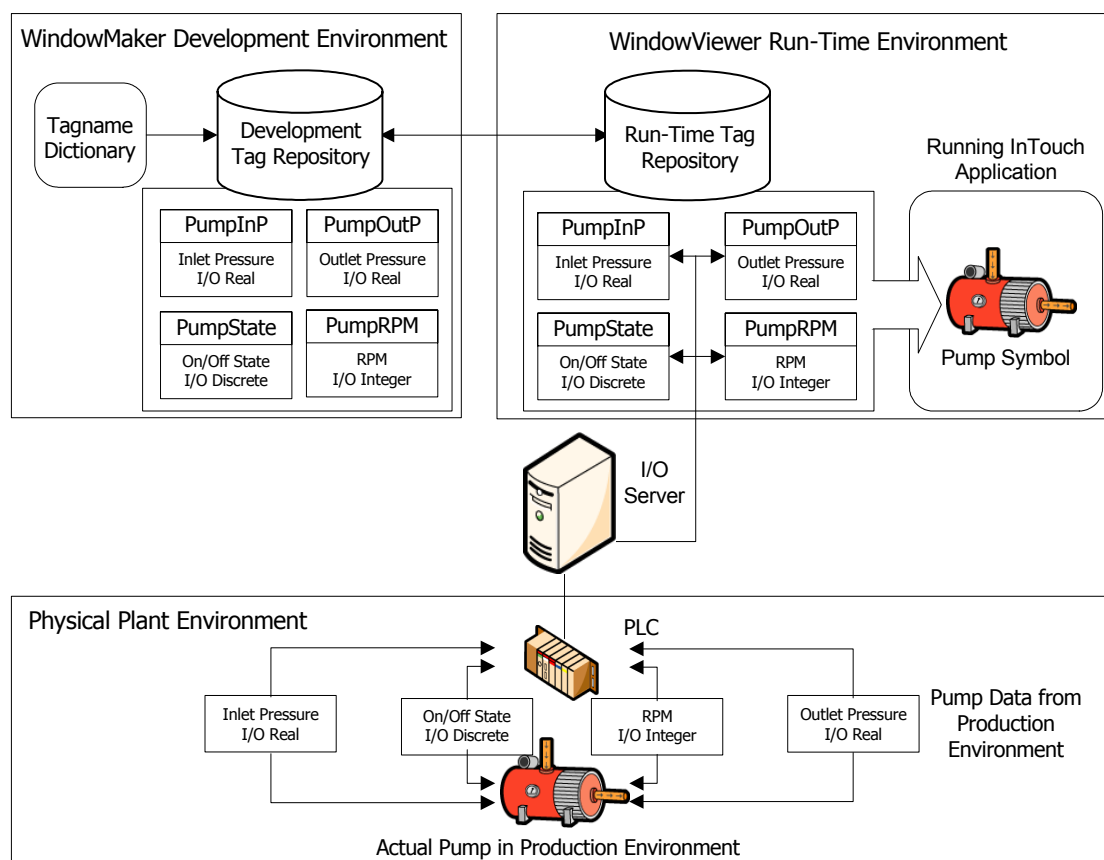
For a reference to an ArchestrA Galaxy, the word Galaxy is removed from the migrated reference. For example *galaxy:pump1.valve1* becomes *pump1.valve1*.

For more information, see the Application Server IDE documentation.

Chapter 4

Data Management

After you create an InTouch application, you define tags for the application using the Tagname Dictionary, which is a WindowMaker tool. The following figure shows how you associate production data with tags to visual objects in a running InTouch application.

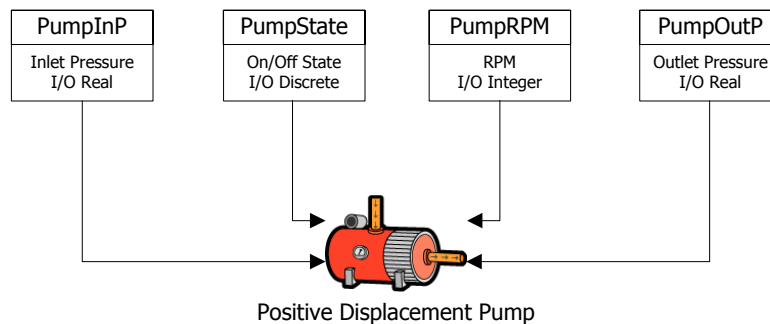


About Tags and Their Properties

A tag represents a data item in an InTouch application. You create tags for those process components whose properties you want to monitor or control with your application.

You assign the name and type of tag with the Tagname Dictionary. For some types of tags, you have other options in the Tagname Dictionary to specify additional properties of tags. For example, I/O type tags include additional options to specify the connection to a remote data source.

The figure below shows an HMI symbol that represents a pump in a manufacturing process. The actual process pump has properties with associated values. Pressure, RPM, and status are pump properties whose values you associate with tags in your HMI.



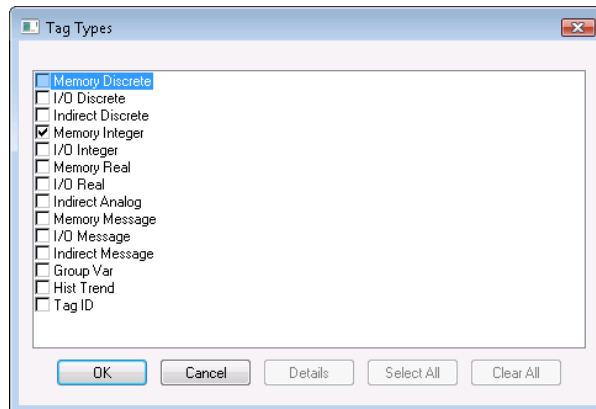
You can use different types of tags for the different types of data collected from a manufacturing component. For example, the PumpState tag returns a Boolean On/Off value to indicate if the pump is running or stopped. You assign the appropriate type of InTouch tag for the type of data that you want to be part of your application.

When WindowViewer starts an application, it reads the tags from the development repository and places them into run-time memory. The InTouch application communicates with the tags placed into run-time memory using animation links or scripts. The InTouch application tracks the current values and other status information from the component properties assigned to tags.

Tag Types

When you define a tag, you assign it to a specific type according to the intended purpose of the tag or the type of data that is associated with it.

In the Tagname Dictionary, you use the **Tag Types** dialog box to assign the tag type to any tag you create.



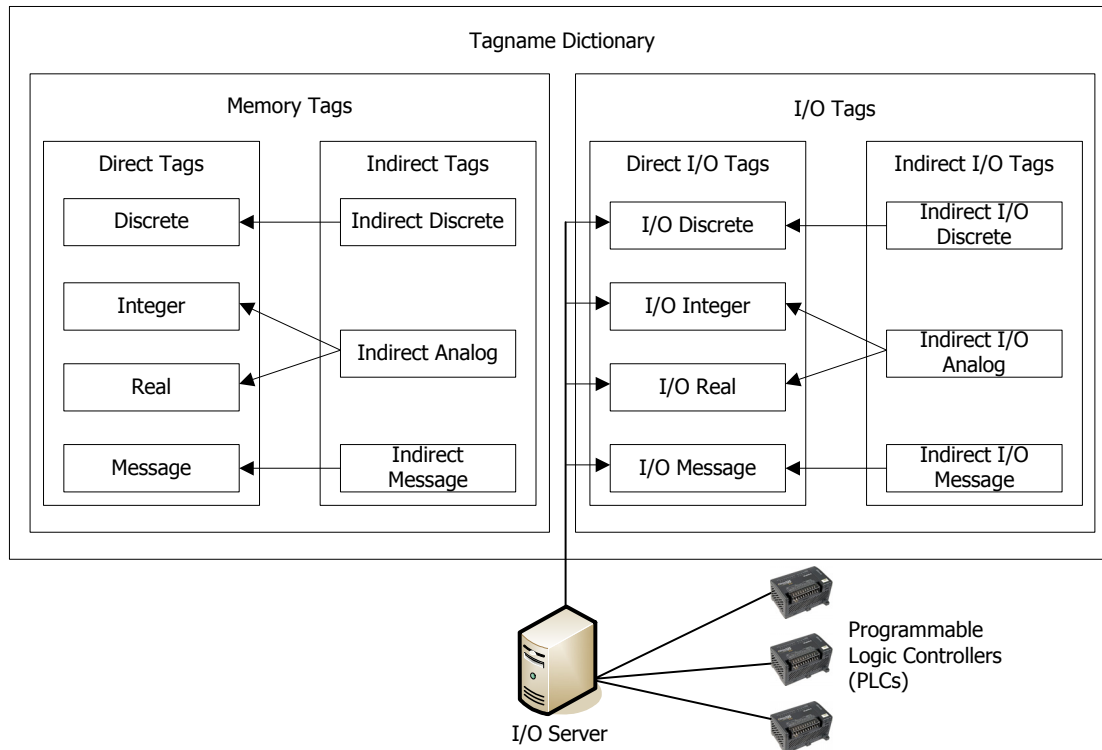
Key Points

- After you assign a tag type, the Tagname Dictionary shows specific properties for the type of tag you selected.
- Memory tags define internal system constants and variables within InTouch applications. Memory tags can also act as calculated variables that are accessed by other programs.
- I/O tags read or write InTouch application data to or from an external source like an I/O Server. External data includes input and output from programmable controllers, process computers, and network nodes.
- Indirect tags serve as pointers to other tags. When you equate an indirect tag to another source tag, the indirect tag acts as if it is the source tag.
- A Hist Trend tag references data shown in InTouch historical trend graphs.
- Tag ID tags are assigned to a specific trend pen or change the tag assigned to a trend pen.

For more information, see Chapter 1, Data Management Overview, in the *InTouch® HMI Data Management Guide*.

Data Types

You can assign your application's tags to four different types based upon the process data associated with the tag. The following figure shows the Tagname Dictionary includes specific types of tags for integer, real, discrete, or message data. The Tagname Dictionary includes data types for memory, I/O, and indirect tags.



Key Points

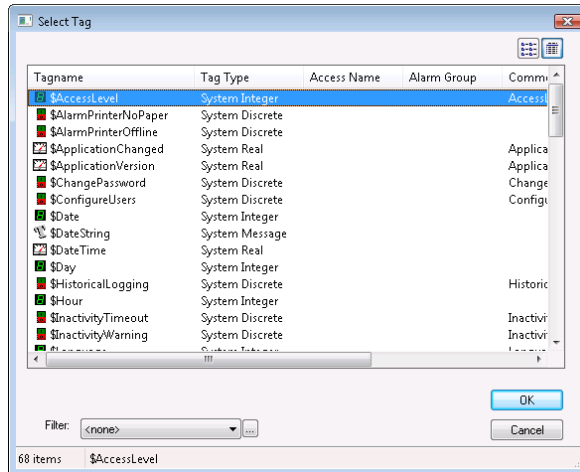
- Discrete tags are associated with process component properties whose values are represented by two possible Boolean states.
- Integer tags can be assigned 32-bit signed-integer numbers between -2,147,483,648 and 2,147,483,647.
- Real tags can be assigned floating decimal point numbers between -3.4×10^{38} and 3.4×10^{38} . All real tag floating point calculations are performed with 64-bit resolution, but the results are stored as 32-bit numbers.
- Message tags can be assigned text strings up to a maximum of 131 single-byte characters.

For more information, see Chapter 1, Data Management Overview, in the *InTouch® HMI Data Management Guide*.

User-Defined Tags and System Tags

Any tag that you create for an InTouch application is a user-defined tag. You can also select predefined system tags from the Tagname Dictionary. System tags are included with all InTouch applications.

You use the **Select Tag** dialog box to select a system tag from the Tagname Dictionary.



You can include system tags in scripts to provide system information for your applications like date and time. You can also use system tags in scripts to manage a running application like:

- Monitoring and managing application security.
- Establishing I/O communications to a remote node.
- Detecting when an alarm occurs.
- Starting and stopping historical logging.
- Updating an application to a new version with NAD.

Key Points

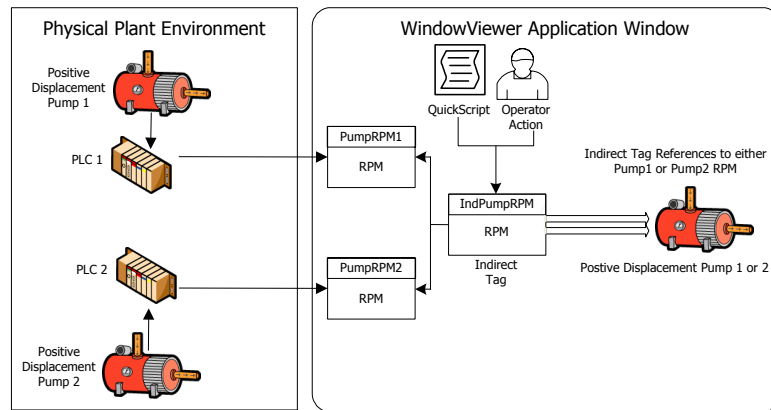
- System tags are identified by a dollar sign (\$) as the first character of a tag's name.
- System tags can be associated with discrete, integer, real, and message data.
- The values associated with a system tag can be read-only, write-only, or read/write.
- System tags cannot be deleted from the Tagname Dictionary. You can only change the comment associated with a system tag.

For more information, see Chapter 3, System Tags, in the *InTouch® HMI Data Management Guide*.

Indirect Tags

Using indirect tags, you can create applications with window objects that show values from multiple tags. Indirect tags are tags that can act on behalf of another tag. In the programming world, these are called pointers. You can point indirect tags at direct tags and then use the direct tags as if you were using the direct tags.

Using indirect tags reduces your application development time. Instead of creating a separate set of tags and window objects for each process component, you can create applications with window objects that show values from multiple tag sources.



Key Points

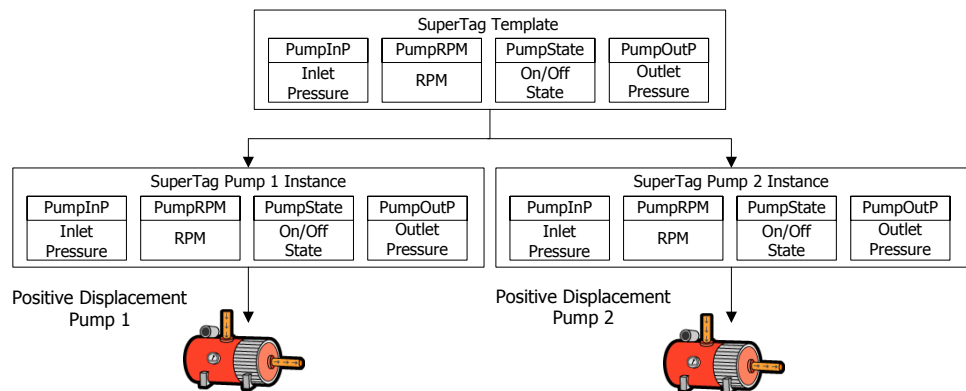
- Indirect tags can be equated to integer and real tags. Indirect tags can be mapped to any other tag as long they are the same tag type.
- When you equate an indirect tag to another source tag, the indirect tag behaves as if it is the source tag. If the value of the source tag changes, the indirect tag reflects the change. If the indirect tag's value changes, the source tag changes accordingly.
- You can use scripts to assign input source tags to an indirect tag. The script can be triggered by an application event or an operator action like clicking a window button.
- You can assign retentive attributes to indirect tags. With retention, the indirect tag retains its most recent tag assignment when the InTouch application starts again.

For more information, see Chapter 6, *Defining Indirect Tags*, in the *InTouch® HMI Data Management Guide*.

SuperTags

A SuperTag is a template structure of tags. The tags that belong to a SuperTag template are associated with the common properties of a component in a manufacturing process.

SuperTags save development time. Instead of creating a set of tags for every component in the manufacturing process, you can replicate a single SuperTag template and create individual instances for all process components that have the same properties.



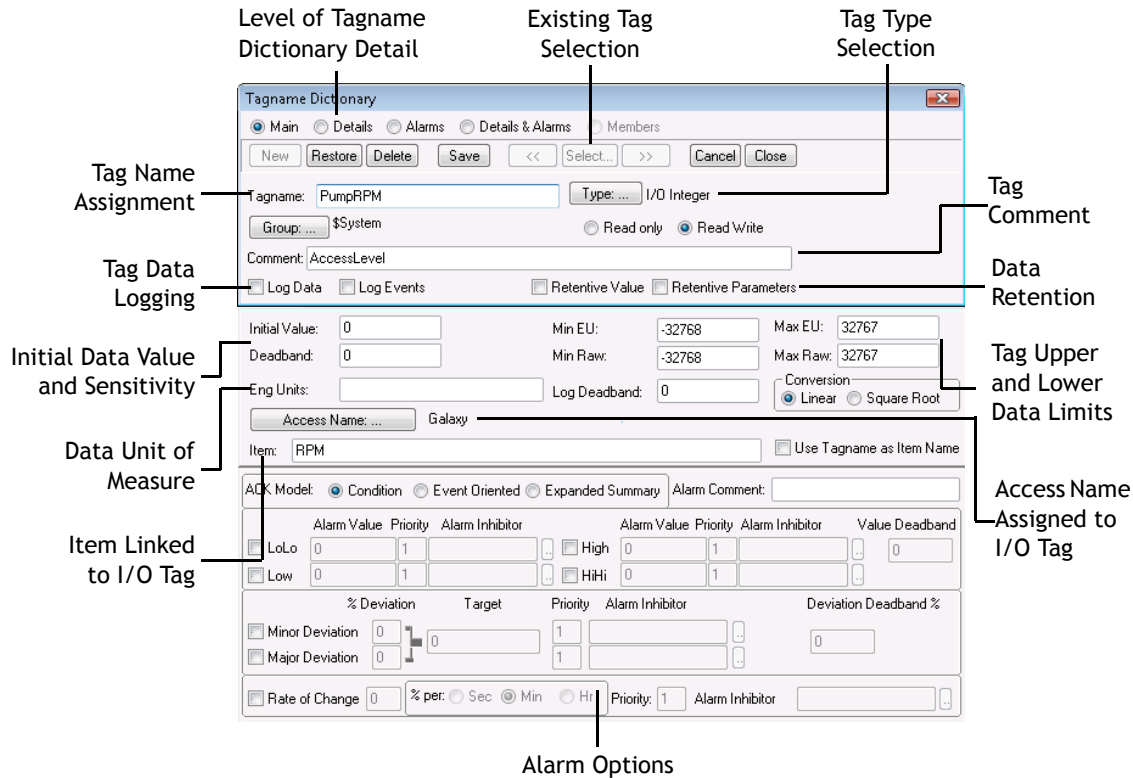
Key Points

- You can create, edit, and delete SuperTags in the Tagname Dictionary or in the TemplateMaker, which is a WindowMaker utility.
- A SuperTag template can organize its member tags in two nesting levels. A SuperTag template can contain up to 64 embedded child tags. Each child tag can contain up to 64 sub-member tags.
- All tags that belong to an instance of a SuperTag can be assigned as InTouch discrete, integer, real, and message data types. The tags within the SuperTag instance support trending, alarming, and all tag dotfields.
- InTouch tag names can be a maximum of 32 characters. The format of a SuperTag name is:
 ParentInstance\ChildMember\Sub-member
 which must be 32 characters or fewer.
- You can use the ArchestrA Bulk Import Utility to migrate SuperTags from the InTouch HMI into an Application Server object structure.

For more information, see Chapter 7, Defining Reusable Tag Structures, in the *InTouch® HMI Data Management Guide*.

Tag Properties

Using the Tagname Dictionary, you create tags for an InTouch application. The figure below shows the **Tagname Dictionary** dialog box with all options to define the properties of an I/O tag.



Key Points

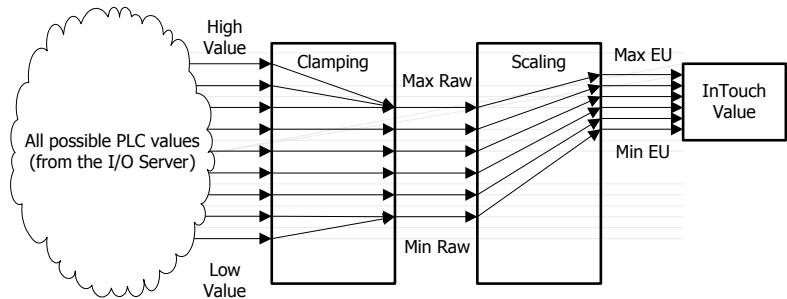
- Thorough planning reduces the time you need to create InTouch applications. You should identify the key requirements of an application's tags in a preliminary planning phase.
- Using the **Tagname Dictionary** dialog box, you can specify common tag properties that are part of every tag definition. You must assign a name to each tag. You can add an optional comment. Both the tag name and comment are common properties of all tags.
- Each type of InTouch tag has unique data properties. After you select a tag type, the **Tagname Dictionary** dialog box expands to show a set of options, based upon the selected tag type.

For more information, see Chapter 2, Managing Tags with the Tagname Dictionary, in the *InTouch® HMI Data Management Guide*.

Tag Data Limits

You must assign I/O integer and real tags a set of attributes that characterize numerical data sent between the InTouch application and external processes.

The InTouch HMI normalizes raw input data from a PLC. The figure below shows the process of clamping raw I/O data values and then scaling them to engineering units that can be shown from an InTouch application.



Key Points

- I/O integer and real tags include attributes that set minimum and maximum limits for raw input data sent by the PLC. The InTouch HMI clamps I/O values that are below or above the raw value range. Clamping reassigns values outside of the range to the minimum or maximum raw value limits.
- I/O integer and real tags include attributes that scale clamped raw values within a range of engineering units. Minimum and maximum engineering unit attributes set the lower and upper boundaries of the scaled values.
- When you define integer and real I/O tags, you specify the type of conversion to scale raw values when calculating engineering units. You can select linear or square root scaling:
 - In linear scaling, the result is calculated using linear interpolation between the minimum and maximum data end points.
 - In square root scaling, the minimum and maximum raw values are used for interpolation. This is useful for scaling inputs from nonlinear devices like pressure transducers.
- You can use a set of dotfields to monitor and modify a tag's raw and engineering units value range.

For more information, see Chapter 2, Managing Tags with the Tagname Dictionary, in the *InTouch® HMI Data Management Guide*.

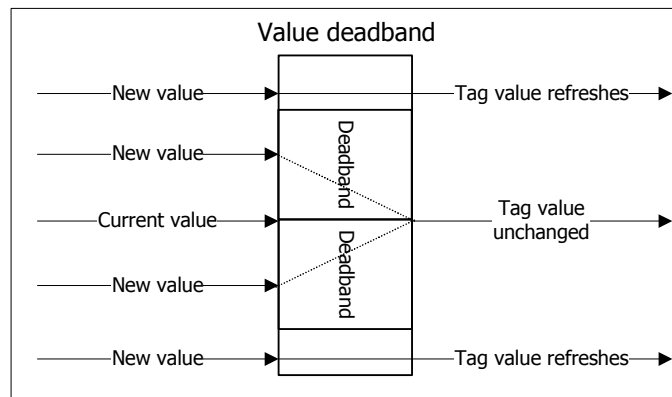
Tag Deadbands

A deadband is a sensitivity setting for tag values. A deadband is usually associated with I/O tags whose values change constantly. A deadband filters out small changes in a tag value to reduce the amount of InTouch data processing.

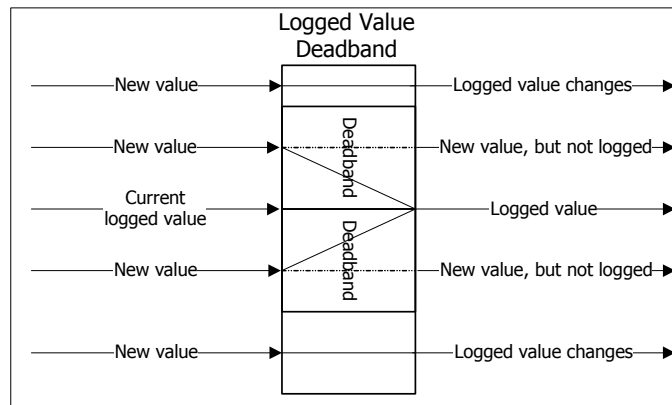
The Tagname Dictionary includes two deadband properties for tags associated with integer and real data.

Key Points

- The value deadband property sets a threshold that must be exceeded before WindowViewer refreshes the tag's value in run-time memory. The following figure shows the absolute deadband range around a current tag value.



- The log deadband sets a threshold that must be exceeded before the tag's value is written to the log file. The following figure shows the deadband around the current value of a logged tag.



Only new tag values outside of the deadband are written to the log file. Small value changes within the deadband range are ignored and not logged.

For more information, see Chapter 2, Managing Tags with the Tagname Dictionary, in the *InTouch® HMI Data Management Guide*.

Tag Retention

Discrete, integer, and real tags are assigned an initial value when the InTouch application starts in WindowViewer. In the case of a discrete tag, the initial value is one of the possible binary states. For integer and real tags, the initial value is the number associated with the tag when the application starts.

You can specify the initial tag value to be the last value of the tag when the application stops running in WindowViewer. By selecting the **Retentive Value** option from the Tagname Dictionary, the tag is assigned its last active value as the initial value when the application starts again.

Key Points

- By default, when you start WindowViewer, the application's tags are initialized to their values defined in the Tagname Dictionary.
- There are the two types of retention:
 - Value retention - when WindowViewer starts, the tag is initialized with its last value when the application was stopped.
 - Parameter retention - when WindowViewer is started, tag parameters, such as alarm limits that can be changed at run time, retain their values.

For more information, see Chapter 2, Managing Tags with the Tagname Dictionary, in the *InTouch® HMI Data Management Guide*.

Tag Dotfields

A dotfield identifies a tag property. All InTouch tags have properties that describe the characteristics of data or possible conditions associated with a tag.

Some dotfields are similar across all types of tags. For example, the .Name dotfield is always associated with a tag's name. Other dotfields apply only to the unique properties of a specific type of tag.

Key Points

- Using dotfields in a script, expression, or user input, you can monitor and modify a tag's properties while an application is running.
- You concatenate the name of the dotfield to the tag name in a script or expression in the form:

```
tag_name.property_dotfield
```

- You can use dotfields to allow input and output of data associated with a tag and you can use historical dotfields to modify the historical trend currently shown from a running application.

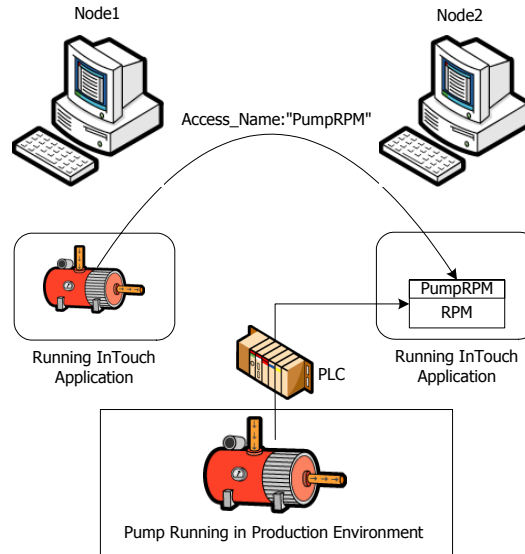
For example, the .UpdateInProgress dotfield is typically used as the expression in a visibility link on a text object of the scroll buttons of a historical trend. You can use the .UpdateInProgress dotfield in a message value display animation link to show "Busy" on the window while historical data is retrieved:

```
DText (HistTrend1.UpdateInProgress, "Busy", "Ready")
```

For more information, see Chapter 4, Using Tag Dotfields to View or Change Tag Properties, in the *InTouch® HMI Data Management Guide*.

Remote Tags

You can create distributed InTouch applications with a tag server running on a separate node from the node running the InTouch application. The following figure shows an InTouch application that makes a remote reference to the PumpRPM tag from a tag server running on another node.



Key Points

- You can reference remote tags by two methods:
 - Associate I/O tags with an Access Name that identifies a remote server as the tag source.
 - Use a remote reference directly to the tag. For example, PLC1:PumpRPM.
- When you want to directly reference a remote tag in any other application, only *AccessName:Item* is required. You do not have to define the remote tag in your local Tagname Dictionary. Remote references can also access data from any I/O data source such as a Wonderware DAServer or Microsoft Excel.
- You can also make a remote reference to SuperTags. The valid syntax for a remote tag reference to a SuperTag is:

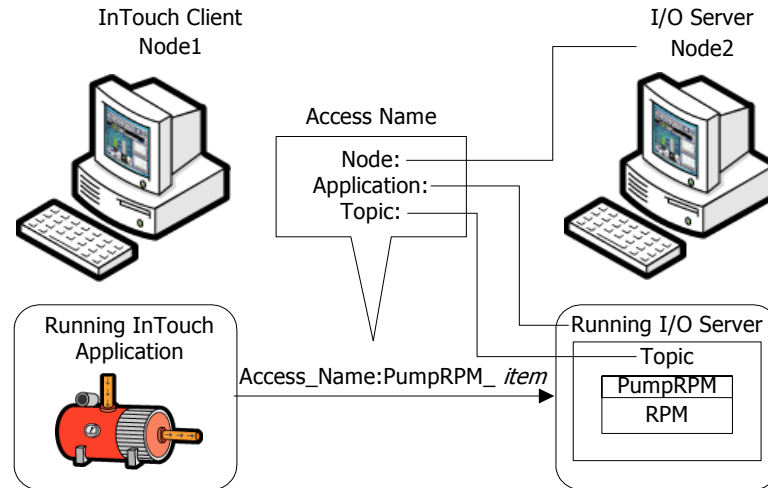

```
Access_name:Parent_Instance\ChildMember\SubMember.
```
- You can redirect ArchestrA object references or remote references to InTouch tags at run time with a script. You can switch object instances for a graphic symbol based on certain conditions being met or directly by an operator action.

For more information, see Chapter 5, Data Access with I/O, in the *InTouch® HMI Data Management Guide*.

Access Names

You can develop distributed applications in which the functional components of an InTouch system are located on different network computers. I/O data sources are usually programs that run on a remote computer and use DDE or the SuiteLink protocols to communicate with other computers in an InTouch system.

An Access Name defines a communication link with another InTouch I/O data source.



Key Points

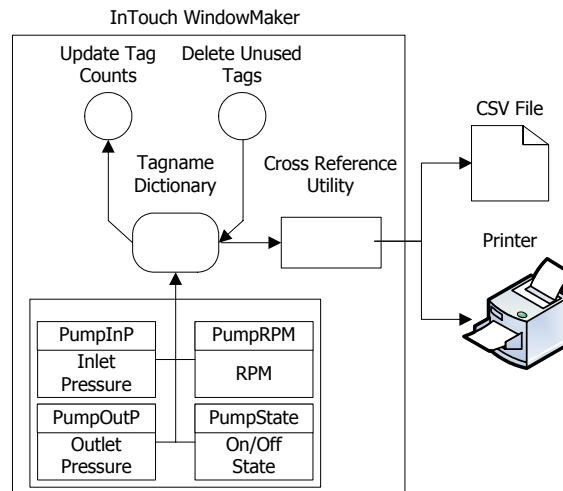
- An Access Name consists of:
 - Node name, which is the name of the computer the I/O data source program or service is running on.
 - Application name, which is the name of the source program or service that provides I/O data to the InTouch application.
 - Topic name, which is an application-specific sub-group of data elements. In the case of data from a Wonderware DAServer program, the topic name is the same name configured for the topic in the DAServer program.
- You can configure an I/O tag by associating it with an Access Name and an item. The item is an individual I/O data source reference, for example a specific value or sensor.
- InTouchView applications cannot connect to I/O sources other than the ArchestrA Application Server Galaxy. You can only use the default Galaxy Access Name with an InTouchView application.

For more information, see Chapter 5, Data Access with I/O, in the *InTouch® HMI Data Management Guide*.

Managing Tags in an Application

The maximum number of tags you can use in an InTouch application is determined by your Wonderware license. InTouch HMI maintains a use count for all tags defined in the local Tagname Dictionary.

The figure below shows how the Cross Reference Utility produces a tag usage report by analyzing the contents of the Tagname Dictionary. You can track tag usage in an InTouch application and then delete any unused tags to minimize your tag count.



Key Points

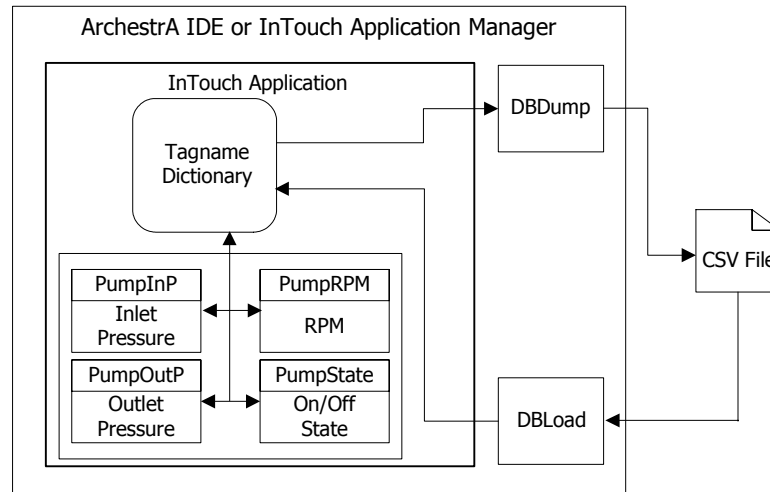
- You can update the count of local tags that are currently defined in an application's Tagname Dictionary. You can also update the count of an application's references to remote tags located on another node.
- The Cross Reference Utility report lists the name of the window where each tag is defined and the window coordinates of all objects linked to a tag. The report also allows you to view any QuickScript or QuickFunction that includes a listed tag.
- You can also print the contents of the Tagname Dictionary, which shows the database entries, windows, and scripts used in the application.
- You can delete unused tags from an InTouch application after updating the use count. The tag count is not automatically updated when a window containing tags is deleted from an application or tags are changed in link scripts. The InTouch HMI regards the tags as being in use and prevents them from being deleted.

For more information, see Chapter 8, Reducing Tag Usage, in the *InTouch® HMI Data Management Guide*.

Exporting and Importing InTouch Tags

You can use the DBDump and DBLoad utilities to export and import the contents of an application's Tagname Dictionary. Both the Application Manager and the ArchestrA IDE include the DBDump and DBLoad utilities.

DBDump exports the contents of an application's Tagname Dictionary to a Comma Separated Values (.csv) file. DBLoad imports the contents of the .csv file into the Tagname Dictionary of a selected InTouch application.



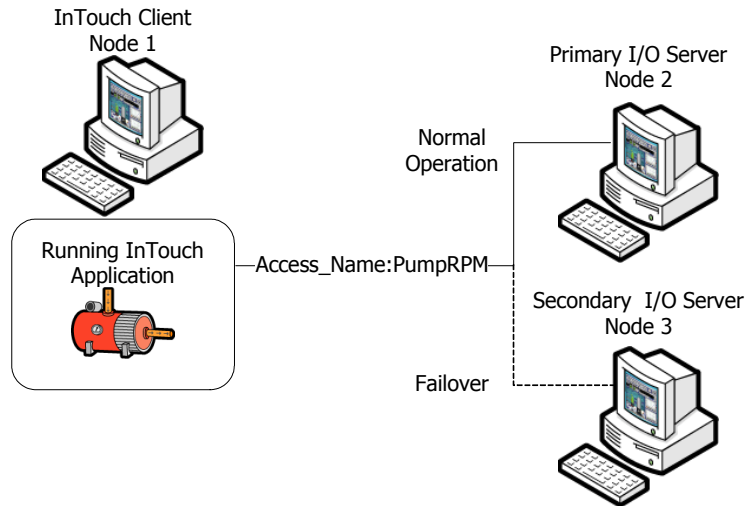
Key Points

- DBDump creates a file containing the contents of the Tagname Dictionary and a set of keywords that identify the Access Names, alarm groups, and each type of InTouch tag. Tag records begin with a keyword line that identifies the type of tag. Each tag type has unique keywords to specify the characteristics of the data associated with the tag.
- You can use any application that supports the .csv file format to add, modify, or delete tags. Typically, you use Microsoft Excel or Notepad to edit the contents of the file.
- After you have made your changes to the data, you then import the contents of the file with DBLoad into the same InTouch application or another InTouch application.
- An application's Tagname Dictionary is updated with new tags based upon values you set for the :mode keyword, which is part of the import file. The value you assign to the :mode keyword indicates how duplicate tags should be handled during the import process.

For more information, see Chapter 4, Exporting and Importing Tag Definitions, Windows, and Scripts, in the *InTouch® HMI Application Management and Extension Guide*.

I/O Failover

You can set up your InTouch application to switch to a failover secondary I/O Server when it can no longer communicate with the primary I/O Server. You can set up server switching by optionally defining a secondary I/O data source when you create an Access Name.



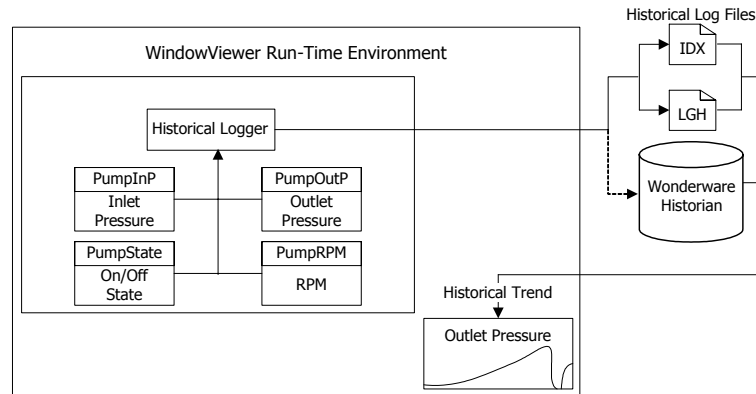
Key Points

- The switching from the primary to the secondary data source can be triggered with a script or a failover expression. A failover expression can be a tag or expression that triggers failover when it evaluates to True.
- You can also manually switch between the primary and secondary data sources of an Access Name without experiencing a failover. This is called a *forced failover*.
- When you define a secondary data source in an Access Name, you specify the failover deadband. The failover deadband is the delay in seconds before switching from the primary to the secondary data source.
- You can write scripts that include functions that return the status of the primary, secondary, and active I/O sources of an Access Name. Typically, operators run a script to determine the status of an Access Name's secondary I/O source before forcing a failover.

For more information, see Chapter 5, Data Access with I/O, in the *InTouch® HMI Data Management Guide*.

Historical Data

While an InTouch application is running, the values of selected tags can be logged and permanently saved. You can use this log data to create historical trend graphs that show some aspect of your plant processes over time.



Key Points

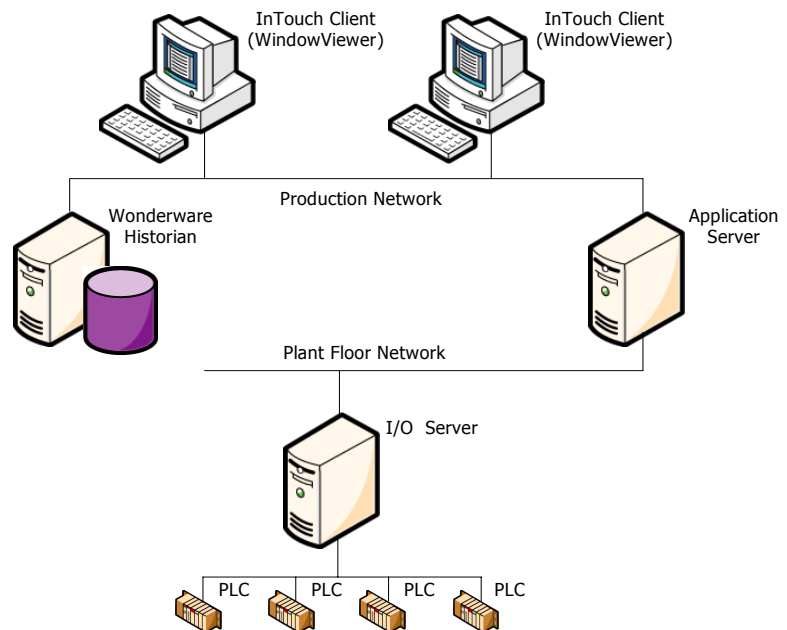
- You select tags for historical logging from the Tagname Dictionary. The selected tags are considered active as long as the application is running in WindowViewer. When a tag's value changes, the Historical Logger determines if an entry should be written to the log based upon each tag's logging deadband and its current value.
- There are two basic ways to save historical data based upon the amount of data to be logged:
 - InTouch historical log files for smaller applications with modest logging requirements.
 - Wonderware Historian, which can retrieve historical data directly from the running InTouch application.
- The InTouch HMI creates two log files. One file contains logged data stored in a proprietary format. The other file is an index to the data.
- A daily logging cycle begins and ends at midnight. The Historical Logger writes the last entries to the active log files at midnight and archives them. Two new files are created for the next day and the Historical Logger begins logging to them.
- The Historical Logger writes historical records at a periodic sampling interval, which is usually once each hour. At all other times, the InTouch Historical Logger writes a log entry each time a tag's value changes more than its specified log deadband.

For more information, see Chapter 9, Recording Tag Values, in the *InTouch® HMI Data Management Guide*.

InTouch and the Wonderware Historian

The Wonderware Historian is a real-time and historical database that stores plant process data. It combines the storage capacity of a relational database with the speed of a real-time system.

As an enhancement to Microsoft SQL Server, the Wonderware Historian acquires plant data at dramatically increased speeds. It also integrates plant data with event, summary, production, and historical data from the InTouch HMI and other Wonderware products.



Key Points

- The InTouch HMI can store historical tag data to remote historical repositories. In this case, you must share the InTouch application directory and configure Wonderware Historian to access the InTouch application.
- If you use the Wonderware Historian to store InTouch historical data, you must use the Distributed Name Manager from WindowMaker to specify a connection to the database.
- The InTouch HMI can show historical data stored in a Wonderware Historian database in a Trend Wizard. You need to install SQL Server client components and browse the Wonderware Historian computer to access historical tag data.

For more information, see the Wonderware Historian documentation.

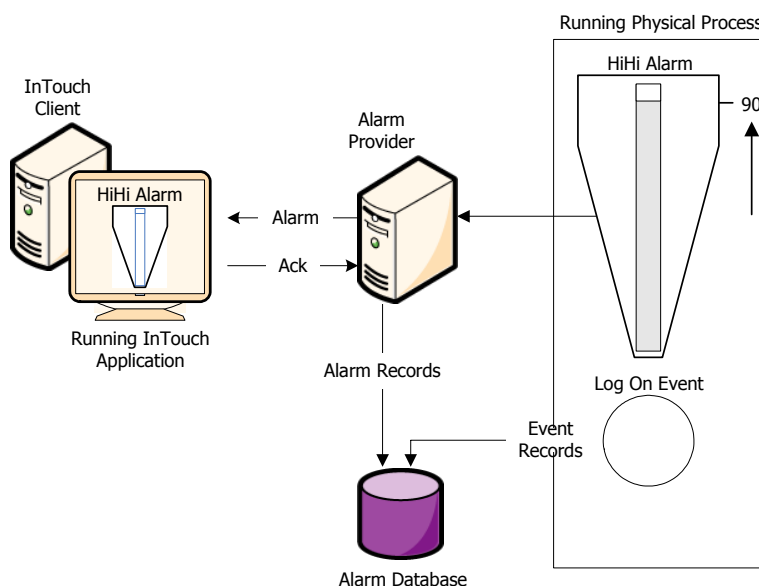
Chapter 5

Alarms and Events

An InTouch application generates alarms and events to notify operators about the status of process activity. Alarms represent warnings about process conditions that can potentially cause problems. Typically, an alarm is triggered when a process value exceeds a defined limit. Operators must acknowledge an alarm.

Events represent normal system status messages. A typical event indicates when a system condition takes place, such as an operator logging on to an InTouch application. Operators do not have to acknowledge events.

The following figure shows how InTouch handles alarms and events while an application is running. Both alarm and event data can be stored to the alarm database.



Types of Alarms

Within the InTouch HMI, alarms are classified into general categories. Alarm categories are based on the characteristics of tag data that represents an alarm condition.

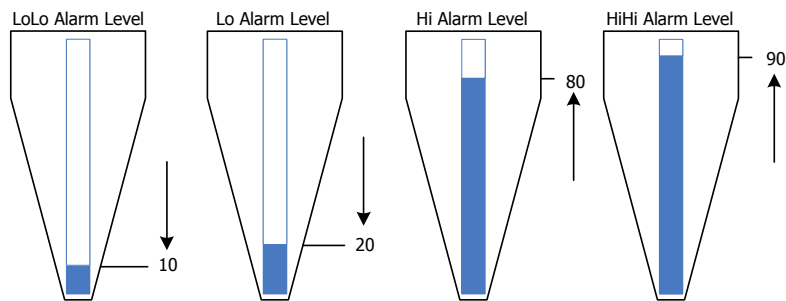
Discrete Alarms

A discrete alarm is set by one of the two possible values of a discrete tag. When you create a discrete tag, you configure whether the alarmed state corresponds to the tag's true or false state.

Value Alarms

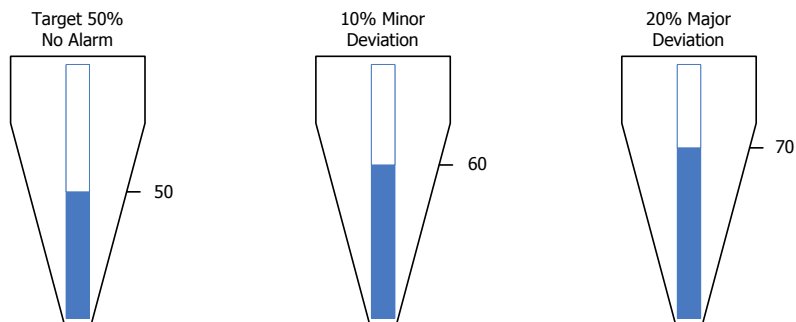
In a value alarm, the current tag value is compared to one or more predetermined limits. If a tag's value exceeds these limits, an alarm occurs.

You can individually configure values and priorities for a LoLo limit, Lo limit, Hi limit, and HiHi limit. Also, you can specify whether these value limits indicate an alarm condition.



Deviation Alarms

The current tag value is compared to a target value. Then, the absolute value of the difference is compared to one or more alarm limits expressed as a percentage of the tag's possible value range.

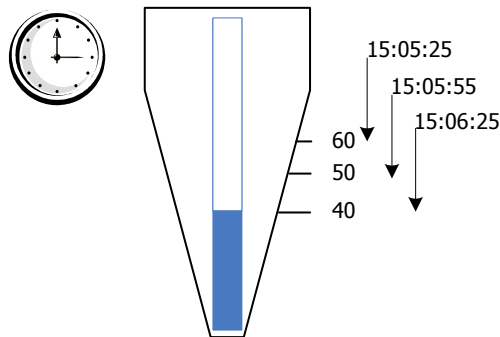


There are major and minor severity levels of a deviation alarms. You can individually configure values and priorities for the major and minor deviation limits.

You can also configure a value for a deviation deadband, which is expressed as a percent of the tag's total value range. This deadband specifies the percentage of the total range that a tag value must change to be regarded in an alarm state.

Rate-of-Change Alarms

In a rate-of-change alarm, a tag's current and previous values are compared over a measured period. If the absolute value of the rate of change within the period exceeds a defined limit, an alarm occurs.

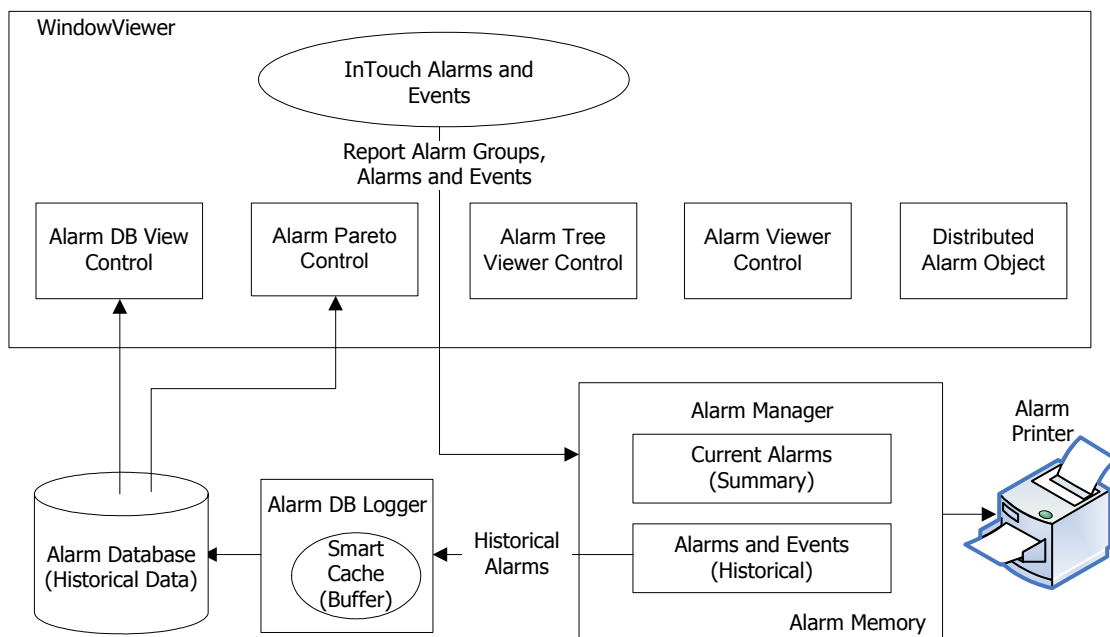


You can configure the value and priority for the rate-of-change limit, and whether or not the limit is to be used. The limit is expressed as a percent of the previous value over a time interval. The interval can be defined as per second, per minute, or per hour.

InTouch Distributed Alarm System

The InTouch Distributed Alarm system consists of a set of integrated components that collect, store, and show alarms.

- The Alarm Manager manages currently active alarms (summary alarms) and historical alarms and events.
- The Alarm DB Logger writes historical alarm records to the alarm database.
- The Alarm Printer prints historical alarms.
- A set of ActiveX controls retrieves alarm records from either the internal alarm memory or the alarm database and shows alarm details at run time.



Key Points

- Summary and historical alarm records are temporarily stored in the Smart Cache before being written to the alarm database.
- On any given node there can be a collection of Alarm Providers (Publishers) and (Subscribers). The InTouch Distributed Alarm system provides the communication link to distribute alarm information between nodes.
- The Distributed Alarm System supports external alarm providers, such as Wonderware I/O Servers, alarm print servers, Galaxy alarms, and other software tools.

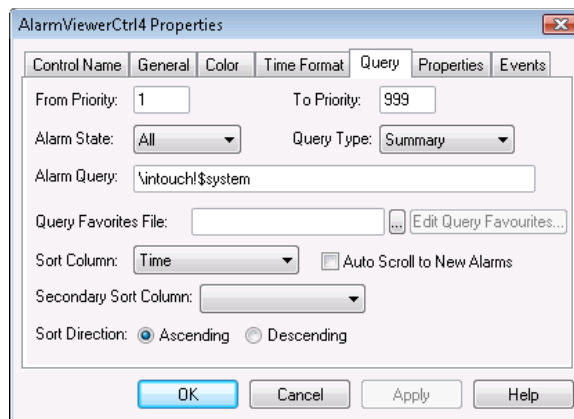
For more information, see Chapter 1, Overview of Alarms and Events, in the *InTouch® HMI Alarms and Events Guide*.

Alarm Queries

An alarm query retrieves either:

- Alarms and events (historical alarms) from the InTouch internal alarm memory or the alarm database.
- Current alarms (summary alarms) from the InTouch internal alarm memory.

When you configure an InTouch alarm ActiveX control, you specify the query source. For example, you use the **Query** tab to configure the query for the Alarm Viewer ActiveX control.



You build your queries by selecting alarm or event attributes from the list shown in the left pane of the dialog box. Then, you assign a value to the selected attributes. Finally, you can combine attributes using Boolean operators to set your query filter conditions.

In this example, the alarm query retrieves alarm data that meets the following conditions:

- Alarm priority (1-999)
- Alarm state (All, Acknowledged, or Unacknowledged)
- Query type (Summary or Historical)
- Alarm group (Local or remote data sources)

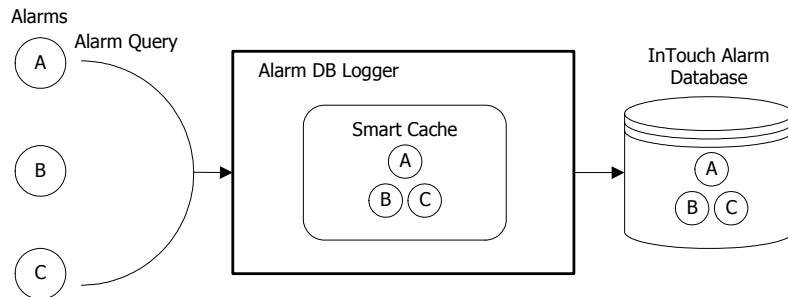
Key Points

- You can save your alarm queries to a query favorites file, which is an XML file. During run time, you can update the alarm display with new alarm data by running another query using selection criteria saved to the file.
- You can write QuickScripts that include query functions or dotfields to select records from alarm memory.

For more information, see Chapter 3, Alarm Queries, in the *InTouch® HMI Alarms and Events Guide*.

Saving Historical Alarm Data

The InTouch Distributed Alarm system includes the Alarm DB Logger utility that logs alarms and events to the alarm database. You can configure the Alarm DB Logger with one or more queries to select alarms from InTouch alarm providers.



Alarms and events are stored in a transient memory cache called the Smart Cache. The Alarm DB Logger writes the contents of the Smart Cache to the alarm database as alarm and event records at a periodic interval.

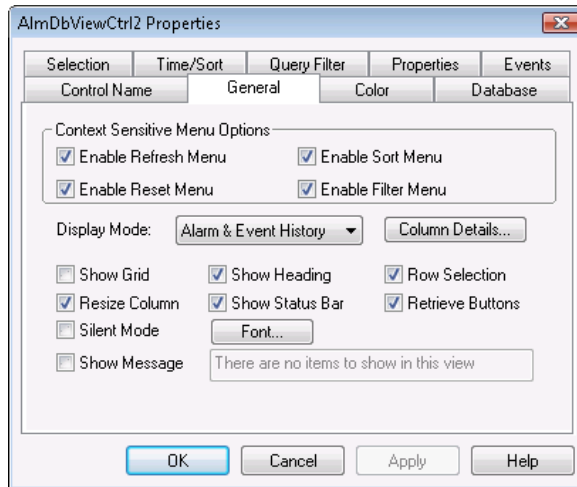
Key Points

- You use the Alarm DB Logger Manager to start and stop logging to the alarm database.
- You can configure the Alarm DB Logger to run as a Windows Service on the computer running the InTouch application, or as a normal application.
- You use the Alarm DB Logger Manager configuration wizard to define one or more queries to select alarms from InTouch alarm providers.
- You use the Alarm DB Logger Manager to set the frequency at which alarm records are written from internal memory Smart Cache to the alarm database.
- The Alarm DB Logger can auto-reconnect to the alarm database when the connection is lost. Logging resumes when the connection to the alarm database is re-established.
- The Alarm DB Logger only works with SQL Server authentication and the SQL Server authentication must be set to mixed mode.

For more information, see Chapter 9, Recording Alarms into an Alarm Database, in the *InTouch® HMI Alarms and Events Guide*.

Viewing Historical Alarms

You can use the Alarm DB View ActiveX control to show historical alarm and event data stored in the alarm database.



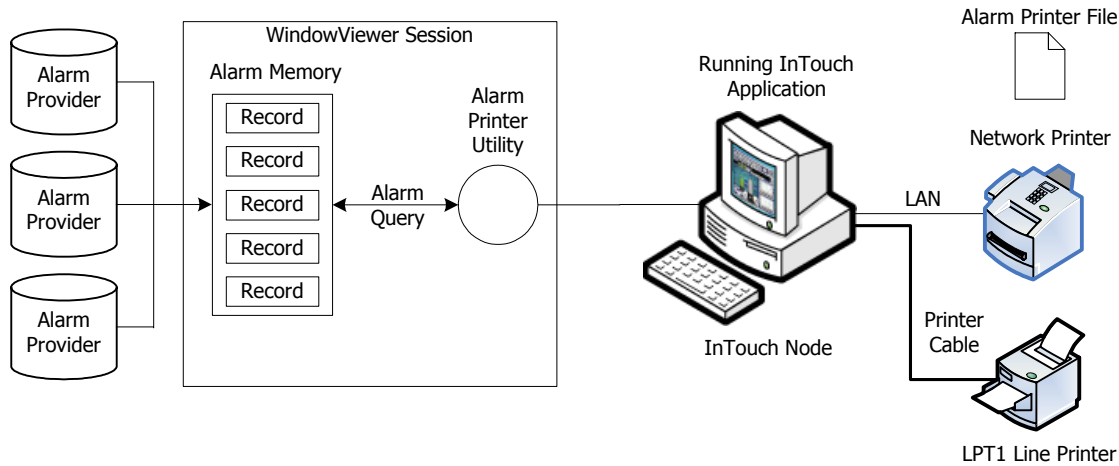
Key Points

- You must configure a connection to the alarm database. Use an account with appropriate read-only access to the alarm database and not a system administrator account.
- You can configure an alarm database query with the Alarm DB View control to show alarm records, event records, or both.
- You can set query values to select alarm records based on the selected time. You can also configure the maximum number of records to view, the start and end time of the alarm query, and the query time zone.
- You can set a filter to select which records are included in your query results. For example, you can select a filter by the date of a record or the state of the alarm.
- During run time, users can change Alarm DB View control options to:
 - Sort alarm and event information within a column.
 - Update the historical alarm display with recent historical data.
 - Perform a different alarm database query.
 - Resize the width of a column.

For more information, see Chapter 10, Viewing Recorded Alarms, in the *InTouch® HMI Alarms and Events Guide*.

Printing Historical Alarms

You use the InTouch Alarm Printer utility to print alarms from multiple nodes. You can print alarm records stored in the alarm memory on an event-by-event basis using a dedicated line or network printer. Also, you can use the Alarm Printer to save alarm records to a file.



The Alarm Printer queries the alarm memory to select records to print or save to a log file. The query selects records from the internal alarm memory based upon alarm priority, current alarm state, and alarm group membership.

Key Points

- You can configure the Distributed Alarm system to print certain events on a line printer as they occur. Typically, you print alarms immediately to record information in the event of a catastrophic failure.
- You can run multiple instances of the Alarm Printer. Each instance of the Alarm Printer must be configured to print to a different printer and must be configured with a separate alarm query.
- You can save your printer configuration settings to a file that can be loaded before printing alarms. Each query logs all alarms specified in the Alarm Printer configuration file that is currently open.
- You can use Alarm Printer functions in scripts to control alarm printing. Alarm printer functions return an integer error code that indicates whether the print operation completed successfully or not.

For more information, see Chapter 8, Printing Alarms, in the *InTouch® HMI Alarms and Events Guide*.

Managing Current Alarm Data

When tag data transitions from a normal to an alarm state, a new instance of its alarm is generated. The InTouch Distributed Alarm system tracks each alarm instance through the following states:

- When the tag first enters an alarmed state.
- When the alarm makes a sub-state transition, if the alarm is a multi-state alarm.
- When the alarm returns to normal.
- Whether the alarm is waiting for an acknowledgment.
- When the alarm is acknowledged.

The life cycle of an alarm instance ends when the tag value associated with the alarm returns to a normal, unalarmed state. A subsequent transition to the alarmed state generates a new alarm instance.

Key Points

- You can create scripts that use dotfields to acknowledge all current alarms, selected alarms, or only alarms of a specific type.
- You can use acknowledgment script functions to acknowledge any alarms that the Distributed Alarm system can query (summary display only).
- The InTouch HMI can automatically acknowledge an alarm when an alarmed tag's value returns to its normal state. This option does not apply to expanded summary alarms.
- Operators acknowledge alarms that appear in the Alarm Viewer control from WindowViewer. Acknowledged alarms are removed from the display.
- Operators can include alarm and acknowledgement comments in response to an alarm.
 - An alarm comment is set when the new alarm instance occurs. Alarm comments can be up to 131 characters.
 - An operator writes the comment when acknowledging an alarm. You can use the acknowledgement comment to update the alarm comment in the tag database.

For more information, see Chapter 5, Acknowledging Alarms in Real Time, in the *InTouch® HMI Alarms and Events Guide*.

Viewing Current Alarms

The InTouch Alarm Viewer ActiveX control can show summaries of current alarms or listings of historical alarms.

Time /	State	Class	Type	Priority
12/14/2006 10:15:11 AM	UNACK_RTN	VALUE	HI	1
12/14/2006 10:15:13 AM	UNACK	VALUE	HI	1

Displaying 1 to 2 of 2 alarms. Default Query 100 % Complete

The Alarm Viewer control has scroll bars, sizable columns, multiple alarm selections, and an update status bar. The Alarm Viewer control shows current alarms with specific colors to indicate whether they are acknowledged or not.

Key Points

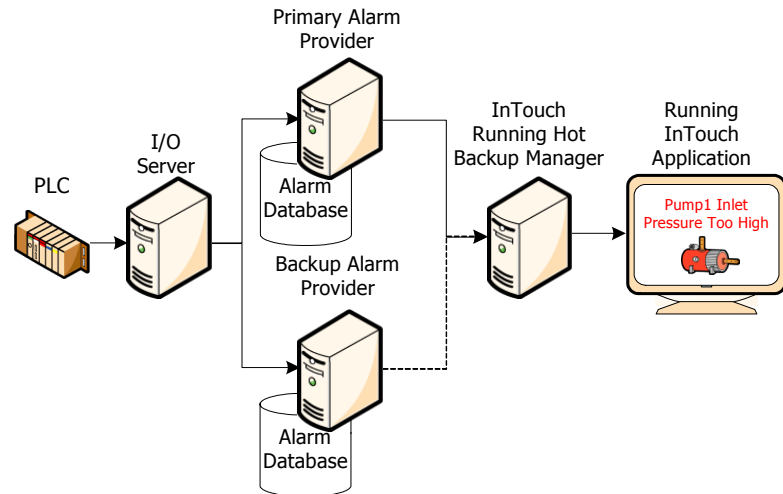
- You should use the Alarm Viewer control to view InTouch current alarms. However, you can continue to use the Distributed Alarm object to view alarms from applications created with versions of InTouch earlier than 7.1.
- You can configure the Alarm Viewer control to run an ActiveX event script when the Alarm Viewer control detects a new unacknowledged alarm.
- The original alarm time shown from the Alarm Viewer control is the date/time stamp of the onset of the alarm. If tag is an I/O tag, then it is the time stamp from the I/O Server if that server is capable of passing time stamps.
- You can sort the alarm records shown in the Alarm Viewer control. By default, alarm records are listed by their date/time stamp in ascending order.
- You can set Alarm Viewer control options that permit users to modify the Alarm Viewer control during run time. When users right-click on the Alarm Viewer control during run time, a context-sensitive (shortcut) menu appears with a list of commands to modify the control.
- Operators can select the **Query Favorites** command on the Alarm Viewer control's shortcut menu to select an alarm query from a list of previously defined alarm queries.

For more information, see Chapter 4, Viewing Current Alarms, in the *InTouch® HMI Alarms and Events Guide*.

Providing Alarm Redundancy

The InTouch Distributed Alarm system issues notifications and receives alarm acknowledgments from applications running on network nodes. Alarm provider applications store alarm data within their designated alarm databases.

For enhanced reliability, you can use the Alarm Hot Backup Manager to create a duplicate alarm provider. The following figure shows how the Hot Backup Manager uses a secondary current alarm repository as a backup provider.



The hot backup provides a single hot backup pair name that points to two alarm providers, the primary and the backup. InTouch HMI's alarm consumers, such as the Alarm Viewer control, reference this name to retrieve alarms from either the primary or the backup alarm provider.

Key Points

- If both provider nodes are operating normally, the alarm consumer receives alarm data from the primary provider. If the primary provider fails, the alarm consumer automatically switches to the backup provider.
- To synchronize alarm acknowledgements between the primary and backup providers, you must identify a combination of tag alarm record fields. This combination of fields generates a unique mapping key to the paired alarm records stored in each provider's current alarm repository.
- The Hot Backup Manager synchronizes alarm acknowledgements between the primary and backup providers.

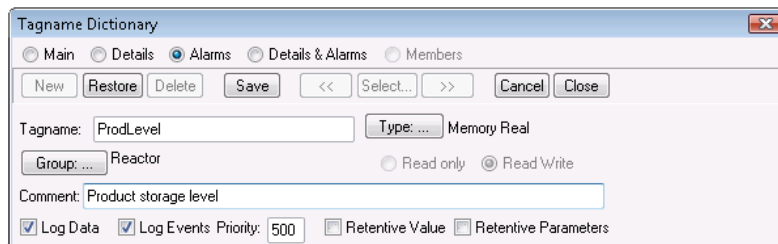
For more information, see Chapter 13, *Enhancing Plant Security Through Alarm Redundancy*, in the *InTouch® HMI Alarms and Events Guide*.

About InTouch Events

An event is a detectable occurrence of something happening within an InTouch application, which may or may not be associated with an alarm. Events represent normal system status messages and do not require an operator response.

An event is different than a condition. A condition can persist for minutes, hours, days, or weeks. An event is momentary; it takes place and is immediately over. An alarm is a condition; an alarm notification is an event.

You specify event logging from the Tagname Dictionary, which includes options to select event logging and an event priority for the tag.



Key Points

- When you define a tag to do event monitoring, an event message is logged to the alarm system each time the tag's value changes.
- An event message describes how the tag value changed and whether the operator, I/O, scripts, or the system initiated the change.
- When you define a tag to do event monitoring, you can also choose to have event messages printed each time the tag value changes.
- Events are assigned a priority, which is a number between 1 and 999. The highest event priority is 1. By classifying events with a numerical scale, you can write scripts that select event data within a defined priority range from the alarm system.

For more information, see Chapter 1, Overview of Alarms and Events, and Chapter 2, Configuring Alarms, in the *InTouch® HMI Alarms and Events Guide*.

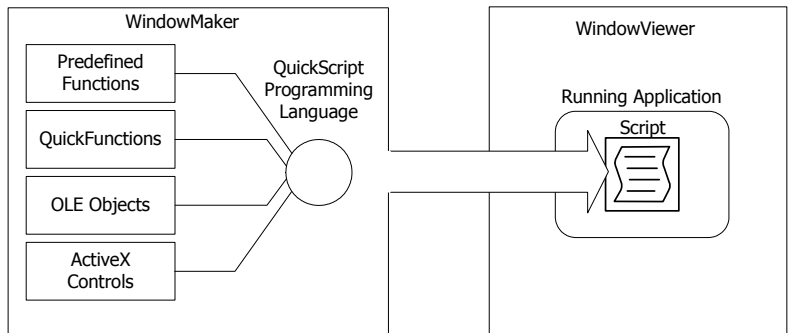
Chapter 6

Scripting and Logic

You can write scripts to monitor and manage aspects of your InTouch applications. A script is a set of programatic instructions that directs an InTouch application to perform an action.

You write InTouch scripts with the QuickScript language. Using QuickScript, you can write scripts that include conditional branching, code looping, and local variables.

The following figure shows the code components that can be included in an InTouch script.



Scripts can be classified by when they are run and whether they run independently of other ongoing application processes.

Scripts can generally be run in two different ways:

- Event-based scripts run once when an event occurs. For example, an event-based script can run after an operator presses a key or a tag value changes.
- Time-based scripts run periodically while a condition is fulfilled. For example, a time-based script can run while a window is open or a button is kept pressed.

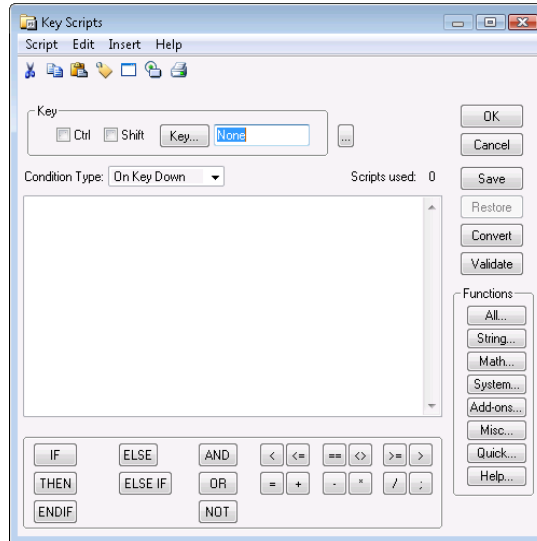
You can configure multiple event-based and time-based scripts to run with the same trigger. For example, you can configure a script to run once when a key is pressed and another script to run periodically every 5 seconds while the same key remains pressed.

For condition scripts you can either run a script synchronously or asynchronously.

- When a synchronous script runs, all InTouch animation and tag value updating stops. Then, animation and tag value updating resumes after the script stops.
- When an asynchronous script runs, all InTouch animation and tag value updating continues during the period when the script runs.

InTouch Script Language

The InTouch script language is used to automate repetitive functionality within an application. You use WindowMaker's **Script Editor** dialog box to create and edit your scripts.



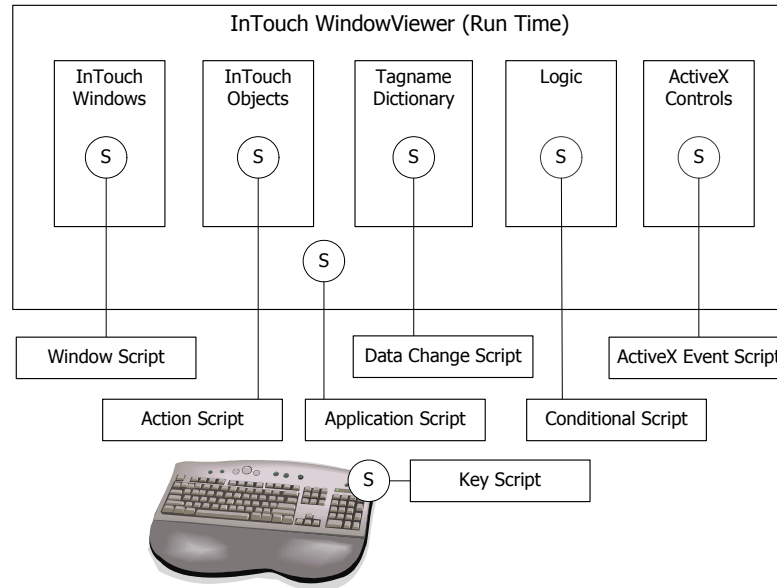
Key Points

- A script statement can be a value assignment, a function call, or a control structure. Each statement must end with a semicolon (;).
- An InTouch script supports the following types of operators:
 - Mathematical (addition, subtraction, multiplication)
 - Boolean (AND, OR, NOT)
 - Bitwise (bitwise AND, bitwise OR)
 - Shift (shift left, shift right)
 - Comparison (less than, equal, greater than)
- An InTouch script only supports conditional branching using the IF-THEN-ELSE control structure.
- An InTouch script only supports FOR loops. A FOR loop monitors the value of a numeric loop variable that increments or decrements with each loop iteration. The script keeps looping until the value of the loop variable reaches a defined limit.
- You can declare local script variables to store temporary or intermediate results. Using local variables increases script performance and minimizes the tag count.

For more information, see Chapter 4, The Script Language, in the *InTouch® HMI Scripting and Logic Guide*.

Script Types

When you create a script in the InTouch HMI, you assign to it a trigger that starts the script. The trigger identifies the type of script. The following figure shows the various types of triggers that can run an InTouch script.



Key Points

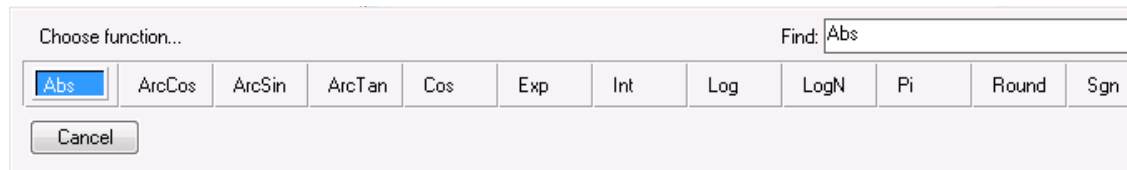
- A window script runs when an InTouch application window opens, closes, or remains open.
- An application script runs when WindowViewer starts, stops, or keeps running.
- An action script runs when an operator interacts with an object in WindowViewer, such as clicking on a window object or moving the mouse pointer over it.
- A data change script runs when the value of a tag changes.
- A conditional script runs when a certain logical condition is true or false, or remains true or false.
- A key script runs when a key or key combination is pressed, released, or while a key remains pressed.
- An ActiveX event script runs when an ActiveX event occurs, such as a click on the ActiveX control.

For more information, see Chapter 3, Script Triggers, in the *InTouch® HMI Scripting and Logic Guide*.

Predefined Script Functions

The InTouch HMI includes a large library of predefined functions you can insert in your InTouch scripts. Predefined functions provide most of the common utilities you need to manage data and your InTouch applications.

Predefined functions are organized by functional groups. After you select a group, then you select the predefined function to insert in your script. You use the **Choose function** dialog box that appears after selecting the predefined math functions group.



The predefined function is placed in your script at the current position of your pointer when you select the function.

Key Points

- You must associate the returned value of a predefined function to a tag or expression in your script. For example:

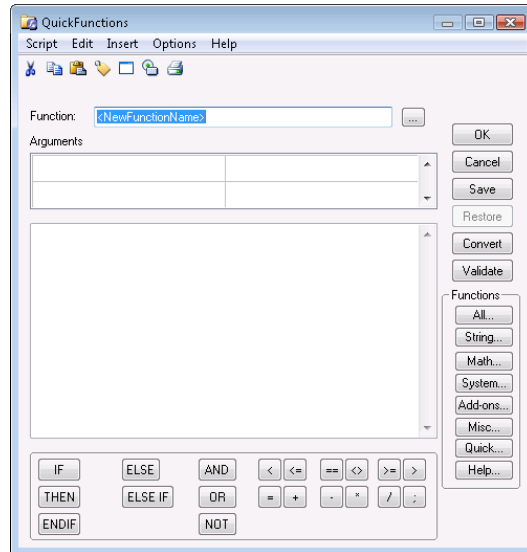

```
NodeName = IOGetNode("ModbusPLC1");
```
- Predefined mathematical functions include trigonometric, log, exponential, and arithmetic functions.
- String functions manipulate text strings associated with message tags, such as converting all characters in a message tag to uppercase.
- You can select functions from the System group to manage files and other non-InTouch applications.
- Recipe Manager, SPCPro, and SQL Access Manager are optional components that can be installed with the InTouch HMI. You can select unique predefined functions for each of these supplementary components.
- You can use miscellaneous functions to:
 - Manage application security.
 - Manage an application's alarms.
 - Configure historical and real-time trends.
 - Establish I/O connections to remote sources.

For more information, see Chapter 6, Built-In Functions, in the *InTouch® HMI Scripting and Logic Guide*.

Custom Script Functions

You can create custom functions known as QuickFunctions and store them in the QuickFunction library for reuse. QuickFunctions are scripts that you call from other scripts and animation links. In other programming environments, a QuickFunction is known as a macro, subroutine, or procedure.

You create your QuickFunction from WindowMaker's **QuickFunctions** dialog box.



When you create a QuickFunction, you must assign a name, any arguments, and the script body with optional return values. Arguments are local variables that exist only within the QuickFunction in which they are defined. You can specify up to 16 arguments for a QuickFunction.

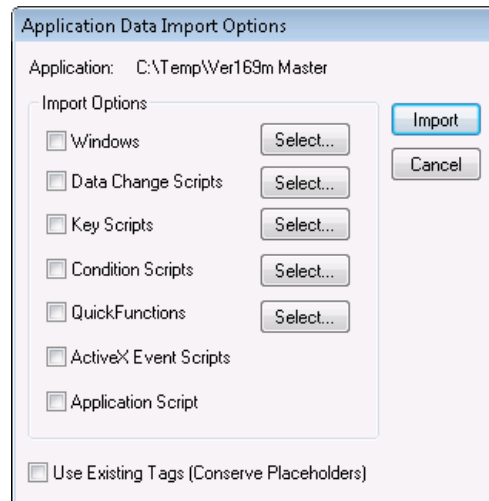
Key Points

- QuickFunctions can run asynchronously. Unlike other scripts, a QuickFunction can run in the background without affecting the main script flow. Running a QuickFunction asynchronously can be used for time-consuming operations like SQL database calls.
- Inserting reusable QuickFunctions minimizes the amount of duplicate code in your scripts.
- You start a QuickFunction using the CALL function in either an animation link or another script.
- To return a result from a QuickFunction, add RETURN *value* to your script. The script stops at the RETURN command and continues at the calling function.

For more information, see Chapter 5, Custom Script Functions, in the *InTouch® HMI Scripting and Logic Guide*.

Importing Scripts

As in the case with windows, you can import scripts from one InTouch application to another to reduce your development time. WindowMaker includes the **Application Data Import Options** dialog box to import scripts from an application's folder.



To import specific scripts, you click on the type of script and then select the script by name from the list of all scripts of the type you selected.

Key Points

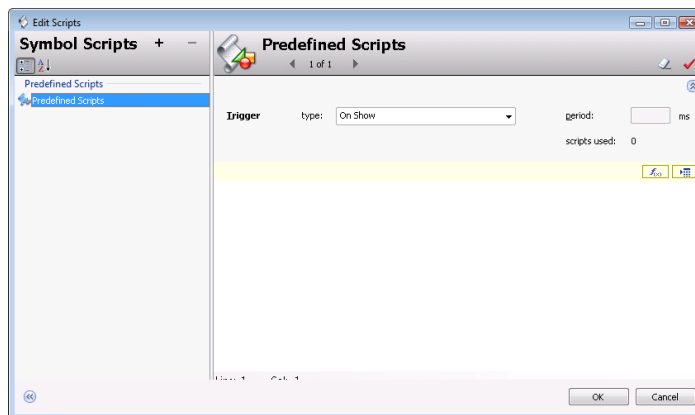
- By default, placeholder tags are created for the tags associated with an imported QuickScript.
- After you import the script, you can convert placeholder tags to local tags or remote tag references.
- If the tags associated with scripts already exist in the target application, you can continue to use the existing tags rather than import tags associated with the scripts.
- To import a window script, you must import the entire window.
- For an imported ActiveX Event script to work properly in the target application, the same ActiveX control and the same event for which the script was originally created must also be used in the target application.

For more information, see Chapter 4, *Exporting and Importing Tag Definitions, Windows, and Scripts*, in the *InTouch® HMI Application Management and Extension Guide*.

Associating Scripts with ArcestrA Symbols

You can associate scripts with the ArcestrA symbols placed in your InTouch applications. You can use scripts to animate your symbols or modify their elements while an InTouch application is running.

After selecting an ArcestrA symbol embedded in an InTouch window, you select the expression or reference whose value triggers the script to run. You use the ArcestrA Symbol Editor to select the trigger that runs the script.



Key Points

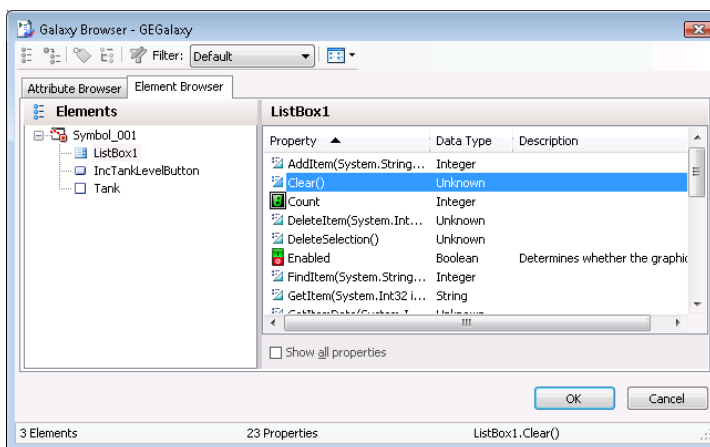
- A symbol script can be either predefined or named. A predefined script runs based on the status of the symbol in the running application. A named script runs when an expression or references associated with the script changes state.
- Predefined scripts are similar to InTouch window scripts. Based upon how you configure the script trigger, a predefined symbol script can run:
 - Once after the symbol opens or is shown.
 - Periodically while the symbol appears in the running application.
 - Once after the symbol closes or is hidden.
- Based upon how you configured the script, a named symbol script can run when the trigger values or expressions are true, false, or transitioning between true and false states. Also, a named symbol script can run when data associated with the trigger expression changes value or its quality state changes value.

For more information, see the Application Server IDE documentation.

Using Methods in ArchestrA Symbol Scripts

Some ArchestrA elements support script methods. These methods can perform various functions on the elements themselves at run time. Typically, you configure an action script to access these methods.

You can see the supported properties and methods of any ArchestrA element by opening the Galaxy Browser and selecting the element.



Key Points

- You can run a script containing an Edit Box control method to load text from a file to the control during run time. You can also run a script to save the current contents of the Edit Box control to a file during run time.
- Edit Box control methods are declared in scripts in the following form:

```
ControlName.SaveText(FileName);
```

where *ControlName* is the name of the Edit box control and *FileName* is the name of the file containing the contents of the control to be loaded or saved. In the example above, SaveText is the name of the method to save the contents of the Edit Box control to a file.

- You can use a script containing the methods of the Combo Box and List Box controls to change the contents of their lists during run time. List items can be added, deleted, or modified.
- Combo Box and List Box control methods are declared in scripts similar to the Edit Box control.

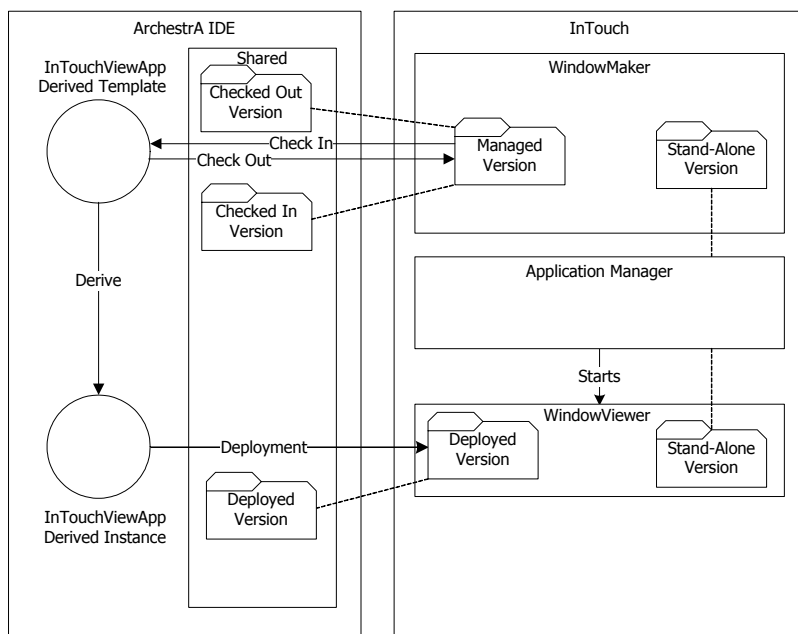
For more information, see the Application Server IDE documentation.

Chapter 7

Applications at Run Time

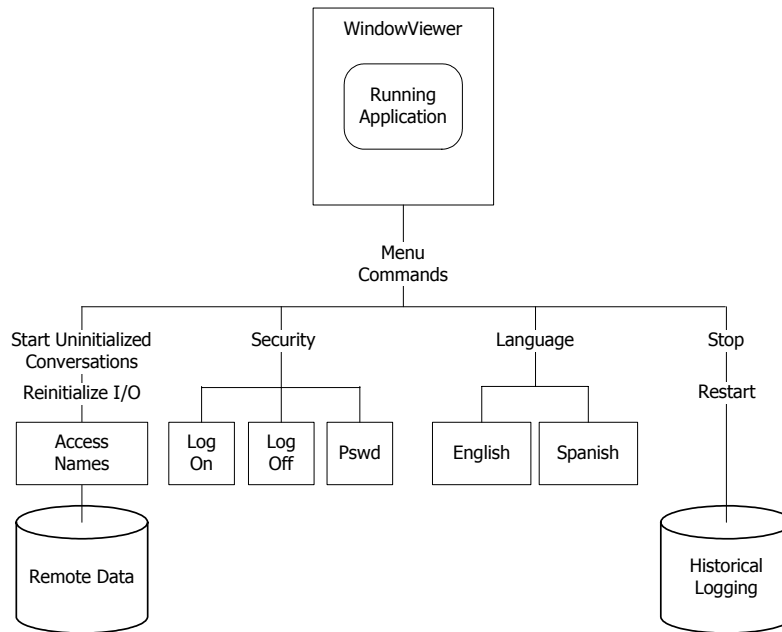
WindowViewer is the run-time environment for your InTouch applications. Based upon your application's requirements, you can configure properties that determine the visual appearance and operational characteristics of WindowViewer.

You can run stand-alone and managed InTouch applications with WindowViewer. The following figure shows that a managed application is deployed from the ArchestrA IDE to the computer running the InTouch HMI. The first time you deploy an InTouchViewApp object, the associated InTouch application files are copied to the node of the platform that hosts the object.



Issuing Commands During Run Time

By default, WindowViewer shows menu commands that control some aspects of a running application. You can use these commands to test a running application. If permitted, operators can select some or all of these commands to change some characteristics of WindowViewer.



Key Points

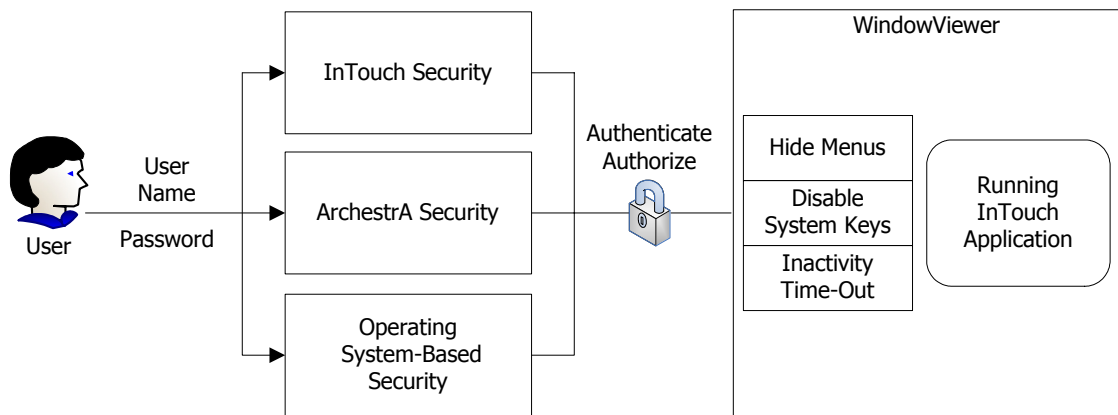
- You can configure properties to hide WindowViewer menus, commands, and standard window components. Hiding menus and commands ensures that operators can only interact with an InTouch application from their computers.
- Operators can select commands to start all uninitiated conversations with data sources that use DDE and SuiteLink communications protocols.
- Operators can select commands to attempt to reinitialize I/O connections to some or all remote sources.
- InTouch applications can be supported in multiple languages. If an application is configured for multiple languages, operators can switch to another language while an application is running.
- Operators can select commands to stop and then restart logging of an application's historical data.

For more information, see Chapter 7, Viewing Applications at Run Time, in the *InTouch® HMI Application Management and Extension Guide*.

Enforcing Security in WindowViewer

You can protect your InTouch applications as they run in WindowViewer by two different forms of security:

- Authenticate and authorize WindowViewer users with ArchestrA, operating system-based, or traditional InTouch security.
- Restrict operator interaction with a running application by setting WindowViewer time-out periods, locking system keys, and hiding window screen elements.



Key Points

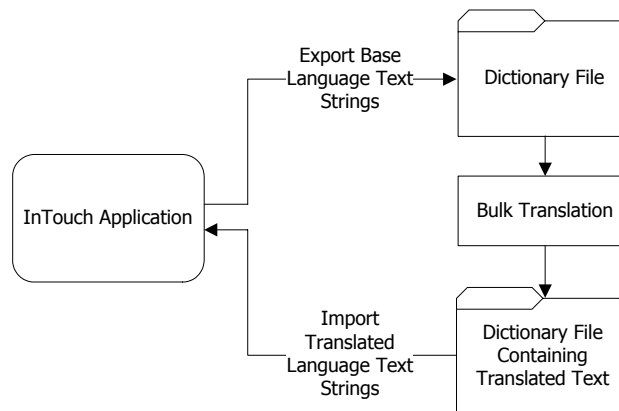
- All types of InTouch security authenticate users during the logon process with a user name and password combination. Each type of security provides a different mechanism to verify the user name and password during the logon process.
- Based upon the type security that protects an application, access to WindowViewer functions is granted according to the user's membership in a group, assigned access level, or priority.
- You can use scripting to automatically log off an inactive operator. This prevents unauthorized access to your InTouch application when operators leave their computers unattended.
- You can disable system keys to prevent operators from using standard Windows keyboard commands to switch to another application from WindowViewer.

For more information, see Chapter 5, *Securing InTouch*, in the *InTouch® HMI Application Management and Extension Guide*.

Switching the Language of an Application

You can develop applications that can be switched to another languages from WindowViewer. Every InTouch application is associated with a base language used to develop the application. You must configure any additional languages that you want to be part of your application.

In addition to switching the run-time language of an application's text strings, you can also configure run-time language switching of alarm comments, alarm states, and alarm types, which are shown in the Alarm Viewer and Alarm DB View controls.



Key Points

- Operators can switch a running application's language by using the **Language** command from the WindowViewer **Special** menu.
- You can also add a button to an application window that operators select to switch to another language. You associate the button with either the `$Language` system tag or the `SwitchDisplayLanguage()` function in a script.
- If your InTouch application contains many strings, you typically export the strings for bulk translation to a dictionary file.
- When you export strings, a dictionary file is created for each language that you export. Also, a separate dictionary file is created for each SmartSymbol in your application.
- You import dictionary files containing translated text into your InTouch application.

For more information, see Chapter 6, Switching a Language at Run Time, in the *InTouch® HMI Application Management and Extension Guide*.

Running InTouch Applications at a Different Screen Resolution

WindowViewer can dynamically change the screen resolution of an InTouch application. Dynamic Resolution allows you to develop a single InTouch application that can run on multiple WindowViewer client computers using a variety of screen resolutions.

Typically, you create and maintain an InTouch application on a single development node in a NAD development environment. Then, the application is distributed to several View nodes. Dynamic Resolution allows the client view nodes to show the application, even if the nodes are running at different screen resolutions.

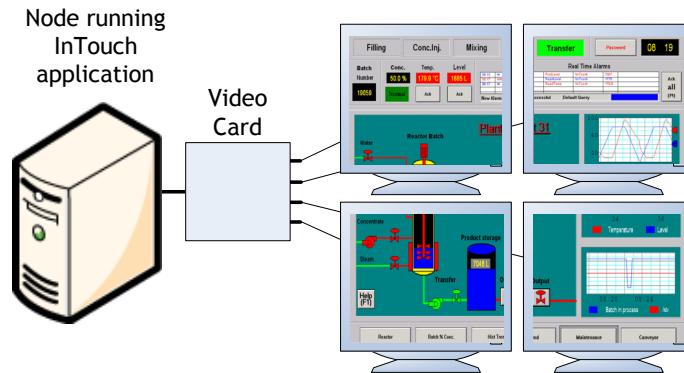
Key Points

- Dynamic Resolution enables each computer running WindowViewer to scale the application screen resolution to the:
 - Original screen resolution.
 - Screen resolution of the machine that WindowViewer runs on.
 - Custom screen resolution of your choice.
- WindowViewer scales the screen resolution while compiling the application and does not require WindowMaker.
- Because each View node can use a different Dynamic Resolution setting, you must configure each individual View node.

For more information, see Chapter 2, Distributing Applications, in the *InTouch® HMI Application Management and Extension Guide*.

Running InTouch Applications on Multiple Monitors

A multi-monitor system shows an InTouch application on several monitors simultaneously. Together, a multi-monitor configuration creates a composite screen composed of all monitors connected to the computer running an InTouch application. Each monitor can show a portion of the screen or only a single window component like a keypad.



Key Points

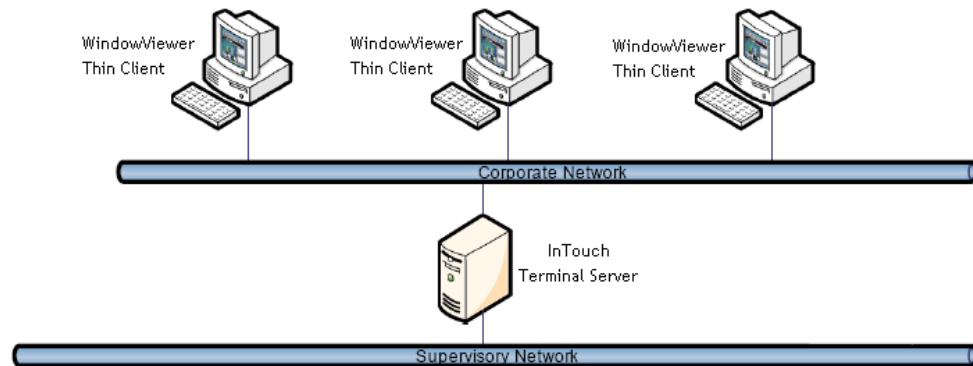
- You can set up two basic multi-monitor configurations: single video card or multiple video card configurations. Each configuration has unique hardware, software, and configuration requirements.
- The composite screen resolution is the sum of the individual horizontal and vertical resolution of each monitor in the configuration.
- You assign values to `MultiScreen`, `MultiScreenWidth`, and `MultiScreenHeight` parameters of the `Win.ini` file to enable multi-monitor support for an application.
- You can specify the `ScaleForResolution` parameter of the `InTouch.ini` file to maintain the current resolution of application windows when migrating between computers that run `WindowViewer` at different screen resolutions.

For more information, see Chapter 8, *Setting Up a Multi-Monitor System*, in the *InTouch® HMI Application Management and Extension Guide*.

Running InTouch in a Terminal Services Environment

InTouch for Terminal Services is a variation of the regular InTouch version and is intended for computers running server versions of Windows with Terminal Services enabled.

You can use InTouch for Terminal Services to run InTouch on one central server and supply InTouch functionality to multiple client computers without imposing any further software or hardware requirements on the client computers. In this environment, the hardware and software requirements for the server are relatively high and those for the clients relatively low. This results in lower total cost of ownership (TCO) and lower ongoing operating expenses.



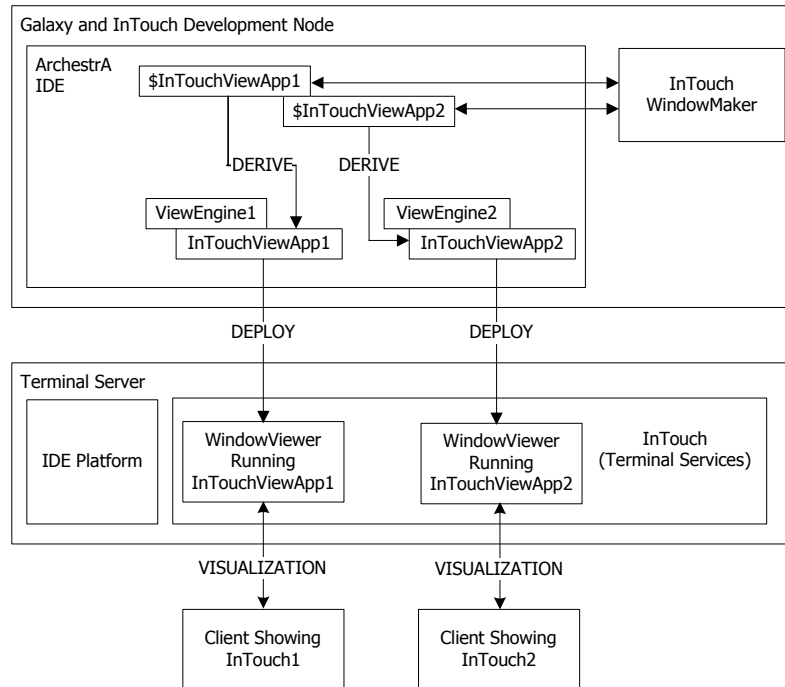
Key Points

- InTouch for Terminal Services uses the Remote Desktop Protocol (RDP) to communicate between clients and the InTouch Terminal Server.
- Each client computer runs an individual InTouch session on the Terminal Server without interacting with other client sessions.
- You can run an application that is developed for standard InTouch with InTouch for Terminal Services. No changes must be made to the application.
- You can use the Distributed Alarm system with Terminal Services for InTouch. Using the alarm client, you can select the alarm data and how to show it from WindowViewer for each Terminal Services session.
- When an alarm is acknowledged in a Terminal Services environment, the Operator Node that gets recorded is the name of the client computer that the respective operator established the Terminal Services session from.

For more information, see Chapter 2, *Distributing Applications*, in the *InTouch® HMI Application Management and Extension Guide*.

Running a Managed InTouch Application with Terminal Services

You can run managed InTouch applications in a Terminal Services environment. The benefit of using Terminal Services is that it allows you to run multiple, autonomous InTouch applications simultaneously on a Terminal Server.



Key Points

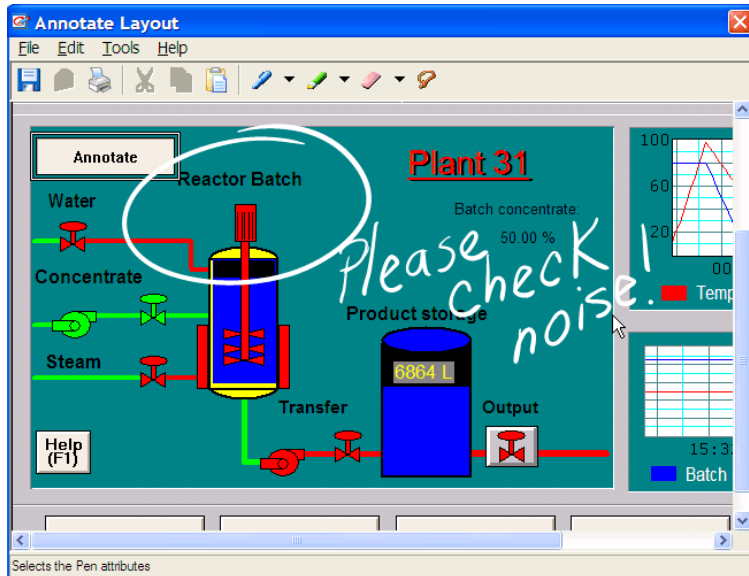
- In a typical Terminal Services architecture, application development, deployment, and client visualization are placed on separate computers.
- You must deploy each InTouchViewApp instance with its individual Archestra ViewEngine host.
- You must deploy each InTouch application to the server running InTouch for Terminal Services.
- You run each managed InTouch application in a separate terminal-services client session.

For more information, see Chapter 4, Using IDE-Managed InTouch Applications at Run Time, in the *InTouch® HMI and Archestra® Integration Guide*.

Running InTouch on a Tablet PC

A Tablet PC uses wireless technology for networking. It is waterproof and vibration resistant, making it suitable for most industrial environments.

You can include the `AnnotateLayout()` function in a script to capture screens shown on a Tablet PC. Running the script takes a screen capture of the visible portion of the active InTouch window. The captured screen appears in the **Annotate Layout** dialog box.



The **Annotate Layout** dialog box includes a toolbar and menu commands. The dialog box shows the screen capture in its client area.

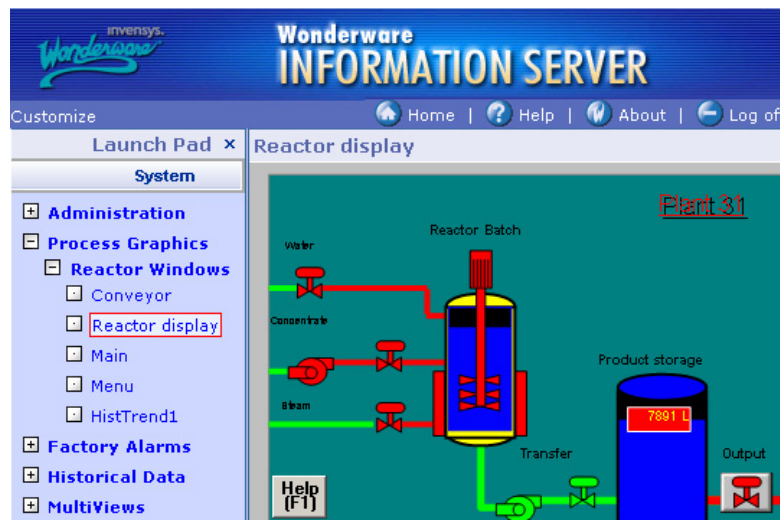
Key Points

- The Tablet PC version of the InTouch HMI can run on the Windows XP Tablet PC Edition operating system. The Windows Vista operating system is also supported with InTouch version 10.
- You can annotate your screen capture by using the Tablet PC's digital pen to write notes and highlight information.
- You can save the screen capture with your annotations as a .gif or .jpeg file.
- If you have the Tablet PC configured with an e-mail client, you can send the screen capture as a file attachment to a message.

For more information, see Chapter 9, Using InTouch on a Tablet PC, in the *InTouch® HMI Application Management and Extension Guide*.

Running InTouch Windows over the Internet

The Wonderware Information Server is a web portal application that can aggregate and present plant production data over the web or company intranet.



Key Points

You can use InTouch with Wonderware Information Server in the following ways:

- Process visualization

You can publish InTouch applications to the Wonderware Information Server portal to show your production process and controls through a Web browser.
- Data interaction

You can use the Wonderware Information Server portal to read values from and write back values to InTouch tags. This enables you to interact with your plant processes without using an InTouch client.
- Alarm Display

You can use the Wonderware Information Server portal to show InTouch real time and historical alarm data.
- Historical Data Display

You can use the Wonderware Information Server portal to show InTouch historical data saved in a Wonderware Historian database.
- Table Weaver display

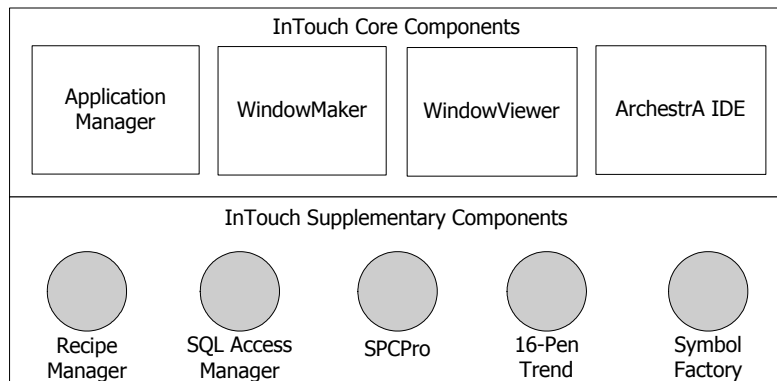
You can use InTouch to create displays for Table Weaver content units.

For more information, see the Wonderware Information Server documentation.

Chapter 8

Supplementary Components

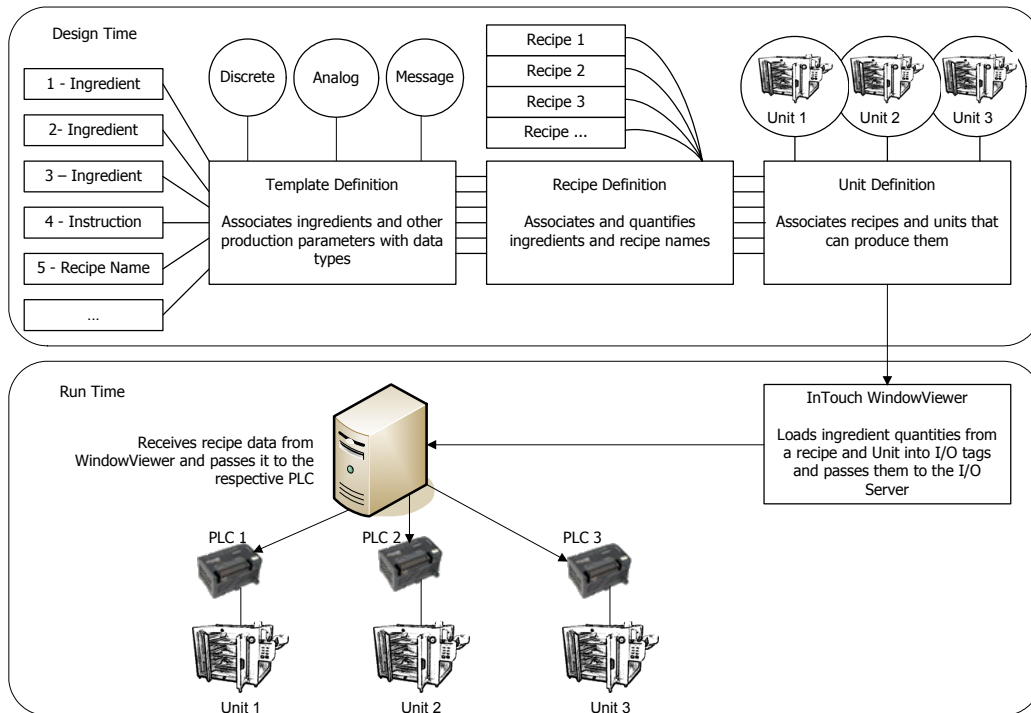
You can optionally install a set of five supplementary components along with InTouch core components. These supplementary components provide additional functions for your InTouch application.



- Recipe Manager includes a set of spreadsheets and script functions to create manufacturing recipes.
- SQL Access Manager consists of a program and a set of SQL functions to store InTouch data to a database.
- SPCPro includes a set of statistical analysis tools to measure the quality of a manufacturing process.
- 16-Pen Trend includes a trend wizard and script functions to create real-time and historical trends.
- Symbol Factory provides a set of industrial symbols that can be placed in InTouch applications to represent process components.

Recipe Manager

Recipe Manager is an optional component that can be installed with InTouch HMI. Recipe Manager simplifies the process of creating manufacturing recipes with a spreadsheet-like editor to create templates and a set of script functions to access data within the templates.



Key Points

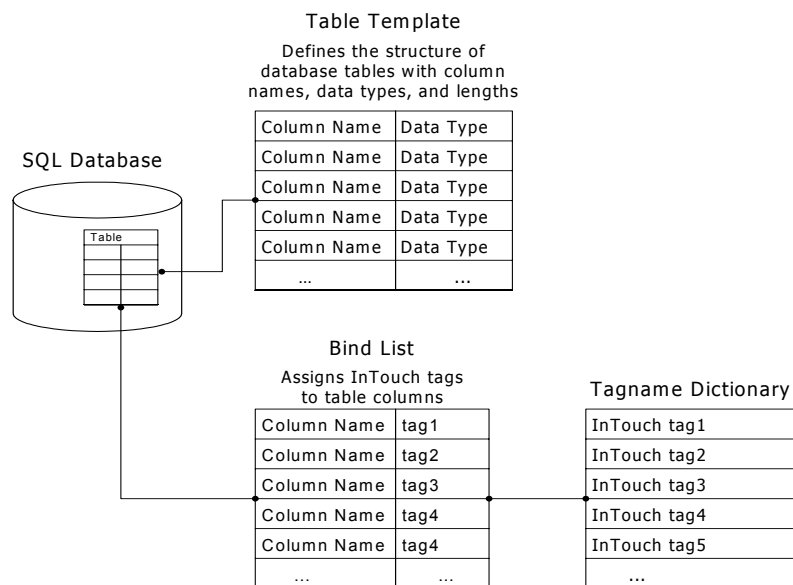
- The Recipe Manager utility provides a spreadsheet-like user interface to create and maintain a recipe template file. A file consists of three templates. You create and edit these templates by adding or modifying data within the cells of each template's spreadsheet
- You use InTouch QuickScripts to interact with your recipe template files. Recipe Manager includes a set of script functions that can be inserted into QuickScripts. Using scripts containing these functions, you can select, modify, insert, or delete records in your recipe template file.
- You can restrict user access to recipes by defining an Item Name in the recipe template file that sets the required security access level required to load, save, or delete a recipe.

For more information, see Chapter 2, Using Recipe Manager, in the *InTouch® HMI Supplementary Components Guide*.

SQL Access Manager

SQL Access Manager is an optional component that can be installed with InTouch HMI. You can use SQL Access Manager to transfer data, such as batch recipes from a SQL database to an InTouch application. SQL Access Manager can also be used to transfer run-time data, alarm status, or historical data from an InTouch application to a database.

SQL Access Manager consists of a program and a set of SQL functions. The SQL Access Manager program creates and associates database columns with InTouch tags. Binding is the process of associating database columns to InTouch tags. Binding allows you to use SQL Access Manager to directly manage InTouch data stored in a database.



Key Points

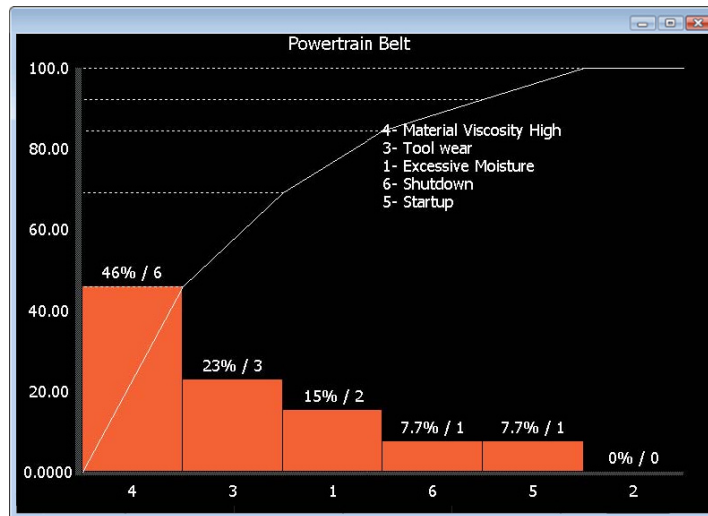
- SQL Access Manager functions can be used in scripts that automatically run based on operator input, a tag value changing, or when a particular set of conditions exist. These functions allow you to select, modify, insert, or delete records in the tables you choose to access.
- SQL Access Manager saves the database field names and their associations in a comma-separated variable file named SQL.DEF.
- SQL Access Manager also creates Table Templates that define the structure and format of the database used with the InTouch HMI.

For more information, see Chapter 3, Working with SQL Databases from InTouch, in the *InTouch® HMI Supplementary Components Guide*.

SPCPro

SPCPro is an optional Statistical Process Control (SPC) analysis utility that can be installed with InTouch HMI. SPCPro extends InTouch process management with a set of statistical analysis tools to measure manufacturing quality.

With SPCPro, you can monitor real-time process quality using numerical methods and alarms for rule violations. SPCPro also provides open access to its data for additional post-process analysis or reporting.



Key Points

- Before you can use SPCPro, you must select a database to store your SPC configuration and collection data. SPCPro can be used with Microsoft Access or SQL Server.
- Use SPCPro wizards to create three types of SPC charts: control charts, histograms, and Pareto charts.
- You collect SPCPro data to a dataset automatically using time-based or event-based sampling.
 - Time-based collection saves data to the dataset at a fixed interval for a specified duration.
 - Event-based data collection requires a unique discrete tag that serves as a trigger to initiate a sampling period.
- You can use SPCPro wizards to manually collect data.
- For a more robust and flexible SPC tool, consider using the Wonderware QI Analyst product.

For more information, see Chapter 1, Monitoring Manufacturing Quality with SPCPro Statistical Process Control, in the *InTouch® HMI Supplementary Components Guide*.

16-Pen Trend

The 16-Pen Trend is an optional trend wizard that can be installed with InTouch HMI. With the 16-Pen Trend wizard, you can create real-time and historical trends capable of showing data from up to 16 tags or expressions.

The 16-Pen Trend Wizard can be configured much like other InTouch chart wizards. The 16-Pen Trend wizard includes a properties dialog box to configure your trends.

	Color	Tagname	EU Text	Min EU	Max EU	Min Scale	Max Scale	Dec.Pos	Width
1		???	0	100	0	1	1	1	1
2		???	0	100	0	1	1	1	1
3		???	0	100	0	1	1	1	1
4		???	0	100	0	1	1	1	1
5		???	0	100	0	1	1	1	1
6		???	0	100	0	1	1	1	1
7		???	0	100	0	1	1	1	1
8		???	0	100	0	1	1	1	1
9		???	0	100	0	1	1	1	1
10		???	0	100	0	1	1	1	1
11		???	0	100	0	1	1	1	1
12		???	0	100	0	1	1	1	1
13		???	0	100	0	1	1	1	1
14		???	0	100	0	1	1	1	1
15		???	0	100	0	1	1	1	1
16		???	0	100	0	1	1	1	1

Key Points

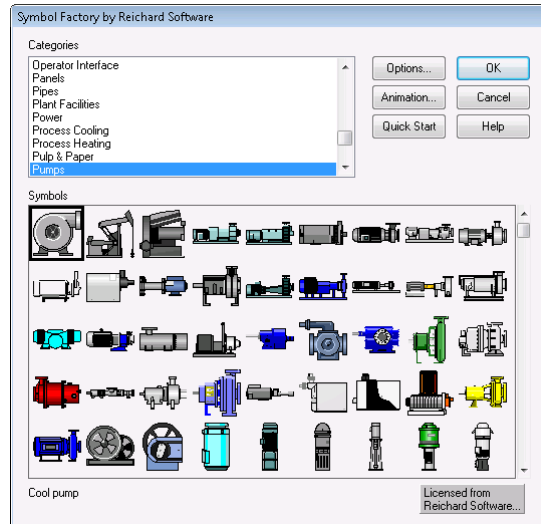
- You can configure the following for a 16-Pen Trend:
 - Tag or expression assigned to each trend pen
 - Trend line width and color
 - Starting and ending dates and times for historical trends
 - Update rate and time span for real-time trends
 - Engineering units assigned to a tag or expression
- You can allow operators to change some characteristics of a 16-Pen Trend while the InTouch application is running.
- You can use script functions to control the 16-Pen Trend at run time. For example, you can write scripts that connect pens to the chart, add new events to the chart, or change the time range shown on the trend's horizontal axis.

For more information, see Chapter 4, Using the 16-Pen Trend Wizard, in the *InTouch® HMI Supplementary Components Guide*.

Symbol Factory

Symbol Factory can be optionally installed with InTouch HMI. Symbol Factory is a collection of over 4,000 industrial symbols that you can use in your applications. For more robust and graphically rich symbols, use the symbols provided with the ArchestrA Symbol Library.

Symbols are organized by their functional categories. After you select a category, you can see the symbols from the category that you can place into an application.



Key Points

- From WindowMaker, you can insert Symbol Factory symbols into both stand-alone and managed applications.
- Any Symbol Factory symbol can be animated. Symbol Factory provides the most common animation links. If you want to use another type of animation link, you can break the symbol and animate it using standard InTouch animation links.
- You can add InTouch objects to Symbol Factory. If you add an InTouch object that has animation links associated with it, the links are also stored with the object. Any tags associated with the object are automatically converted to placeholder tags.
- You can import graphic Windows Meta Files (.wmf) or bitmap files (.bmp) into Symbol Factory. When you import either type of graphic file, it is automatically converted into a wizard.

For more information, see Chapter 5, Symbol Factory, in the *InTouch® HMI Supplementary Components Guide*.

Index

Symbols

\$AccessLevel system tag 27
 \$ApplicationChanged system tag 25
 \$ConfigureUsers system tag 27
 \$Language system tag 96
 \$Operator system tag 27

Numerics

16-Pen Trend 44, 107

A

Access Names
 creating 64
 description 64
 setting primary and secondary I/O sources 67
 action script 86
 ActiveX controls
 Alarm Pareto 42
 Alarm Tree Viewer 42
 Alarm Viewer 42
 AlarmDB View 42
 description 37
 using in InTouch applications 42
 AddPermission() function 28
 Alarm DB Logger
 description 74
 starting and stopping alarm logging 76
 storing records to a SQL Server database 76
 Alarm DB View control 77
 Alarm Manager
 description 74

Alarm Pareto ActiveX control 42
 Alarm Printer
 description 74, 78
 saving configuration files 78
 Alarm Tree Viewer ActiveX control 42
 Alarm Viewer ActiveX control 42, 80
 AlarmDB View ActiveX control 42
 alarms
 acknowledging 79
 adding an acknowledgment comment 79
 backup provider 81
 historical 77
 Hot Backup Manager 81
 managing current data 79
 printing 78
 queries 75
 setting query filter 77
 types 72
 using a query favorites file 75
 animation links
 description 37, 45
 interacting with 46
 AnnotateLayout() function 101
 Application Manager
 description 14
 introduction 14
 overview 15
 Application Publisher 31
 application script 86
 Application Server
 architecture 24
 description 24

- applications
 - converting older 32
 - converting screen resolution 33
 - default folder location 31
 - exporting and importing 31
 - setting security 30
 - types 16
- ArchestrA
 - animation links in symbols 45
 - application workflow 10
 - Graphic Toolbox 41
 - reusing SmartSymbols 49
 - securing InTouch applications 29
 - setting up security 29
 - symbols 41
 - ViewEngine Galaxy object 21
 - WinPlatform Galaxy object 21
 - workflow with InTouch 10
- ArchestrA Bulk Import Utility
 - description 57
 - migrating SuperTags 57
- ArchestrA IDE
 - creating a managed application 21
 - importing stand-alone application 31
 - overview 10, 15
- ArchestrA security
 - description 26
 - roles 29
 - Secured Write 29
 - Verified Write 29
- authentication
 - using operating system-based security 28

B

- basic objects
 - description 37
 - selecting 39

C

- cells 40
- clamping 59
- complex objects
 - creating 40
 - description 37
 - predefined 42
 - types 40

- condition scripts
 - description 86
 - run modes 84
- custom functions 88

D

- data change script 86
- DBDump 66
- DBLoad 66
- description 72
- deviation alarms 72
- discrete alarms 72
- discrete tags 54
- display links 46
- Distributed Alarm System
 - component parts 74
 - description 74
 - Smart Cache Buffer 74
- Distributed Name Manager 69
- documentation conventions 7
- dotfields 62
- Dynamic Resolution 97
- Dynamic Resolution option 33

E

- EnableDisableKeys() function 30
- events
 - description 82
 - priority 82

F

- functions
 - acknowledging alarms 79
 - AddPermission() function 28
 - Alarm Printer 78
 - AnnotateLayout() function 101
 - custom script 88
 - EnableDisableKeys() function 30
 - inserting into scripts 12
 - predefined 87
 - RestartWindowViewer() function 25
 - returning status of I/O sources 67
 - selecting alarm records from database 75
 - SwitchDisplayLanguage() function 96

G

graphic primitives 39
 Graphic Toolbox 41, 48

H

historical trends 44
 HMI, See Human Machine Interface
 Hot Backup Manager 81
 Human Machine Interface 9

I

I/O failover 67
 I/O tags
 alarm date/time stamp 80
 associating with an Access Name 64
 deadbands 60
 description 53
 scaling raw values 59
 setting remote references 63
 indirect tags
 description 53, 56
 types 56
 integer tags
 clamping and scaling 59
 description 54
 InTouch security
 access level 27
 authenticating users 27
 description 26
 InTouchView application
 creating 21
 description 20
 InTouchViewApp template
 creating a managed application 23
 deploying an instance 21
 deriving 21
 exporting a package file 31
 importing a stand-alone application 31

K

key scripts 86

L

language switching
 bulk translation 96
 description 96
 dictionary file 96
 licensing 35
 linear scaling 59
 links
 animation types 46
 creating movement 46
 display 46
 location links 46
 touch 46
 logging
 deadband 60
 historical data 68
 historical log files 68
 sampling interval 68
 starting and stopping 94
 starting and stopping with
 WindowViewer commands 13
 system tags 55
 logging deadband 60

M

managed application
 converting from a stand-alone
 application 34
 creating 21, 23
 creating symbols with Symbol
 Editor 18
 deploying 21
 description 16, 18
 editing 12
 exporting to another 31
 fast switching 18
 publishing a managed application 31
 memory tags 53
 message tags
 description 54
 string functions 87
 mode keyword 66
 multi-monitor system 98
 description 98

N

NAD, See Network Application Development
 Network Application Development
 change modes 25
 description 25

O

operating system-based security
 description 26
 inheriting Windows password policies 28
 overlay window 38

P

parameter retention 61
 popup window 38
 predefined functions 87
 published application
 description 16, 19
 publishing from a managed application 31

Q

QuickScripts
 database query functions 75
 description 87
 security 26
 using with Recipe Manager 104

R

rate-of-change alarms 73
 real tags
 clamping and scaling 59
 description 54
 equating to indirect tags 56
 retention 61
 Real Time trend 44
 Recipe Manager
 description 104
 templates 104
 Remote Desktop Protocol 99
 remote references
 description 63
 using Access Name 63
 replacement window 38
 RestartWindowViewer() function 25
 retention 61

S

scaling
 description 59
 linear 59
 square root 59
 scripts
 associating with ArcestrA symbols 90
 asynchronous 84
 conditional branching 85
 custom functions 88
 event-base 84
 importing 89
 operators 85
 predefined functions 87
 run modes 83
 supported programming constructs 85
 synchronous 84
 time-based 84
 triggers 86
 types 86
 security
 authenticating users 95
 authentication 27
 generic InTouch type 27
 locking system keys 30
 protecting InTouch applications 26
 restriction features 30
 supported types for InTouch 26
 using operating system-based authentication 28
 Smart Cache Buffer
 description 74
 writing contents to alarm database 76
 SmartSymbols
 description 40
 reusing 49
 SPCPro
 description 103
 sampling 106
 SQL Access Manager
 binding 105
 description 105
 square root scaling 59
 stand-alone application
 creating 22
 description 16, 17
 importing to ArcestrA IDE 31

- SuiteVoyager
 - description 102
 - use with InTouch 102
 - SuperTags
 - description 57
 - migrating with ArchestrA Bulk Import Utility 57
 - remote reference syntax 63
 - supplementary components
 - 16-Pen Trend 107
 - Recipe Manager 104
 - SPCPro 106
 - SQL Access Manager 105
 - summary 103
 - Symbol Factory 108
 - SwitchDisplayLanguage() function 96
 - Symbol Editor
 - associating with automated objects 41
 - description 18
 - graphic primitives 39
 - InTouch symbols 40
 - restrictions for a published application 19
 - re-using SmartSymbols 49
 - Symbol Factory 108
 - symbols
 - ArchestrA 41
 - description 40
 - user-defined complex objects 40
 - system tags
 - characteristics 55
 - description 55
- T**
- Tablet PC
 - description 101
 - sending annotated screen capture as an e-mail attachment 101
 - use with InTouch 101
 - Tagname Dictionary
 - creating tags overview 52
 - exporting tags 66
 - importing tags 66
 - setting tag properties 58
 - setting value ranges, measurement units, and an initial value 61
 - tags
 - \$AccessLevel system tag 27
 - \$ApplicationChanged system tag 25
 - \$ConfigureUsers system tag 27
 - \$Language system tag 96
 - \$Operator system tag 27
 - accessing I/O data by remote references 63
 - boundaries 59
 - clamping raw I/O data values 59
 - configuring for historical logging 68
 - converting from placeholder 47
 - data types 54
 - defining alarm threshold values 12
 - defining an Access Name 64
 - description 52
 - direct 56
 - discrete 54
 - exporting 66
 - I/O 53
 - importing 66
 - indirect 53, 56
 - initial value 61
 - integer 54
 - logging deadbands 60
 - measurement units 61
 - memory 53
 - message 54
 - modifying properties with dotfields 62
 - real 54
 - retention 61
 - scaling I/O values 59
 - SuperTags 57
 - system 55
 - types 53
 - user-defined 55
 - using in an application 12
 - value deadbands 60
 - value ranges 61
 - technical support, contacting 8
 - Terminal Services
 - description 99
 - running a managed application 100
 - use with InTouch 99
 - using Remote Desktop Protocol 99
 - touch links 46

trends

- 16-Pen 44, 107
- historical 43, 44
- objects 43
- Real Time trend 44
- script functions 87

V

- value alarms 72
- value deadband 60
- value retention 61
- ViewEngine Galaxy object 21

W

- window script 86
- WindowMaker
 - description 12–13
 - exporting and importing windows 47
 - introduction 12
 - modifying external links of an ArchestrA symbol 41
- windows
 - exporting and importing 47
 - properties 37
 - types 38

WindowViewer

- controlling data access 94
 - controlling logging 94
 - Dynamic Resolution option 33
 - enforcing security 95
 - issuing commands during run time 94
 - loading tags into run-time memory 52
 - running applications on multiple monitors 98
 - switching the language of an application 96
 - switching the screen resolution of an application 97
 - using security 95
- WinPlatform Galaxy object 21
- wizards
 - description 37
 - trend 44
 - Wonderware Historian
 - description 69
 - saving historical data 69
 - use with InTouch 69
 - using to save InTouch log data 68