AVEVA™ InTouch HMI
formerly Wonderware

Application Management and Extension Guide

Publication date: Friday, September 24, 2021

Contact Information

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

https://sw.aveva.com/

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

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

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Contents

Chapter 1

About InTouch HMI

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 manufacturing processes.

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.

Types of InTouch Applications

InTouch HMI, formerly Wonderware applications are categorized by how they are managed, the types of symbols they support, and where they were published from:

- A standalone application is the default application created in Application Manager that allows the flexibility of InTouch symbols and Industrial Graphics.
- A managed application is created and managed with the ArchestrA IDE. For more information, see Managed Applications.
- You can create a published application by exporting a managed application from the derived InTouchViewApp template. For more information, see *Published Applications*.
- InTouchView application, which can be created either from Application Manager or the ArchestrA IDE. For more information, see *InTouchView Applications* on page 15.

InTouch HMI supports the migration and upgrading of older Modern applications.

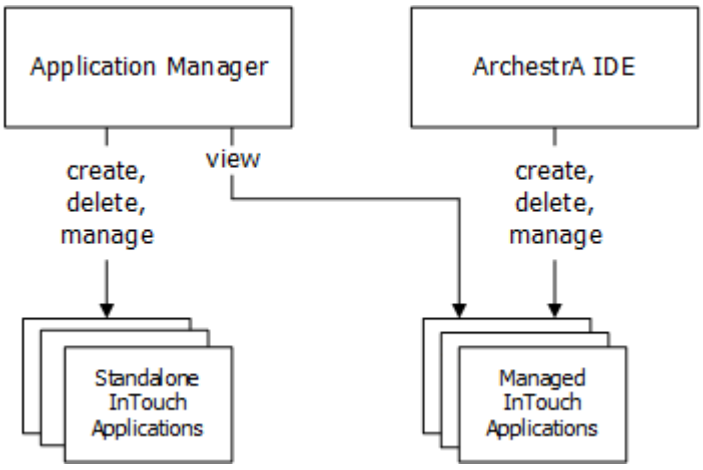
Standalone Applications

A standalone application is created and managed by Application Manager. A standalone application consists of a set of files maintained by the InTouch HMI in the directory file system. It is built entirely with WindowMaker and run with WindowViewer and can also contain Industrial Graphics. A standalone application can be deployed across multiple network nodes and is not restricted to a single node. It can be imported to the ArchestrA IDE and converted to a managed application.

Managed InTouch Applications

You can manage InTouch applications using the ArchestrA Integrated Development Environment (IDE) if it is installed on the same computer as the InTouch HMI. These applications are called managed InTouch applications. Unlike standalone InTouch applications, they are more integrated into the ArchestrA environment and support advanced graphics.

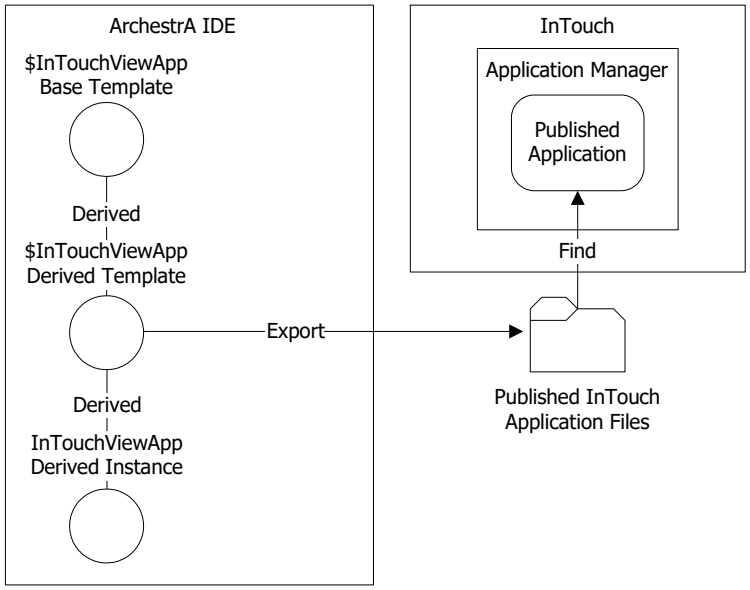
Managed InTouch applications appear in the InTouch Application Manager as *Managed* and can be edited only by starting WindowMaker from within the IDE. You can only start a managed application in WindowViewer, from the Application Manager. When WindowViewer starts, it copies a managed application's files to a folder during run time.



Each managed application is associated with an ArchestrA InTouchViewApp object, which is derived from a base template. 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 ArchestrA 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. The ArchestrA IDE includes the Industrial Graphic Editor, which you can use to create symbols that represent production processes in your InTouch HMI application. You can fast switch from WindowMaker and WindowViewer to test a managed application only if the WindowMaker was opened from the IDE.

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 Industrial graphics 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.



You can use WindowMaker to migrate a published application from a version of InTouch prior to version 11.1 (2014 R2 Patch 01) to version 11.1 Patch 01. You can then modify the published application by updating the application source files and re-publishing. After you publish a managed application, you can still use embedded Industrial graphics to write data to a Galaxy or visualize data. A published application cannot be imported again into a Galaxy. After migrating a published Managed application, you need to republish the application. The ArchestrA Embedded Alarm Control will be upgraded to the new version upon republishing.

InTouchView Applications

InTouchView applications show visual interfaces for use in an Application Server environment. InTouchView applications run in WindowViewer, with Application Server providing most of the HMI functionality. An application can also be configured to serve as InTouch Tag Server application, providing other nodes with secure tag information. InTouchView applications are useful in scenarios where a client node only needs to access data and does not need the full functionality of a development node.

InTouchView applications offer only some of the standard functions available from full-featured InTouch applications. InTouchView applications:

- Cannot connect to I/O sources other than the ArchestrA Application Server Galaxy or InTouch Tag Server.
- Cannot generate alarms. However, you can display and acknowledge alarms from remote alarm providers, such as ArchestrA objects and InTouch alarms.
- Do not log application data or events. An InTouchView application generates only SYS and USER-related events.
- When the application consumes data from a Galaxy, it can be secured only with ArchestrA security.

- Cannot reference tags within embedded Industrial Graphics.

You develop InTouch applications with WindowMaker, and run them in WindowViewer. You can then change an InTouch application to an InTouchView application that will allow you to manage your InTouch applications through the Application Server. Likewise, you can change an InTouchView application to an InTouch application.

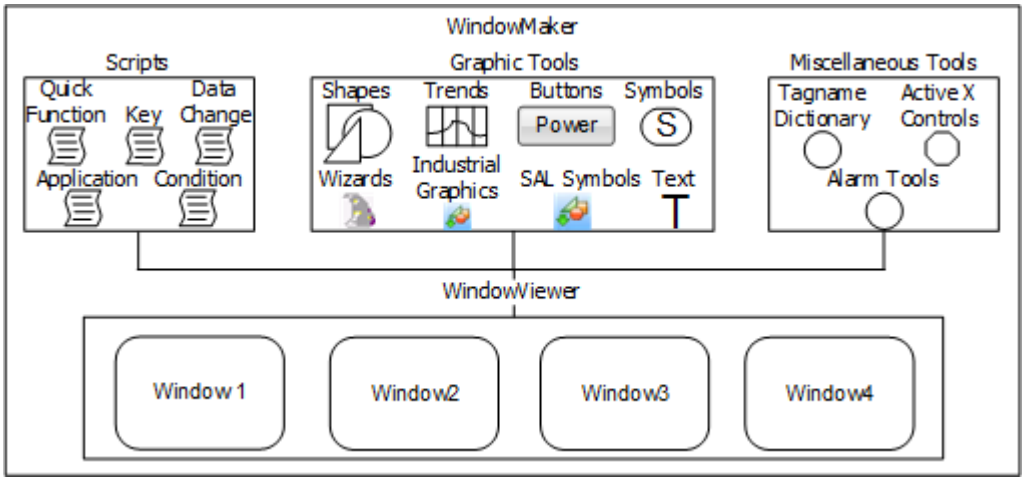
The following lists show which WindowMaker commands and Tagname Dictionary options are unavailable when creating InTouchView applications.

- Unavailable Special menu commands:
  - Access Names
  - Alarm Groups
  - Configure...Alarms
  - Configure...Historical Logging
  - Configure...Distributed Name Manager
- Unavailable Tagname Dictionary options:
  - Alarms
  - Details & Alarms
  - Log Data
  - Log Events
  - Priority

Building Applications

WindowMaker provides graphic tools, a scripting language, and tag management utilities to 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 InTouch HMI applications.



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 Industrial Graphics 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.

Running Applications

You use WindowViewer to run all types of InTouch applications. After you deploy a managed application from the ArchestrA IDE, you open it in WindowViewer from Application Manager. 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 and preventing 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.

Chapter 2

Licensing in InTouch HMI

About InTouch HMI Licensing

InTouch HMI uses the AVEVA Enterprise License Server to make licenses available to InTouch. The AVEVA Enterprise License Manager manages one or more License Servers.

To make licenses available to InTouch HMI, complete the following steps:

1. Import the entitlement XML file received upon purchase of the license.
2. Using the License Manager interface, select the licenses on the entitlement that you want to activate on the License Server.
3. Once the licenses are activated, they becomes available to WindowMaker or WindowViewer upon start up.

The activated licenses appear in the License Manager under the License Grid.

InTouch releases and returns the consumed license to the License Server when:

- The machine running InTouch is shutdown or
- The InTouch application is shutdown

Note: In the event of InTouch HMI shutting down abnormally, licenses will not be returned. InTouch HMI must be restarted and manually shut down to release licenses.

The License Manager and License Server are installed with InTouch HMI. InTouch HMI will point to your local License Server by default. You can change this configuration in the post-install Configurator. Refer to the *AVEVA Enterprise Licensing Guide* for the detailed procedure.

Licenses Available for InTouch HMI

InTouch HMI provides different types of licenses to manage various scenarios. Licenses are determined based on various parameters such as:

1. **Console Type**: specifies the console type; Remote Desktop Services/Terminal Services or non-RDS nodes. RDS/TSE is a console running on a machine that is configured with terminal server, while Non-RDS is a console running on a machine that is not configured with terminal server. For more information, see Deploying and Working with Terminal Services and Remote Desktop Services.
2. **Access Type**: specifies the access type the node is configured as - Read-Only or ReadWrite. For more information see, *Configuring User Access to Applications Running in Remote Sessions* on page 234.

3. **Data Source**: specifies the data source the application will use - Galaxy or InTouch Tag Server. For more information, see *InTouchView Applications* on page 15.

InTouch HMI Unlimited RDS License

The InTouch HMI Unlimited RDS License is consumed by WindowViewer for unlimited client sessions, only in an RDS enabled machine. When WindowViewer acquires the license, the application is licensed. The Read/Write access depends on the Remote Access configuration. The same unlimited license will provide both Read-Only and ReadWrite access. If the license is not acquired, then the licensing will depend on the existing RDS handling and Remote Access configuration.

InTouch Licensing in RDS and non-RDS Environments

If an InTouch application is running on a server node enabled with Remote Desktop Services (RDS), the console will behave the same as an RDS client session. Each session will consume a license. Each session will also consume a separate InTouch development license.

In this case, the InTouch application's ReadWrite capability is defined by its remote access configuration and confirmed by the consumed license. For example, if an application with ReadOnly remote access configuration is launched in WindowViewer in an RDS client session, it will look for a ReadOnly InTouch license. If an RDS ReadOnly license is not available in the License Server, startup license validation will fail.

On a node without RDS enabled, you can also login with a RDS client session that is allowed by the operating system. If an InTouch application is running in this non-RDS environment, the client session will behave the same as a console. In this case, the application's remote access configuration does not determine the ReadWrite access. ReadWrite access is determined only by the license in non-RDS environments.

About InTouchView Application Licensing

An InTouchView application shows visual interfaces designed specifically for use in an Application Server environment. See *InTouchView Applications* on page 15 for details on this application type.

The type of license an InTouchView application consumes depends on whether it is running in an RDS environment.

If an InTouchView application is running in a RDS client session, it will look for a ReadOnly or ReadWrite client connection license, depending on the remote access type configuration of the application.

Only one connection license will be consumed per RDS session.

Note: An InTouchView application consumes the same license as the Graphic Run Time Module's ViewApp application, if configured with a Galaxy data source.

InTouchView Application Licensing

You can configure an InTouch HMI application as an InTouchView Application, where it can be used as a client to an InTouch Tag Server or Galaxy.

If you configure the InTouchView application to connect to data from an InTouch Tag Server then the licenses available are:

	InTouch HMI Client ReadWrite License	**InTouch HMI Client Read-Only License**	**InTouch HMI Unlimited Client License***
Remote Access	ReadWrite	Read-Only	ReadWrite & Read-Only
RDS/TSE	Yes	Yes	Yes
Non-RDS	Yes	Yes	No
Supports MarkAppReadOnlyNonRDS	Yes	Yes	N/A

* This license serves unlimited number of RDS clients.

If you configure the InTouchView application to connect to data from a Galaxy then the licenses available are:

	Supervisory Client ReadWrite License	**Supervisory Client Read-Only License**	**Supervisory Client Server License**
Remote Access	ReadWrite	Read-Only	ReadWrite & Read-Only
RDS/TSE	Yes	Yes	Yes
Non-RDS	Yes	Yes	No
Shared with OMI	Yes	Yes	Yes
Supports MarkAppReadOnlyNonRDS	Yes	Yes	N/A

Best Practices for Administering InTouch Licenses on the Server

There are several best practices to follow when administering InTouch licenses that will ensure InTouch license consumption is deterministic. Deterministic license consumption allows you to consume appropriate licenses on demand for a particular system. This type of license consumption will make it easier to administer InTouch licenses using the server-based AVEVA Enterprise Licensing system.

The two best practices for deterministic license consumption are license reservations and floating licenses. Refer to the sections below for details.

Reserving Licenses

You can reserve licenses to specific devices in the License Manager. Reserving a license to a particular device ensures that the license cannot be acquired by another InTouch application and interrupt or prevent your application from running.

Reserving Licenses

You can reserve licenses to specific devices in the License Manager. Reserving a license to a particular device ensures that the license cannot be acquired by another InTouch application and interrupt or prevent your application from running.

User-based License Reservation

In the AVEVA Enterprise License Manager license reservation page, it is possible to mark a license to be reserved to a specific user. While the reservation page allows this particular configuration, it's important to know that neither InTouch OMI nor InTouch HMI ViewApps support user-based license reservations. The end-result will be the inability for the software to acquire the license reserved. Therefore, only use device-based reservations for Supervisory Client licenses.

Device-based License Reservation

When reserving a Supervisory Client license for a specific device, the Device Name needs to be the name of the computer running the InTouch HMI/OMI ViewApp. In the case where the ViewApp is running inside of an RDS or Terminal Server, the Device Name needs to follow this naming pattern:

<RDSHostName>-<RDPClientName>-<index>

where RDSHostName is the name of the RDS or Terminal Server, and RDPClientName is the name of the PC running the RDP client software, and "index" is 1, unless there will be multiple RDP sessions from a single client machine, in which case the index should be incremented (starting at 1) for each reservation for that specific RDP client, up to the total number of RDP sessions from that specific RDP client.

Example 1: A computer with a hostname of "ControlRoomA" runs InTouch OMI

Device Name: "ControlRoomA"

Example 2: A computer with a hostname of "ControlRoomB" running a single Remote Desktop Client (RDP), connecting to the Remote Desktop Server (aka: Terminal Server) with a hostname of "PrimaryRDS"

Device Name: "PrimaryRDS-ControlRoomB-1"

Example 3: Two computers with hostnames "SupervisorPC1" and "LineMgrA", respectively, each running a single Remote Desktop Client (RDP) connecting to the Remote Desktop Server (aka: Terminal Server) with a hostname of "PrimaryRDS"

Device Names:

License Reservation 1: "PrimaryRDS-SupervisorPC1-1"

License Reservation 2: "PrimaryRDS-LineMgrA-1"

Situation: A computer with a hostname of "ExecutiveDesktop" running four (4) Remote Desktop Clients (RDPs), connecting to the Remote Desktop Server (aka: Terminal Server) with a hostname of "PrimaryRDS"

Device Names:

License Reservation 1: "PrimaryRDS-ExecutiveDesktop-1"

License Reservation 2: "PrimaryRDS-ExecutiveDesktop-2"

License Reservation 3: "PrimaryRDS-ExecutiveDesktop-3"

License Reservation 4: "PrimaryRDS-ExecutiveDesktop-4"

For RDS load balancing support, all RDS licenses can be activated on a single License Server that multiple RDS client sessions can point to. The licenses on the server must be of the same capability so that the licenses can be shared amongst each RDS client session. Licenses are considered to be of the same capability if their internal parameters have the same value. No reservations are needed in this scenario. If different license types for different RDS client sessions are required, then a License Server must be installed on each RDS server.

Refer to the *AVEVA Enterprise Licensing Guide* for detailed license reservation procedures.

Floating Licenses

Floating licenses are not reserved to any machine. It is recommended to have floating licenses of the same product name and capabilities on a single License Server. For example, you could have a License Server with several activated InTouch 2017 Runtime 60K tags licenses with the same capabilities. This is a recommended practice to ensure deterministic license consumption.

However, it is not recommended to have licenses of the same product name but different capabilities activated on the same License Server. For example, you could have a mix of InTouch 2017 Runtime 60K tags activated licenses with InTouch 2017 Runtime 500 tags activated licenses on the same license server. In this scenario, there is no way to ensure which instance of WindowViewer will consume the license with the higher tag count.

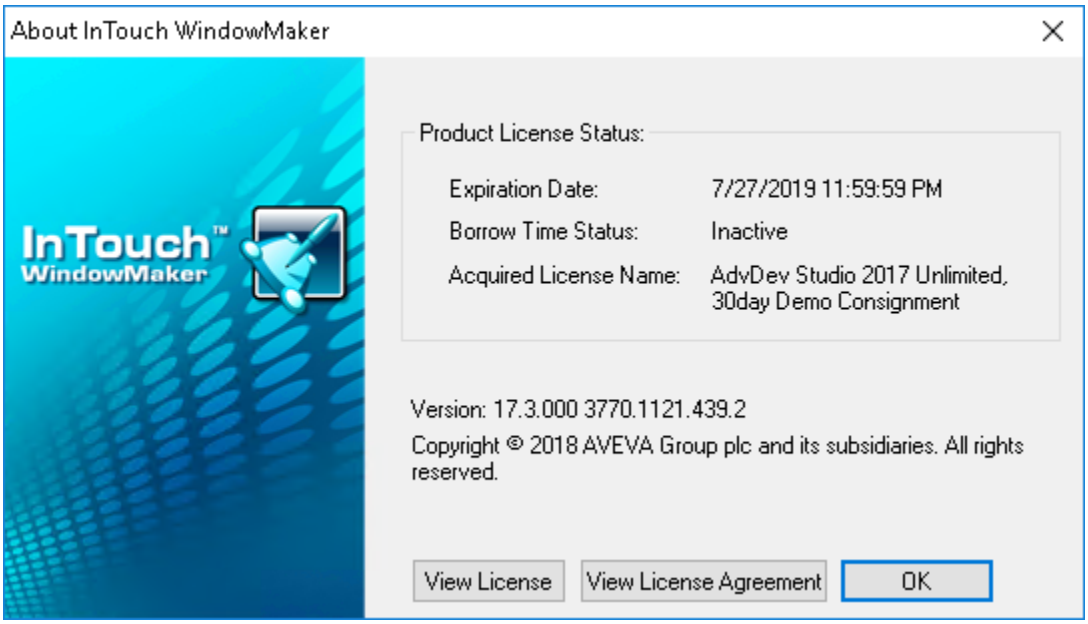
Viewing License Information

You can view the specific information for the current license consumed by WindowMaker or WindowViewer.

To View WindowMaker or WindowViewer License Information:

1. Do one of the following:

a. From WindowMaker, click **Help**, **About WindowMaker**. The **About WindowMaker** dialog box appears.

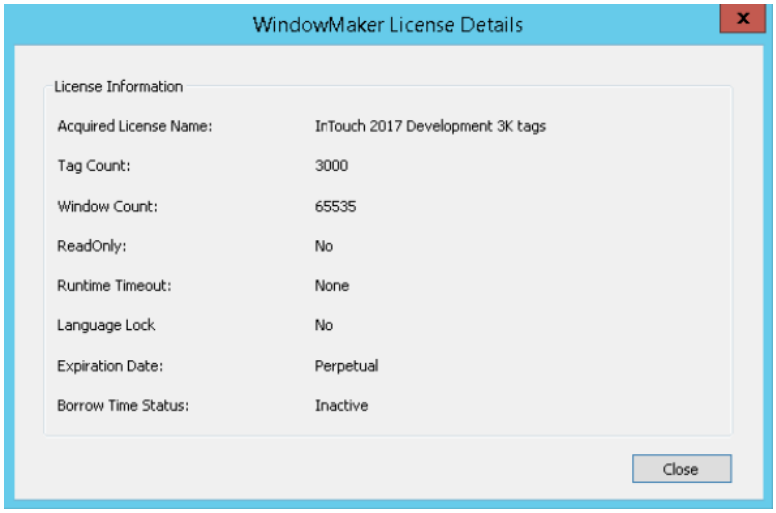


The company name and license serial number do not appear in this dialog. This information appears in the AVEVA Enterprise License Manager interface.

b. From WindowViewer, click **File**, **About WindowViewer**. The **About WindowViewer** dialog box appears.

Note: The **View License** option is disabled for InTouch Application Manager. Because Application Manager does not consume a license, you can only view the EULA. The **About Application Manager** dialog box does not display any license-related information.

2. For both WindowMaker and WindowViewer, click **View License Agreement** to view the End User License Agreement.
3. Select **View License** to view the details for the license. The License Information dialog box displays.



The parameters displayed in the license information for WindowMaker and WindowViewer are as follows:

- **Acquired License Name:** the full name of the license consumed by the product from the License Server

- **Tag Count:** the number of tags allowed by the consumed license.
- **Window Count:** the number of windows allowed by the consumed license.
- **ReadOnly:** displays the I/O Read/Write permissions allowed by the license. No indicates that the application can write to I/O tags.
- **Runtime Timeout:** the application runtime allotted by the license. The InTouch session will end when the timeout period elapses.
- **Language Lock:** applies to licenses for InTouch on Chinese operating systems only. A Chinese license must be consumed for InTouch to run on a Chinese operating system.

  The Language Lock restriction does not apply for a connection license.
- **Expiration Date:** the date the consumed license expires.
- **Borrow Time Status:** intended to notify the user when the license is 50% past whatever the allotted borrow time is. The status will change to **Active** when the 50% has been reached. If the license is not renewed when the borrow time has fully elapsed, InTouch will become unlicensed.

Managing Consumption of a Different License After Startup

As part of standard upgrade and maintenance activities, different licenses can be deactivated or activated on the License Manager. If WindowMaker or WindowViewer consumes a valid license after startup time and cannot renew the license, it can consume a different license, from the one initially assigned or consumed. If this occurs, InTouch must then validate if the capabilities of the newer license is appropriate. The comparison in license capabilities between the last good license and the currently consumed licenses will determine if the WindowMaker or WindowViewer can continue running without entering the Grace Period. See *Working with the Grace Period* on page 26 for details on how to exit the Grace Period.

The two possible scenarios for a different license being consumed after startup are described below.

Scenario 1: A less capable license is consumed after startup

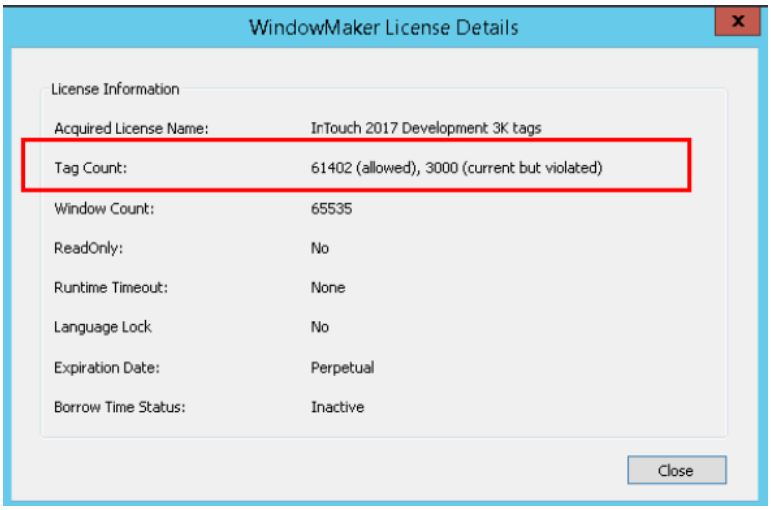
In this scenario, the license consumed after startup time is of lesser capabilities than the last good license. This is considered a license downgrade, and will result in WindowMaker or WindowViewer entering the Grace Period.

A license downgrade is considered when the new license's parameters fall into the below check:

The parameter changes that will trigger the Grace Period are described below.

- Tag Count: if the tag count is reduced, the Grace Period is triggered.
- Window Count: if the window count is reduced, the Grace Period is triggered.
- Runtime Timeout: if the Timeout value changes from **None** to any other value, Grace Period is triggered.
- Language Lock: if the Language Lock value changes from **No** to **Yes**, or vise versa, Grace Period is triggered
- ReadOnly: if the ReadOnly parameter is changed from **No** to **Yes** or vise versa, Grace Period is triggered.

If a license downgrade occurs, the downgraded, or "violated" parameters display in the **View License** dialog box. For example, if WindowMaker originally consumed a sixty-thousand tag count license and downgraded to a three thousand tag count license, the tag count parameter would display as follows:



Important: The application will continue to run with the capabilities of the last good license. In this example, the original sixty-thousand tag count will remain when entering the Grace Period.

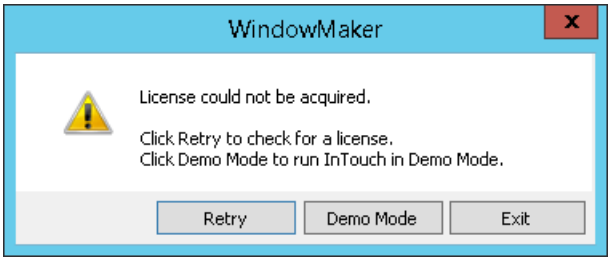
Scenario 2: A more capable license is consumed after startup

In this scenario, the license consumed after startup time is of greater or equal capability than the original. This is a license upgrade, and will not cause WindowMaker or WindowViewer to enter the Grace Period.

If a license upgrade occurs, all parameters in the **View License** dialog box are updated to display the higher capability values.

Working in Demo Mode

InTouch will run in Demo Mode when it cannot consume a valid license at startup time. If you attempt to start WindowMaker or WindowViewer and a valid license is not available, you can select to run in Demo Mode. You will be prompted with the following dialog:



In Demo Mode, InTouch will display a message stating that it will run in demo mode. WindowViewer will only run in Demo Mode for two hours before timing out. WindowMaker will run in Demo Mode indefinitely.

In Demo Mode InTouch will:

- Allow 32 Local tags (Excluding system tags)
- Allow maximum of 32 windows

When Demo Mode timeout is reached, InTouch will prompt to exit.

Note: While in Demo Mode, even if you activate a valid license, you need to exit WindowMaker or WindowViewer and restart InTouch to consume the valid license.

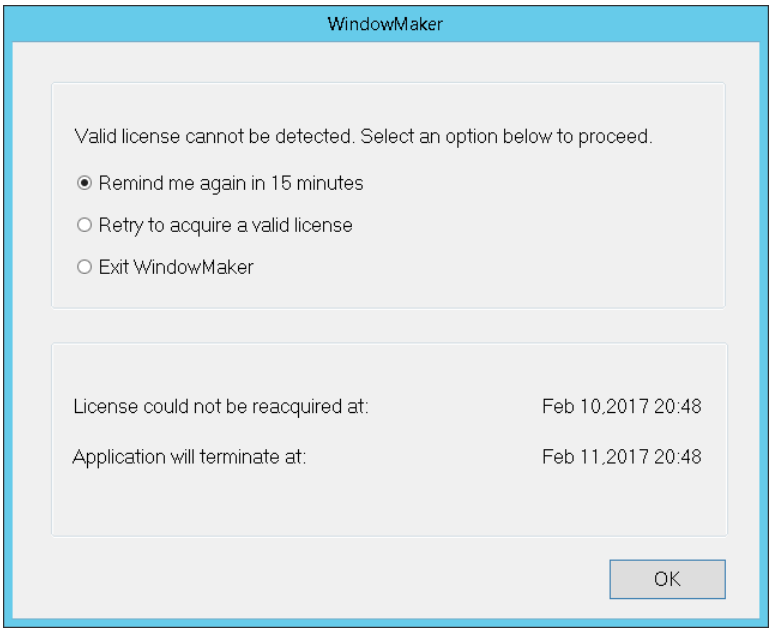
Configuring the ViewLicenseRetryCount key in the InTouch.ini file, will instruct WindowViewer to continue to attempt acquiring a license in the background for the number of times specified in the parameter. If a license is acquired, the dialog box will close and WindowViewer will be launched.

Working with the Grace Period

The Grace Period is a twenty-four hour period in which InTouch can continue to run with the last good license's capabilities after certain conditions have occurred. Both WindowMaker and WindowViewer can enter the Grace Period. At the end of the Grace Period InTouch will terminate if a valid license is not reacquired within the allotted time.

The Grace Period will be triggered by the following scenarios. If valid license is not reacquired within twenty-four hours, WindowMaker or WindowViewer will terminate.

When WindowMaker or WindowViewer enters the Grace Period, you will be prompted with the following dialog:



You have the options to be reminded again, to retry to acquire the license, or to exit the application.

Scenario 1: Consumed License is Lost

WindowMaker or WindowViewer will go in to Grace Period if it consumes a valid license from the License Server and the license is deactivated while the product is still running.

To exit the Grace Period and resume normal operation, activate a valid license on the License Server. When the license is consumed, WindowMaker or WindowViewer will exit the Grace Period and normal operation will resume.

Scenario 2: License Expired

WindowMaker or WindowViewer will go in to Grace Period if it consumes a valid license from the License Server and the license expires. To exit the Grace Period and resume normal operation, activate a valid license on the License Server.

If WindowMaker or WindowViewer fails to acquire the license, the first dialog will appear again.

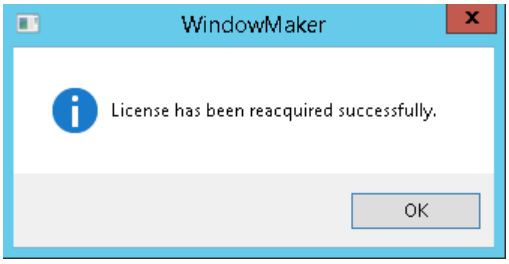
Scenario 3: License is Downgraded

AVEVA Enterprise License Manager is a server-based licensing system, which means that licenses need to be renewed periodically. If a WindowMaker or WindowViewer license is downgraded during this renewal to one of lesser capability, it will enter the Grace Period. See *Managing Consumption of a Different License After Startup* on page 24 for a detailed description of downgrade scenarios.

To exit the Grace Period and resume normal operation, retry to acquire the last good license or a better license.

Important: The functionality enabled by your last good license will persist in Grace Period mode.

In all of the above scenarios, if you select the option to retry to acquire the license and an appropriate license is successfully acquired, you will see the following dialog:



If InTouch fails to acquire the license, the first dialog will appear again.

Note: Configuring the ViewLicenseRetryCount key in the InTouch.ini file, will instruct WindowViewer to continue to attempt acquiring a license in the background for the number of times specified in the parameter. If a license is acquired, the dialog box will close and WindowViewer will be launched.

Chapter 3

Managing InTouch HMI Applications

About Managing InTouch HMI Applications

When managing InTouch HMI applications, you:

- Create or delete InTouch applications. See *Creating an InTouch Application* on page 30 and *Deleting an InTouch Application from the Application Manager* on page 57.
- Open applications in either WindowMaker or WindowViewer. See *Opening an Application in WindowMaker and WindowViewer* on page 53.
- Search for applications. See *Finding InTouch Applications* on page 57.
- Move an application to a different computer. See *Publishing Applications to Remote Nodes* on page 66.
- Distribute applications among multiple computers. See *About Distributing Applications* on page 75.
- Manage InTouch services. See *About Managing InTouch Services* on page 105.
- Import or export tag definitions, windows, and scripts. See *Exporting and Importing InTouch Components* on page 112.
- Configure security. See *About Securing InTouch* on page 174.

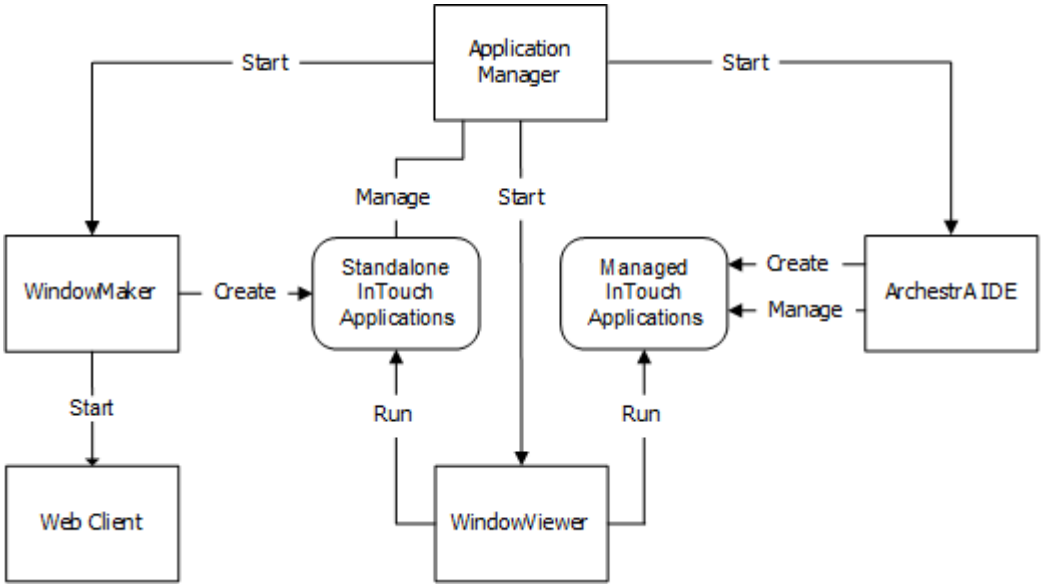
You can extend your application by:

- Translating text strings and alarm comments into different languages. See *About Switching a Language at Run Time* on page 213.
- Integrating an application with ArchestrA. See *Managed InTouch Applications* on page 14 and *About Viewing Applications at Runtime* on page 228.
- Displaying applications on multiple monitors. See *About Setting Up a Multi-Monitor System* on page 254.
- Using applications on a Tablet PC. See *About Using InTouch on a Tablet PC* on page 263.

For the Windows Vista, Windows 7, and Windows Server 2008 operating systems, a standard user can use the InTouch Application Manager to find an application and open WindowViewer. Application properties will be read-only for the standard user. All read/write or configuration operations from Application Manager require administrative privileges.

About the InTouch Application Manager

You use the InTouch Application Manager to manage most global tasks such as creating, deleting, and modifying your InTouch applications. Application Manager shows a list of your current InTouch applications. You select an application from the list to open in WindowMaker or WindowViewer.



Starting the Application Manager

You can start the Application Manager from the **Start** menu or from a shortcut placed on your computer's desktop.

To start the Application Manager for the first time

- On the taskbar, click **Start**, point to **Programs**, point to **AVEVA InTouch HMI**, and then click **InTouch HMI Application Manager**. The **AVEVA Application Manager** appears.

The window shows InTouch applications on your computer that you created or found using the Application Manager. In the Details view, the **Version** column shows the InTouch version that was last used to save the application.

Using the Application Manager

The Application Manager is the first step in creating InTouch HMI applications and provides multiple options for application management.

The interface is divided into tabs and each tab presents with different functionality depending on the context. The menu options and other elements on the screen will be different for each tab.

- InTouch - All InTouch HMI related options.
- Web Client - Configure settings for the Web Client

Creating an InTouch Application

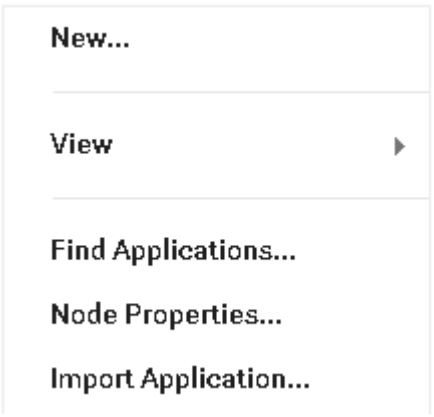
You can create a new InTouch application using the Application Manager. The application path cannot exceed 114 characters, including the network drive letter, colon, and all backslashes. If the limit is exceeded, you cannot open the application in WindowMaker. Application name must be 32 characters or less.

The INTOUCH.ini file is created when you create an application. The INTOUCH.ini file contains the default configuration settings for your application. As you configure your application, the new settings are written to the INTOUCH.ini file.

After you create an application, you can copy an existing INTOUCH.ini file to the application folder. This way, you do not need to configure customized InTouch parameters each time you create a new application.

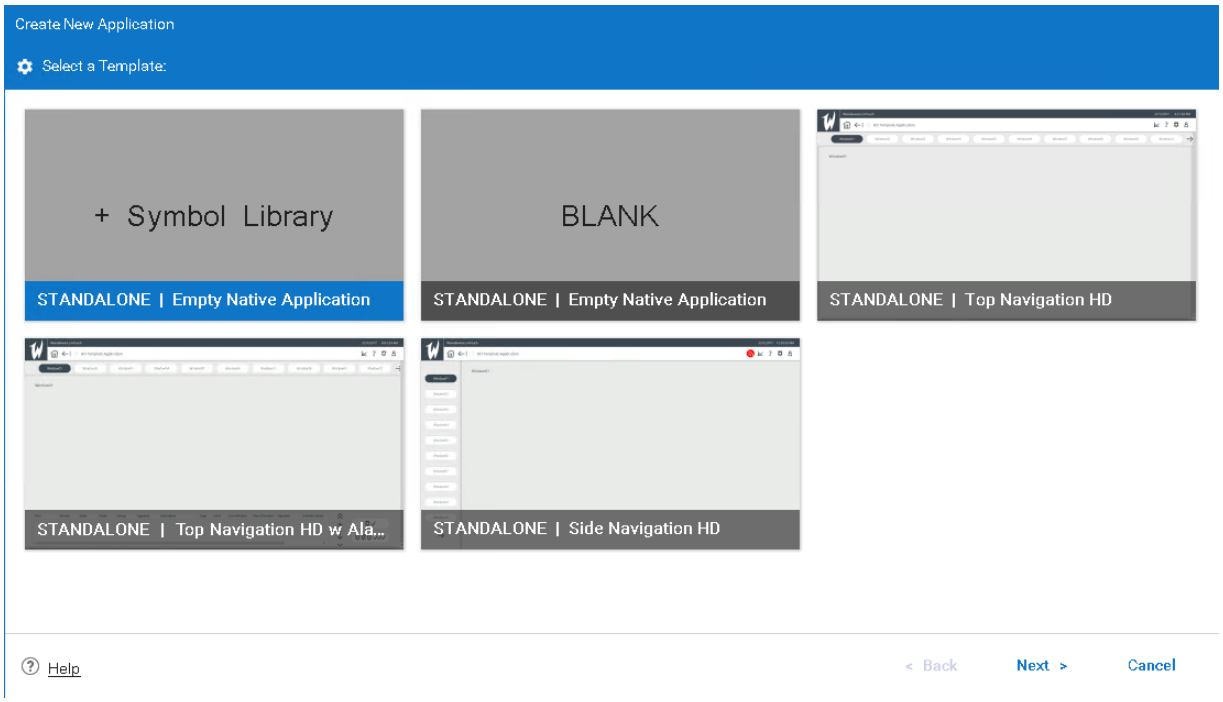
To create a new application

1. You can create a new application, using different options on the Application Manager.
   a. Click File menu, click New.
   b. Click  on the toolbar.
   c. Click  at the right bottom.
   d. Right-click in the empty space, and select New….



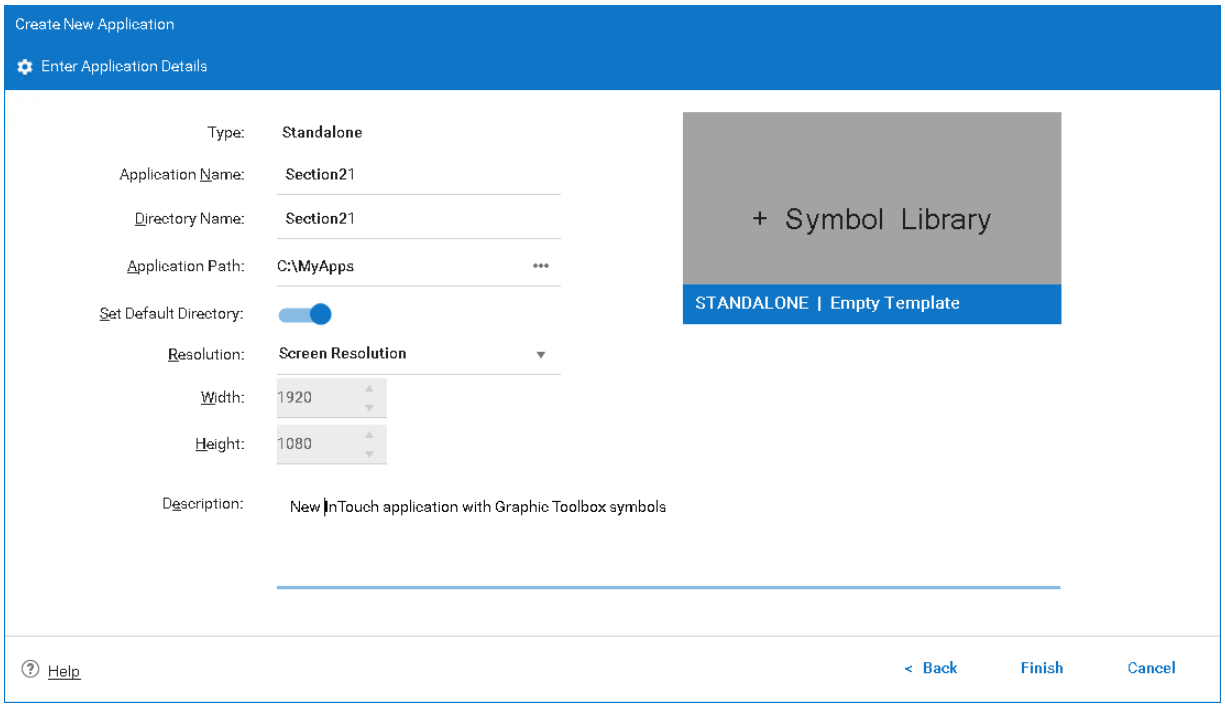
   The **Select a Template:** page appears.
2. Select the Template. Click **Next**.

The Blank template will not include the default Graphic toolbox symbols and SAL symbols. In all templates you can use both InTouch Symbols and Industrial graphics in your application. If you have created any application templates and they are available in the correct folder, those templates will also be displayed.



3. In the Enter Application Details, enter the following details:
   - **Type**: Will display the type of application selected in the previous screen: Standalone.
   - **Application Name**: Enter the name of the application
   - **Directory Name**: Type the application folder name.
   - **Application Path**: Click the **…** button to browse a folder other than the default location.
   - **Set Default Directory**: Use the toggle to set the application path as the default directory.
   - **Resolution**: Select the application target resolution from the dropdown list, if it is different from the default Screen Resolution option. To edit the width and height, select Custom from the Resolution dropdown list.
   - **Width**: Specify the width of the application.
   - **Height**: Specify the height of the application.
   - **Description**: Type an optional description up to a maximum of 255 characters.

4. Click **Finish**.



A horizontal bar displays the progress of creating the application. After the application is created, it appears in the Application Manager's list of applications.

Creating a New InTouchView Application

You identify an InTouchView application by setting an option from Application Manager when you create the application. You can configure an InTouch application to an InTouchView application and vice versa. A full InTouch license is required to run applications converted from InTouchView to InTouch. When an application is changed to an InTouchView application, the license acquired will differ based on the data source selected. For more information, see *InTouchView Applications* on page 15.

For detailed instructions on how to create a new InTouchViewApp, see the *InTouch HMI and ArchestrA Integration Guide*.

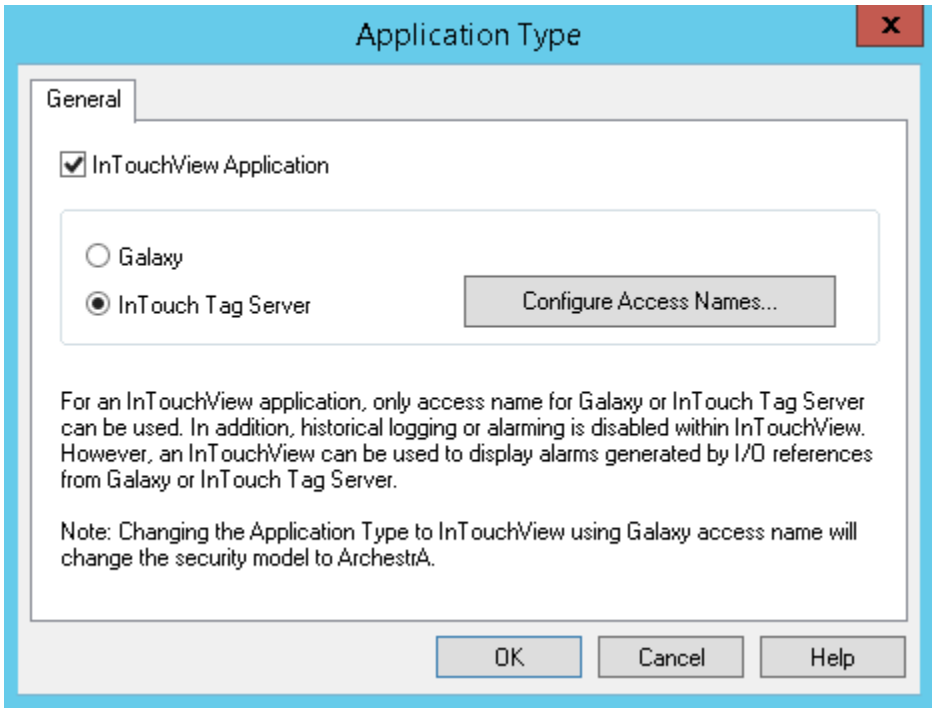
Changing an InTouch Application to an InTouchView Application

You can change an InTouch application to InTouchView if the application only needs to connect to data from an Application Server or InTouch Tag Server. Some WindowMaker functions are unavailable in an InTouchView application.

To change an InTouch application to an InTouchView application

1. Open the InTouch application in WindowMaker.

2. On the **Special** menu, click **Application Type**.The **Application Type** dialog box appears.



3. Select the **InTouchView Application** check box.
4. To connect to data from a Galaxy, select **Galaxy**. To connect to data from InTouch Tag Servers, select **InTouch Tag Server**.
5. Select **Configure Access Names**.
   a. If you select **Galaxy**, you must remove all Access Names other than Galaxy before converting an InTouch application to InTouchView. If they are not removed, a message is shown during the conversion attempt.
   b. If you select **InTouch Tag Server**, you can configure new access names, Click **Add**, and provide the Access Name and Node Name.

      The Application Name and Topic Name are grayed out. All Access Name other than Galaxy or referencing to InTouch Tag Server must be removed prior to changing this application to InTouchView. Only I/O references from access name(s) where Application is configured as *view* will be binded, I/O reference from access name *Galaxy* will not be binded.
   c. Click **OK**.

      For more information on configuring Access Names, see *Setting Up Access Names* in the Data Management guide.
6. Click **Close**.
7. Click **OK.** When a message appears, click **OK**.

Configuring and Using the InTouch OPC UA Server

InTouch HMI supports the OPC UA (Unified Architecture) protocol for machine-to-machine communications.

This OPC UA Server functionality allows third party clients to interact with InTouch HMI and leverage industry standards, such as OPC UA. When OPC UA Server functionality is enabled on InTouch HMI, a client can utilize the built-in OI Gateway or a third party client to connect to InTouch HMI, and use OI Gateway or the client to securely browse the OPC UA namespace and interact with InTouch HMI.

OPC UA Configuration Checklist

Required tasks for end-to-end configuration of the OPC UA server and OPC UA client

The configuration tasks are shown in the order in which they must be completed.

1. **Configure the System Management Server:** The System Management Server is used for establishing a trust relationship between machines, and must be configured to ensure secure communications between nodes. The System Management Server is normally configured during initial System Platform installation. See the *System Platform Installation Guide*, "Configuring the System Management Server," for details.

   Note: When enabled, all connecting clients must be using the same System Management Server as the InTouch from which this OPC UA Server instance has been deployed. Also, InTouch HMI must be run in the context of a user with Administrative privileges, which gives InTouch HMI access to the encryption certificates that enable secure communications.

2. **Configure the OPC UA server:** Set the configuration options and deploy the OPC UA server to a run-time node.
3. **IT compliance/firewall validation:** Firewall configuration and verification must be completed at this point of the configuration. The node to which the OPC UA Server has been deployed must have Inbound Rules for the firewall configured and verified.

   IMPORTANT! A firewall test must be successfully performed before proceeding with the remaining configuration tasks.

4. **Configure the OPC UA Client:** Client configuration may include the following:
   - Define the OPC UA server address (in the format `opc.tcp://<ServerName>:<PortNumber>`).
   - Select the correct OPC UA server security policy (`Basic256Sha256`).
   - Add the users to the InTouchHMIOPCUAWriteUsers user group.
   - Enter the configured OPC UA User Credentials (username and password)
   - Anonymous Connections are supported only for reading InTouch tags. To avoids any security risks, it is recommended to access the data using authenticated credentials.
5. **Security Certificate:** Download and configure the OPC UA security certificate on the run-time node.
6. **Validate connectivity:** Open the OPC UA client and verify that you can connect to the OPC UA Server, and can view items in the namespace.

Configuring the InTouch OPC UA Server

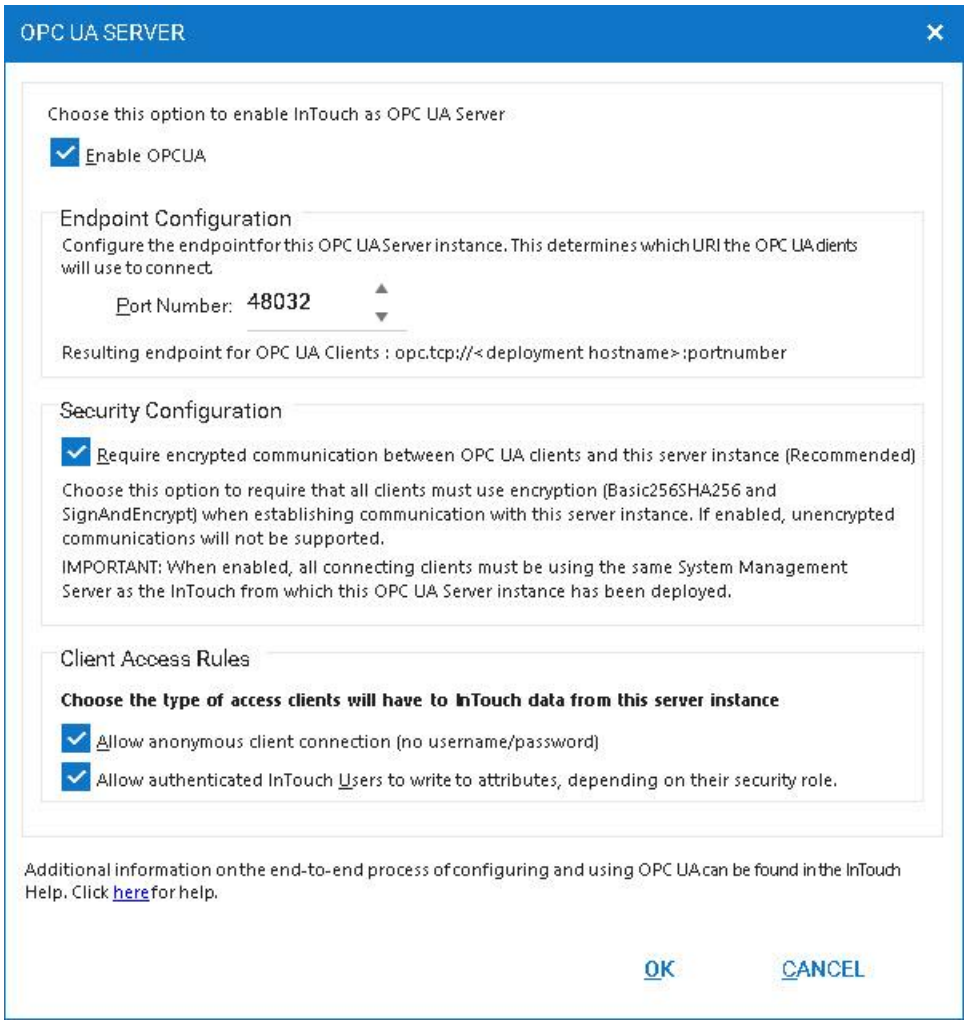
The InTouch OPC UA Server provides access from an OPC UA client to InTouch HMI data, without the need for a gateway, or other protocol translation mechanism.

1. Launch **AVEVA InTouch HMI Application Manager**.

2. Click  .

   The OPC UA Server dialog box appears.

3. The OPC UA server will be disabled by default. To configure the server, click **Enable OPCUA**.



4. Edit the **Port Number**. By default the port will be set to 48032.

Note: The port number is unique for a user and RDS session. Assign alternate port numbers for additional RDS sessions.

5. **Security Configuration**: It is strongly recommended that you enable this option, as this will encrypt the payloads across the connection. Note that the client must match this configuration.

   The Client Access Rules options allow you to determine the type of access the client will have to InTouch data.

   a. Select the **Allow anonymous client connection (no username/password)** checkbox to allow anonymous access.

   b. Select the **Allow authenticated InTouch Users to write to attributes, depending on their security role** checkbox to allow only authenticated users.

6. Click **OK**.
7. After configuration, start the WindowViewer to launch the OPC UA Server. OPC UA clients can now access InTouch HMI data.

Configuring the Firewall for the OPC UA Service

OPC UA communications in InTouch HMI require that the run-time node firewall allows a connection with an OPC UA client node. Before changing firewall settings, however, it is recommended that you perform a *Firewall Test* on page 39.

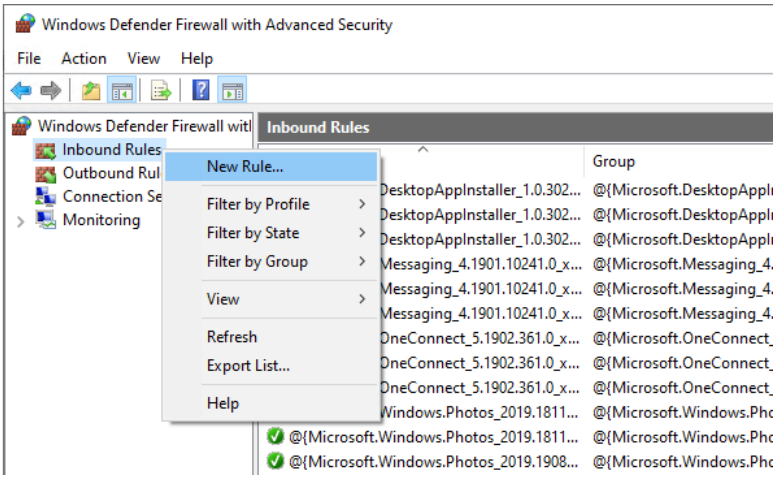
Configure the Run-Time Node Firewall

Important: The firewall rules must be added to the node to which the OPC UA Server Service is deployed.

To configure the run-time node firewall:

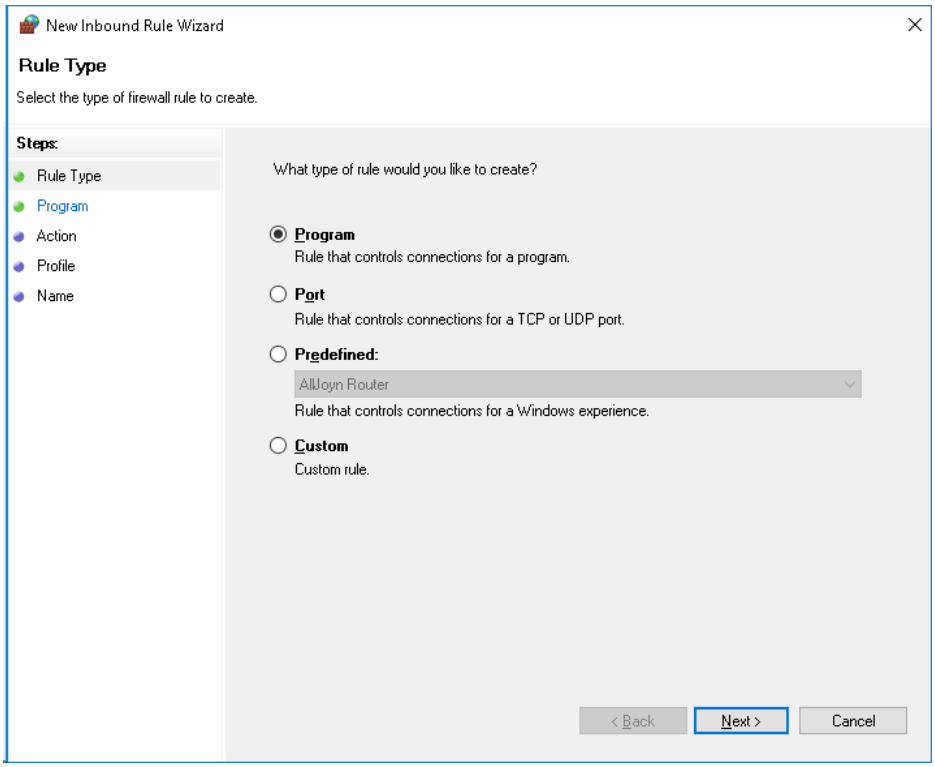
On the run-time node(s) where the OPC UA Server Service is deployed, open the Windows firewall and configure it as follows:

1. In the Windows Search bar, open **Windows Firewall**.
2. Select **Advanced Settings** and create an **Inbound Rule**.

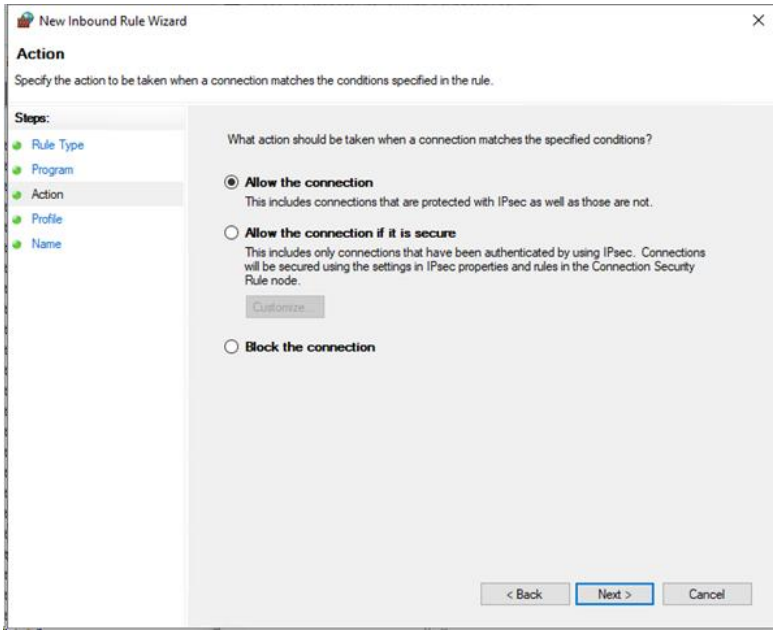


3. Click "**New Rule**." The **Rule Wizard** opens. .

4. Select **Program** for the Rule Type and click **Next**.

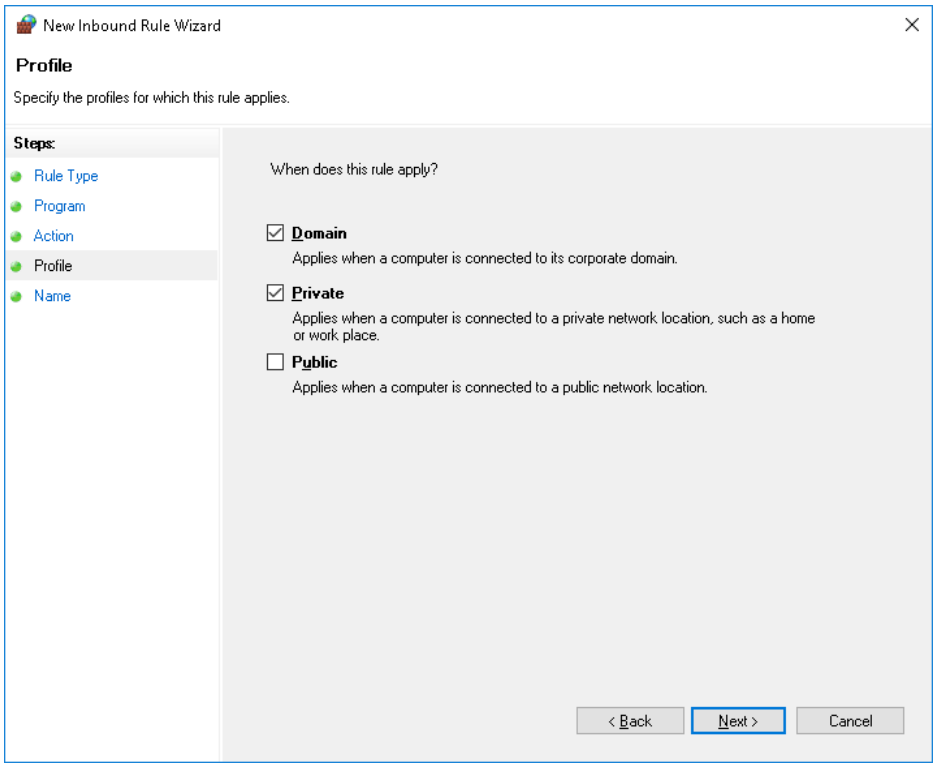


5. On the next screen, select the option "**Allow the connection**." Click **Next**.

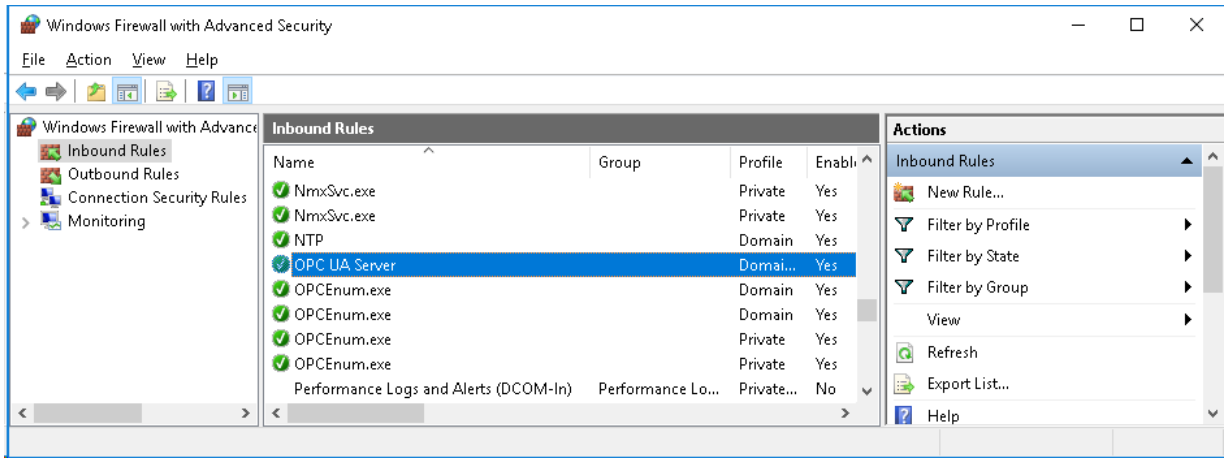


6. The wizard will ask when the rule applies.
   - For **Domain** environments: Select **Domain** and **Private**. We recommend that you deselect **Public**.

- For **Workgroup** environments: Select **Public**. The Domain and Private settings have no affect in a Workgroup environment.



7. Finally, provide a name for this rule (for example, "OPC UA Server"). If you will be configuring multiple OPC UA services, be sure to use names that differentiate each service from the others.
8. Now, check that the new rule has been added to the list of InBound Rules in the Windows Firewall and that it is enabled.



9. Verify that you can connect to the run-time node from the OPC UA client node by repeating the Firewall Test.

Firewall Test

To perform a firewall test with Telnet

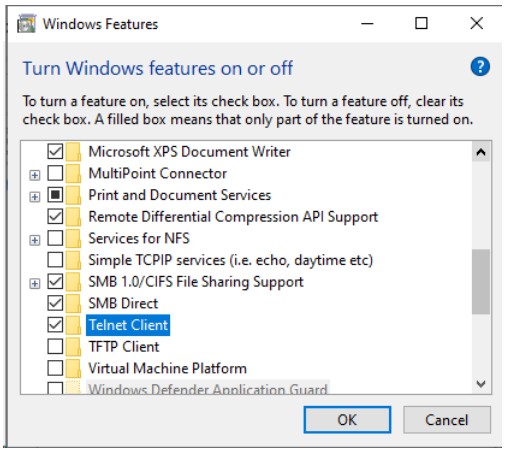
1. From a separate node from where the OPC UA Server service is running, enable **Telnet** by turning on the **Telnet Client** Windows Feature.

   Note: This test is ideally run from the OPC UA client node.

   a. Open the Windows **Control Panel**.

   b. Go to **Programs and Features**, then select **Turn Windows features on or off.**

   c. Scroll down the list of features to **Telnet Client** and enable it. Telnet is disabled by default.



2. Prior to running this test, verify that WindowViewer is running. If the OPC UA Service Host is configured, WindowViewer will start the InTouch OPC UA Service. See *Configuring the InTouch OPC UA Server* on page 34 for details.

3. Run Telnet in a command window on the OPC UA client node by entering the following:

   telnet <nodeName or ipAddress> <portNumber>

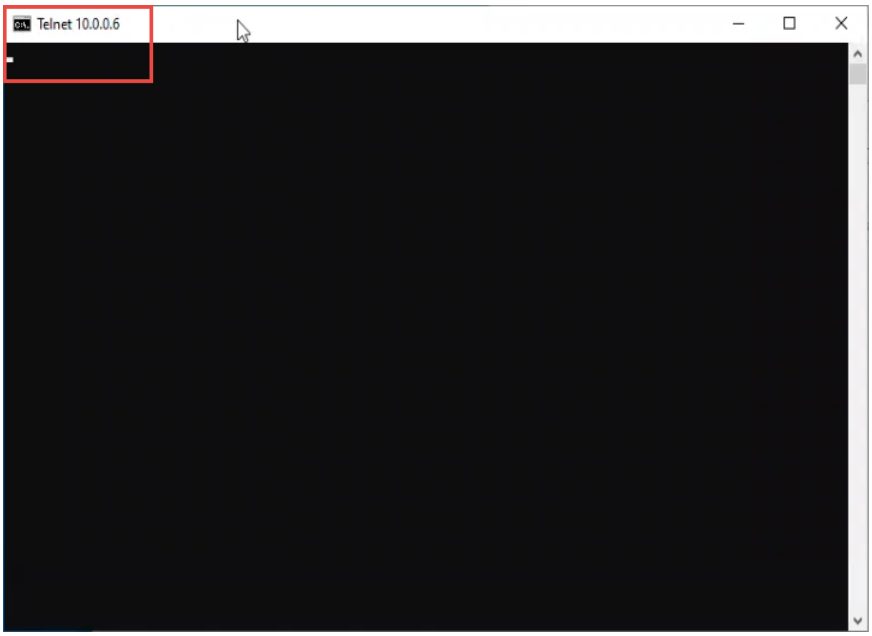
   where

   - **nodeName** is the machine name of the InTouch HMI run-time node. Use nodeName or ipAddress, not both.
   - **ipAddress** is the IP address of the InTouch HMI run-time node. Use nodeName or ipAddress, not both.
   - **portNumber** is the port number you configured in InTouch HMI for the OPC UA Service. The default port number 48032.

   Example: `telnet 10.10.10.06 48031`

   - If the command is not successful, it will time out with a message stating that the connection failed. In this case, go to *Configure the Run-Time Node Firewall* on page 36.

- If the telnet command is successful, the command prompt changes to a Telnet prompt.



- If the Firewall Test is successful, configure the OPC UA client and OPC UA server certificates. The next step in setting up your OPC UA connection depends on if you are using a third-party OPC UA client application, or the OPC UA connection available with OI Gateway.

Configuring Server and Client Certificates for Third-Party OPC UA Client Applications

IMPORTANT! These procedures apply ONLY if you are using a third party OPC UA client. If you are using OI Gateway as the OPC UA client, skip to *Using OI Gateway to Configure the Client Security Certificate* on page 46.

To configure encrypted communications between the InTouch OPC UA server and a third-party OPC UA client, both computers will need access the following certificates:

- <Computer Name> ASB OPC UA Server
- The client certificate from your OPC UA client.

To complete this setup, you will need to perform three steps:

1. Copy the certificate from the OPC UA server node to the OPC UA client node. This step includes the following:
   - Exporting the OPC UA server certificate.
   - Installing the certificate on the client node.
2. Copy the certificate from the OPC UA client node to the OPC UA server node.
3. Make sure that the firewall has been configured to allow the InTouch.OPCUA.ServiceHost.exe application.

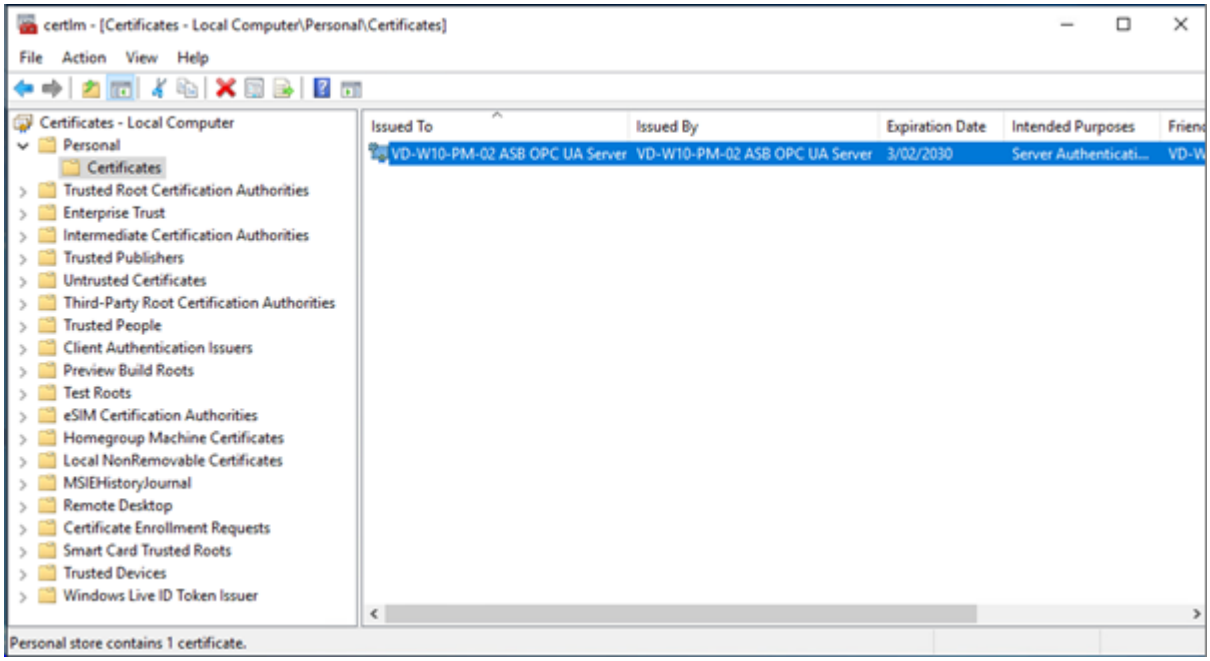
Export the OPC UA server certificate to the OPC UA client node

To export the OPC UA server certificate

1. Open the Windows Certificate Manager on the OPC UA server node.

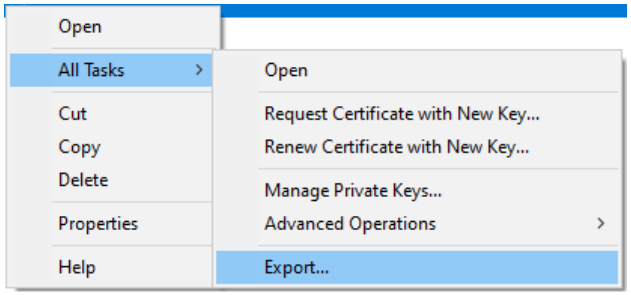
   To open the Certificate Manager, either type "`Manage Computer Certificates`" in the Windows search box and select, or open the command prompt and run "`certlm.msc`."

2. In the tree view, expand the **Personal** node, then click on **Certificates**.

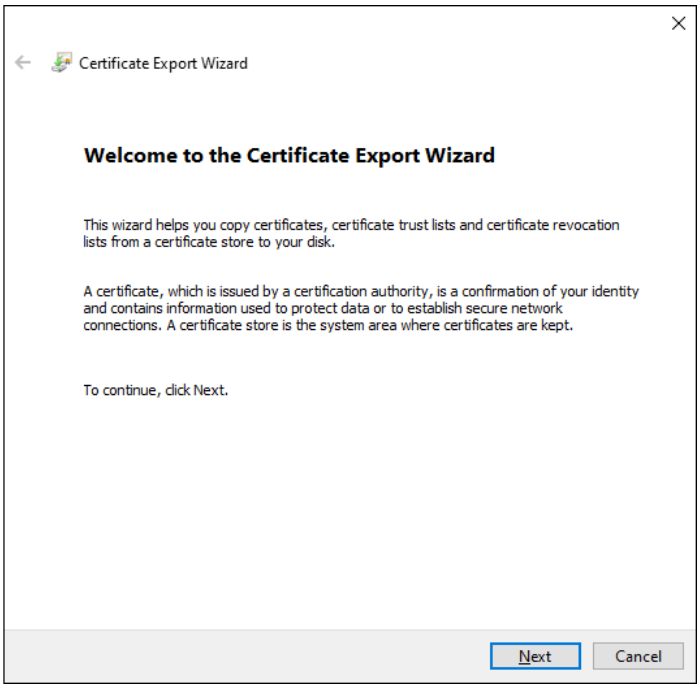


3. Locate the certificate "**<computer name> ASB OPC UA Server**".

4. Right-click on the certificate and select 'All Tasks\Export…"



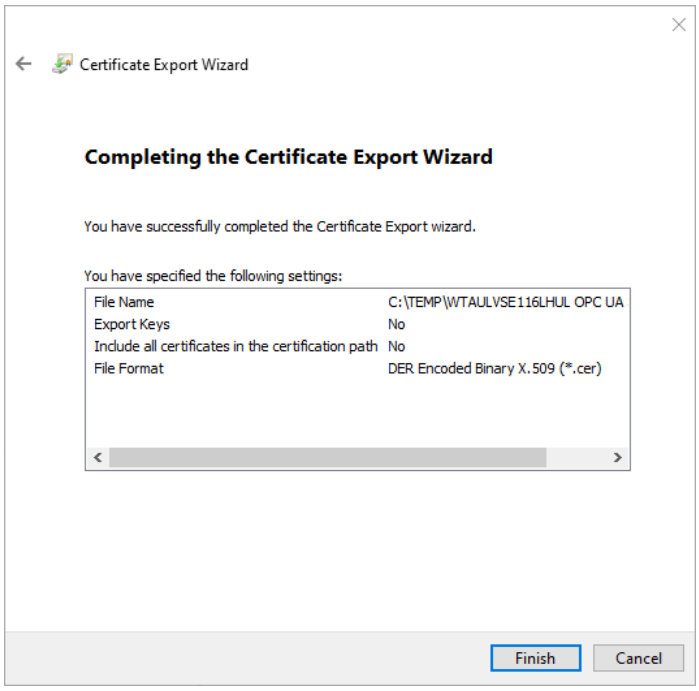
The **Certificate Export Wizard** opens.



5. Depending on type of certificates that your client application uses, you will need to export the certificate as one of two certificate file types:
   - **.cer file**, if the client uses the Windows Certificate Store
   - **.der file**, if the client uses file-based certificates.

   Even though the file extensions are different, the file formats are the same.
6. In the Certificate Export Wizard, select the following options.
   - **Export Private Key:** select "No, do not export the private key" (default), then click **Next**.
   - **Export File Format:** select either "DER encoded binary X.509 (.CER)" or "Base-64 encoded X.509 (.CER)," depending on the requirements of your client application. Make the selection, then click **Next**.
   - **File to Export:** enter a file name to export the Root CA, for example "`c:\temp\<machine name> OPC UA Server.cer`," then click **Next**.

7. When the **Export Wizard** finishes, you may need to change the file extension of the certificate from ".cer" to ".der," depending on what your client application is expecting. You will need to do this if your OPC UA Client application stores certificates in a specific folder, rather than in the Windows Certificate Store.



Import the OPC UA Server Certificate on the Client Computer

To complete the process of copying and installing the OPC UA server certificate to the OPC UA client node, you need to import the OPC UA Server certificate to the OPC UA client computer.

Each OPC UA client application has its own mechanism to manage certificates. Generally, an OPC UA client will use one of two mechanisms to manage certificates:

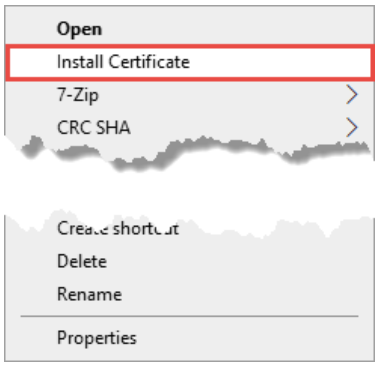
- Utilizing the Windows Certificate Store
- Storing certificates in a specified folder, defined by the OPC UA Client application.

Refer to the documentation for your OPC UA client applications for more details on how to import a server certificate.

To import a certificate into the OPC UA client Windows Certificate Store

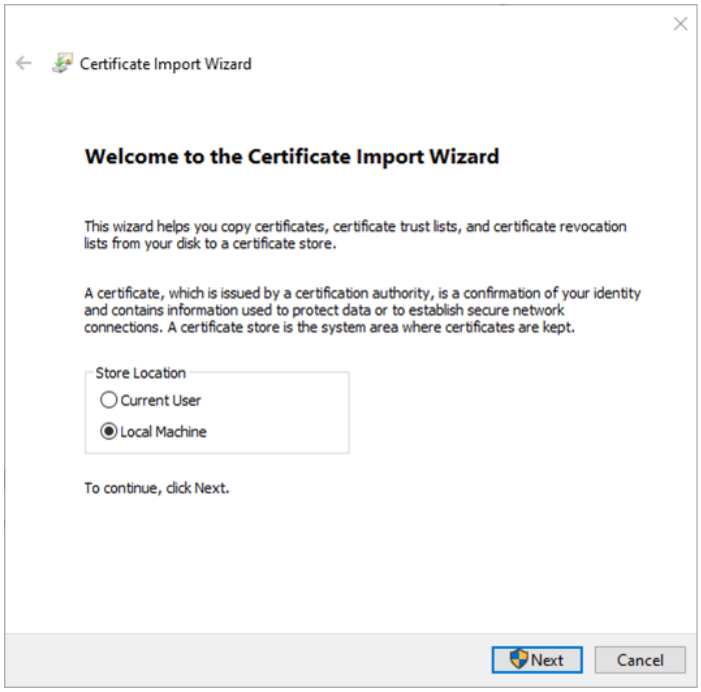
1. Copy the certificate file (<machine name> OPC UA Server.cer) to the OPC UA client node. Where you copy the file is not important.

2. Right click on the certificate file and select Install Certificate" from the context menu.



This opens the Certificate Import Wizard.

Note: Administrator privileges are required to import the certificate.



3. In the Certificate Import Wizard, select the following options.
   - **Store location:** Select "Local Machine," then click **Next**.
   - **Certificate store:** Browse to "Personal," then click **Next**.
   - **Completing the Certificate Import Wizard:** Review the settings, then click **Finish**.

To copy a certificate to specified location on the OPC UA client

1. Copy the certificate file (<machine name> OPC UA Server.cer) to the folder that is specified by your OPC UA client applications. The following table shows the location of the certificate folder for a number of common OPC UA clients.

OPC UA Client	Manufacturer	Folder
Datafeed OPC UA Client	Softing	C:\ProgramData\Softing\OpcClient\pki\trusted\certs
UaExpert	UnifiedAutomation	C:\Users\Admin\AppData\Roaming\unifiedautomation\uaexpert\PKI\trusted\certs
UA Client Getting Started	UnifiedAutomation	C:\ProgramData\unifiedautomation\UaSdkNetBundleEval\pkiclient\trusted\certs
Matrikon	Matrikon	C:\Users\Admin\Appdata\Local\Matrikon\OPCUAExplorer\pki\DefaultApplicationGroup\trusted\certs
KEPServer	Kepware	C:\ProgramData\Kepware\KEPServerEX\V6\UA\Client Driver\cert
Top Server	Software Toolbox	C:\ProgramData\Software Toolbox\TOP Server\V6\UA\Client Driver\cert

Refer to the documentation for the OPC UA client application for additional information about managing certificates.

2. Configure the OPC UA certificate, as described below.

Configure OPC UA Client Certificates on the OPC UA Server

The next step in the of configuring OPC UA server and client certificates is to trust the OPC UA client certificate that has been installed on the OPC UA server node.

The easiest way is to attempt to connect an OPC UA client to the server. Since trust of the client certificate has not yet been established, the connection is expected to fail.

Once the connection fails, any OPC UA client certificates that are not installed on the OPC UA server are placed in the “Rejected Certificate” folder on the OPC UA Server.

By default, the folder location is:

C:\ProgramData\AVEVA\PCS\OPC UA Rejected Client Certificates\certs

Note: Access to this folder requires administrator rights, and the folder is hidden by default.

To import certificates placed in the Rejected Certificate folder

1. To import the OPC UA Client certificates, browse to the rejected certificate folder.
2. Right click on the certificate for the OPC UA Client that you want to trust and select “Install Certificate”. This opens the **Import certificate Wizard**.
3. Select the following options in the wizard:
   - **Store location:** Select "Local Machine," then click **Next**.
   - **Certificate store:** Select “Trusted People,” then click **Next**.
   - Completing the Certificate Import Wizard: Review the settings, the click **Finish**.

Port Usage

OPC UA communicates via a single TCP port. This is specified in the **Endpoint Connection** setting in the configuration of the OPC UA Server, and defaults to port 48032. For details, see *Configuring the InTouch OPC UA Server* on page 34.

If you will run multiple OPC UA Servers on the same computer using RDS sessions, you must specify a different port number for each subsequent server.

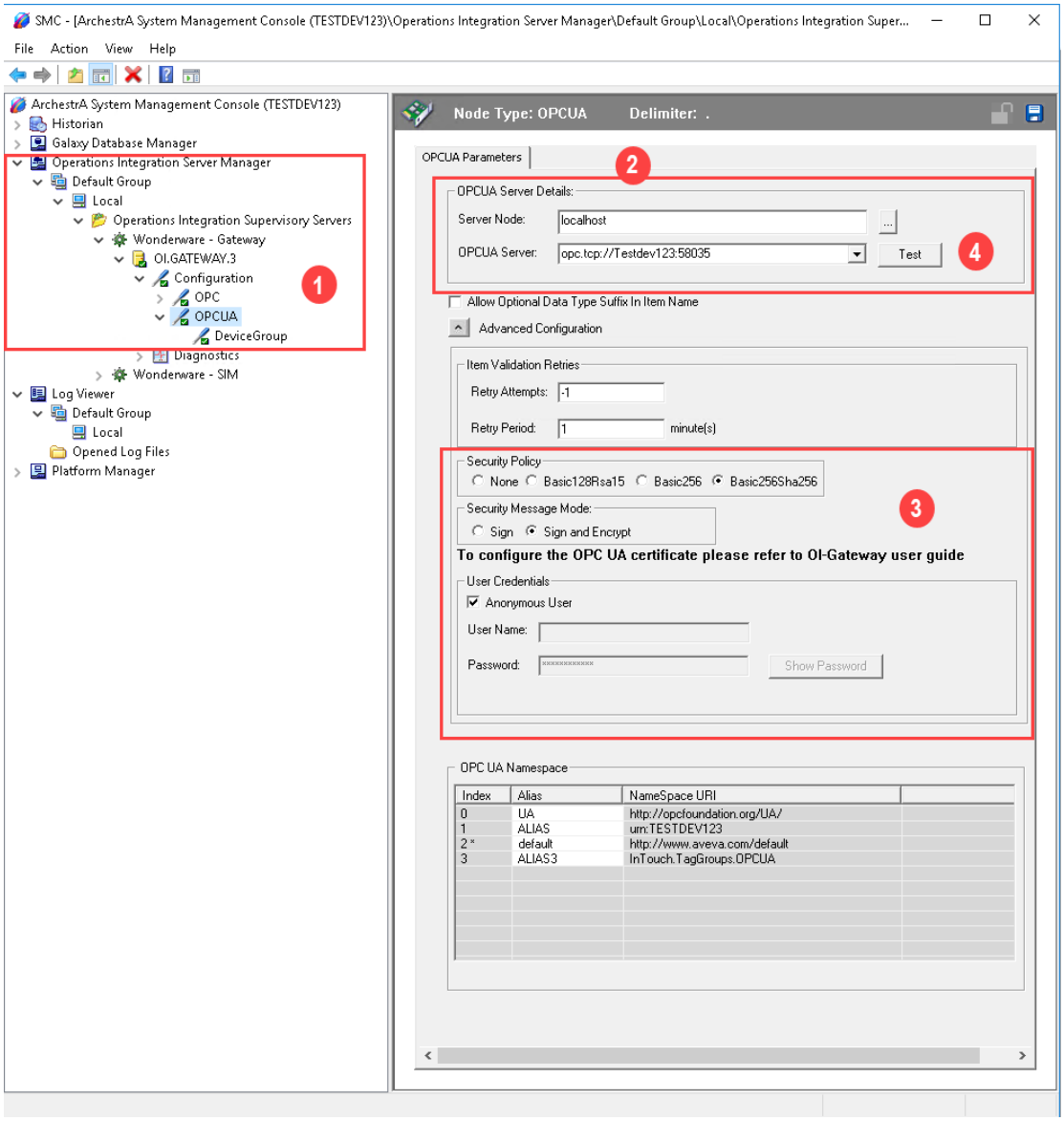
Confirm that this port is not blocked by any firewall software installed on your computer. For information about testing and configuring communications through firewall, see *Configuring the Firewall for the OPC UA Service* on page 36.

Using OI Gateway to Configure the Client Security Certificate

OI Gateway provides a convenient method for configuring security certification on the server and client OPC UA nodes.

To configure the security certificate through OI Gateway

1. From the Start Menu on the run-time node, open the System Platform Management Console (SMC) (**Start, AVEVA, System Platform Management Console).**



2. In the console tree, navigate to the **OI.GATEWAY.3** node under Operations Integration Supervisory Servers (**1**).
3. Create an OPC UA connection.
4. Configure OPC UA Server Details (**2**).
   - **Server Node:** Enter the machine name of the run-time node.
   - **OPC UA Server:** This is the URI (uniform resource identifier) for the OPC UA server (the run-time node). The address must be entered manually because it is not currently discoverable. Enter it in the format opc.tcp://<machine name>:<OPC UA port number>

Use the OPC UA port number that you entered when configuring the InTouch OPC UA Server in InTouch HMI Application Manager. The default port number is 48032.

5. Enter the authorization and authentication credentials (**3**).

   You must match the authorization settings configured in the OPC UA server dialog. If the Require Security Authentication checkbox is checked, then you must select the following settings:

   - **Security Policy:** Basic256Sha256.
   - **Security Message Mode:** Sign and Encrypt.
   - Under User Credentials, select **Anonymous User** to allow anonymous access. You can also provide user credentials of authenticated users if the corresponding option was selected during OPC UA configuration. The user credentials provided must be part of the InTouchHMIOPCUAWriteUsers user group.

6. Click the **Test** button (**4**). The test will fail, but it will download the OPC UA certificate.

   IMPORTANT! The reason for this initial test failure is because the certificates between the client and server applications must be trusted. Installing the certificates will fix this.

7. Go to the next section.

Once the certificates are trusted, the OPC UA client configuration will need to be validated.

Trusting the Certificate between the OPC UA Server and OPC UA Client

In this release of the OPC UA Server service, the operation of creating the trust between the OPC UA Server and the OPC UA client must be done manually. The **Test** operation causes the OI Gateway to submit its own certificate to the OPC UA Server node so it can be trusted. The following steps show how to then trust the client certificate from the OPC UA Server node. If you are not using OI Gateway, follow the procedures listed in *Configuring Server and Client Certificates for Third-Party OPC UA Client Applications* on page 40.

1. Access the folder **C:\ProgramData\AVEVA\PCS\OPC UA Rejected Client Certificates**.

   This is the location where the certificate from the client is initially placed by default as an attempt to connect to the OPC UA Server.

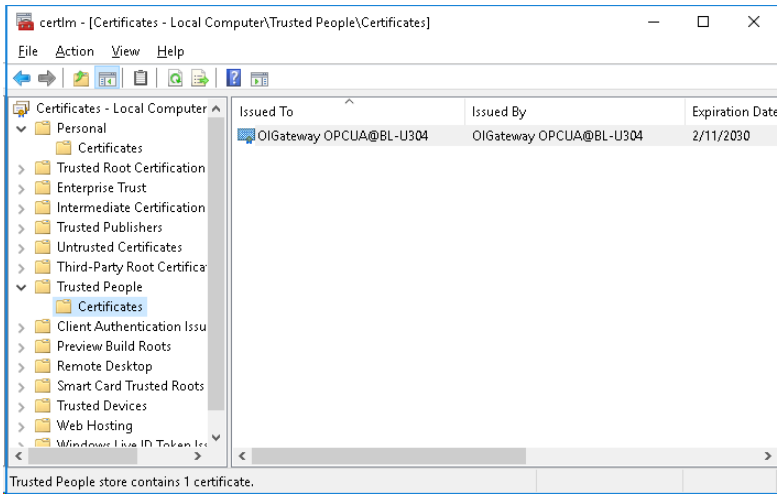
   Note: The ProgramData folder is hidden by default. You may need to enable the hidden items option in Windows Explorer in order to view it.

2. Right-click on the certificate name, for example, **OI Gateway OPC UA@OPCUA client node{long hex ID}.der**.
3. Select **Install Certificate** from the context menu. This opens the **Certificate Import Wizard**.
4. Select **Local Machine** for the Store Location, then click **Next**.
5. From the **Select Certificate Store** list, select **Trusted People** as the certificate store. This is the only choice that will work with the OPC UA certificate.
6. Close the wizard to complete installation.
7. Finally, delete the certificate from the **OPC UA Rejected Client Certificates** folder, since the certificate is now installed.

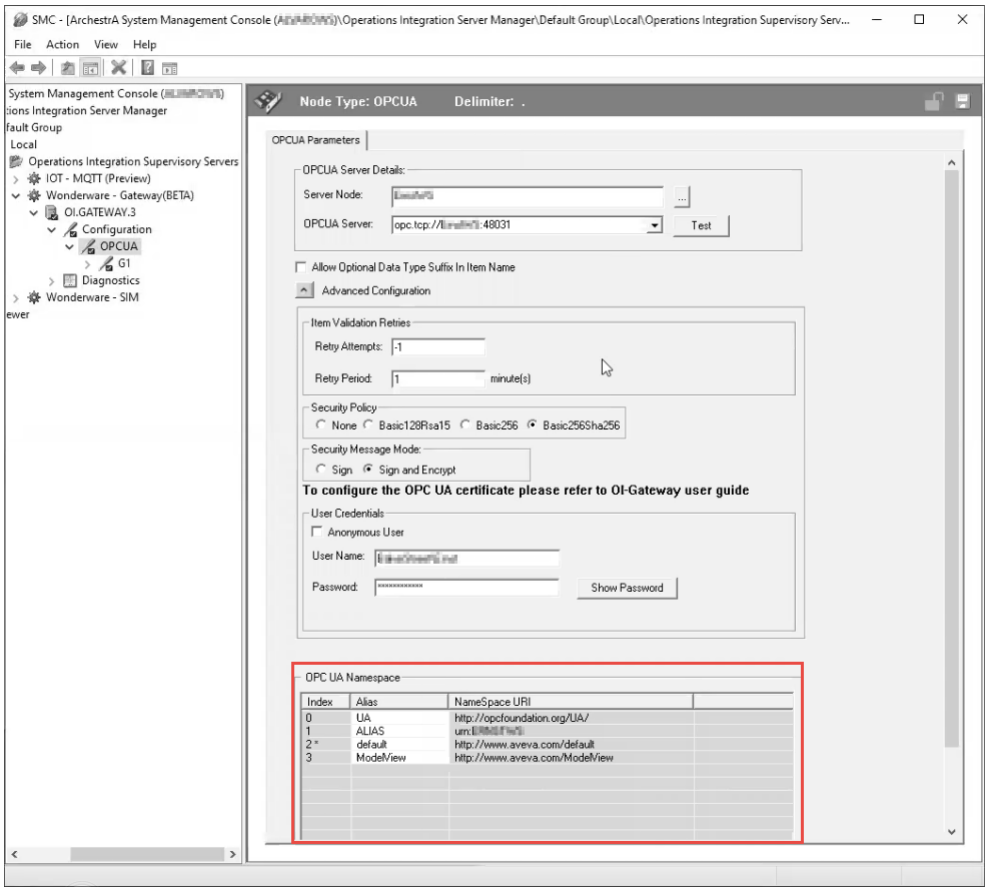
Verify OPC UA Certificate Installation

To verify certificate installation

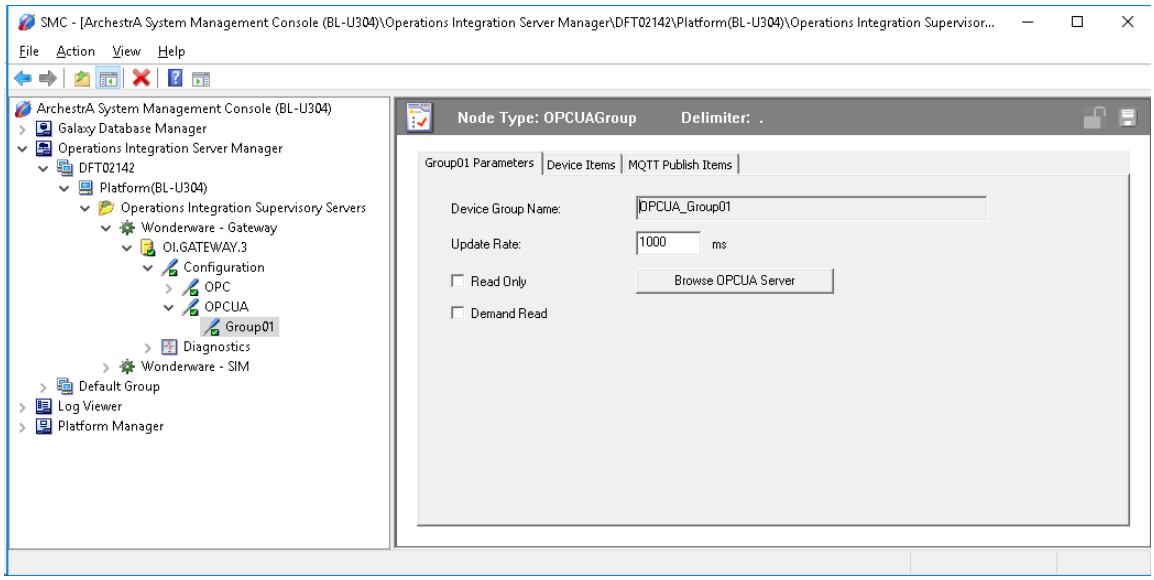
1. Open the certificate manager by typing **certificate** in the Windows search window, then select **Manage Computer Certificates** from the search results. The **Certificate Manager** opens.
2. Navigate the **Trusted People** folder and check that the OPC UA certificate has been installed.



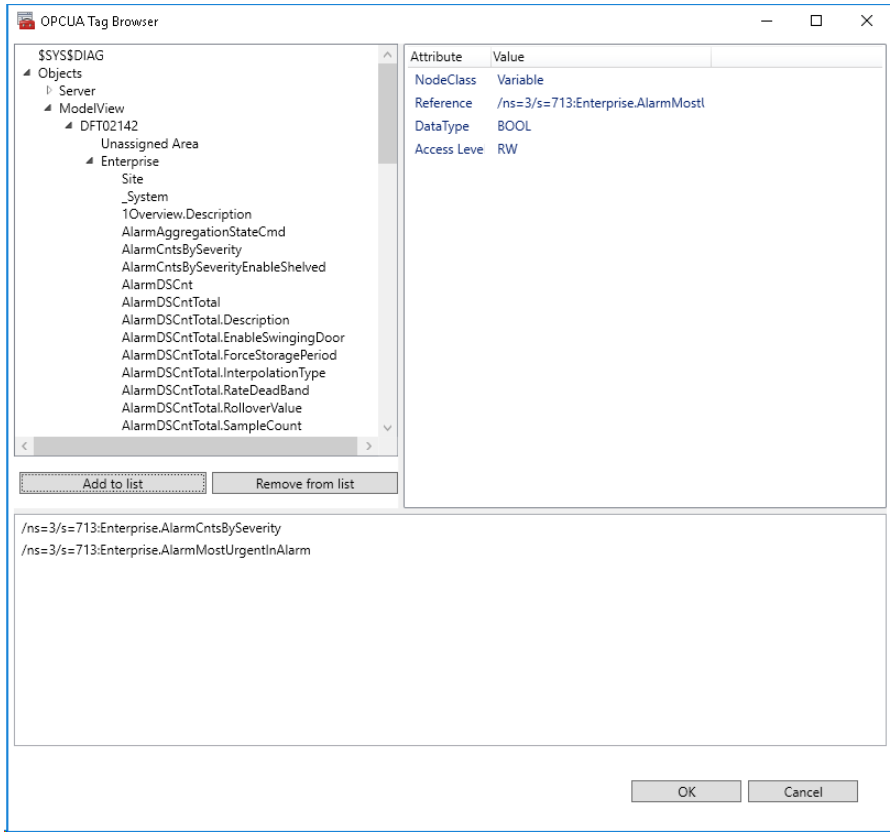
3. From the OPC UA client node, open the **System Platform Management Console** again, and click the **Test** button in OPC UA parameters window. At the bottom of the window, the **OPC UA Namespace** alias list will be automatically populated, indicating the connection was successful.



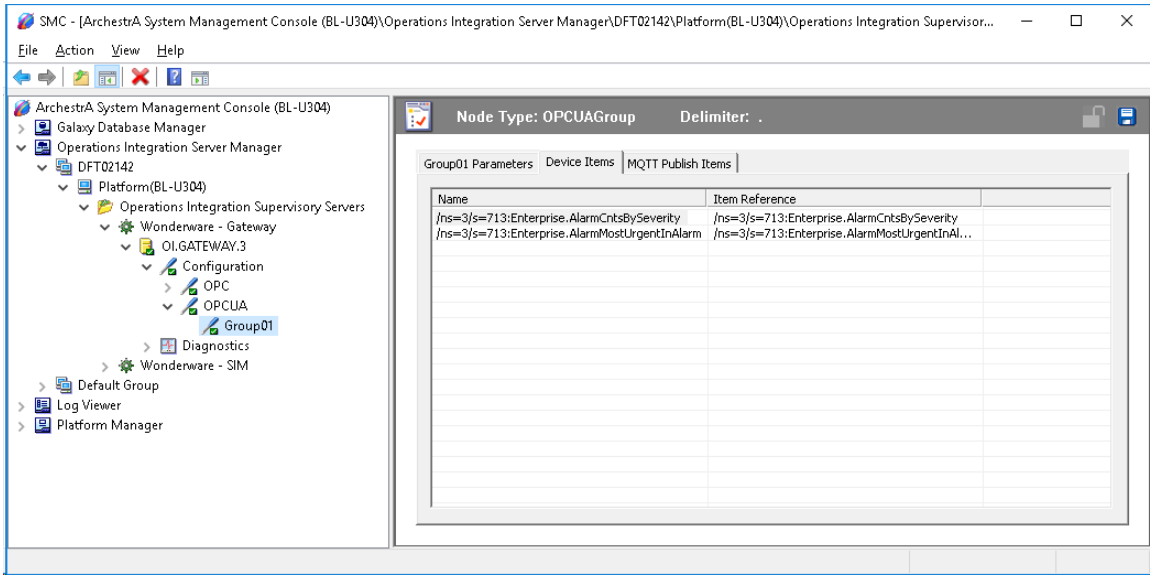
4. Add a group under the OPC UA Connection (named **Group01** in the following figure). The device group name prefixes **OPCUA_** to the group name you enter.



5. Click the **Browse OPC UA Server** button to browse the OPC UA hierarchy.
6. Optional: Add OPC UA tags to monitor in the alias list to help ensure that you can browse the namespace.



7. Switch back to OI Gateway configuration in the **System Platform Management Console**, and notice that the OPC UA tags are now listed in the **Device Items** window.



8. By default, item names in the list duplicate the complete item reference path. Rename the items as needed to be more user-friendly.

Launching an InTouch OMI ViewApp

Use the OMI tab in the Application Manager to launch the Viewer for InTouch OMI ViewApps.

1. Develop and deploy an InTouch OMI app.

   Once the ViewApp is deployed it will appear in the Application Manager OMI tab.

2. Select the ViewApp and click **Viewer** from the toolbar.

   The Viewer is launched and the ViewApp is displayed.

Updating Web Client Settings Using the Application Manager

The Web Client tab of the AVEVA Application Manager provides options for the user to configure Web Client related settings.

Updating the Web Client Settings:

1. In AVEVA Application Manager, click the **Web Client** tab. Configure the following settings:
   a. *Graphic Refresh Rate (ms)*: Set the rate on how frequently the web browser will query the web server for graphic data. The default is 1 second. For more information, see the System Platform Installation Guide.
   b. *Alarm Refresh Rate (ms)*: Set the rate on how frequently the web browser will query the web server for alarm data. The default is 1 second. For more information, see the System Platform Installation Guide.
   c. *Web Client Site Name*: Provide a string that will replace the standard URL.
   d. *Show Header*: Select the checkbox to display the Title Bar.
   e. *Enable NavBar*: Select the checkbox to display the Navigation Bar.

      This setting will be disabled if the Show Header setting is not selected.

   f. *Allow Anonymous Access*: Select the checkbox to allow users access to web client without authentication.

      For more information on settings c - f, see the *Viewing Application Graphics in a Web Browser* guide.
2. On modifying the settings, click **Apply**.

   Additionally, you can enable and disable the web client using the  and  icons.
3. Use the  icon to launch the web client.

Registering with the AVEVA Identity Manager

Using the AVEVA Identity Manager you can configure the Web Client to use Single Sign On, instead of the default Windows OS based authentication.

The steps to configure the Identity Server are:

1. In the System Platform Configurator, configure the *Common Platform > System Management Server*.

2. In AVEVA Application Manager, register the AIM server with user credentials.

The AIM Registration dialog box can also be used to configure the reverse proxy server:

1. Setup the reverse proxy server.
2. In the System Platform Configurator, configure the *Common Platform > System Management Server*.
3. Provide the Secure Gateway address in the AIM registration dialog box.

You can select a remote or local System Management Server. For more information on configuring the System Management Server, see the *System Platform Installation Guide*.

To register with the Identity Server:

1. In Application Manager, click the **Web Client** tab.
2. Click **Tools** and then **AIM Registration...**.

   The **AIM Registration** dialog box appears.
3. Click the **Use AIM Server as the authentication server** checkbox to enable the use of the AVEVA Identity Manager.

   The Identity Server field displays the Identity Server, configured in the Configurator.
4. Update the following settings:

   a. *User Name*: Provide the user name to connect to the Identity Server. The user must be part of the Administrators user group.

   b. *Password*: Provide the password for the corresponding user name.
5. To complete the reverse proxy setup, provide the URL of reverse proxy or DMZ server in the **Secure Gateway** field.

   This is an optional setting, and needs to be set only if the We Client is hosted behind a reverse proxy server.
6. Optionally you can also click the **Allow Industrial Graphics to be embedded in any website** checkbox to view the web client within an HTML iframe in runtime.
7. Click **OK**.
8. Click **Apply**.

Opening an Application in WindowMaker and WindowViewer

You must open a new application in WindowMaker before you can open it in WindowViewer.

To open an application in WindowMaker

1. Select the application in the Application Manager window.
2. On the **File** menu, click WindowMaker.

Tip You can also double-click an application to open it in WindowMaker.

To open an application in WindowViewer

1. Select the application in the Application Manager window.

2. On the **File** menu, click WindowViewer.

Customizing the Application Manager Window

You can hide the toolbar and status bar. You can also change how the applications are listed in the Application Manager window. Applications can be shown as Details, List and Icon.

Application Manager Views

- **Details** – This view lists the application in a tabular form with all the application related properties like path and resolution.
- **List** – This view displays the application name, type, thumbnail and the ribbon.
- **Icon** – This view displays the application name, type, resolution, description, thumbnail and ribbon.

To hide the toolbar

- On the **View** menu, click **Toolbar** so that no check mark is shown.

To hide the status bar

- On the **View** menu, click **Status Bar** so that no check mark is shown.

To change the view for the application list

- On the **View** menu, click the appropriate command or click the corresponding button on the toolbar.

To rearrange the order and availability of the Toolbar options

1. To customize the order of the options, click the icon and select **Customize…**
2. Under the **Items** tab, select an option (for example WindowMaker), and click the **Move Up** or **Move Down** buttons to change the order the options appear on the toolbar.

To remove option from the Toolbar

- Click on the checkbox against the option.

For example, clear the checkbox against the option WindowMaker and click Close. The Window Maker option will not appear on the toolbar.

Using the Application Ribbon

In the Icon or List view, each application displays a ribbon with application specific options.

The options are:

- **Change Thumbnail**: Click the icon, and select the image from the Browse dialog box. The image will be displayed in the Icon and List view.
- **WindowMaker**: Click to launch WindowMaker.
- **WindowViewer**: Click to launch WindowViewer.
- **Delete**: Click to delete the application.
- **Open application folder**: Click to navigate to the application folder.

- **Application properties**: Click to launch the application properties.

Click the  icon to see additional options.

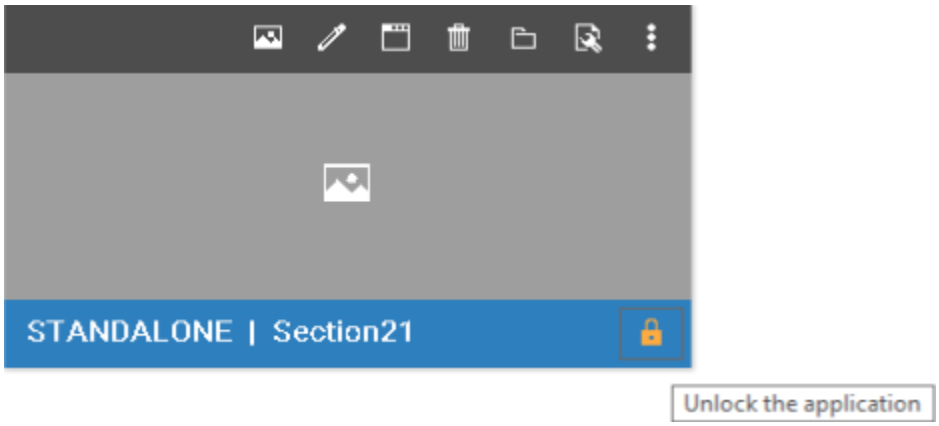
- **DBLoad**: Click to launch the DBLoad utility.
- **DBDump**: Click to launch the DBDump utility.
- **Rename**: Click and enter the new name of the application.
- **Export as Template**: See *Exporting InTouch Applications to use an Template* on page 60.

Locking and Unlocking InTouch Applications

During the use of WindowMaker and WindowViewer if the application was not closed correctly the application may be locked. You must unlock the application using the Application Manager.

To unlock an InTouch application

- Launch Application Manager, select the application, and click the lock icon.



The application is unlocked and available for use. For more information, see *Application Editing Locks* on page 91.

Modifying an InTouch Application

You can rename an application and change the application properties.

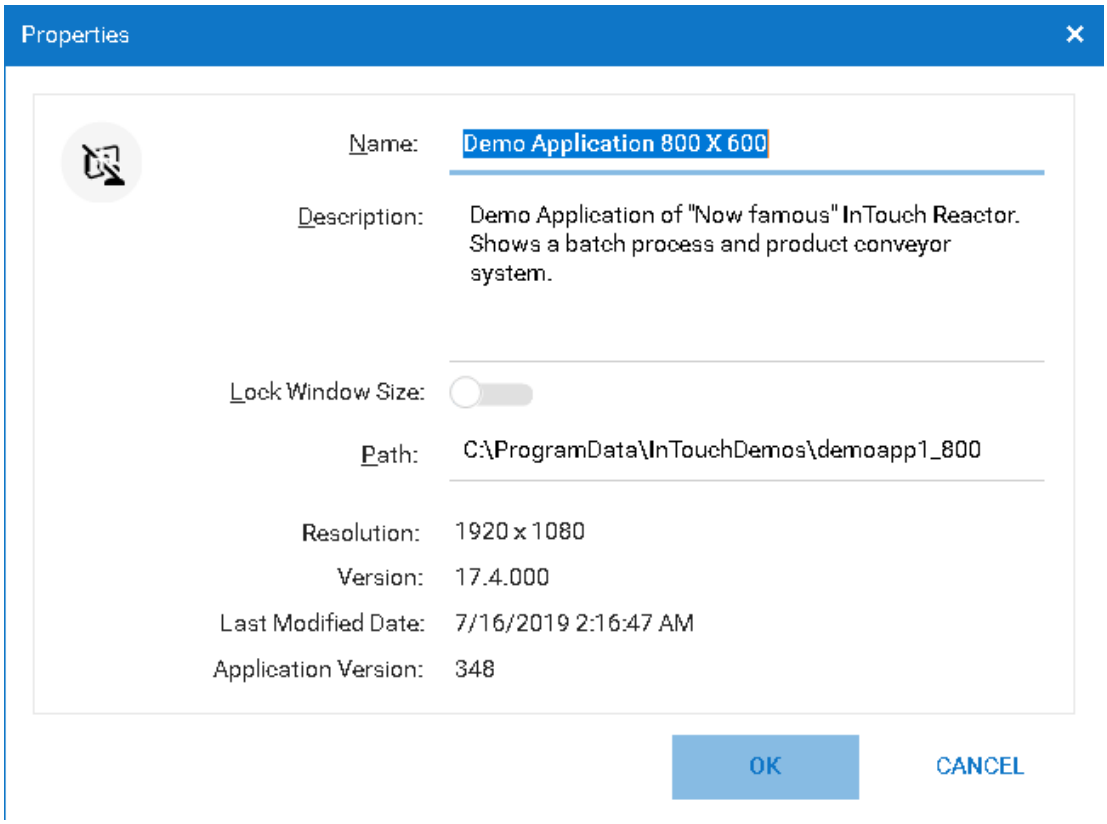
To rename an application

1. Select the application in the list.
2. On the **File** menu, click **Rename**.

To modify application properties

1. Select the application in the list.

2. On the **File** menu, click **Properties**. The **Properties** dialog box appears.



3. Configure the application properties.
   - In the **Name** box, type a new name for the application.
   - In the window, type another description for the application.
4. Click **OK**.

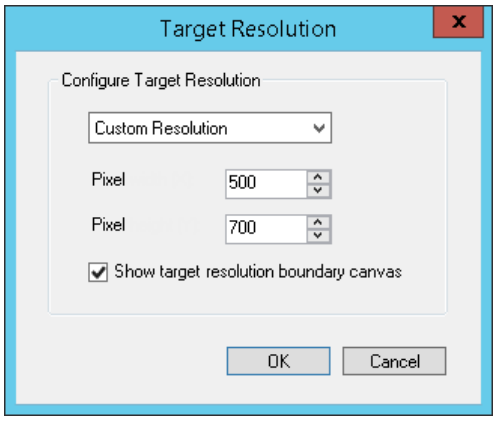
To Change Target Resolution

You can change the specified target resolution while editing the application in WindowMaker. This functionality is not available using command line.

Do the following:

1. Click **Special -> Configure**, and select **Target Resolution**.

The **Target Resolution** dialog box appears.



2. Edit the target resolution as needed and click **OK**.

   The boundary canvas will be modified to reflect the change. Graphics, window size and window controls will remain the same.

Note: The Show target resolution boundary canvas is checked by default.

Deleting an InTouch Application from the Application Manager

When you delete an application from the Application Manager, the application files remain on your computer.

To delete an application

1. Select an application in the list.
2. On the **File** tab, in the **Application** group, click **Delete**.
3. In the message that appears, click **Yes**.
4. You can also delete an application in the following ways:

- Right-click the application and select **Delete** from the context menu.
- Click the **delete** icon from the application ribbon.
- Select the application and press the **delete** key on the keyboard.

  You can select and delete multiple applications.

Deleting earlier Modern Applications

If the node contains older Modern application and the node is upgraded to the current Product version, you can select to delete the older Modern application. This action will permanently delete the Modern application folder and repository.

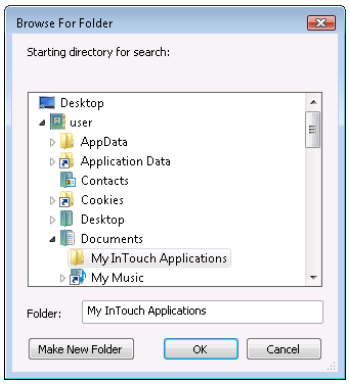
Finding InTouch Applications

You can search for existing InTouch applications. Application Manager shows any applications that are found.

An application cannot be opened in WindowMaker if the full path exceeds 114 characters (including the network drive letter, colon, and all backslashes). If an application exceeds the character limit, rename the folders in the path or move the application to a different location.

To find applications

1. On the **Tools** menu, click **Find Applications**. The **Browse For Folder** dialog box appears.



2. Select the folder in which you want to search for applications.
3. Click **OK**.

Working with Application Templates

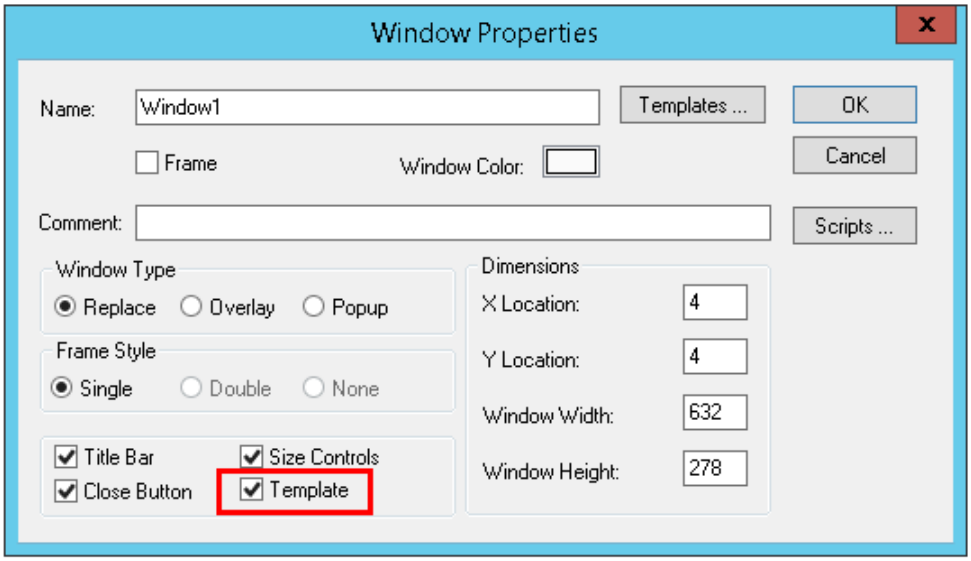
You can develop your own application template from a Standalone application. Developing an application template is a three-step process.

1. Create an application and set appropriate windows in the application to template windows.
2. Create an application thumbnail for preview in the Application Template Browser.
3. Export the application as an .aaPKG file to make it available as a template in the Browser.

To set application windows as template windows:

1. Create an application.
    a. Develop your application using graphics, scripts and windows.
2. Do the following for all windows in the application:
    a. Right-click each window and select **Properties**.

b. In the **Window Properties** dialog box, select the **Template** checkbox.



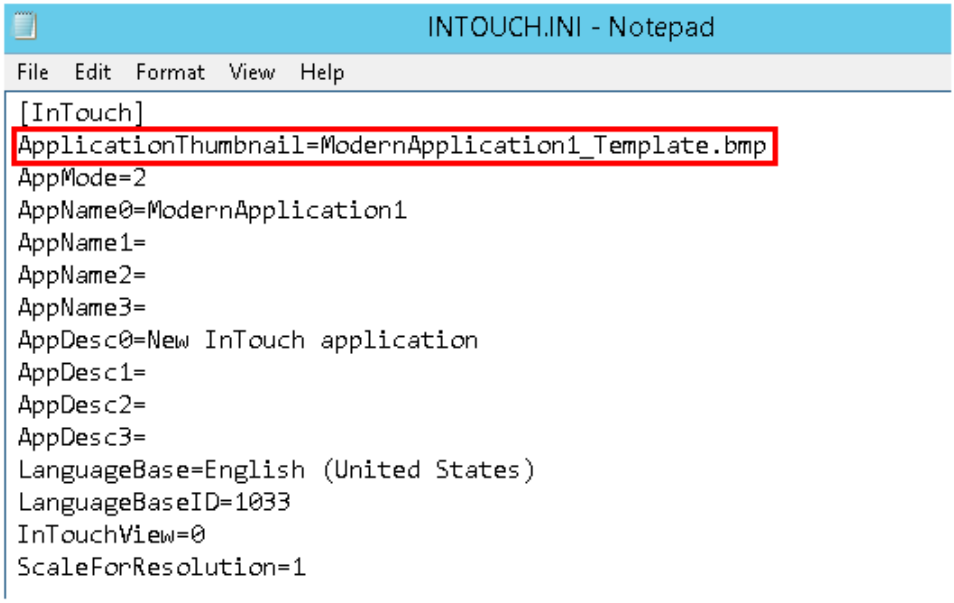
Upon click of **OK**, each window is automatically placed in the Template Windows folder in the Windows & Scripts pane.

You must now create and assign a thumbnail to the application you want to make into a template.

To create and assign an application thumbnail:

1. Using any screen capture program, take a screen capture of your application at either configuration or run time.
   a. Save the image to any picture file format, such as a .bmp or .png file, and copy it to your application folder.
2. From your application folder, open the INTOUCH.INI file with a standard text editor such as Notepad.
3. Edit the INTOUCH.INI file to include the file name of the image in the **ApplicationThumbnail** field.

Note: The **ApplicationThumbnail** field is case sensitive and must exactly match the name and extension of the thumbnail image.



4. Save your changes and close.

The application must now be exported as a .aaPKG file that will populate in the **Application Template Browser**.

To export application and create template:

1. In the Application Manager, export your application to create an .aaPKG file.

   For details on how to export an Application, see *Exporting InTouch Applications to use an Template* on page 60.

2. Copy the exported .aaPKG file into the following directory:

   ```
   C:\Program Files (x86)\Wonderware\InTouch\ApplicationTemplates
   ```

   Your application is now available as a template in the Application Template Browser.

Note: The application template thumbnail you created in the previous procedure is extracted from the exported .aaPKG file. If your application template appears in the **Application Template Browser** with a blank thumbnail, a valid image could not be extracted. Be sure a valid image file format was used to save the thumbnail and the exact filename is entered into the ITOUCH.INI.

Exporting InTouch Applications to use an Template

The Export As Template option allows you to export an .aapkg file that contains all the graphic related contents (symbols, client controls, script library, language, styles) and save it as a template. This template file can then be used to create other Standalone or Managed Application.

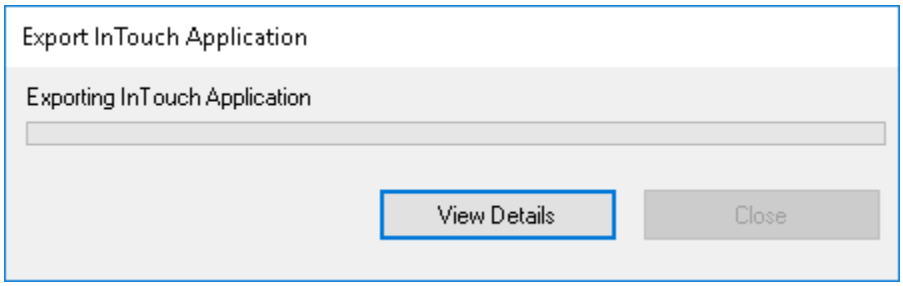
To export an application as a template:

1. In the Application Manager, on the File menu select **Export As Template**.

The Export InTouch Application dialog box appears.

2. Provide a location to store the .aapkg file.

   The Export InTouch Application progress window appears.



3. After the export is complete, click **Close**. Click **View Details**, to check for any errors.
4. The .aapkg file will be available in the location specified, and can now be used as an application template to create new applications.

Converting InTouch Windows to Industrial Graphics

You can convert the windows of a managed InTouch application to Industrial Graphics. The converted Industrial Graphics appear in the WindowMaker Graphic Toolbox and the IDE Graphic Toolbox. In addition to graphics that appear in windows, InTouch scripts are converted to ArchestrA scripts.

Preparing to Convert Windows

Before you convert InTouch windows:

- Only windows from InTouch standalone and managed applications can be converted to Industrial Graphics.
- Windows must be closed in WindowMaker to be converted.

Converting Windows

Only the symbols and scripts of a window are converted. The window's color, type, frame, title bar, size control, and **Close** button are excluded from the converted symbol.

Based on the InTouch graphic type, window graphics are converted as follows:

- All InTouch graphic primitives are converted to corresponding Industrial Graphic primitives.
- An InTouch Smart Symbol is converted to an Industrial embedded graphic.
- An Industrial Graphic within a window is converted to an embedded symbol. No new symbol is created for an embedded symbol.
- An InTouch symbol is converted to a group with the property TreatAsIcon=True.
- An InTouch cell is converted to a group with the property TreatAsIcon=False.
- InTouch Windows controls are converted to ArchestrA Windows controls.
- Some InTouch graphic components cannot be converted to Industrial Graphics:

- InTouch Real-time and Historical trends cannot be converted.
- ActiveX controls and the Distributed Alarm Display cannot be converted.

Converting Animation Scripts

All InTouch animation links embedded in windows are converted to the corresponding ArchestrA animation.

No validation warning or error messages are logged during the conversion. You should validate converted scripts with Industrial Graphic script validation to find any unsupported script syntax.

The following exceptions occur when converting InTouch animation scripts:

- Discrete Alarm and Analog Alarm of Line Color and Fill Color animation links are converted to Boolean and Truth Table of Line Style and Fill Style animations.
- The ShowWindow animation link is converted to an action script with the ShowGraphic script function. The HideWindow animation link is converted to an action script with the HideGraphic script function.
- All InTouch tags configured in animation link expressions have the prefix "InTouch" added to the tag name. For example, Tag1 is converted to Intouch:Tag1.
- The prefix "galaxy:" of ArchestrA attribute references configured in animation links are removed. For example, galaxy:UD001.Value is converted to UD001.Value.
- All InTouch script functions configured in action scripts are not converted. All scripts are copied over except the InTouch tag and ArchestrA attribute reference handling.

Known Limitations of Windows Conversions

Converting an InTouch window to an Industrial Graphic does not always achieve complete fidelity. In some cases, window components cannot be converted to a symbol. This section describes known limitations when converting an InTouch window to an Industrial Graphic and any alternative solutions.

- Converting a Window Containing a Vertical Mouse Release Slider SmartSymbol

  A Vertical Mouse Release Slider SmartSymbol contains two types of animation. The symbol uses fill animation to show the current measured value against a scale and a movable slider knob to set a value. A window containing an embedded Vertical Mouse Release Slider SmartSymbol does not retain the movable slider animation when converted to an Industrial Graphic.

- Converting a Window Containing a Symbol Factory Symbol

  Not all types of animations incorporated in Symbol Factory symbols can be converted to an Industrial Graphic. The following types of Symbol Factory symbol animations cannot be converted:

  - Percent Fill
  - Line Color
  - Horizontal Movement
  - Vertical Movement

- Migrating InTouch Translation Strings to Industrial graphics

InTouch translation strings are stored in an XML file in the application's folder. The translated strings of each language are placed in separate XML files. Translations of all windows and SmartSymbols strings are available from the XML file in the application directory after importing localized strings.

The following procedure explains how to use the Language Assistant tool to migrate InTouch window localization strings to an Industrial graphic.

a. Convert an InTouch window to an Industrial Graphic.

b. Import the contents of an InTouch language XML file to Language Assistant to create a global dictionary of phrases.

c. Export an Industrial Graphic to an XML file.

d. Import the symbol XML file to Language Assistant, which will automatically apply a global dictionary translation to the Industrial Graphic phrases.

e. Publish the Industrial Graphic XML file from Language Assistant.

f. Import the XML file to the Industrial Graphic Toolbox containing the translated text strings.

- Migrating InTouch Scripts to Industrial Graphics

  Scripts containing InTouch or QuickScript functions cannot be converted to an Industrial Graphic.

- Migrating InTouch History Objects to Industrial Graphics

  InTouch windows containing an InTouch Historical Trend are not completely converted to Industrial Graphics. Only some parts of the Historical Trend may appear in the converted symbol. Trend components like scooter bars may not appear in the converted Industrial Graphic.

After Converting Windows

After conversion, the symbol is added to the ArchestrA IDE **Toolbox** and the WindowMaker **Industrial Graphic Toolbox**. The original converted InTouch window is backed up. A new InTouch window is created in the InTouch application, which embeds the newly created Industrial Graphics.

If only a subset of the window is converted, then the user must manually verify that the converted windows will work with the non-converted window.

The name of the converted window is assigned by default as the name of the new Industrial Graphic. If the InTouch window name contains unsupported ArchestrA characters, an underscore (_) replaces each unsupported character. If the symbol already exists, a numerical suffix is appended to the name of the converted symbol. For example, Main_001.

The special characters dollar ($), pound (#) and underscore (_) are the only exceptions.

If an application is migrated, editing the window or modifying any of the window properties will replace any special characters in the window name to underscore (_). This is applicable for all types of windows including application, frame and, template.

A new toolset is created and assigned the InTouch application name for all converted symbols. The InTouch folder hierarchy is maintained by the toolset after the windows conversion.

- A converted symbol from an unassigned InTouch window is added to a toolset assigned the InTouch application name.

- If the window belongs to an assigned area, the original folder structure is created in both toolboxes and the name of the InTouch application is assigned as the top root folder name.

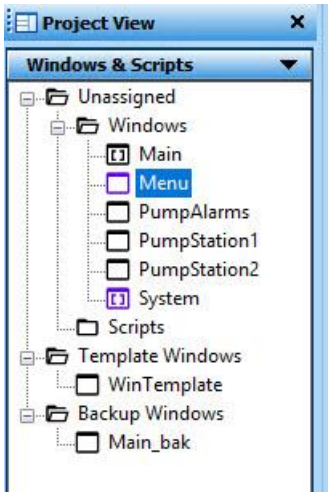
Completing the Window Conversion Procedure

To convert windows to Industrial Graphics

1. Close all InTouch windows that will be converted to symbols.
2. Convert one or more windows to Industrial Graphics by one of the following methods:

   Shortcut Menu Method

   a. Select the window to be converted from the **Windows & Scripts** pane.



   b. Right-click to show the shortcut menu and select **Convert To Industrial Graphic...**.

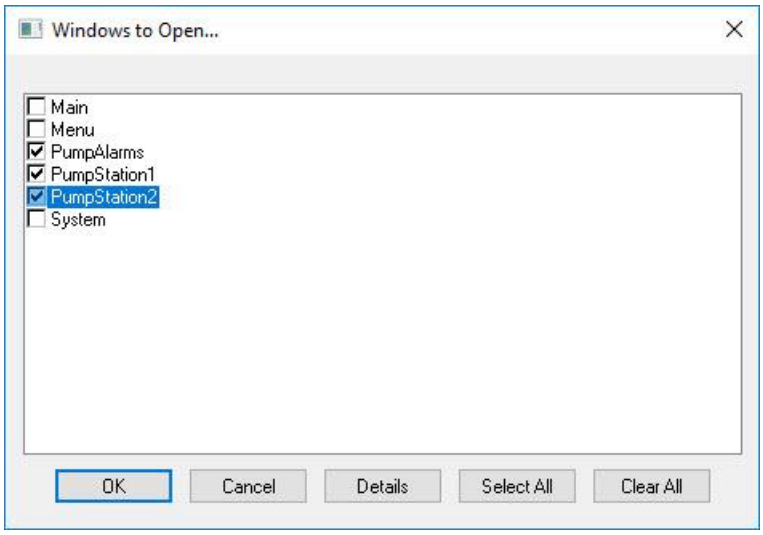
   c. Continue at Step 3.

   File Menu Method

   d. Click **File** from the menu bar and select **Convert To Industrial Graphic...**.

3. The **Windows to Convert** dialog box appears, select the windows that you want to convert.

By default, the window selected in the previous screen will be checked. You can now select additional windows.

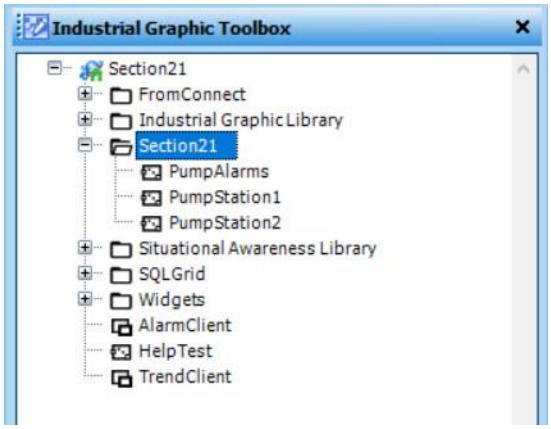


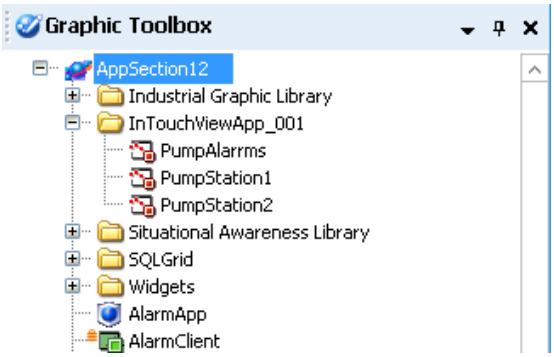
4. Click **OK**.

   A message appears and indicates the windows are being converted in succession. After the windows are converted, a succession of **Check In** dialog boxes appear to enter an optional comment for each window that was converted.

5. Observe the WindowMaker **Industrial Graphic Toolbox** and the ArchestrA IDE **Graphic Toolbox.**

   The converted windows should appear as Industrial Graphics in both tool boxes.





Diagnosing Window Conversion Errors

During conversion, a progress bar shows warning or error messages that occur during the window conversion. A custom log flag is created containing useful diagnostic information for every element, animation, or script while converting windows.

After conversion, click **Window conversion Report** from the conversion progress bar to export all messages to an HTML conversion report that can be viewed in a message window.

Both the conversion progress dialog box and the conversion report include the following information about a window conversion:

- Window name
- Conversion status that indicates if the window was converted successfully or if errors occurred.
- Warning messages that occurred during the conversion.
- Error messages that occurred during the window conversion.

Publishing Applications to Remote Nodes

Using Application Publisher, you can create a compressed, self-extracting package file that contains all relevant files and setup procedures to install an InTouch application on another computer. You use Application Publisher to publish standalone InTouch applications. You publish managed InTouch applications using the ArchestrA IDE.

You have two options to publish applications:

- **Run-time only.** A run-time only package includes the files needed to run the application, but not to edit the application.
- **Design-time and run-time.** A design-time and run-time package includes all files needed to edit and run the application. Some run-time files, such as compiled *.wvw files, are excluded because they can be re-created from the design-time files.

You can post published applications to a web server where they can be downloaded and installed. The following package information is shown for posted applications:

- Package description
- Publisher name
- Published file name (executable)
- Application resolution

For example:

Description	Dairy Processing Application
Publisher	Navin Johnson

File Name	Dairy.exe / Video Resolution…(1024x768)

Description	Dairy Processing Application
Publisher	Navin Johnson
File Name	Dairy_2.exe / Video Resolution…(800x600)

Contents of a Published File

The following table lists the included folders, files, and excluded files for all published stand-alone InTouch applications.

Included Folders	Included Files	Excluded Files
Main application folder	All	Backup files. These files have the .?bk file name extension.
	Files with these extensions: .win, .dat, .lgh, .idx, .log, .fsm, .stg, .$$$	Subfolders not in the Special Directories list
	retentiv.x retentiv.d retentiv.a retentiv..s (two dots) retentiv.h wm.ini db.ini linkdefs.ini tbox.ini group.def itocx.cfg	The appedit.lok file, which indicates that the application is open in WindowMaker.
	Any files with names of the form SSD_*.xml.	Compiled window files with the file name extension .wvw.
Dictionary subfolders for run-time language switching	All files with the .xml extension.	
Symbol subfolders	All files and subfolders.	

Included Folders	Included Files	Excluded Files
	wiz.ini file, if there are wizards installed.	
	Copy of the wizard executable.	
	.dll files, .wdo files .wdf files	

For a run-time only application, all files with a file names of SSD_*.xml are excluded.

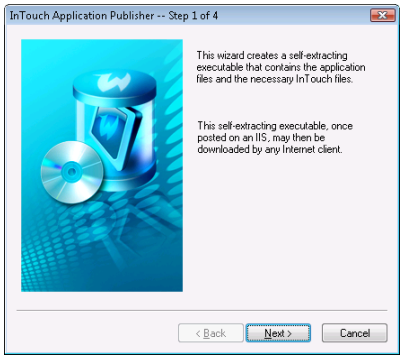
Publishing a Standalone InTouch Application

Use the Application Publisher to publish a standalone InTouch application. If you want the published application to run at a specific screen resolution, set the original application to that resolution before you publish it. To publish a managed InTouch application, use the ArchestrA IDE.

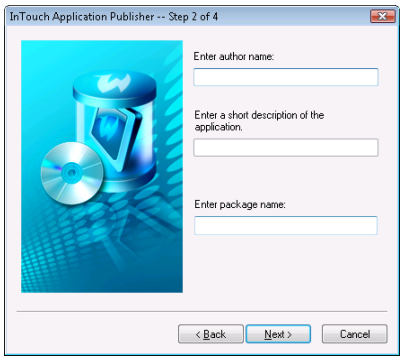
To publish a standalone InTouch application

1. Start the Application Publisher.
   a. Open WindowMaker.
   b. Show the Classic View and expand the **Tools** pane.
   c. Expand **Applications**.
   d. Double-click **Application Publisher**.

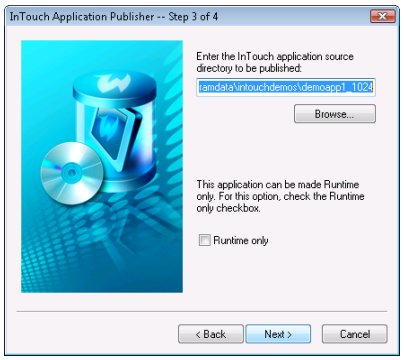
      The InTouch **Application Publisher – Step 1 of 4** dialog box appears.



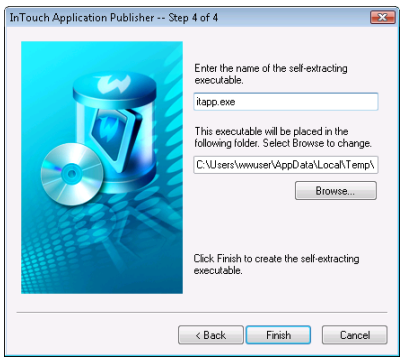
2. Click **Next**. The InTouch **Application Publisher – Step 2 of 4** dialog box appears.



3. Configure the package details.
   - In the **Enter author name** box, type the name of the person to contact regarding the application. The name limit is 256 characters.
   - In the **Description** box, type a description of the application. The limit is 256 characters.
   - In the **Package Name** box, type a unique name for the published application package. The limit is 32 characters. If you use the name of an existing package, the existing package is overwritten.
4. Click **Next**. The InTouch **Application Publisher – Step 3 of 4** dialog box appears.



5. Configure the publishing details.
   - In the box, type the path to the InTouch application folder. The default path is the WindowMaker application folder.
   - Select the **Runtime only** check box to exclude the development WindowMaker files in the published file.
6. Click **Next**. The InTouch **Application Publisher – Step 4 of 4** dialog box appears.



7. Configure the details for the executable application package.
    - In the first box, verify the executable name in the first box is correct. By default, the executable name is the same as the package name.
    - In the second box, type the path to the folder in which to save the executable file, or click **Browse** to select a different folder. By default, the executable is saved in your current temporary folder.
8. Click **Finish**.

Publishing Applications to Insight

Using the Insight Publisher you can publish an application to the Insight website. You can use the Application Manager or WindowMaker. In the Application Manager, select the AVEVA Insight Publisher icon from the Toolbar. In InTouch WindowMaker, you can use the **AVEVA Insight Publisher** from the **Tools** pane under **Applications**.

You can select one of the following options and proceed with the onscreen instructions.

- **Publish** - Create a new Insight data source from an existing InTouch application.
- **Import** - Import an Excel spreadsheet listing items from an OPC, MQTT, or OI Server.
- **Authorize** - Create a data source.

For more information, refer to the Historian Documentation.

Note: You will need an Insight Account to publish the application.

AVEVA

Chapter 4

Migrating and Upgrading Applications

Moving from a Legacy Application to the New Standalone Application

Prior to System Platform 2020, InTouch HMI users could create the following types of applications:

- Standalone
- Modern
- Managed
- Published

Standalone applications were built using legacy symbols and controls. Modern applications supported the use of Industrial Graphics (formerly known as ArchestrA graphics/symbols) in addition to legacy symbols. Managed applications were built using the IDE and Galaxy Objects. Standalone applications could be published into a package and then distributed to other nodes, resulting in Published applications..

In InTouch HMI 2020, modern applications have been redesigned as more comprehensive standalone applications. The new standalone application offers many improvements over legacy standalone applications.

- Easy distribution – Copy and paste the application folder to a different node. No import or export operations needed.
- Use of Industrial Graphics – The new standalone application combines the ease of use of earlier legacy applications with modern industrial graphics.
- Ready for the cloud – Applications created on-premise nodes can now be viewed on a HTML5 compliant browser.
- Light weight – The applications files are light weight and allow for better performance and use.

There is no change in behavior of Managed and Published applications.

Migrating and Upgrading Older Applications

To support applications created in earlier versions of InTouch HMI, you can use two workflows to transition to the new standalone application.

- In-place migration of older modern applications: If the node contains old Modern applications and the product version on the node is upgraded, then you migrate the application using the Application Manager.

- Importing .aapkg files of modern applications exported from earlier versions of InTouch HMI.

Migrating Earlier InTouch Applications to the Current Version

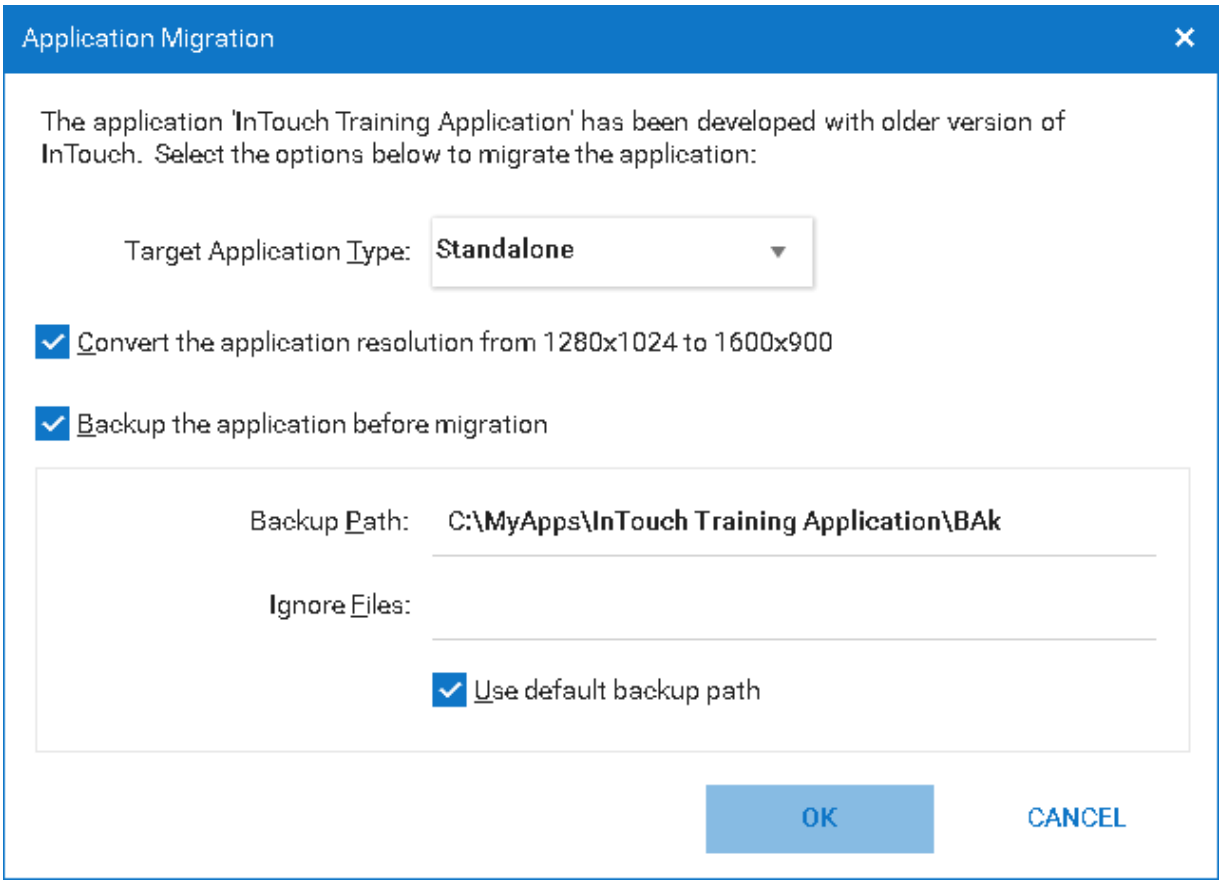
You can migrate applications developed with older versions of the InTouch HMI to the current version. When you attempt to open an older application with either WindowMaker or WindowViewer, you are shown the Application Migration dialog box. Here you can:

- Choose to convert the application resolution.
- Create a backup copy before migrating the old application to the current version of the InTouch HMI.

You can migrate existing standalone, modern, or published InTouch applications to the current InTouch version. You must specify the folder to create the backup copy and if you want to exclude any files from the backup.

1. From the application list, double-click on an application. The Application Migration dialog box appears.
2. To convert the application resolution from the original to the current resolution, select the Convert the application resolution from <existing resolution> to <new resolution> checkbox.
3. To change the default backup path (<Application Directory>\Bak), clear **Use Default Backup Path** box. Then, type the path to the folder in the **Backup Path** box where you want to save the backup. If the folder does not exist, you must create it, and then create the backup.
4. In the **Ignore Files** box, you can specify any files that you want to exclude from the backup. By default, all files in the application directory are backed up. Type the file names you want to exclude separated by a semicolon (;). Or, use standard wild card characters ('*' and '?') to exclude a set of files by the common characters in their names.

5. After configuring the necessary options, click **OK**.



Converting Legacy Alarm Displays

When you open an application built with a version before InTouch 7.11 in WindowViewer, a dialog box appears prompting you to run WindowMaker to convert the application. If you continue with the conversion, all of the Standard Alarm Objects are converted to Distributed Alarm Objects with default values. Colors, fonts, expressions, and alarm query settings are not preserved.

Managing Application Settings

InTouch application settings, such as the application path, are stored in the Win.ini file. The Win.ini file is located in the below directory:

C:\Users\<User Name>\AppData\Local\Wonderware

WindowMaker runs as an administrative user and WindowViewer can run as an administrative or standard user. The standard user cannot access the Win.ini directory of the administrative user profile. Therefore, as the application developer, you need to copy the common Win.ini attributes to the standard user's Win.ini profile when you develop the application. This ensures that all the attributes that are set under the administrative user are also available when WindowViewer is started by the standard user. You must copy the attributes each time you make changes to the common Win.ini attributes.

Importing InTouch Applications

You can import existing modern applications using the Application Manager, which will be converted to a Standalone application.

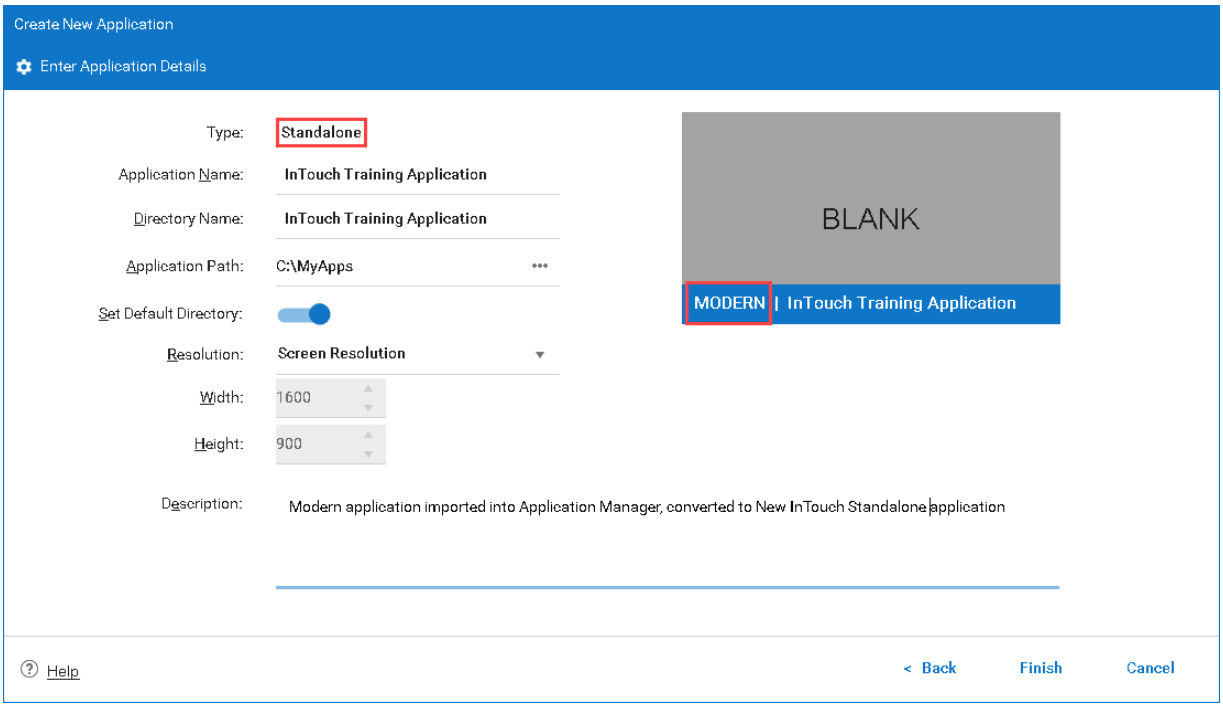
Note: Standalone applications can be copied from one node to another and found using the Find Applications option, they do not need to be imported or exported.

To import an existing modern application:

1. On the **File tab** , in the Main group, click **Import**.

   The **Create New Application: Select an application to import** dialog box appears.

2. Use the Find Applications section to search for the application you want to import. Search for a folder or a file to import.
3. Select the application and click **Next >**.
4. Make any changes to the settings. You will notice that the modern application is imported as a standalone application.
5. Click **Finish**.



A new application is created and displayed in the Application Manager.

Chapter 5

Distributing Applications

About Distributing Applications

Distributed applications typically have a central development station, central data storage, and client stations. You use InTouch Network Application Development (NAD) to build and maintain distributed applications. NAD allows many client stations to maintain a copy of a single application without restricting the development of that application. Using an individual copy of the application provides Viewer redundancy. Client stations are automatically notified when the application changes. You can create single computer, client-based, and server-based InTouch applications.

You can also manage and deploy InTouch applications using the ArchestrA IDE. For more information about using the ArchestrA IDE with the InTouch HMI, see *About InTouch HMI and ArchestrA Integration* in the InTouch® HMI and ArchestrA® Integration Guide.

Supported InTouch Architectures

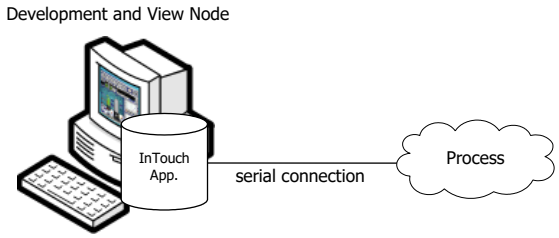
Supported InTouch network architectures are:

- Single computer
- Client-based
- Server-based
- NAD

Single Computer Architecture

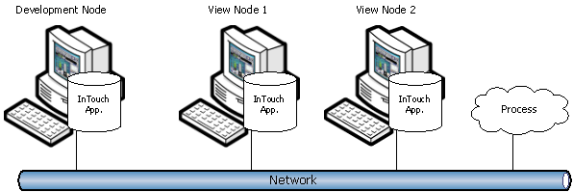
A single computer application typically consists of one non-networked computer that functions as the primary operator interface. This computer is connected to the industrial process with a direct connection, such as a serial cable.

In this architecture, you develop the InTouch application on the single computer. You can copy the application to another computer to modify it and then copy it back to the original computer.



Client-Based Architecture

In a client-based architecture, there is a unique copy of one InTouch application for each computer running WindowViewer (View node) or in a unique location on a network server. In the following example, an application is developed and tested on the development node and then copied to each View node.



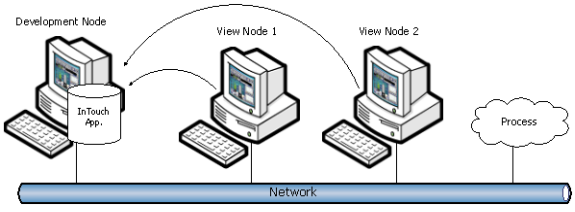
There is inherent redundancy because each node can be self-sufficient, and there is no limit to the number of View nodes you can use.

Each View node must have an identical copy of the application and identical access to any network data sources, such as I/O Servers or the IndustrialSQL Server. However, each View node maintains a separate conversation with the shared server, which can result in increased network loading.

You can modify and test the application on the development node without affecting the running process. However, you must distribute the modified application to the View nodes. You must shut down each View node locally, copy the new application to it, and then restart.

Server-Based Architecture

A server-based architecture distributes a common InTouch application to several View nodes. In the following figure, two View nodes access the same application from the development node.



For each View node:

- A logical drive must be mapped to the shared network drive of the development node.
- The shared application must be registered with the InTouch program.

- The computer must have identical access to any data sources referred to by the application. There are also ways to define the data source locations by using a combination of scripts to identify the node name and change each data location based on that name.

In this architecture, there is a single application to maintain. View nodes are automatically updated when the application changes and WindowViewer restarts.

Disadvantages of this architecture are:

- Development of application is restricted
- No redundancy if the development node goes down
- All nodes must have the same screen resolution

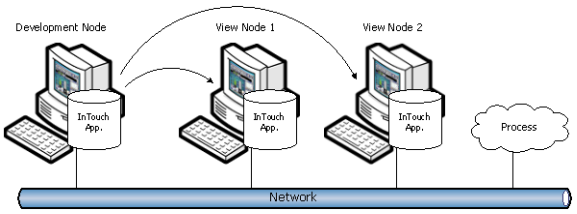
Network Application Development (NAD)

In the Network Application Development (NAD) architecture, you maintain a master copy of an application on a central network location, which is usually the development node. Each View node copies the application to a user-defined location and runs it.

When you notify clients of application changes (using the Notify Clients command on the WindowMaker Special menu), a flag is set in the application directory, which is then read by the View nodes.

You can configure how you want application changes handled for the View nodes. These range from ignoring the changes to automatically shutting down and restarting the View node, which reloads the master application.

In the following figure, the two View nodes have the master application registered from the development node, but actually run it locally on their computers.



Note: If you configure your application to write historical data to the master application node's application directory, all NAD nodes attempt to write their historical data to the master application. To avoid this, on each NAD node, configure historical data to write to a local directory, not the master application node.

If you are distributing a large, complex application to numerous nodes, slow system response time may be apparent on the initial download. Updates, however, are optimized. Application transfer may be a problem for slow networks or over serial connections.

Also, be aware of other network constraints, such as the user of routers that filter out certain types of network traffic and addresses.

Planning Considerations for Networked Applications

Regardless of the architecture you choose when building your InTouch application, it is important to consider:

- Access to I/O data sources.
- Access to shared files.

- Where data is logged.
- Any special network requirements.

I/O Data Access for Networked Applications

The InTouch HMI uses Access Names to reference real-time I/O data. Each Access Name equates to an I/O address, which consists of a node name, an application, and a topic. In a distributed application, I/O references can be set as global addresses to a network I/O Server or local addresses to a local I/O Server.

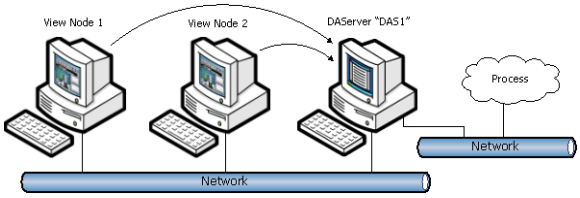
Note: InTouchView is restricted to the single Galaxy Access Name. You cannot create other Access Names for InTouchView. For more information about the restrictions of InTouchView, see *Viewing Applications at Run Time* on page 228.

The View node must have the same access to data sources as the development node.

Using Global I/O Addresses

Global addresses to I/O data allow all View nodes to share a common network-based I/O Server. This eliminates the need for multiple I/O Servers, but is less fault-tolerant and can result in lower overall performance.

In the following figure, two View nodes are running a copy of the same application. Both View nodes refer to the same I/O data source. Because each application uses a fully qualified I/O address for the data source, all references point to the same I/O Server.



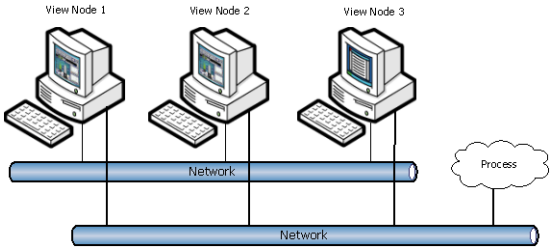
You can set up an InTouch application to identify an element of data stored on another node by using a three-part addressing convention in an Access Name. The Access Name addressing convention includes the node name, application name, and topic name where the remote data is located. An InTouch application obtains remote data using the Access Name in combination with an item name. For more information about defining an Access Name for a remote I/O Server, see Data Access with I/O in the InTouch® HMI Data Management Guide

Note: When you create Access Names in WindowMaker, if the Access Name uses the SuiteLink protocol, the software prevents Access Names from accessing the same node, application and topic. Do not use the IOSetAccessName() function to redirect Access Names to duplicate ones during run time or else the redirected Access Name will not work.

Using Local I/O Addresses

Local addresses to I/O data are used when each View node has its own I/O Server. This architecture provides fault-tolerant operation, as each View node can continue to run independently if the network goes down.

In the following figure, two View nodes run copies of the same application that refer to their own I/O data source. Because each application uses a local I/O address for the data source, each reference points to the local I/O Server.



Using a local I/O Server significantly increases the load on the process connection network. For example, three nodes triples the traffic created by one node, as each node's requests must be separately processed.

For more information about defining an Access Name for a local I/O Server, see Data Access with I/O in the InTouch® HMI Data Management Guide.

SuiteLink

The SuiteLink communications protocol is based on the TCP/IP protocol. Use SuiteLink for your high-speed industrial applications, as it provides these features:

- Value Time Quality (VTQ), in which a timestamp and quality indicator are associated with all data values delivered to VTQ-aware clients. The InTouch HMI is a VTQ-aware client whose tag data is delivered with a VTQ indicator.
- Extensive diagnostics of the data throughput, the server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows operating system performance monitor.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large number of nodes.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. Each connection between a client and a server depends on your network requirements.

Access to Shared Files

In a distributed application, file references can be set up as:

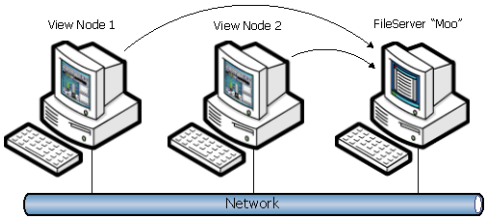
- Global addresses to a network file server.
- Local addresses to local files.

The View node must have the same access to data sources as the development node.

Using Global Addresses to File Data

You can set up global addresses to file data so that all View nodes share a common network-based set of files. This provides single-source maintenance of the files, but it is less fault-tolerant than local copies.

In the following figure, two View nodes are each running a copy of the same application, but reference the same recipe file. Because each application uses a drive letter mapped to a fully-qualified network path for the file, all references point to the same file.

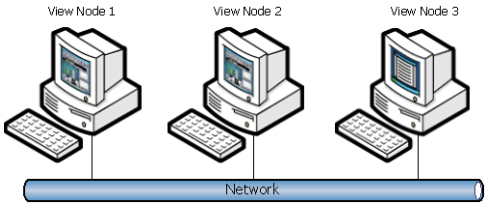


To set up a shared file

1. Map a network drive to the shared path containing the referenced files. For example, G:\Directory\Recipe.csv, where "G:\" is the mapped drive letter that refers to \\Moo\Share. You must map this same drive on every View node.
2. In scripts, reference the shared path. For example:
```
RecipeSelectRecipe("G:\Directory\Recipe.csv", "review", "RecipeName");
```

Using Local Addresses to File Data

You can use local addresses to file data when each View node has its own copy of the file. In the following figure, three View nodes are each running a copy of the same application and reference the local copy of a recipe file.



In this example, the local address is:
```
C:\Directory\Recipe.csv
```

where "C:\" is the local drive.

In scripts, reference the local path. For example:
```
RecipeSelectRecipe("C:\Directory\Recipe.csv", "review", "RecipeName");
```

This architecture is fault-tolerant. However, you must copy any file changes to all the View nodes.

Any file access should be "Read Only" and modification to the local file should not be permitted.

Access to Shared Files through UNC

You can use a Universal Naming Convention (UNC) address anywhere that you would normally enter a file path, such as for application directory entries, configuration items, and distributed alarms. If you use UNC names, you do not need to create mapped drives.

A UNC address is in the form of \\Node\Share\Path, where:

- Node is the name of the computer that contains the file share.
- Share is the logical name assigned to the shared folder on that computer.
- Path is the normal path to that file with respect to the share.

Note: If you are using SuiteLink, the node name is limited to 15 characters.

Before you can access a file through UNC, you must create a file share on the computer you want to access. For more information, see your Windows documentation.

For example, assume that you have a computer with the network name of "EngineRm" that you have shared the root drive "C:\" with the share name of "Root". To set up a UNC path to the "C:\IT\Apps\Boiler" application you must use the following UNC:

\\EngineRm\Root\IT\Apps\Boiler

If the "Boiler" directory itself was shared as "Boiler," the UNC could be shortened to:

\\EnginerRm\Boiler

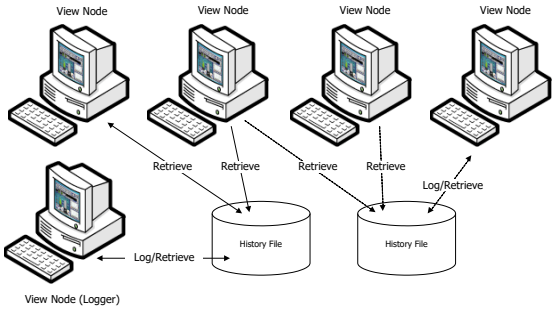
No path is required if the share is a path specified in the PATH environment variable.

Note: If you need to write to a file referred to by a UNC address, the share must be a read/write share, even on a local node. If you create a share that is password-protected, you will not be able to access the share with a UNC unless you first set up a network drive mapping. You can set up a drive mapping from the remote node by using Windows Explorer.

Logging Data in a Distributed Environment

You can use the InTouch distributed history system to retrieve historical data from any InTouch application on the network. This system also allows for remote retrieval of data from multiple history databases simultaneously. These databases are called history providers.

Only one InTouch node can log to a distributed history file. However, an unlimited number of InTouch nodes can view the contents of the file.



A remote node retrieving data from a history file may not see data for the last hour of data (based on the logger node's time). Remote trends can only view data that has been written to the logging node's disk.

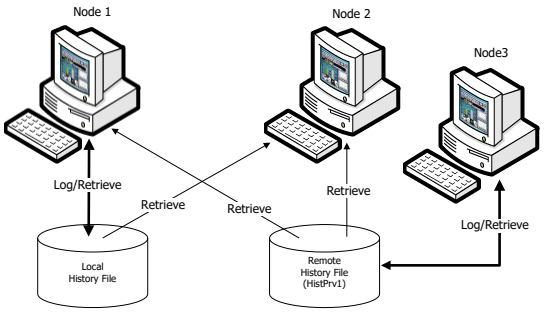
Data for each tag checked for 'Log Data' is automatically written to disk after 22 samples for that tag have been collected. If the HTUpdateToCurrentTime() function is called, data is written to disk regardless of the number of samples collected. By default, data is written to disk once an hour. You can change this interval by adding the following line to the INTOUCH.ini file:

```
ForceLogging=X;
```

Where X is minutes and can be set to any interval between 5 and 120.

Note: The NetDDE Helper service must be running when you use the distributed history system.

The following figure shows the configuration of a typical distributed history system using Network Application Development (NAD) to distribute the application.



Nodes 1 and 2 contain copies of the same InTouch application; however, the application is configured to allow only Node 1 to log to a local history file, whereas either node can retrieve from the local history file or the remote history file. Node 3 is also logging to and retrieving from the remote history file location. Node 3, the history provider, is assigned the name HistPrv1. Node 1 is both a development and run-time station, while Node 2 is just a run-time station.

Do the following major steps to create this type of application:

1. Create a history provider list. See *Configuring Remote History Providers* on page 82.
2. Create and configure a historical trend object. For more information, see Trending Tag Data in the InTouch® HMI Data Management Guide
3. Configure the application for distributed logging. See *Configuring Distributed Historical Logging* on page 84.
4. Distribute the application. See *Configuring an InTouch Application for NAD* on page 86.

You can distribute your application manually or by using NAD. When you distribute your application, the historical provider list file is distributed as part of the application.

After you have distributed your application, you can run the View nodes and retrieve both local tags and tags from a remote history provider. While the application runs on all the View nodes, only the logging node logs to the historical log file; other nodes can only read from it.

Configuring Remote History Providers

You must specify a name and network location for each remote history provider that you want to use with the InTouch HMI. You can use either a remote InTouch history provider or a remote IndustrialSQL Server history provider.

Note: A remote history provider cannot be configured for an InTouchView application. For more information about the limitations of InTouchView applications, see *InTouchView Applications* on page 15.

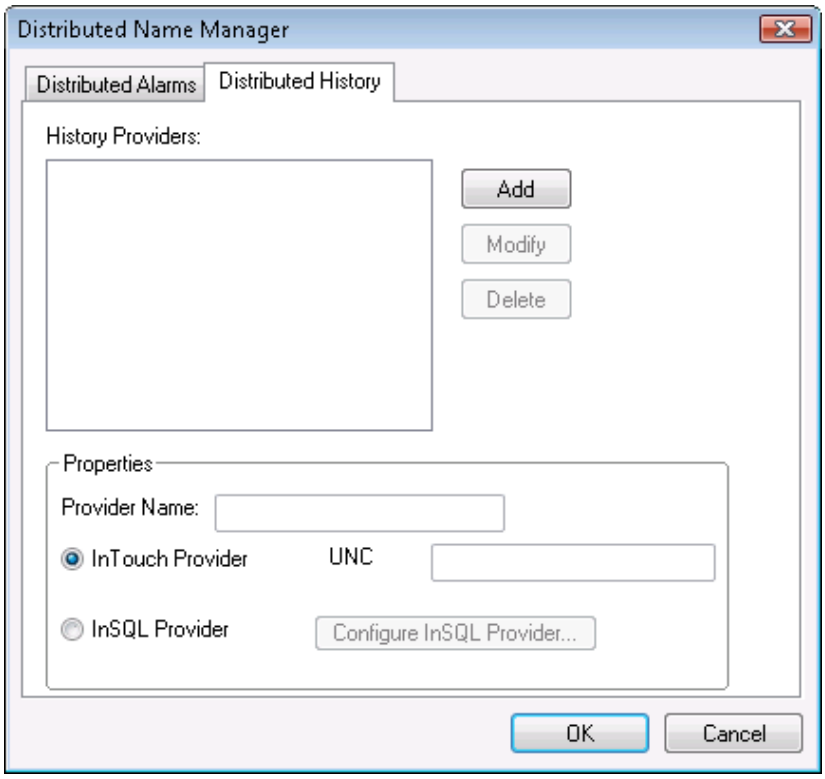
While the local InTouch application is considered a history provider, you do not need to define it for your application.

If you reference an undefined history provider in an application, WindowViewer ignores the reference and an error message is written to the Logger.

The HistData utility cannot retrieve historical information from a Historian provider.

To configure a history provider

1. On the **Special** menu, point to **Configure**, and then click **Distributed Name Manager**. The **Distributed Name Manager** dialog box appears.
2. Click the **Distributed History** tab.



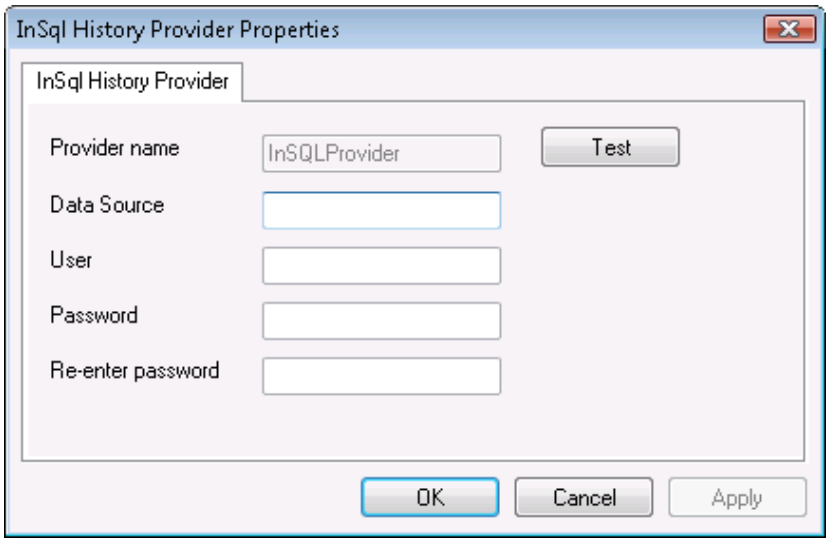
3. In the **Provider Name** box, type the name you want to use for the new historical provider.

   A provider name can be 16 alphanumeric characters or fewer.
4. To configure an InTouch history provider, do the following:
   a. Click InTouch **Provider**.
   b. In the**UNC** box, type the UNC path to the InTouch application directory and then click **Add**.

      The UNC path format is:
      ```
      \\node_name\volume_name\directory\
      ```

      If the UNC location is password-protected, you must first establish a node connection using Windows Explorer.
5. To configure a IndustrialSQL Server history provider, do the following:
   a. Click **InSQL Provider**.

b. Click **Configure InSQL Provider**. The **InSql History Provider Properties** dialog box appears.



c. In the Data Source box, type the name, up to 35 characters, of the node where the IndustrialSQL Server database resides.

d. In the User box, type the user name for the logon account. The user account must have database permissions to retrieve data.

e. In the Password and Re-enter password boxes, type the password for the logon account.

f. Click **Test** to validate the connection to the IndustrialSQL Server. When a message appears, click **OK**.

g. Click **OK** to close the **InSql History Providers Properties** dialog box.

6. Click **OK**.

Dynamically Configuring Remote History Providers

At run time, you can also dynamically configure a historical trend's remote history provider by creating a script that specifies the remote history provider tag references in the HTSetPenName() function. For example:

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

Where a 1 specifies the trend pen that plots the specified remote history provider tag.

The run-time **Historical Trend Setup** dialog box and .Pen dotfield are not supported for remote history providers.

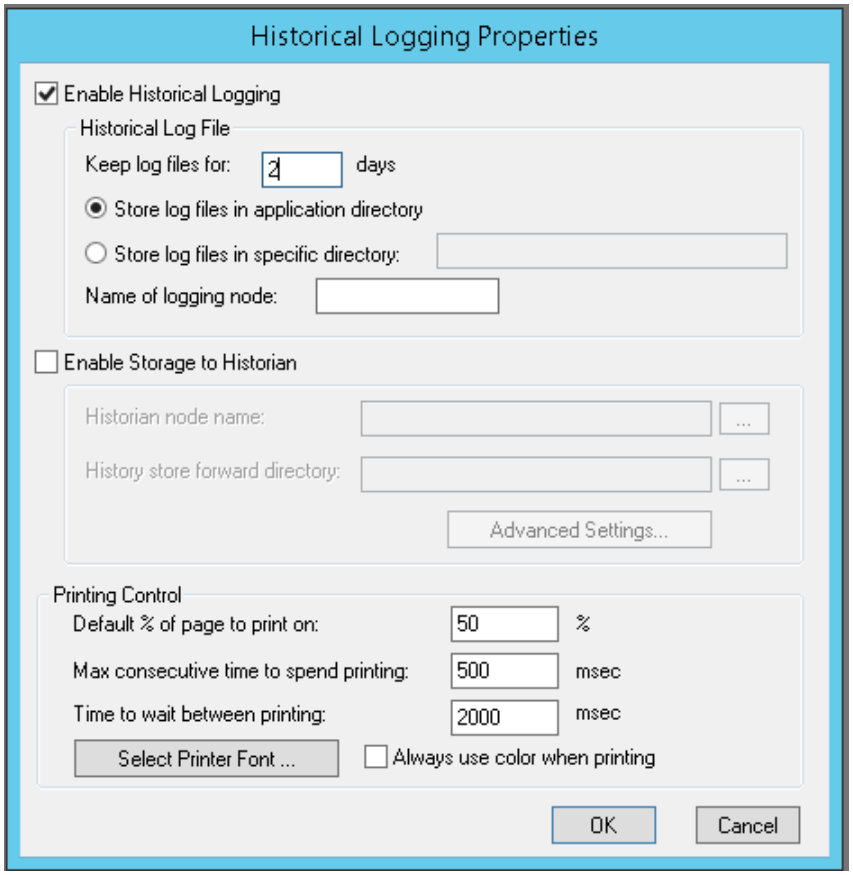
Configuring Distributed Historical Logging

Only one InTouch node can log to the history file. However, multiple InTouch nodes can view the file.

Note: Historical logging cannot be configured for an InTouchView application. For more information about the limitations of InTouchView applications, see *InTouchView Applications* on page 15.

To configure distributed historical logging

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



2. Select the **Enable Historical Logging** check box to turn on global tag logging.
3. Select **Store Log Files in Specific Directory**, and then in the input box, type the path of the location where the log files are stored.

   You must type a valid Universal Naming Convention (UNC) path. For example, `\\Node\Share\Path`

   If you are using NAD, make sure the path points to a folder other than the application folder.
4. In the **Name of Logging Node** box, type the name of the node that will be logging to the history log file.

   This setting only allows the node named here to log to the file.
5. Click **OK**.

Note: When an application with the **Enable Historical Logging** option selected is distributed to a WindowViewer node, that node checks this option to determine if it should log or not. If **Enable Historical Logging** is selected, the possible settings are:**Field equals name of Node - Logging enabled Field does not equal name of Node - Logging disabled.**

Considerations for Special Networks

If you are working on a slow network and the InTouch HMI takes a long time to start or save information, modify the `win.ini` settings on the NAD client:

```
ViewNadClearNADCopyDirectory=0
ViewNADCopyApplicationOnStartup=1
ViewNADOnApplicationChanged=3 ( or 4)
ViewNADThreadPriority=2
```

For the ViewNADOnApplicationChanged parameter, a setting of 3 corresponds to the **Load changes into** WindowViewer option on the **Node Properties** dialog box in the InTouch Application Manager. A setting of 4 corresponds to the **Prompt user to load changes into** WindowViewer option. These settings allow the application to continue to run while NAD downloads take place in parallel, on a separate execution thread.

When NAD performs an update to an application, it copies only the changed files from the master. NAD does not copy the SmartSymbol design-time dictionary files for run-time language switching.

Configuring an InTouch Application for NAD

Network Application Development or NAD is an architecture that combines the best of the client-based and server-based architectures. NAD provides automatic notification of application changes and can automatically distribute updated applications to View nodes

When configuring an application for NAD, you must specify the folder that you want WindowViewer to copy the master application to.

- If this is the development node, you can type a local folder path, such as c:\InTouch\NAD. You can also type a networked remote UNC path, such as \\node\share\path. This is convenient for file server-based networks where most file storage is kept in a central location.
- If this is a client node (run-time only), you typically use a local folder path.

We recommend that you use a local folder whenever possible to prevent network delays and failures from affecting the operation of WindowViewer.

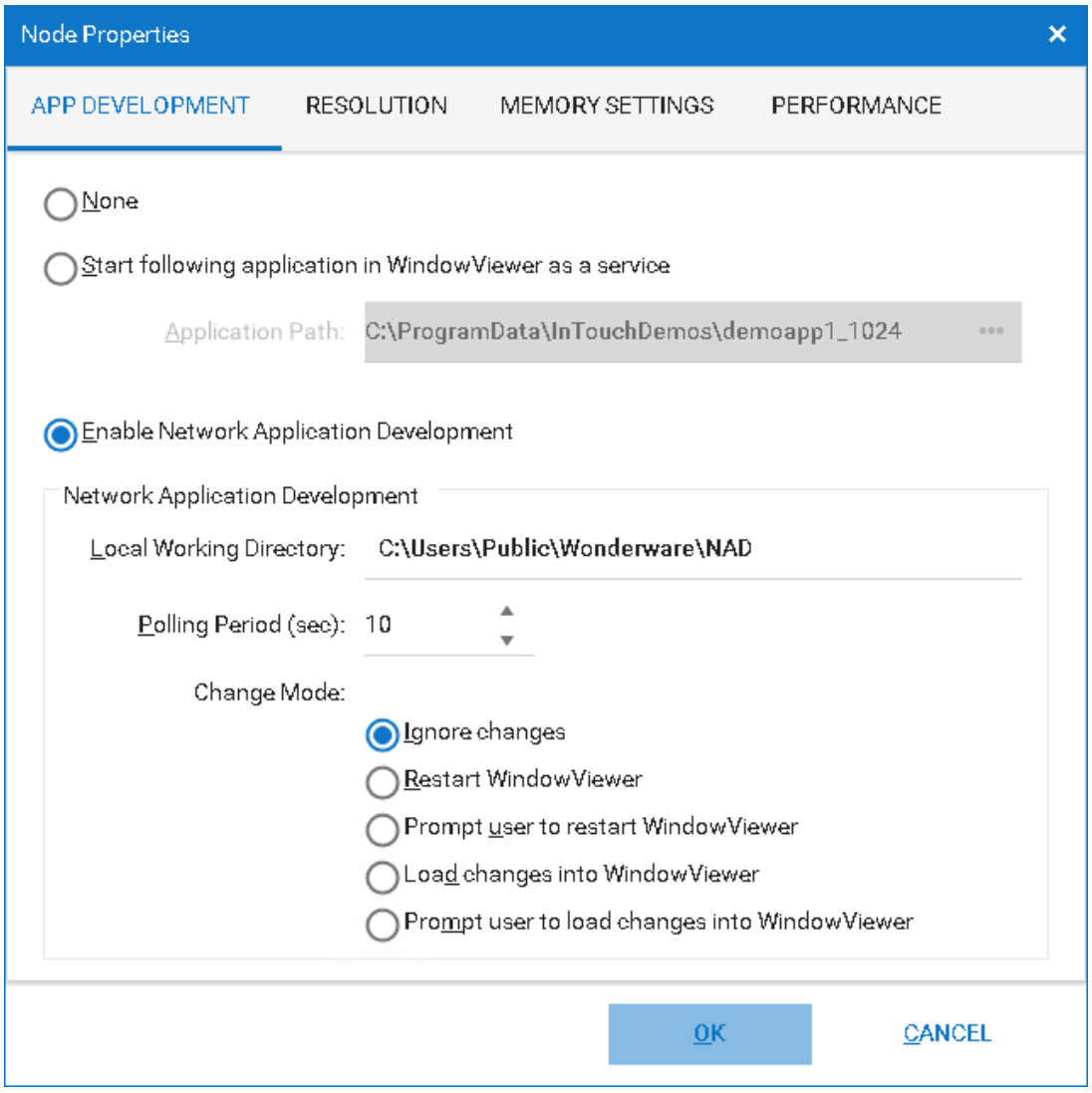
Caution: Do not use a root folder or a UNC pathname that points to a root folder. The View node recursively deletes all files and subfolders in the specified destination application folder before copying the master application directory. Therefore, never use the path of the master application folder or a UNC to the master application folder.

If you do not specify a folder, WindowViewer automatically creates a local subfolder named NAD in the folder from which WindowViewer is launched. The NAD folder should be considered a temporary folder and no other files should be saved to it except those copied by NAD itself.

To configure an application for NAD

1. Start Application Manager.

2. On the **Tools** menu, click Node Properties. The **Node Properties** dialog box appears.



3. Select the **Enable Network Application Development** radio button.
4. In the **Local working directory** box, type the path to the folder that you want WindowViewer to copy the master application.
5. In the **Polling period (sec)** box, type the interval, in seconds, at which the View node checks the development node for updates.
    - Be careful that you do not set this value too small. If WindowViewer checks for master application changes too often, it can interfere with servicing the running application.
6. In the **Change Mode** area, select the option that determines the action WindowViewer takes when the master application changes.
    - Click **Ignore changes** to have the WindowViewer node ignore any changes made on the development node.
    - Click **Restart WindowViewer** to have the WindowViewer node copy over the updated master application (if configured to do so) and then restart itself.

- Click **Prompt user to Restart WindowViewer** to show the operator a message that the application has changed. The operator can either restart WindowViewer with the application updates or continue using the current application.
- Click **Load Changes into WindowViewer** to dynamically load in WindowViewer the changes made in the development node. This may affect performance for large updates.

Note: It is recommended that you use the **Load Changes into** WindowViewer option only if the application changes are minor and few in number. Examples of minor changes include changes made within an existing window, resizing of graphic toolbar elements, adding new graphic toolbar elements, and reference substitutions. When making changes that require that WindowViewer be restarted, such as adding new tags, adding new windows, or changing the configuration—or if in doubt—use one of the Restart options instead.

- Click **Prompt user to load changes into WindowViewer** to show the operator a message that the application has changed. The message prompts the operator to load the changes.

7. Click **OK**.

Performing an Automatic NAD Update

You can start an automatic NAD update during application development.

When you run the **Notify Clients** command, a flag is set to notify all remote View nodes that the master application has changed. Clients can automatically start an update process based on the **Change Mode** option defined for each node.

The first time a standalone application (with embedded Industrial graphics) is opened on a View node, graphics may not appear and errors are logged in the SMC Logger. To avoid this, run the **Notify Clients** command from the master node and the Industrial graphics will be loaded on the View node based on the **Change Mode** option.

To perform an automatic update

1. Open the application in WindowMaker.
2. On the **Special** menu, click **Notify Clients**.
3. Click **Notify Clients Now** to notify clients immediately.
4. Click **Prompt to Notify Clients on Close**, to be reminded to notify NAD clients, when WindowMaker is closed.

Note: If the **Prompt to Notify Clients on Close** option is selected, every time WindowMaker is closed it will verify if there are any changes from the last notification. If there are any changes, a dialog box with the prompt 'Do you want to notify the NAD clients?' will appear. Click **Yes** to notify the clients, click **No** to ignore the changes.

Performing a Manual NAD Update

You can write scripts that allow operators to manually start a NAD update on the View nodes in which they work.

To manually update an application with NAD, you must set the **Change Mode** option to **Ignore Changes** in the **Node Properties** dialog box. For more information, see *Configuring an InTouch Application for NAD* on page 86.

Use the following system tags and functions in your script to perform a manual NAD update:

- *$ApplicationChanged System Tag*
- *$ApplicationVersion System Tag*
- *RestartWindowViewer() Function*
- *ReloadWindowViewer() Function*

$ApplicationChanged System Tag

Signals that the master application has changed in a Network Application Development (NAD) architecture.

Category

application

Usage

```
$ApplicationChanged
```

Remarks

This system tag changes to 1 every time the update signal is generated by selecting **Notify Clients** on the WindowMaker **Special** menu. $ApplicationChanged is reset to 0 when the application is updated. This tag can be used to generate a message that informs the operator that the master application has changed.

You can also use the $ApplicationChanged system tag in a data change script to build a node update notification script. This script can launch your own dialog boxes or stop running processes. Then, you could use the ReloadWindowViewer() function to start the update process.

Data Type

Discrete (read only)

Example

Using the following statement in the tagname box of a data change script causes the body of the script to run. The script body could show a window informing the user to restart WindowViewer for the change to take effect.

```
$ApplicationChanged
```

See Also

$ApplicationVersion

$ApplicationVersion System Tag

Contains the current version number of the application. This number changes with every change that can be saved or undone.

Category

application

Usage

```
$ApplicationVersion
```

Remarks

The value associated with the $ApplicationVersion system tag is set to the current version of the InTouch application. The version changes with every change to the application that can be saved or undone. This tag can be used to generate a message that informs the operator that the master application has changed.

Data Type

Real (read only)

Example

If used in an analog display link, this system tag shows the current version of the application that is running within WindowViewer.

```
$ApplicationVersion
```

See Also

$ApplicationChanged

RestartWindowViewer() Function

Shuts down WindowViewer, copies the updated master application (if configured to do so), and then restarts WindowViewer.

Category

system

Syntax

```
RestartWindowViewer();
```

Remarks

This function is used to update an application when the automatic update Network Application Development (NAD) functions are not used.

Use the $ApplicationChanged system tag to determine when a NAD update has occurred.

You use the **Notify Clients** command to initiate a NAD update. However, the operator may want to delay the update until a later time. You can use this function with a button action script so that the operator can restart WindowViewer when it is convenient.

You could instead use the ReloadWindowViewer() function, which updates the View node without shutting down WindowViewer.

See Also

$ApplicationChanged, ReloadWindowViewer()

ReloadWindowViewer() Function

Dynamically updates WindowViewer with the updated master NAD application without any interruption in service.

Category

system

Syntax

```
ReloadWindowViewer();
```

Allows the user control over reloading WindowViewer.

Remarks

Use this function to update an application when the automatic update Network Application Development (NAD) functions are not used.

Use the $ApplicationChanged system tag to determine when a NAD update has occurred.

You use the **Notify Clients** command to initiate a NAD update. However, the operator may want to delay the update until a later time. You can use this function with a button action script so that the operator can reload the application in WindowViewer when it is convenient.

See Also

$ApplicationChanged

Application Editing Locks

To prevent multiple developers from trying to edit an application, WindowMaker locks an application during the edit session. If you try to open a locked application, an error message is shown. The name of the node editing the application is included in the message.

If WindowMaker is abnormally shut down with an application loaded, the appedit.lok file may not be deleted. You can manually remove the lock by deleting the appedit.lok file from the application directory.

Changes to an Application During a NAD Update

When the WindowViewer node updates an application, it makes every attempt to retain the attributes (read-only, system, hidden, and so on) of the master application during the copy process.

WindowViewer also copies all files and subfolders of the master application, except for these files: *.WVW, *.DAT, *.LGH, *.IDX, *.LOG, *.LOK, *.FSM, *.STG, *.DBK, *.CBK, *.HBK, *.KBK, *.LBK, *.NBK, *.OBK, *.TBK, *.WBK, *.XBK, *.$$$, RETENTIV.X, RETENTIV.D, RETENTIV.A, RETENTIV.S, RETENTIV.H, RETENTIV.T, SSD_, WM.INI, DB.INI, LINKDEFS.INI, TBOX.INI, GROUP.DEF, and ITOCX.CFG.

Note: WindowViewer recursively deletes all files and sub folders in the destination application folder except those required for run-time language switching. This folder should be considered a temporary folder. No other files should be placed in it.

The NAD client starts an update by creating a local list of files and sub-directories that appear in the client application directory. As it looks for updates in the list of master files, the NAD client removes the corresponding client file for each master file from the local list. The remaining entries in the local list are obsolete files and sub-directories that should be deleted from the application.

All downloaded files are copied to a temporary sub-directory called NAD_Temp. Files are only copied from NAD_Temp to the application directory if all of the new and updated files are copied successfully within the re-try limits. If the NAD client has to abandon an update, the running application is not corrupted by the partial introduction of new or updated files.

If contact with the NAD master fails after all new and updated files have been downloaded, the update can still be completed by copying the updates from NAD_Temp and deleting the obsolete files. This ensures that files are not erased simply because a lost connection makes it impossible to confirm their existence on the master application.

NAD can detect whether additional changes have been made to the master application during application download. If such a situation arises, NAD abandons the download of the application. If you run the **Notify Clients** command after the latest update, NAD automatically begins downloading the latest application files at the next polling period. Otherwise, it waits until the next **Notify Clients** command issued before an application download takes place.

Scaling the Application Resolution at Run Time

You can use Dynamic Resolution Conversion (DRC) so that the distributed applications you create can run on different screen resolutions.

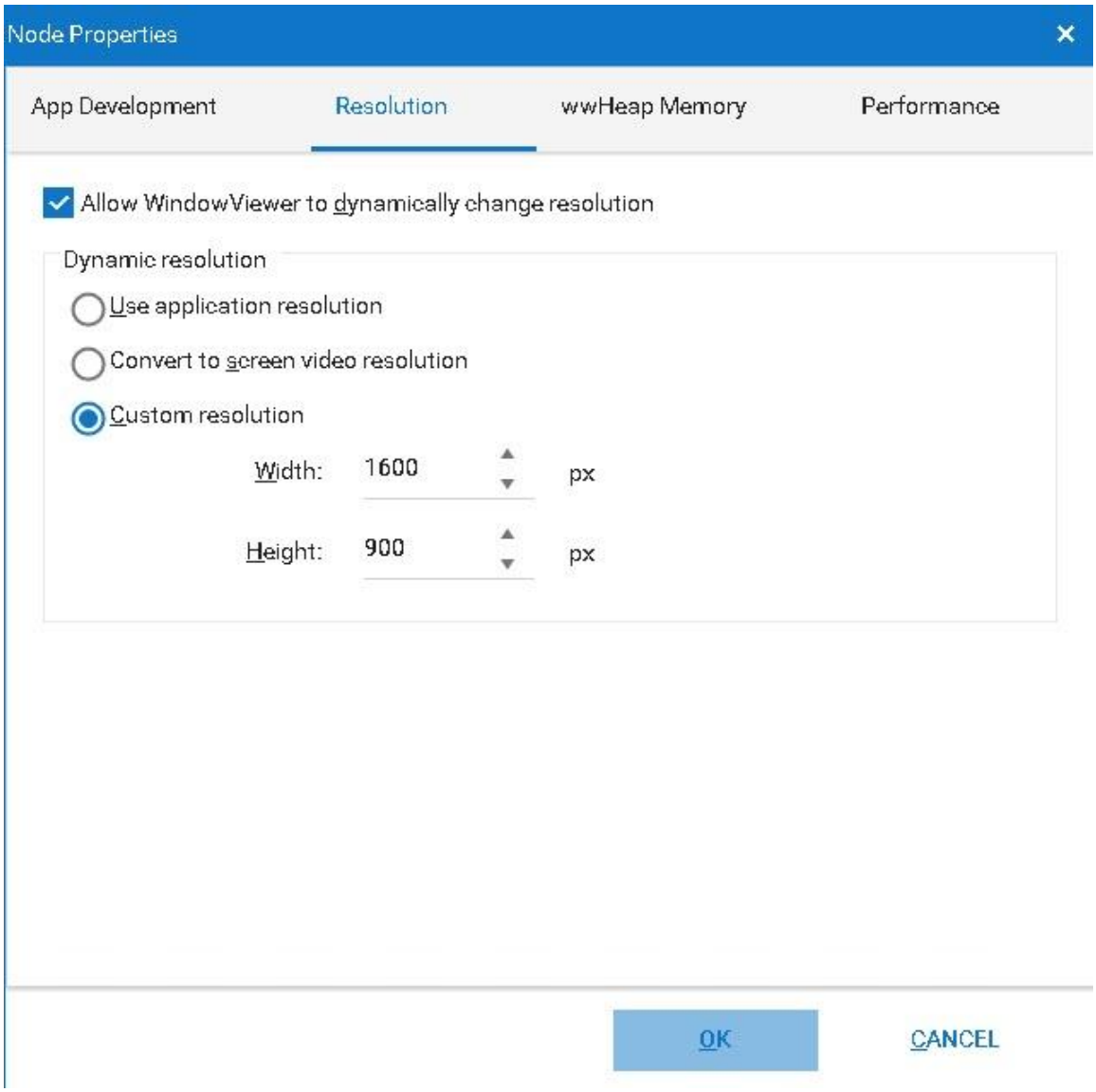
Each View node can scale the application appropriately, including scaling to a custom resolution. This scaling takes place while WindowViewer compiles the application and does not require WindowMaker. Because each View node can use a different DRC setting, each View node must have its own settings configured.

Caution: If you do not use DRC to scale the application, WindowViewer only runs the application if the node's screen resolution is identical to the screen resolution of the application development node. If the resolutions are different, WindowViewer prompts the operator to run WindowMaker to convert the application to the node's resolution. Use caution when doing this if you have set up a UNC path to the master application directory, as this will only modify the original application.

To configure an application for DRC

1. Start Application Manager.
2. On the **Tools** menu, click **Node Properties**. The Node Properties dialog box appears.

3. Click the **Resolution** tab.



4. Select the **Allow WindowViewer to dynamically change resolution** check box if you want WindowViewer to locally scale the master application.
5. In the **Dynamic Resolution** area, select one of the following:
   - Select **Use application resolution** if you want WindowViewer to run the application at the resolution it was developed for and ignore the node's resolution. For example, if the application was developed at 800x600 and the node's resolution is 1024x768, WindowViewer does not dynamically scale the application. Instead, the application resolution remains at 800x600.
   - Select **Convert to screen video resolution** if you want WindowViewer to run the application at the node's resolution and ignore the resolution the application was developed at. For example, if the node is running at 800x600 and the application was developed at 1280x1024, WindowViewer dynamically scales the application to fit the node's 800x600 resolution.

- If the target resolution is different from the screen resolution when the application was created, then WindowViewer will scale to the current screen resolution from the original application resolution instead. The original application resolution is the screen resolution when the application was created regardless of the target resolution settings. For example, if the application was developed at 1920x1080 with a target resolution of 1280x1024 and the view node is running the application at resolution of 800x600, WindowViewer will dynamically scale the application to use the original application resolution of 1920x1080. For more information, see *Original Application Resolution* on page 230.

- Select **Custom resolution** if you want WindowViewer to run the application at a specific resolution you specify in the **Width (X)** and **Height (Y)** (must be integer values) boxes. The application's resolution and the node's resolution are both ignored. For example, if **Width (X)** and **Height (Y)** are set to 512 and 384, respectively, the application is dynamically scaled to fit in a 512x384-pixel area on the node's screen.
  - If the target resolution is different from the screen resolution when the application was created, then WindowViewer will scale to the current screen resolution from the original application resolution instead. The original application resolution is the screen resolution when the application was created regardless of the target resolution settings.

6. Click **OK**.

Locking the Application Resolution

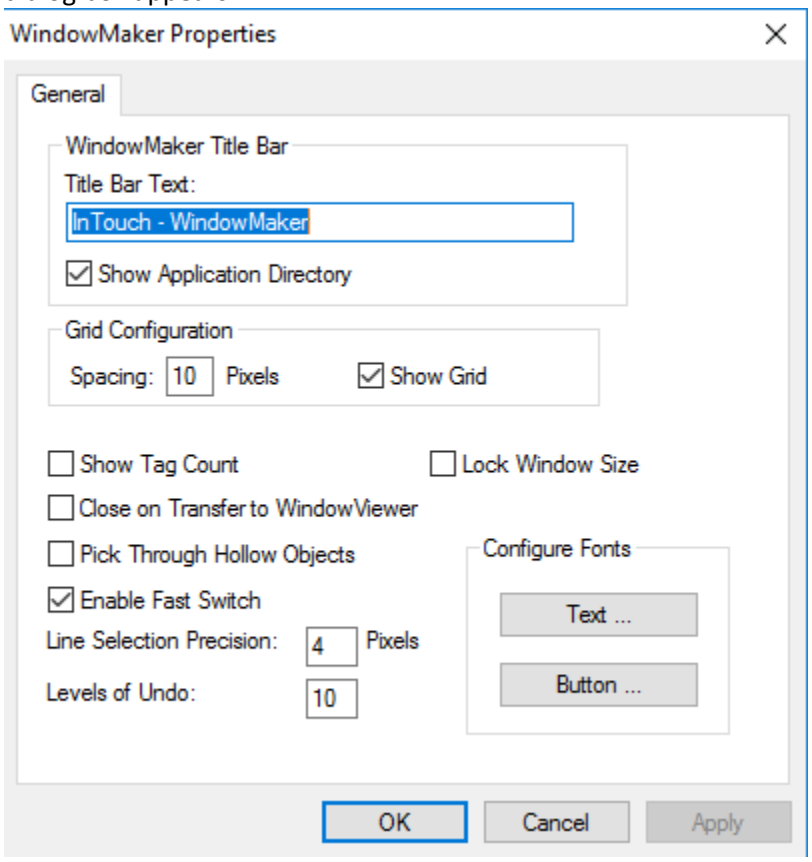
You can configure the WindowMaker properties to lock the size of InTouch application windows. This allows you to convert applications to a different resolution without scaling the windows and graphics.

If you select this option, the next time you open an application in a computer with a different resolution, the system prompts you to specify whether you want to convert the application to the new resolution without scaling the windows and graphic.

You can lock the application resolution from inside WindowMaker or from the Application Manager.

To lock the application resolution from WindowMaker

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

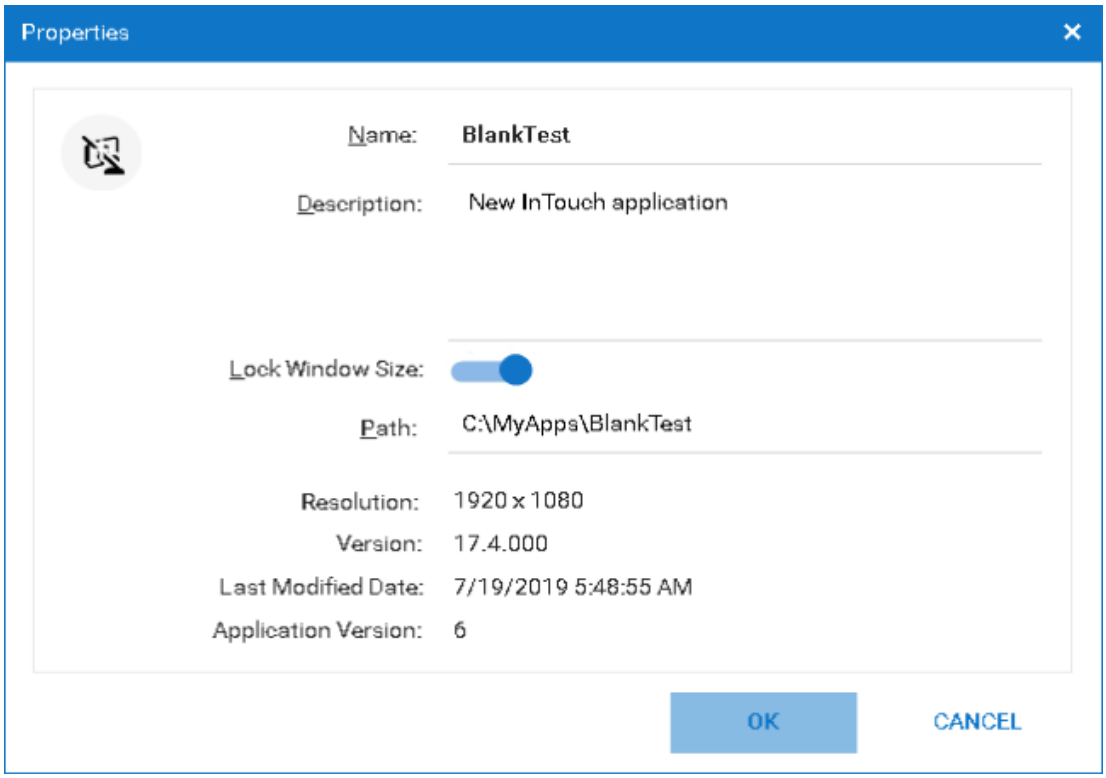


3. Select the **Lock Window Size** check box. By default, the check box is not selected.
4. Click **OK**.

To lock the application resolution from Application Manager

1. Open Application Manager.Click to select the application you want to configure.
2. Click **File** on the menu bar, then click **Properties**. The **Properties** dialog box appears.

3. Select the **Lock Window Size** switch. By default, the check box is not selected.



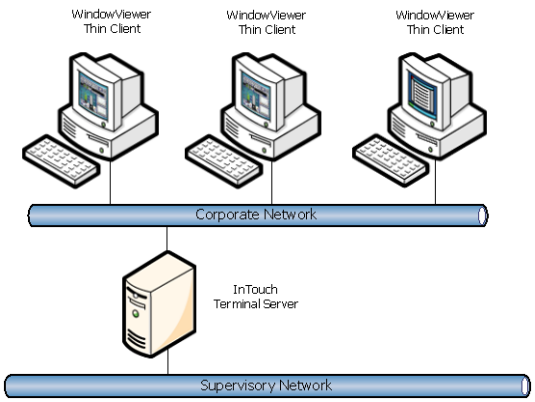
4. Click **OK**.

Chapter 6

Deploying and Working with Terminal Services and Remote Desktop Services

Terminal Services Overview

Terminal Services is a configurable service included in the Microsoft Windows Server operating systems that runs Windows-based applications centrally from a server. In Terminal Services, client computers access the server node, where multiple instances of InTouch software applications run simultaneously.



The Terminal Services environment has three main parts:

- **Terminal Services Server**. The server manages the computing resources for each client session and provides client users with their own unique environment. The server receives and processes all keystrokes and mouse actions performed at the remote client and directs all display output for both the operating system and applications to the appropriate client. All Terminal Services application processing occurs on the server.
- **Remote Desktop Protocol (RDP).** A Remote Desktop Protocol (RDP) client application passes the input data, such as keystrokes and mouse movements, to the server.
- **Client**. The Terminal Services client performs no local application processing; it just shows the application output. You access Terminal Services from a client by running the **Terminal Services Client** command on the Windows Program menu. When you connect to the Terminal server, the client environment looks the same as the Windows server. The fact that the application is not running locally is completely transparent.

For more information about Terminal Services, including features and benefits, see your Microsoft documentation.

Planning Considerations for Terminal Server Applications

Important: We recommend that you install applications on a test server before you deploy them in your production environment.

Before you install Terminal Services:

- Identify the client applications (for example, InTouch) that you need to install on the server.
- Identify the hardware requirements for your clients.
- Determine the server configuration required to support clients.
- Identify the licenses required for Terminal Services as well as other applications that you will be running.
- Understand how some aspects of InTouch applications run under Terminal Services, such as alarms, security, I/O, and scripts.

Deploying InTouch Applications in a Terminal Services Environment

When deploying InTouch applications in a Terminal Services environment, a separate InTouch application should be deployed for each node.

Alarms in a Terminal Services Environment

By using the Distributed Alarm System with Terminal Services for InTouch, alarm clients running on different terminal sessions can select what alarm to show and how to present it.

Alarm Providers identify themselves by a name that uniquely identifies their application, and the instance of their application. This information is made available to the Distributed Alarm System when the Alarm Provider or the Alarm Consumer registers with the Distributed Alarm System.

The node on which an Alarm Provider is running is identified by a name that uniquely identifies the computer node in the system. This information is made available to the Distributed Alarm System when an instance of it starts up on the computer node.

When an alarm event is logged, the node and complete Alarm Provider name identify the source of the alarm.

When an alarm is acknowledged in a Terminal Services environment, the Operator Node that gets recorded is the name of the client computer running the Terminal Services session used by the operator. If the node name of the computer cannot be retrieved, its IP address is used instead.

Note: Alarm Providers are not supported on Terminal sessions. They are only supported on the Terminal Console.

Security in a Terminal Services Environment

Use application security to secure your InTouch application, IndustrialSQL Server, and other sensitive information systems.

- Use the $Operator system tag to secure your application. You can then control operator access to specific functions by linking those functions to internal tags.

  For more information about using the $Operator system tag, see *About Securing InTouch* on page 174.

- Replace the GetNodeName() function with the newer TseGetClientId() function to identify the client computer. When using Terminal Services, the GetNodeName() function returns the name of the terminal server, not the name of the client computer.

Use security auditing to monitor intrusion attempts. If you suspect that your system is under any sort of attack, then you can enable logging for an array of auditable events. By default, security logging/auditing is disabled because it usually requires excessive processing resources.

Caution: Security auditing requires significant resources. Enable auditing when you evaluate your pilot server to accurately estimate your InTouch application hardware requirements.

I/O in a Terminal Services Environment

The InTouch HMI cannot start I/O Servers in a Terminal Services environment. Depending on the sequence that view sessions start, you may need to use the IOReinitialize() function. All servers (I/O devices or view applications) must be running before starting an application that reads values from these servers.

To avoid receiving an "initializing I/O" error message when WindowViewer starts, clear the **Start Local Servers** check box on the **General** tab on the WindowViewer **Properties** dialog box.

Script Execution in a Terminal Services Environment

Because all applications running in Terminal Services use a single timing reference (server clock), scripts may not run during periods of excessive CPU loading. Abnormal CPU loading can be caused by excessive video processing or when several applications have the same script triggers defined (such as an End-of-Shift event). It is possible, therefore, that if the server is busy processing scripts from many clients, it may not start a script on another client during the interval when the timer would normally start the script. This can prevent the script from running on the client.

To ensure scripts run correctly, combine scripts with common triggers and move them to a single application, such as a tag server. This is one of the primary reasons for pilot deployment. Pilot deployment gives you an opportunity to conduct stress testing to determine if your hardware selection is adequate.

Logging on to a Terminal Session Properly to Run InTouch

Each session must be logged on with a unique account. This can be done manually or Terminal Services can be configured to enforce unique logons.

Note: Running with the same logon account on multiple sessions can cause corruption and other unexpected results.

Alarm Query Syntax in a Terminal Service Environment

The alarm query syntax for a session's alarms is:

```
\\ServerNodename\InTouch!$System
```

The alarm query syntax for console alarms includes a colon (:) after the node name; for example:

```
\\ServerNodename:\InTouch!$system
```

Miscellaneous Limitations in a Terminal Services Environment

The following table describes the limitations and suggested solutions to run applications on a terminal server.

Feature	Supported?	Comment
WindowViewer	Yes	WindowViewer is not supported running as a service under Terminal Services.
DDE to an I/O Device or MS Office (for example, Excel)	No	Use a tag server (console or separate computer). This includes DDE QuickScripts: WWExecute(), WWpoke() and WWRequest()
DDE from MS Office (for example, Hot-link configured in Excel)	Yes	Excel and the InTouch HMI must be running in the same session.
Historical Trending	Yes	Use a tag server or NAD to log values. Multiple sessions may read the same historical files, but only a console can write to historical files.
InTouch Alarm Logger	Yes	--
MEM OLE Automation	Yes	--
Printing Alarms	No	--
Retentive tags	Yes	Must use NAD.
SQL Access (ODBC)	Yes	Database should be on a separate computer.
SuiteLink to an I/O Device or another InTouch application.	Yes	When communicating to another view session, include the Terminal Server node name and append the IP address of the desired session to the application name. For example, view10.103.25.6. I/O Servers are not supported in client sessions.

Retrieving Information About the InTouch Client Session Using Scripts

You can use the following InTouch QuickScript functions for Terminal Services.

- *TseGetClientId() Function* on page 101
- *TseGetClientNodeName() Function* on page 101
- *TseQueryRunningOnConsole() Function* on page 101

- *TseQueryRunningOnClient() Function* on page 101

TseGetClientId() Function

Returns a string version of the client ID (the TCP/IP address of the client) if the View application is running on a Terminal Server client. This client ID is used internally to generate SuiteLink server names and logger file names. Otherwise, the TseGetClientId() function returns an empty string.

Syntax

```
MessageResult=TseGetClientId();
```

Example

The client IP address 10.103.202.1 is saved to the MsgTag tag.

```
MsgTag=TseGetClientID();
```

TseGetClientNodeName() Function

Returns the client node name if the View application is running on a Terminal Server client assigned a name that can be identified by Windows. Otherwise, the TseGetClientNodeName() function returns an empty string.

Syntax

```
MessageResult=TseGetClientNodeName();
```

Example

The client node name is returned as the value assigned to the MsgTag tag.

```
MsgTag=TseGetClientNodeName();
```

TseQueryRunningOnConsole() Function

The TseQueryRunningOnConsole() function can be run from a script to indicate whether the View application is running on a Terminal Services console.

Syntax

```
Result=TseQueryRunningOnConsole();
```

Return Value

Returns a non-zero integer value if the View application is running on a Terminal Services console. Otherwise, the TseQueryRunningOnConsole() function returns a zero.

Example

IntTag is set to 1 if WindowViewer is running on a Terminal Services console.

```
IntTag=TseQueryRunningOnConsole();
```

TseQueryRunningOnClient() Function

Returns a non-zero integer value if the View application is running on a Terminal Services client. Otherwise, it returns a zero.

Syntax

```
Result=TseQueryRunningOnClient();
```

Return Value

Returns 0 if View is not running on a Terminal Services client.

Example

IntTag is set to 1 if WindowViewer is running on a Terminal Services client.

```
IntTag=TseQueryRunningOnClient;
```

Remote Desktop Services Overview

Remote Desktop Services, formerly Terminal Services, is a server role in Windows Server® 2008 R2 and later versions that provides technologies that enable users to access Windows-based programs that are installed on a Remote Desktop Session Host (RD Session Host) server, or to access the full Windows desktop. With Remote Desktop Services, users can access an RD Session Host server from within a corporate network or from the Internet.

When a user accesses a program on an RD Session Host server, the program runs on the server. Each user sees only their individual session. The session is managed transparently by the server operating system and is independent of any other client session. Additionally, you can configure Remote Desktop Services to use Hyper-V™ to either assign virtual machines to users or have Remote Desktop Services dynamically assign an available virtual machine to a user upon connection.

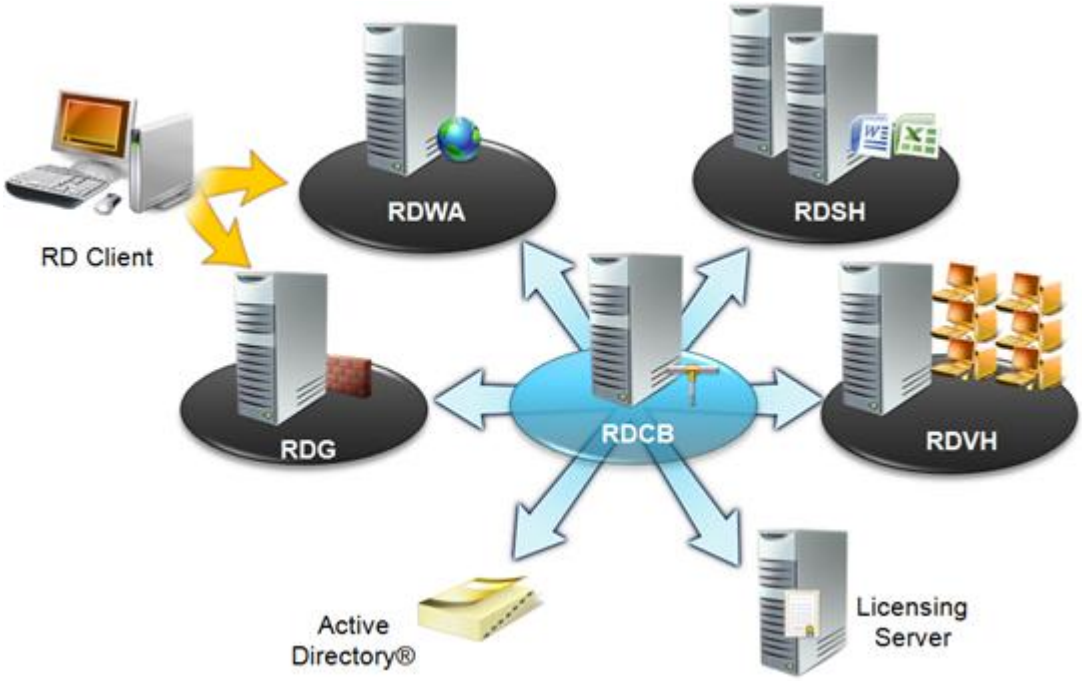
For more information about Remote Desktop Services, see the Remote Desktop Services page on the Windows Server 2008 R2 TechCenter (http://go.microsoft.com/fwlink/?LinkId=138055).

Remote Desktop Services Role Services

Remote Desktop Services is a server role that consists of several role services. In Windows Server 2008 R2 and later versions, Remote Desktop Services consists of the following role services:

- **RD Session Host**: Remote Desktop Session Host (RD Session Host), formerly Terminal Server, enables a server to host Windows-based programs or the full Windows desktop. Users can connect to an RD Session Host server to run programs, to save files, and to use network resources on that server.
- **RD Web Access**: Remote Desktop Web Access (RD Web Access), formerly TS Web Access, enables users to access RemoteApp and Desktop Connection through the Start menu on a computer that is running Windows 7 or through a Web browser. RemoteApp and Desktop Connection provides a customized view of RemoteApp programs and virtual desktops to users.
- **RD Licensing**: Remote Desktop Licensing (RD Licensing), formerly TS Licensing, manages the Remote Desktop Services client access licenses (RDS CALs) that are required for each device or user to connect to an RD Session Host server. You use RD Licensing to install, issue, and track the availability of RDS CALs on a Remote Desktop license server.
- **RD Gateway**: Remote Desktop Gateway (RD Gateway), formerly TS Gateway, enables authorized remote users to connect to resources on an internal corporate network, from any Internet-connected device.
- **RD Connection Broker**: Remote Desktop Connection Broker (RD Connection Broker), formerly TS Session Broker, supports session load balancing and session reconnection in a load-balanced RD Session Host server farm. RD Connection Broker is also used to provide users access to RemoteApp programs and virtual desktops through RemoteApp and Desktop Connection.

- **RD Virtualization Host**: Remote Desktop Virtualization Host (RD Virtualization Host) integrates with Hyper-V to host virtual machines and provide them to users as virtual desktops. You can assign a unique virtual desktop to each user in your organization, or provide them shared access to a pool of virtual desktops.



Securing your Remote Desktop Services (RDS) Connections

To safeguard against attacks, we recommend the following security practices:

1. Use strong passwords

   Use a strong password on all accounts with access to Remote Desktop.

2. Update your software

   Make sure you are running the latest versions of both the client and server software by enabling and auditing automatic Microsoft Updates.

3. Set an account lockout policy

   By setting your computer to lock an account for a period of time after a number of incorrect guesses, you will help prevent "brute-force" attack.

4. Use Two-factor authentication

   RD Gateways support smartcard two-factor authentication.

5. Change the listening port for Remote Desktop

   Prevents network attacks and worms that attempt to access the default Remote Desktop port (TCP 3389).

6. Use RD Gateways

RD Gateway restricts access to Remote Desktop ports while supporting remote connections through a single "Gateway" server. When using an RD Gateway server, all Remote Desktop services on your desktop and workstations are routed through the RD Gateway. The RD Gateway server listens for Remote Desktop requests over HTTPS (port 443), and connects the client to the Remote Desktop service on the target machine. Refer to the steps here: http://technet.microsoft.com/en-us/library/cc770601.aspx

7. Configure Network Level Authentication for Remote Desktop Services Connections

   Network Level Authentication requires that the user be authenticated to the RD Session Host server before a session is created. Network Level Authentication increasing availability of the RD Session Host server (reduces the risk of denial-of-service attacks of the RD Session Host server). https://technet.microsoft.com/en-us/library/hh831778.aspx

8. Configure Server Authentication and Encryption Levels

   By default, Terminal Services sessions use native Remote Desktop Protocol (RDP) encryption. However, RDP does not provide authentication to verify the identity of a terminal server. You can enhance the security of Terminal Services sessions by using Transport Layer Security (TLS) 1.0 for server authentication and to encrypt terminal server communications. The RDS and the client computer must be correctly configured for TLS to provide enhanced security. By default, RDS connections between the client and server are encrypted at the highest level of security available (128-bit), ensuring integrity and confidentiality of the data transmitted.

Windows Server 2016 Remote Desktop Services Best Practices

For the Windows Server 2016 environment you can implement the below best practices for Remote Desktop Services:

- Use Multi-Factor Authentication

  Leverage the power of Active Directory with Multi-Factor Authentication to enforce high security protection. Refer to the Microsoft documentation here: https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-plan-mfa

- Secure data storage with User Profile Disks (UPDs)

  User Profile Disks (UPDs) allow user data, customizations, and application settings to follow a user within a single collection. A UPD is a per-user, per-collection VHD file saved in a central share that is mounted to a user's session when they sign in - the UPD is treated as a local drive for the duration of that session. Refer to the Microsoft documentation here: https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-plan-secure-data-storage

Chapter 7

Managing InTouch Services

About Managing InTouch Services

A service is a Windows process that performs a specific unattended background system function without a user interface or a required user logon.

The following startup options are available for Windows services:

- **Automatic**. When Windows restarts, the service automatically starts without any user intervention.
- **Manual**. A user or an application process must explicitly start the service.
- **Disable**. The service is prevented from starting. This is useful for troubleshooting.

Note: The parameters option in the InTouch WindowViewer service is not supported.

Services are started without compromising the Windows security system.

The InTouch HMI includes the following Windows services:

- Alarm DB Logger
- Alarm DB Purge/Archive
- NetDDE Helper
- SuiteLink
- WindowViewer

Running WindowViewer as a Service

If you configure WindowViewer to run as a Windows service, WindowViewer automatically starts when the computer on which the application is installed starts. The WindowViewer service runs in the background. If the WindowViewer service is running you cannot start another instance of WindowViewer.

Running WindowViewer as a service provides the following benefits:

- Most disaster recovery plans require that essential computer systems start immediately after electrical power is restored. Microsoft Windows Servers can restart automatically after power is restored. When WindowViewer runs as a service, your plant automation system can begin running immediately. The last InTouch application that was opened in WindowViewer automatically starts when the computer restarts.

- WindowViewer continues to log historical data, gather alarm information, process scripts, act as an I/O Server, and write values as an I/O client, even as different operators log on and off.

Note: A logged on user must have proper access to the network location if a network application is used to run as a service or a network path is used as a historical logging folder.

If WindowViewer is already running as a service and you attempt to start it again from a shortcut icon or by clicking WindowViewer on the Windows **Start** menu, a message is logged in the ArchestrA Logger. The message describes the restrictions to starting WindowViewer when it has been configured to run as a service.

If WindowViewer is already running as a service and you attempt to launch Application Manager or WindowMaker, a warning message will be logged in the ArchestrA Logger.The message explains that Application Manager and WindowMaker cannot open when WindowViewer is running as a service.

Important: When running WindowViewer as a service, the user account privileges have been set to non-interactive to reduce the potential security exposure of running a service with administrator privileges.

Configuring WindowViewer to Start as a Service

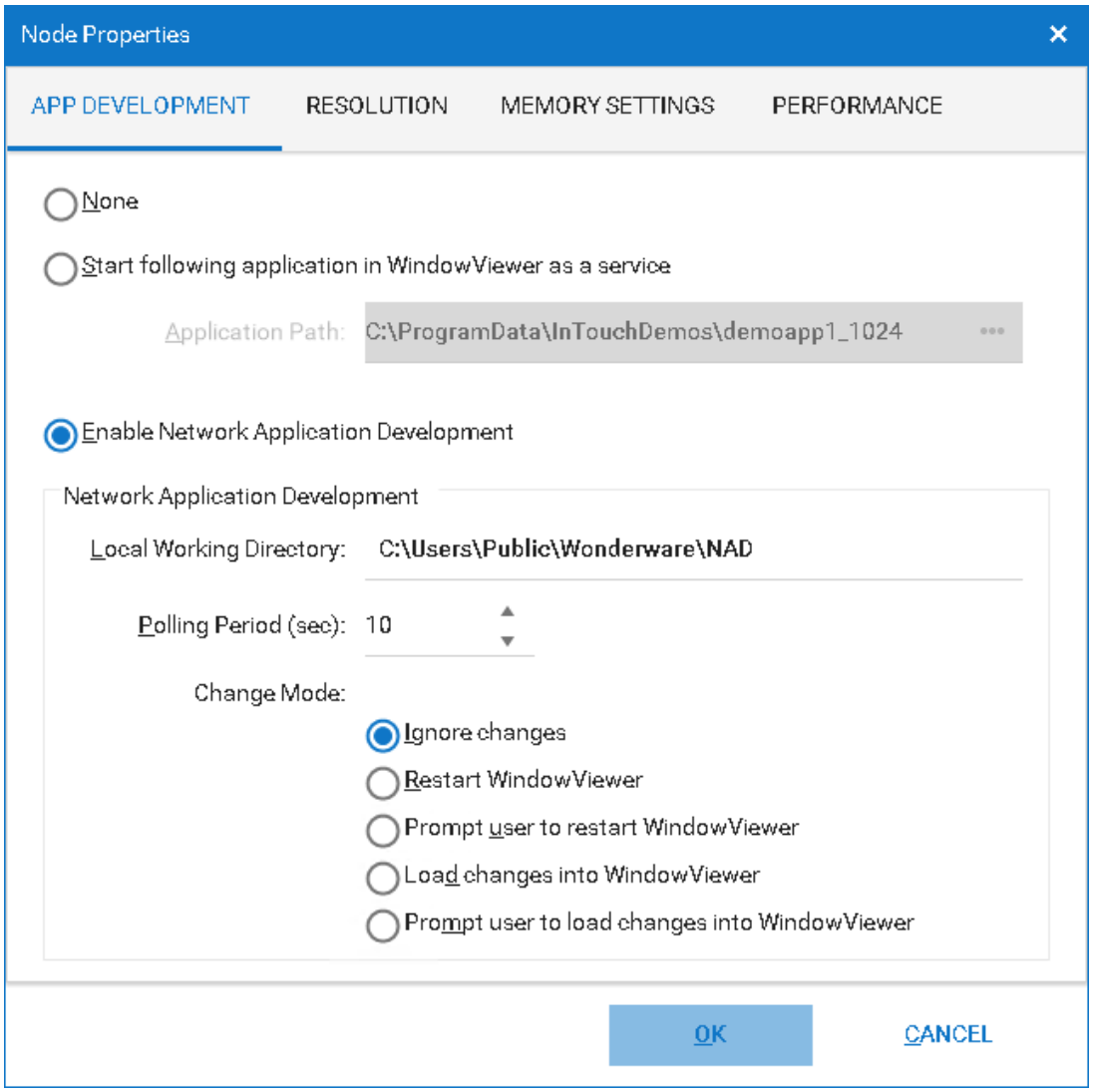
Running WindowViewer as a Windows service provides continuous operation after the operator logs off and automatic start up at system boot time without operator intervention. This allows unmanned station start up of WindowViewer without compromising operating system security.

When WindowViewer is configured to start as a service, an InTouch application must also be specified to run in WindowViewer as a service. You can specify the application directory in the **Node Properties** dialog box or manually enter it into the WIN.INI file.

To configure WindowViewer to start as a service

1. Launch the InTouch Application Manager appears.

2. On the **Tools** menu, click **Node Properties**. The **Node Properties** dialog box appears.



3. Select the **Start** WindowViewer **as a service** check box to configure WindowViewer to automatically run as a service.

   The **Application to run as a service group box** will become enabled.

4. Click the ellipsis button to prompt a file explorer and navigate to your InTouch application.

   The application directory will populate in the group box.

5. Click **OK**.

6. Click the WindowViewer icon in the Application Manager toolbar.

   WindowViewer will now run as a service for the specified InTouch application.

Note: You can also fast switch from WindowMaker to WindowViewer to start the WindowViewer service for the InTouch application if you have configured the Node Properties as described in the above steps. You can do this in place of starting WindowViewer from the Application Manager.

Editing WIN.INI to Run Application as Service in WindowViewer

If Start WindowViewer as a service is enabled in the **Node Properties**, you can manually enter the application directory into the WIN.INI file. If you update the WIN.INI file before selecting the application in Application Manager, WindowViewer runs as a service for the application once selected.

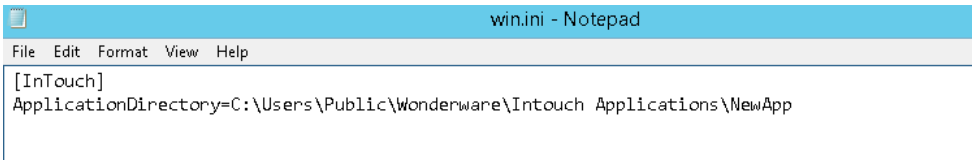
You can also update the WIN.INI with the application open in WindowMaker. If you then fast-switch to run time, WindowViewer runs as a service for the application.

Note: The above functionality is not supported for Managed InTouch applications. If you attempt to fast-switch a Managed application from WindowMaker to run as a service in WindowViewer, a warning message will be logged in the ArchestrA Logger.

The WIN.INI is located here:

```
C:\ProgramData\Wonderware\InTouch\Service\win.ini
```

Enter the directory of the application you want to run as a service, as in the example below:



Manually Starting a Service

You can manually start the InTouch WindowViewer service using the Windows Control Panel.

WindowViewer does not appear in the Services Control Panel unless you configured it to start as a service. For more information, see *Configuring WindowViewer to Start as a Service* on page 106.

To start the WindowViewer service using Control Panel

1. Start the Control Panel.
2. Double-click **Administrative Tools** and then double-click **Services**. The **Services** dialog box appears.
3. In the details pane, right-click WindowViewer service and then click **Start**.

Important: The command prompt cannot be used to start WindowViewer as a service.

Stopping a Service

You can manually stop the WindowViewer service using the Control Panel.

To stop the WindowViewer service using the Control Panel

1. Start the Control Panel.
2. Double-click **Administrative Tools** and then double-click **Services**. The **Services** dialog box appears.
3. In the details pane, right-click WindowViewer and then click **Stop**.

Configuring the User Account for InTouch Services

By default, Windows services run using the local system account. InTouch services require a user account with administrative privileges, which may not be provided by the local system account.

When you install the InTouch HMI, you specify an administrative account that all ArchestrA services run under, if the account was not created already. This account is considered the master ArchestrA account. The InTouch services use the master ArchestrA account to automatically start up.

Note: The master account is also called the impersonation account. An impersonation account is the user or group account that provides access to the restricted resource "area" of your site or server.

If you want to change the master account, use the ArchestrA Change Network Account Utility.

Caution: Changing the master account affects all ArchestrA services, not just InTouch services.

To change the ArchestrA master account

1. On the **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Change Network Account**. The **Change Network Account** dialog box appears.
2. Change the user account. For more information, see the Change Network Account documentation.
3. Click **OK**.

Troubleshooting InTouch Services

If a service depends on other services starting before it can start, Windows verifies if the prerequisite services are running before starting the service.

Depending on your requirements for running WindowViewer, be aware of the following dependencies:

- The NetDDE Helper service must be running if you plan to use Distributed Alarming or Distributed History or if you intend to access network DDE data.

  The NetDDE Helper service also depends on both the Network DDE and Network DDE DSDM services being installed and configured for either Manual or Automatic startup. During installation, the NetDDE Helper service is configured for Manual startup. WindowViewer automatically starts this service when the computer starts.

- If you need WindowViewer to act as a SuiteLink server or client, then the SuiteLink service must be running.

  The SuiteLink service also requires that Microsoft TCP/IP be installed.

- If you want to store any messages or errors while WindowViewer is running, you must make sure that the ArchestrA Logger service is installed.

  Both the SuiteLink and ArchestrA Logger services should be installed and configured to run in automatic startup.

Viewing Error Messages for Services

Use the Windows Event Viewer to troubleshoot error messages related to services. For example, you may see the error "One or more services failed to start …" The Windows Event Viewer lists informational messages, warnings, or errors that occurred while starting Windows services. For more information about the Event Viewer, see your Microsoft documentation.

You can see any warning or error messages that resulted from an InTouch service failing to start. If the Event Viewer indicates that the WindowViewer service failed to start, the most likely cause is a dependency on a prerequisite service that is not running.

Troubleshooting Problems with the Services User Account

If InTouch services fail to install or start after you install the InTouch HMI, you could have a problem with the user account that they run under.

To troubleshoot services user account problems

1. Open the Windows User Manager window and create a new master user account.

   This user account must have administrative privileges on the local computer to start an InTouch component as a service. If you do not see your computer's node name in the domain list, then manually type in the node name.

   For more information, see *Configuring the User Account for InTouch Services* on page 109.

2. Verify that your computer's node name is no longer than 14 characters. If the node name contains underscore characters (_) or dashes (-), remove them.
3. During installation when you are prompted to enter the domain name, type in the node name of your computer, not the domain name. Then, type the user name that was created in step 1 and your password.
4. If you already installed the InTouch HMI, you can still specify the domain name, user name, and password by running the ArchestrA Change Network Account Utility.
5. Reboot your computer.
6. Log onto your network domain with any valid user account. Even if your domain goes down, it does not affect your InTouch application that is running on the local computer.

Deactivating Advised I/O Items

When you start up the Windows operating system, the services that are configured to automatically start will start in the "background" with no visible user interface appearing on the desktop. The services in this situation are running in the system context. When an operator logs on the system, any services that are running in the system context that have an associated user interface automatically appear on the desktop. In this situation, the services are now running in the desktop context.

If you configure the WindowViewer service to automatically start, the service runs in the system context when the operating system starts. Then, when a user logs on, the WindowViewer service continues to run but in the desktop context, and the WindowViewer user interface automatically appears.

If you have InTouch Access Names defined with the **Advise only active items** option turned on, and have I/O tags that are active only in certain InTouch application windows (the tags are not used anywhere else in the application), it is possible to "deactivate" those tags. For example, if WindowViewer is running as a service, and you close an application window using a script, the window automatically is unloaded from memory, thus terminating the link to those tags.

Registry Keys for the InTouch Services

The InTouch services are listed as keys in the Windows registry:

SuiteLink:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SLS
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\slssvc
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SuiteLink

NetDDE Helper:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWNetDDE

WindowViewer:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VIEW

Chapter 8

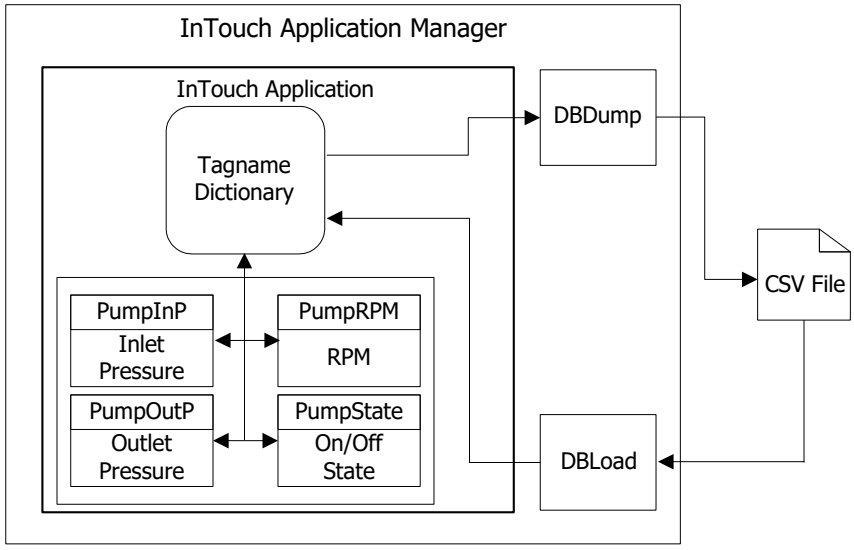
Exporting and Importing InTouch Components

About Exporting and Importing InTouch Components

You can build InTouch applications more quickly by importing or exporting some or all of the components of an existing application. You can import tag definitions, windows, scripts, application style libraries, Industrial graphics, client controls, localization strings, HTML5 widgets, and script function libraries from your existing application to a new application. Tag definitions are imported and exported from the Application Manager, other components are imported and exported via WindowMaker.

Exporting Tag Definitions

The figure below shows the steps to export and import tag definitions between an interim export file and an application's Tagname Dictionary.

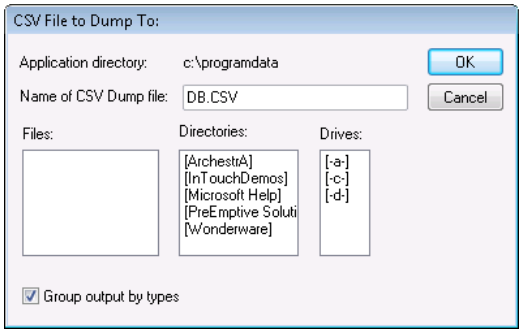


You use the DBDump utility within the Application Manager to export the contents of the Tagname Dictionary to a Comma Separated Value (CSV) file. You can view and edit the exported file with Microsoft Notepad or Microsoft Excel. After making edits, you then import the tag definitions to an InTouch application with the DBLoad utility, which is also an Application Manager utility.

You must convert an application to the current version of the InTouch HMI software before you can export the tag definitions.

To export tag definitions

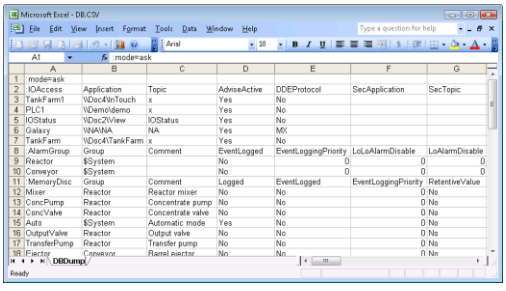
1. Close WindowMaker and WindowViewer.
2. Start Application Manager. The **Application Manager** dialog box shows a list of InTouch applications.
3. Select the application from the list.
4. Click the DBDump icon. The **CSV File to Dump To**: dialog box appears.



5. In the **Name of CSV Dump file** box, type a name for the file with a .csv file name extension.
6. Select the type of data grouping in the export file.
   - Select the **Group output by types** check box to group the data by the types of tags in the export file. This is the default.
   - Clear **Group output by types** to save the output to the export file alphabetically by tag name.
7. Click **OK** to save the contents of the Tagname Dictionary to the selected file. A message appears indicating the contents were saved successfully to the file.

Viewing Exported Tag Definitions

If you use Microsoft Excel to view an export file created with the DBDump utility, each data record appears in a separate spreadsheet cell.



The file consists of keywords, their attributes, and data from the Tagname Dictionary arranged in column order beneath keyword attributes.

Notice the :MemoryDisc keyword in the example of the Excel spreadsheet. This keyword identifies memory discrete tags that were exported from a Tagname Dictionary. On the same spreadsheet row, the attributes of a memory discrete tag appear in separate spreadsheet columns. For example, the Logged attribute column shows whether a memory discrete tag's data is logged or not.

Immediately beneath the keyword and attributes row are the exported tags and their associated properties. In the example of the Excel spreadsheet, OutputValve is a memory discrete tag whose data is not logged.

You can view or edit the export file created by DBDump with any program that supports the .csv file format. Typically, Excel is used because its columnar spreadsheet format makes it easy to organize tag data. But, you can also use Microsoft Notepad if you prefer to view or edit the file's contents in its native comma-delimited string format.

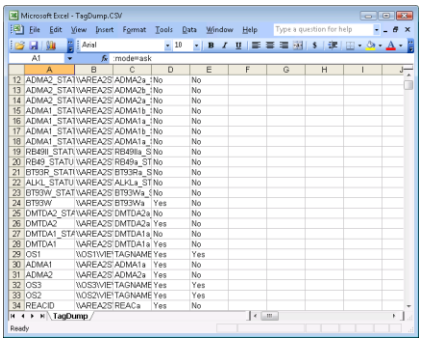
Importing Tag Definitions

You can use the DBLoad utility within the Application Manager to import a .csv file of tag definitions into an application's Tagname Dictionary. You can import a definition file that you originally created with the DBDump utility. Or, you can create your own import file.

You can also use the DBLoad utility instead of the InTouch TemplateMaker to create SuperTag instances. For more information, see *Creating SuperTag Instances* on page 146.

Tagname Dictionary Import File Format

You can manually create DBLoad import files with any program that supports a .csv file format. If you use Excel to create an import file, each entry is placed in a separate spreadsheet cell. This makes it much easier to read, and there is less chance of error.



For more information on creating import files, see *Creating an Import File Template* on page 115.

The DBLoad import file contains a set of keywords that organize Access Names, alarm groups, and tag data within the file.

- A colon (:) precedes all keywords.
- To continue a line, enter a backslash (\) at the end of the line.
- To enter comments, precede them with a semi-colon (;).

The following table lists the keywords within a DBLoad import file. The table lists the keywords in the order they are specified when you create the file with DBDump. But you can specify keywords in any order within the file.

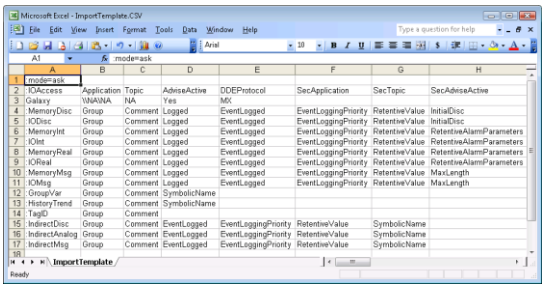
Keyword	Description
:mode	Specifies how duplicate tag records are handled when importing the contents of the DBLoad file to an application's Tagname Dictionary.
:IOAccess	Access names defined for the InTouch application.
:AlarmGroup	Alarm groups defined for the InTouch application.
:MemoryDisc	Memory discrete tags.
:IODisc	I/O discrete tags.
:MemoryInt	Memory integer tags.
:IOInt	I/O integer tags.
:MemoryReal	Memory real tags.
:IOReal	I/O real tags.
:MemoryMsg	Memory message tags.
:IOMsg	I/O message tags.
:GroupVar	Group Var tags.
:HistoryTrend	Hist Trend tags.
:TagID	Tag ID tags.
:IndirectDisc	Indirect discrete tags.
:IndirectAnalog	Indirect analog tags.
:IndirectMsg	Indirect message tags.

Each keyword includes a set of associated attributes that specify the properties of Access Names, alarm groups, and tags. For example, the :IOAccess keyword includes attributes to specify the application, topic, and communication protocol, which are properties of every InTouch Access Name.

Creating an Import File Template

You can manually create Tagname Dictionary import files with any application that supports the .csv file format. But, creating an entire import file can be time consuming and prone to errors. Using an existing .csv file as a template is faster and more reliable.

The following figure shows a template import file created by DBDump. The figure shows a file created from an InTouch application that has no windows nor tags. The resulting file only includes the required keywords and attributes without tag data.



After creating a template, you then manually add tag data beneath the keyword that identifies the type of tag. You insert the properties of your tags in the corresponding attribute columns associated with the tag type keywords.

To create a template import file

1. Open the Application Manager.
2. Create a new InTouch application.

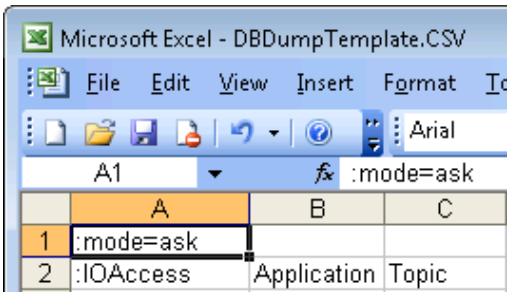
   For more information about the steps to create an application, see *Creating an InTouch Application* on page 30.
3. Select the new application from the list shown in Application Manager.
4. Export the contents of the application's Tagname Dictionary with the DBDump utility.

   For more information about exporting tags, see *Exporting Tag Definitions* on page 112.
5. Edit the file to insert tag data that you want to import.

Setting the Operating Mode for Dictionary Import Files

You must specify how DBLoad handles duplicate tag records while loading data from the import file into an application's Tagname Dictionary.

If you use a import file template created with DBDump, the first line of the file contains the :**mode** keyword. For example, you can assign the value ask to the :mode keyword in cell A1 of the Excel application.



You can assign the following values to a :mode keyword:

```
:MODE=REPLACE
:MODE=UPDATE
:MODE=ASK
:MODE=IGNORE
```

```
:MODE=TERMINATE
:MODE=TEST
```

:MODE=REPLACE

If a duplicate tag is encountered, the DBLoad utility deletes the existing tag in the Tagname Dictionary and replaces it with the tag from the import file with the same name.

:MODE=UPDATE

If a duplicate tag is encountered, the DBLoad utility overwrites the existing tag definition in the Tagname Dictionary only with data explicitly specified from the import file. All other data associated with the tag remains unchanged in the Tagname Dictionary.

Fields are considered explicitly defined if the field is in the record and entered by you or is set by the ":KEYWORD=value" mechanism. If a field is not specified in the record, and the keyword has been reset using the ":KEYWORD=" command, the current field value is not updated.

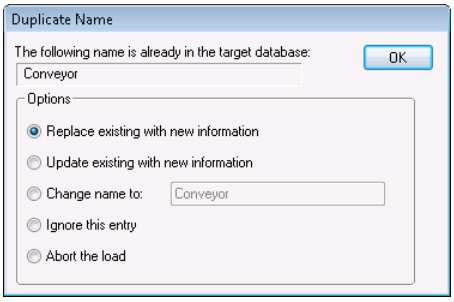
The following is an example of what occurs when an import file in the update mode is loaded/merged into the Tagname Dictionary:

```
:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
; Tagname1's Group updated to Group2 only
Tagname2,,
; Tagname2's Group updated to Group1 and the DConversion left as is
Tagname3,,Reverse
; Tagname3's Group updated to Group1 and the DConversion to "Reverse"
; the following line "resets" the Group field to its default value
:Group=
;  Data field "Group" is reset to its default value
Tagname4,,
; Tagname4 will be left alone
```

The tag types must be compatible if the type is being changed and the tag is in use. For example, an existing historical trend tag cannot be changed to an I/O Integer if the tag is in use by the application. Also, a tag cannot be changed to ReadOnly=yes if the tag is being used on an input link in the application. Because of these restrictions, update the use counts for the target application before running the DBLoad utility.

:MODE=ASK

DBLoad stops when a duplicate tag is encountered while loading the Tagname Dictionary. The **Duplicate Name** dialog box appears and shows a list of options to handle duplicate tags. This is the default import mode.



Options for handing duplicates are:

- Click **Replace existing with new information** to replace the existing tag record with the record from the import file.
- Click **Update existing with new information** to overwrite the existing tag record with only the fields that are explicitly defined in the import file.
- Click **Change Name to** replace the name of the imported tag with the name you type in the box of the **Duplicate Name** dialog box.
- Click **Ignore this entry** to ignore the tag and continue importing the contents of the file.
- Click **Abort the Load** to cancel the import process.

:MODE=IGNORE

The DBLoad import utility ignores the duplicate tag and continues processing the remaining records of the import file.

:MODE=TERMINATE

The DBLoad import operation stops when a duplicate tag is encountered.

:MODE=TEST

DBLoad scans the import file for errors and does not attempt to load tag definitions into the Tagname Dictionary. DBLoad generates a report that identifies any format errors by line number and location in the import file.

Run DBLoad with :**mode=test** first to identify any errors in the import file. After correcting any errors, change the **mode** keyword value to :**mode=replace** or :**mode=update** before running DBLoad.

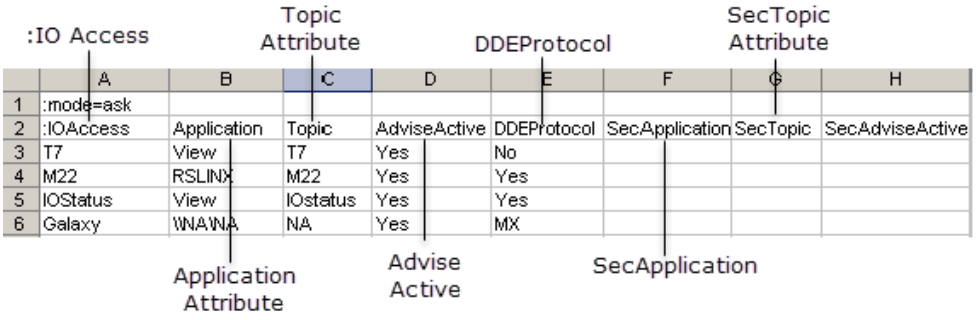
Setting Access Names and Alarm Groups

The DBLoad import file includes keywords that specify an InTouch application's defined Access Names and alarm groups.

:IOAccess Keyword Attributes

The **:IOAccess** keyword identifies the Access Names defined for an InTouch application. The **:IOAccess** keyword includes a set of attributes that describe the characteristics of a defined InTouch Access Name.

The following figure shows how Access Names are defined in an Excel spreadsheet with the :IOAccess keyword. Attributes are specified left to right in separate spreadsheet columns.



	A	B	C	D	E	F	G	H
1	:mode=ask							
2	:IOAccess	Application	Topic	AdviseActive	DDEProtocol	SecApplication	SecTopic	SecAdviseActive
3	T7	View	T7	Yes	No			
4	M22	RSLINX	M22	Yes	Yes			
5	IOStatus	View	IOstatus	Yes	Yes			
6	Galaxy	WAWA	NA	Yes	MX			

The following table shows the list of attributes associated with the :IOAccess keyword. The table lists the attributes in the order they are specified when using a template import file created with the DBDump utility.

String Position	Attributes	Acceptable Values	Default Values
1	Application	Application name defined for the Access Name	None
2	Topic	Topic name defined for the Access Name	None
3	AdviseActive	What information to poll from the server No = Advise all items Yes = Advise only active items	Yes
4	DDEProtocol	Communication protocol defined for the Access Name No = Suitelink Yes = DDE MX = Message Exchange	No
5	SecApplication	Application name defined for the secondary source of the Access Name	None
6	SecTopic	Topic name defined for the secondary source of the Access Name.	None

String Position	Attributes	Acceptable Values	Default Values
7	SecAdviseActive	When to poll information stored on the secondary server NO = Advise all items YES = Advise only active items	None
8	SecDDEProtocol	Communication protocol defined for the secondary source of the Access Name NO = Suitelink YES = DDE MX = Message Exchange	None
9	FailoverExpression	Failover expression that switches the Access Name to the secondary source when TRUE	None
10	FailoverDeadband	Integer number of seconds before starting failover to the secondary source defined by the Access Name	None
11	DFOFlag	Disable Failover flag Yes = Disable Failover flag set No = Disable Failover flag not set	None
12	FBDFlag	Switch back to Primary flag YES = Switch back to the Primary source after the failover condition clears NO = Do not switch back to the Primary source after the failover condition clears	None
13	FailbackDeadband	Integer number of seconds before switching back to the primary Access Name source after the failover condition clears	No value

:AlarmGroup Keyword Attributes

The DBLoad import file contains a keyword that identifies the alarm groups defined for an InTouch application. The **:AlarmGroup** keyword includes a set of attributes that describe the characteristics of the InTouch application's alarm groups.

The following table shows the list of attributes associated with the :AccessGroup keyword. The table lists the attributes in the order they are specified when using a template import file created with the DBDump utility.

String Position	Attributes	Acceptable Values	Default Values
1	Group	Name of the alarm group	$System
2	Comment	Comment assigned to the alarm group Any text string	None
3	EventLogged	Event logging enabled or disabled Yes or On = Event logging enabled No or Off = Event logging disabled	No
4	EventLoggingPriority	Priority assigned to events 1 to 999, 0 = not logged	0
5	LoLoAlarmDisable	LoLo alarm disabled or enabled 0 = LoLo alarm enabled 1 = LoLo alarm disabled	0
6	LoAlarmDisable	Low alarm disabled or enabled 0 = Low alarm enabled 1 = Low alarm disabled	0
7	HiAlarmDisable	High alarm disabled or enabled 0 = High alarm enabled 1 = High alarm disabled	0
8	HiHiAlarmDisable	HiHi alarm disabled or enabled 0 = HiHi alarm enabled 1 = HiHi alarm disabled	0
9	MinDevAlarmDisable	Minor Deviation alarm disabled or enabled 0 = Minor Deviation alarm enabled 1 = Minor Deviation alarm disabled	0
10	MajDevAlarmDisable	Major Deviation alarm disabled or enabled 0 = Major Deviation alarm enabled 1 = Major Deviation alarm disabled	0

String Position	Attributes	Acceptable Values	Default Values
11	RocAlarmDisable	Rate of Change alarm disabled or enabled 0 = ROC alarm enabled 1 = ROC alarm disabled	0
12	DSCAlarmDisable	Discrete alarms disabled or enabled 0 = Discrete alarm enabled 1 = Discrete alarm disabled	0
13	LoLoAlarmInhibititor	Name of the tag used to inhibit LoLo alarms Tag reference: any discrete or analog tag	None
14	LoAlarmInhibitor	Name of the tag used to inhibit Low alarms Tag reference: any discrete or analog tag	None
15	HiAlarmInhibitor	Name of the tag used to inhibit High alarms Tag reference: Any discrete or analog tag	None
16	HiHiAlarmInhibitor	Name of the tag used to inhibit HiHi alarms Tag reference: Any discrete or analog tag	None
17	MinDevAlarmInhibitor	Name of the tag used to inhibit Minor Deviation alarms Tag reference: Any discrete or analog tag	None
18	MajDevAlarmInhibitor	Name of the tag used to inhibit Major Deviation alarms Tag reference: Any discrete or analog tag	None

String Position	Attributes	Acceptable Values	Default Values
19	RocAlarmInhibitor	Name of the tag used to inhibit Rate of Change alarms Tag reference: Any discrete or analog tag	None
20	DSCAlarmInhibitor	Name of the tag used to inhibit discrete alarms Tag reference: any discrete or analog tag	None

Defining Tag Type Keywords and Attributes

Tag records begin with a keyword line that identifies the type of tag. Each tag keyword includes a unique set of attributes to specify the characteristics of the data associated with the type of tag.

In the following example, the :**IODisc** keyword identifies the I/O discrete tag type. The remaining values in the keyword line identify the attributes of the data associated with an I/O discrete tag. This example shows the contents of the file with Notepad in its native comma-delimited string format.

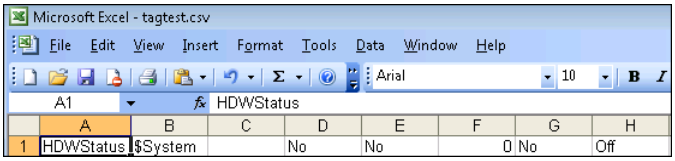
```
:IODisc,Group,Comment,Logged,EventLogged, EventLoggingPriority,RetentiveValue,InitialDis,
OffMsg,OnMsg,AlarmState,AlarmPri,DConversion, AccessName,ItemUseTagname,ItemName,ReadOnly,
AlarmComment,AlarmAckModel,DSCAlarmDisable, DSCAlarmInhibitor,SymbolicName
```

Beneath the tag type keyword line, individual rows specify the tags of that type with a set of attribute values. In the following example, the **HDWStatus** tag belongs to the I/O Discrete tag type in the import file.

```
"HDWStatus","$System","",No,No,0,No,Off,"","",,1,Direct,"HistdataViewstr",No,"Status",No,""
,0,0,"",""
```

The record uses quotation marks to identify a blank string.

The following figure shows the same import file data in an Excel spreadsheet. The **Comment** cell is blank because no tag comment is specified in the import file.



Tag Keyword Attributes

The following table lists all attributes associated with InTouch tag keywords. The table include columns that describe the type of data associated with each tag attribute and its default value.

These tag attributes can be specified in any order in your DBLoad import file as long as the accompanying tag data matches its corresponding attribute. For example, if you insert a :IODisc keyword in an Excel import file, then all I/O discrete tags' engineering units must be placed in the same Excel column as the EngUnits attribute.

Attribute	Acceptable Value	Default Value
AccessName	InTouch Access Name assigned to tag	None
AlarmAckModel	Alarm acknowledgement model Integer 0 = Condition 1 = Event Oriented 2 = Expanded Summary	0
AlarmComment	Alarm comment assigned to the tag Text string	None
AlarmDevDeadband	Tag deviation alarm deadband Real	None
AlarmPri	Alarm priority assigned to the tag 1 to 999	1
AlarmState	Tag alarm state On, Off, or None	None
AlarmValueDeadband	Tag alarm deadband Real	0
Comment	Comment assigned to the tag Text string	None
Conversion	Tag value conversion Linear or Square Root	Linear
Deadband	Value deadband assigned to the tag Real	0
DevTarget	Tag deviation target value Real	0
DSCAlarmDisable	Discrete alarms disabled or enabled 0 = Discrete alarm enabled 1 = Discrete alarm disabled	0
DSCAlarmInhibitor	Name of the tag used to inhibit a discrete alarm	None

Attribute	Acceptable Value	Default Value
EngUnits	Engineering units assigned to tag Text string	None
EventLogged	Event logging enabled or disabled Yes or On = Event logging enabled No or Off = Event logging disabled	No
EventLogging	Tag event logging enabled or disabled No or Off = Logging disabled Yes or On = Logging enabled	No
EventLoggingPriority	Priority assigned to events 1 to 999, 0 = not logged	0
Group	Name of the alarm group in which the tag belongs	$System
HiAlarmDisable	High alarm disabled or enabled 0 = High alarm enabled 1 = High alarm disabled	0
HiAlarmInhibitor	Name of the tag used to inhibit High alarm Any discrete or analog tag	None
HiAlarmPri	Priority assigned to High alarm 1 to 999	1
HiAlarmState	High alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No
HiAlarmValue	High alarm point assigned to tag Real	0
HiHiAlarmDisable	HiHi alarm disabled or enabled 0 = HiHi alarm enabled 1 = HiHi alarm disabled	0
HiHiAlarmInhibitor	Name of the tag used to inhibit HiHi alarm Any discrete or analog tag	None

Attribute	Acceptable Value	Default Value
HiHiAlarmPri	Priority assigned to HiHi alarm 1 to 999	1
HiHiAlarmState	HiHi alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No
HiHiAlarmValue	HiHi alarm point assigned to tag Real	0
InitialDisc	Initial value assigned to discrete tag 0, Off, False, or No = Off 1, On, True, or Yes = On	0
InitialMessage	Initial tag message Text string	None
InitialValue	Initial value assigned to the tag Real	0
ItemName	Name of the item assigned to the tag Text string	None
ItemUseTagname	**Use Tagname as Item Name** option enabled or disabled No or False = Disabled Yes or True = Enabled	No
LoAlarmDisable	Low alarm disabled or enabled 0 = Low alarm enabled 1 = Low alarm disabled	0
LoAlarmInhibitor	Name of the tag used to inhibit Low alarm Any discrete or analog tag	None
LoAlarmPri	Priority assigned to Low alarm 1 to 999	1
LoAlarmState	Low alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No

Attribute	Acceptable Value	Default Value
LoAlarmValue	Low alarm point assigned to tag Real	0
LogDeadband	Logging deadband assigned to the tag Real	0
Logged	Tag value logging enabled or disabled No or Off = Logging disabled Yes or On = Logging enabled	No
LoLoAlarmDisable	LoLo alarm disabled or enabled 0 = LoLo alarm enabled 1 = LoLo alarm disabled	0
LoLoAlarmInhibitor	Name of the tag used to inhibit LoLo alarm Any discrete or analog tag	None
LoLoAlarmPri	Priority assigned to LoLo alarm 1 to 999	1
LoLoAlarmState	LoLo alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No
LoLoAlarmValue	LoLo alarm point assigned to tag Real	0
MajDevAlarmDisable	Major Deviation alarm disabled or enabled 0 = Major Deviation alarm enabled 1 = Major Deviation alarm disabled	0
MajDevAlarmInhibitor	Name of the tag used to inhibit Major Deviation alarm Any discrete or analog tag	None
MajorDevAlarmPri	Priority assigned to Major Deviation alarm 1 to 999	1

Attribute	Acceptable Value	Default Value
MajorDevAlarmState	Major deviation alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No
MajorDevAlarmValue	Major deviation alarm percentage assigned to tag Real	0
MaxEU	Maximum engineering units value assigned to the tag Real	32767
MaxLength	Maximum message length Real	131
MaxRaw	Maximum raw value assigned to tag Real	32767
MaxValue	Maximum value assigned to the tag Real	32767
MinDevAlarmDisable	Minor Deviation alarm disabled or enabled 0 = Minor Deviation alarm enabled 1 = Minor Deviation alarm disabled	0
MinDevAlarmInhibitor	Name of the tag used to inhibit Minor Deviation alarm Any discrete or analog tag	None
MinEU	Minimum engineering units value assigned to the tag Real	-32768
MinorDevAlarmPri	Priority assigned to Minor Deviation alarm 1 to 999	1
MinorDevAlarmState	Minor deviation alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No

Attribute	Acceptable Value	Default Value
MinorDevAlarmValue	Minor deviation alarm percentage assigned to tag Real	0
MinRaw	Minimum raw value assigned to tag Real	-32768
MinValue	Minimum value assigned to the tag Real	-32768
OffMsg	Discrete tag Off message Text string	None
OnMsg	Discrete tag On message Text string	None
ReadOnly	Tag value read only or read/write Yes = Read Only No = Read/Write	No
RetentiveAlarmParameters	Tag Retentive Parameters enabled or disabled No or Off = Disabled Yes or On = Enabled	No
RetentiveValue	Tag Retentive Value enabled or disabled 0, Off, False, or No = Disabled 1, On, True, or Yes = Enabled	No
RocAlarmDisable	Rate of Change alarm disabled or enabled 0 = ROC alarm enabled 1 = ROC alarm disabled	0
RocAlarmInhibitor	Name of the tag used to inhibit Rate of Change alarm Any discrete or analog tag	None
ROCAlarmPri	Priority assigned to Rate of Change alarm 1 to 999	1

Attribute	Acceptable Value	Default Value
ROCAlarmState	Rate of Change alarm enabled or disabled No or Off = Disabled Yes or On = Enabled	No
ROCAlarmValue	Change in tag value by percent Real	0
ROCTimeBase	Measurement period to calculate rate of change Sec, Min or Hr	Min
SymbolicName	Symbolic name assigned to input data blocks by the S7 Tag Creator product. Symbolic names are listed in the S7 Tag Creator Symbol Table.	None

:MemoryDisc Keyword Attributes

The DBLoad import file includes the :MemoryDisc keyword to define memory discrete tags that can be imported to the Tagname Dictionary. The following table lists the attributes of the :MemoryDisc keyword associated with the properties of a memory discrete tag.

The table shows the order that :MemoryDisc keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg

String Position	Attribute
10	AlarmState
11	AlarmPri
12	AlarmComment
13	AlarmAckModel
14	DSCAlarmDisable
15	DSCAlarmInhibitor
16	SymbolicName

:IODisc Keyword Attributes

The DBLoad import file includes the :IODisc keyword to define I/O discrete tags that can be imported to the Tagname Dictionary. The following table lists the attributes of the :IODisc keyword associated with the properties of an I/O discrete tag.

The table shows the order that :IODisc keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	Conversion
13	AccessName

String Position	Attribute
14	ItemUseTagname
15	ItemName
16	ReadOnly
17	AlarmComment
18	AlarmAckModel
19	DSCAlarmDisable
20	DSCAlarmInhibitor
21	SymbolicName

:MemoryInt Keyword Attributes

The DBLoad import file includes the :MemoryInt keyword to define memory integer tags that can be imported to the Tagname Dictionary. The following table lists the attributes of the :MemoryInt keyword associated with the properties of a memory integer tag.

The table shows the order that :MemoryInt keyword attributes are specified when the DBDump utility is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinValue

String Position	Attribute
13	MaxValue
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHIAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable

String Position	Attribute
42	LoAlarmDisable
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor
55	SymbolicName

:IOInt Keyword Attributes

The DBLoad import file includes the :IOInt keyword to define I/O integer tags that can be imported to the Tagname Dictionary. The following table lists the attributes of the :IOInt keyword associated with the properties of an I/O integer tag.

The table shows the order that :IOInt keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue

String Position	Attribute
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinEU
13	MaxEU
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHIAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState

String Position	Attribute
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
39	MinRaw
40	MaxRaw
41	Conversion
42	AccessName
43	ItemUseTagname
44	ItemName
45	ReadOnly
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor
62	SymbolicName

:MemoryReal Keyword Attributes

The DBLoad import file includes the :MemoryReal keyword to define memory real tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :MemoryReal keyword associated with the properties of a memory real tag.

The table shows the order that :MemoryReal keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinValue
13	MaxValue
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue

String Position	Attribute
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor

String Position	Attribute
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor
55	SymbolicName

:IOReal Keyword Attributes

The DBLoad import file includes the :IOReal keyword to define I/O real tags that can be imported to the Tagname Dictionary. The following table lists the attributes of the :IOReal keyword associated with the properties of an I/O real tag.

The table shows the order that :IOReal keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinEU
13	MaxEU
14	Deadband
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue

String Position	Attribute
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	MinRaw
40	MaxRaw
41	Conversion
42	AccessName
43	ItemUseTagname
44	ItemName
45	ReadOnly
46	AlarmComment

String Position	Attribute
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor
62	SymbolicName

:MemoryMsg Keyword Attributes

The DBLoad import file includes the :MemoryMsg keyword to define memory message tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :MemoryMsg keyword associated with the properties of a memory message tag.

The table shows the order that :MemoryMsg keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged

String Position	Attribute
5	EventLoggingPriority
6	RetentiveValue
7	MaxLength
8	InitialMessage
9	AlarmComment
10	SymbolicName

:IOMsg Keyword Attributes

The DBLoad import file includes the :IOMsg keyword to define I/O message tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :IOMsg keyword associated with the properties of an I/O message tag.

The table shows the order that :IOMsg keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	Logged
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	MaxLength
8	InitialMessage
9	AccessName
10	ItemUseTagname
11	ItemName
12	ReadOnly
13	AlarmComment
14	SymbolicName

:GroupVar Keyword Attributes

The DBLoad import file includes the :GroupVar keyword to define Group Variable tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :GroupVar keyword associated with the properties of a Group Variable tag.

Note: InTouch Group Var tags are obsolete. The :GroupVar keyword is included to support legacy applications only.

The table shows the order that :GroupVar keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	SymbolicName

:HistoryTrend Keyword Attributes

The DBLoad import file includes the :HistoryTrend keyword to define HistTrend tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :HistoryTrend keyword associated with the properties of a HistTrend tag.

The table shows the order that :HistoryTrend keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	SymbolicName

:TagID Keyword Attributes

The DBLoad import file includes the :TagID keyword to define Tag ID tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :TagID keyword associated with the properties of a Tag ID tag.

The table shows the order that :TagID keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment

:IndirectDisc Keyword Attributes

The DBLoad import file includes the :IndirectDisc keyword to define indirect discrete tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :IndirectDisc keyword associated with the properties of an indirect discrete tag.

The table shows the order that :IndirectDisc keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

:IndirectAnalog Keyword Attributes

The DBLoad import file includes the :IndirectAnalog keyword to define indirect analog tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :IndirectAnalog keyword associated with the properties of an indirect analog tag.

The table shows the order that :IndirectAnalog keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	EventLogging
4	EventLoggingPriority

String Position	Attribute
5	RetentiveValue
6	SymbolicName

:IndirectMsg Keyword Attributes

The DBLoad import file includes the :IndirectMsg keyword to define indirect message tags that will be imported to the Tagname Dictionary. The following table lists the attributes of the :IndirectMsg keyword associated with the properties of an indirect message tag.

The table shows the order that :IndirectMsg keyword attributes are specified when DBDump is used to create the import file. See *Tag Keyword Attributes* on page 123 for the data associated with these attributes and their default values.

String Position	Attribute
1	Group
2	Comment
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

Using Blank Strings in an Import File

For a dictionary import file, there is a difference between a field containing a blank string and a field without data. Keyword attributes that can be assigned a blank string are:

Comment	Eng Units	OffMsg
InitialMessage	OnMsg	Application
ItemName	Topic	

In the following example, a blank string is indicated by quotation marks (" "):

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System
```

where:

The value of the Comment field for Tagname1 is Hi, and the value of the Comment field for Tagname2 is a blank comment.

Microsoft Excel ignores quotation marks that denote a blank string when it saves the file, resulting in the following:

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,,$System
```

To ensure that a blank string is used with Excel, type a space in the cell as the attribute value.

Using Default Values for Fields

You can use keywords to set the default values for specific fields of a record. The default values are the original InTouch values for the tag type. For example, a memory discrete tag uses the Group=$System, EventLogging=Off, InitialValue=Off, as default values.

For example:

```
:KEYWORD=value
```

This sets the default value of the referenced field for all subsequent data records. Use this feature to set the default value for fields that should remain unchanged for a number of records. If a field has a default value defined, the default value is used if there is no data in the record for the value.

For example, if you set :GROUP=Reactor_Site, then all tags that have a blank entry for the GROUP column are assigned to the Reactor_Site Alarm Group. If the tag has, for example, $System entered for the GROUP, the tag remains assigned to the Alarm Group $System.

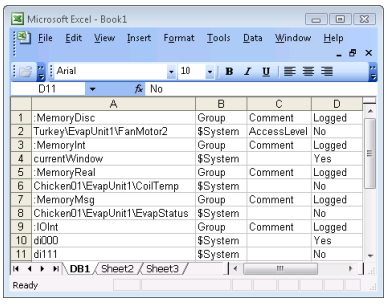
You can reset an individual keyword to its original default value by omitting the value in the equation. For example, :GROUP=.

To reset all keywords, use the :RESET command. This command does not have arguments and affects all entries in the file that occur after the command.

Creating SuperTag Instances

You can create SuperTags using the DBLoad utility within the Application Manager. However, the SuperTag instances you create are not reflected in the SuperTag template definition in the TemplateMaker.

You must use the valid SuperTag format, and the SuperTag instance data records must begin with the valid keyword for the tag type. For example:



The following syntax examples are valid:

```
ParentInstance\ChildMember
```

```
ParentInstance\ChildMember\Submember
```

The following syntax examples are invalid:

```
ParentInstance\
ParentInstance\ChildMember\
```

If you use an invalid format, an error message appears.

When you import the CSV file containing SuperTag instances, the instances are automatically added to the Tagname Dictionary and are immediately available for use in animation links and InTouch QuickScripts.

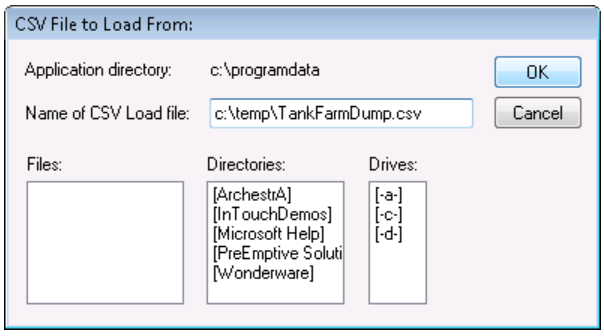
Importing Tag Definitions with DBLoad

When you import the contents of a file with DBLoad, all tag definitions are imported into the Tagname Dictionary of the selected InTouch application.

If the import fails, a message appears describing the reason for the failure. The error messages are written to the Logger.

To import tag definitions into an InTouch application

1. Close WindowMaker and WindowViewer.
2. Back up the application whose Tagname Dictionary will be loaded with tag definitions from an import file.
3. Start **Application Manager.**
4. Select the application from the list whose Tagname Dictionary will receive the imported tag definitions.
5. Click the DBLoad icon. A message appears requesting confirmation that you backed up the InTouch application.
6. Click **Yes** to confirm the application is backed up. The **CSV File to Load From** dialog box appears.



7. In the **Name of CSV Load file** box, locate and select the file you want to import.
8. Click **OK**.

   The next step varies based upon whether DBLoad imports new or existing tag definitions to the Tagname Dictionary.

   - If you are importing new tag definitions, the new tag data is loaded into the application's Tagname Dictionary. A message appears confirming the data was successfully loaded and merged.

- If you are importing existing tag definitions, the import stops if the :mode keyword is set to :mode=ask and the import file contains duplicate tags. You are shown options to handle the duplicate tags or you can cancel the import. For more information about keyword options, see *Setting the Operating Mode for Dictionary Import Files* on page 116.

Importing Windows

Importing windows from an existing InTouch application into your current application allows you to reduce development time because you can reuse your previously created windows, objects, and window scripts.

You must convert an application to the current version of the InTouch HMI software before you can import windows.

By default, placeholders are created for the tags associated with an imported window. After importing, you can convert the placeholders to local tags or remote tag references. For more information, see *Tag Placeholders for Imported Windows and Scripts* on page 154. If the associated tags already exist in the target application, during the import you can select to use these instead.

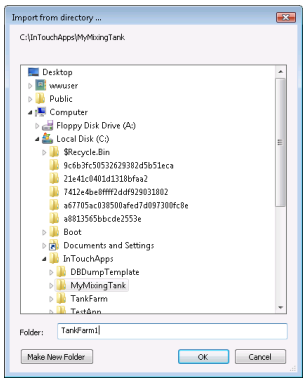
When you import windows containing SmartSymbols and select to use existing tags, the InTouch HMI still keeps placeholders for the recovered symbols, even though the tags are available in the target application.

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

Important: If you move InTouch window files using any method other than importing or exporting them, the contents of the application Tagname Dictionary can become corrupt.

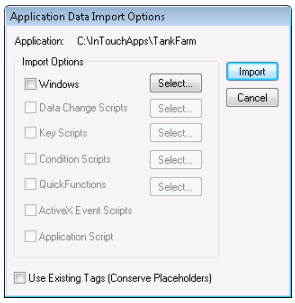
To import a window

1. Close all windows in your current application.
2. On the **File** menu, click **Import**, and then click **Windows and Scripts**. The **Import from directory** dialog box appears.



3. Select the folder for the application containing the windows to import.

4. Click **OK**. The **Application Data Import Options** dialog box appears.



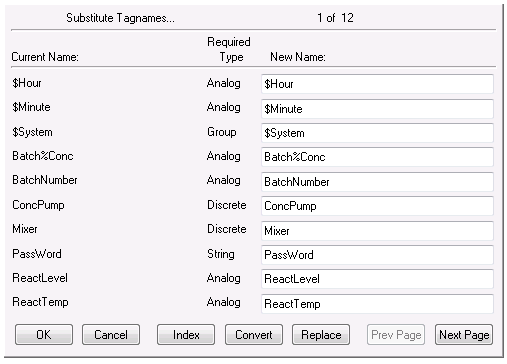
5. Select the **Windows** check box and then click **Select** to select the individual windows to import.
6. Select the **Use Existing Tags (Conserve Placeholders)** check box if the tags associated with the imported windows already exist in your application and you want to use them instead of placeholders.
7. Click **Import**.
8. Convert the placeholder tags to either local tags or remote tag references. For more information, see *Converting Placeholder Tags for an Imported Window* on page 149.
9. If an imported window contains one or more wizards, double-click on each wizard to open its properties panel. If an imported window contains one or more SmartSymbols, edit each SmartSymbol and create new instances.

Converting Placeholder Tags for an Imported Window

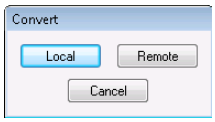
When you import or export a window or QuickScript to or from your current application, all the tags associated with that window or QuickScript are transferred with the window. But, the tags are not added to your new application's Tagname Dictionary. Instead, the tags are automatically marked as placeholder tags unless the **Conserve Placeholders** options is selected on import. You must convert these placeholder tags and, if required, define them in your new application Tagname Dictionary.

To convert tags for a window

1. Open the window in WindowMaker.
2. Press F2 to select all objects in the window.
3. On the **Special** menu, click **Substitute Tags**. The **Substitute Tagnames** dialog box appears.

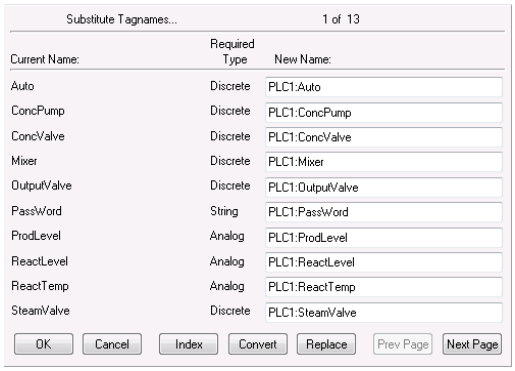


4. Click **Convert**. The **Convert** dialog box appears.



5. Convert the tags.
   - Click **Local** to convert the placeholder tags to local tags. You are prompted to define each tag in the Tagname Dictionary.
   - Click **Remote** to convert the placeholder tags to remote tag references. The **Access Names** dialog box appears. Select the Access Name and click **Close**.

   After the conversion, the **Substitute Tagnames** dialog box shows the new tags.



6. Click **OK**.

Exporting Windows

You can export application windows to:

- Create or maintain a library application of all windows.
- Create remote tag references in another application.

You must convert an application to the current version of the InTouch HMI software before you can export windows.

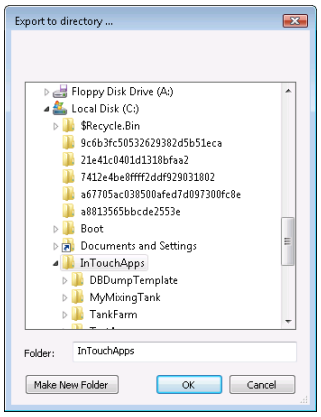
When you export a window, all objects and animation links associated with that window are exported. The tags associated with the objects in the window are converted to placeholder tags to prevent existing tags in the destination application from being overwritten. For more information on converting placeholder tags, see *Converting Placeholder Tags for an Imported Window* on page 149.

Important: If you move InTouch window files using any method other than importing or exporting them, the application's Tagname Dictionary can be corrupted.

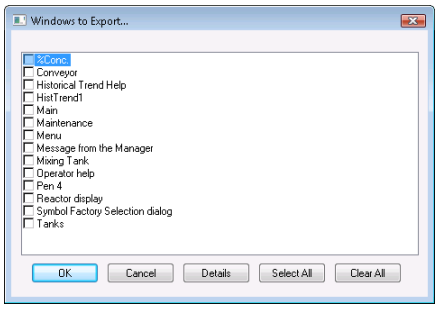
To export a window

1. Close all windows in your current application.

2. On the **File** menu, click **Export**, and then click **Windows**. The **Export to directory** dialog box appears.



3. Select the folder of the application to which to export the windows.
4. Click **OK**. The **Windows to Export** dialog box appears.



5. Select the windows to export.
6. Click **OK**.

   If a problem occurs, the **Problem with Export Operation** dialog box appears. Click the option for the action you want to take and then click **OK**.

Importing Scripts

You can import existing QuickScripts from an InTouch application into your current application to save development time.

You must convert an application to the current version of the InTouch HMI software before you can import scripts.

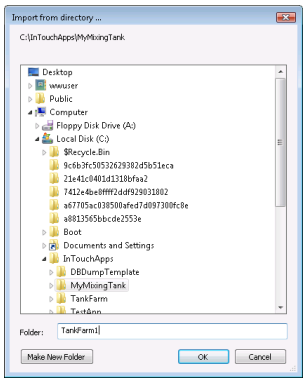
By default, placeholders are created for the tags associated with an imported QuickScript. After importing, you can convert the placeholders to local tags or remote tag references. For more information, see *Tag Placeholders for Imported Windows and Scripts* on page 154. If the associated tags already exist in the target application, during the import you can choose to use these instead.

To import a window script, you must import the entire window.

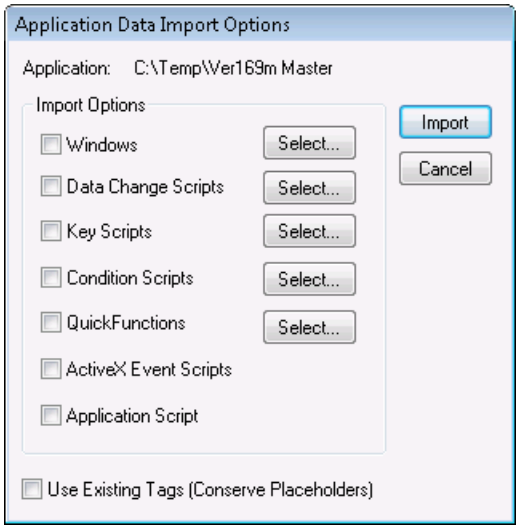
For an imported ActiveX Event script to function properly in the target application, the same ActiveX control and the same event for which the script was originally created must also be used in the target application and it must be loaded into memory. If the window containing an ActiveX control is closed, any scripts associated with it (either ActiveX Event scripts or QuickScripts) do not run properly.

To import a QuickScript

1. Close all windows in your current application.
2. On the **File** menu, click **Import**, and then click **Windows and Scripts**. The **Import from directory** dialog box appears.



3. Select the folder for the application that contains the scripts to import.
4. Click **OK**. The **Application Data Import Options** dialog box appears.



5. Select the check box for the QuickScript type(s) that you want to import and then click **Select** to select the individual script(s) to import.

Note: To import a window script, you must import the entire window. For more information, see *Importing Windows* on page 148.

---

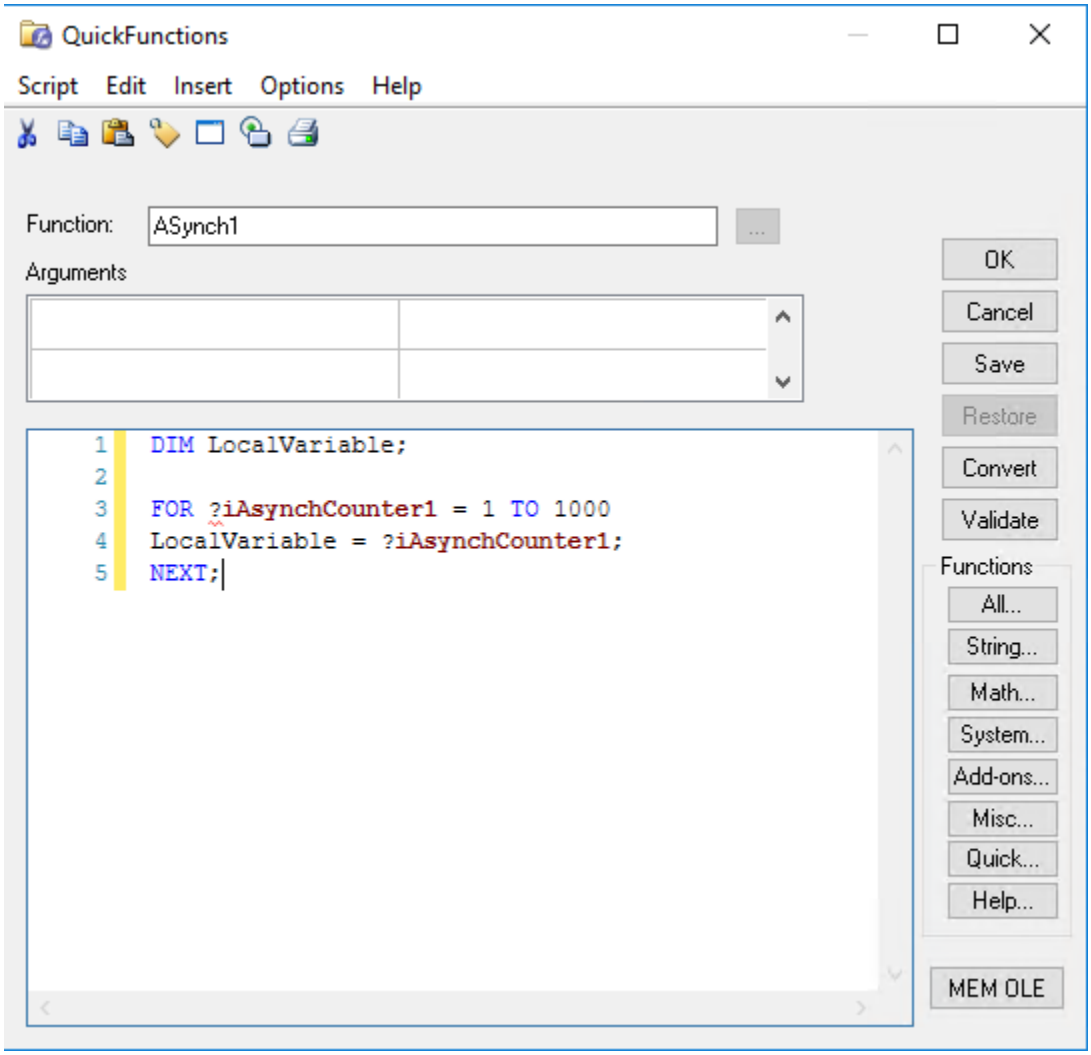
6. Select the **Use Existing Tags (Conserve Placeholders)** check box if the tags associated with the imported script(s) already exist in your application and you want to use them instead of placeholders.
7. Click **Import**. If your application has scripts with identical names, you are prompted to overwrite, skip, or rename.
8. Convert the placeholder tags to either local tags or remote tag references. For more information, see *Converting Placeholder Tags in an Imported Script* on page 153.

Converting Placeholder Tags in an Imported Script

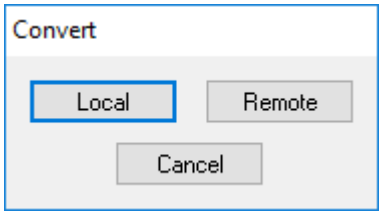
When you import or export a QuickScript to or from your current application, all the tags associated with that QuickScript are transferred. But, the tags are not added to your new application's Tagname Dictionary. Instead, they are automatically marked as placeholder tags. You must convert these placeholder tags and, if required, define them in your new application Tagname Dictionary.

To convert placeholder tags in an imported script

1. On the **Special** menu, point to **Scripts**, and then click the type of QuickScript you imported. The QuickScript editor appears, showing the first QuickScript on file for the selected type of script.



2. Click **Convert**. The **Convert** dialog box appears.



3. Convert the tags.
    - Click **Local** to convert the placeholder tags to local tags. You are prompted to define each tag in the Tagname Dictionary.
    - Click **Remote** to convert the placeholder tags to remote tag references. The **Access Names** dialog box appears. Select the Access Name and click **Close**.
4. After the tags are converted, click **OK** in the QuickScript editor.

Tag Placeholders for Imported Windows and Scripts

When you import a window or QuickScript, you can configure how you want the associated tags to be handled.

- **Use placeholder tags.**

  By default, imported tags are converted to "placeholder" (or "index") tags. A maximum of 4096 placeholders is allowed.

  Placeholder tags include a three-character prefix. For example, if the original tag is WaterHeater, then the placeholder tag is ?d:WaterHeater.

  If you import a tag that contains 30, 31, or 32 characters, the placeholder prefix is still added to the beginning of the tag, and the length of the existing tag is not truncated. For example, for placeholder tags only, a 32 character tag is increased to 35 characters. This increase in tag length is not supported for standard tags.

  To use a placeholder tag in the application, you must either:

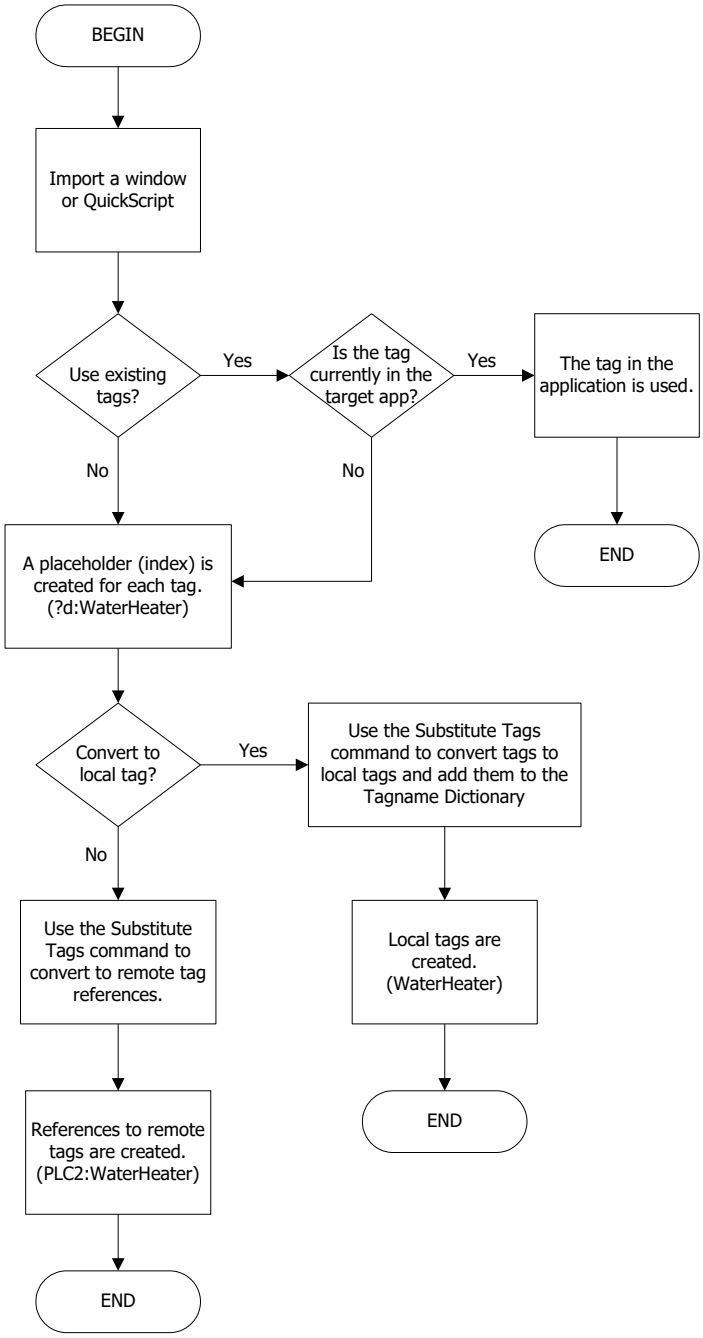
  Convert it to a regular (local) tag and define it in the Tagname Dictionary.

  Convert it to a remote tag reference. An example of a remote tag is PLC2:WaterHeater. Remote tag references allow your application to instantly receive data from a remote tag server and eliminates the need to define a single tag in the local Tagname Dictionary.

- **Using existing tags.**

  During an import, if you select to use existing tags, the InTouch HMI verifies that the imported tags already exist in the Tagname Dictionary. If a tag already exists, then the tag is imported as a fully qualified tag. Using this option reduces the total number of placeholders, allowing you to import applications with larger tag databases.

The following flowchart describes how tags are handled for imported windows and QuickScripts.



Exporting Industrial Graphics from an Application

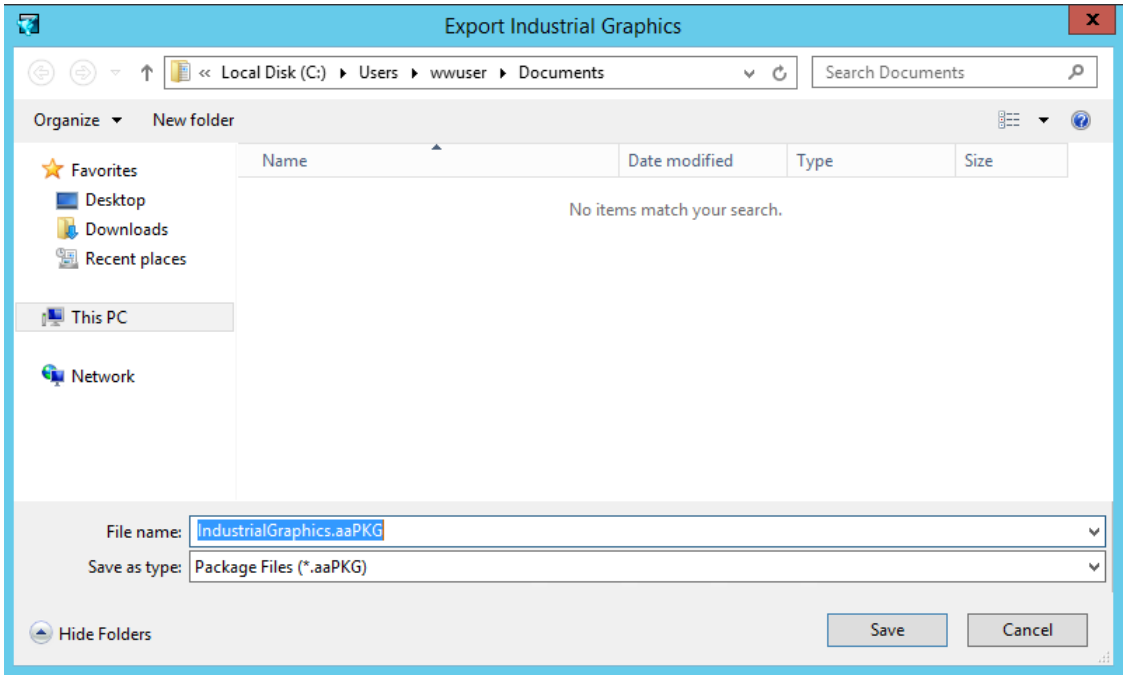
You can export all Industrial graphics from an application to an aaPKG file. You can then import the graphics from the file to another application on the same or different computer.

You cannot select Industrial graphics individually to export from an application. All Industrial graphics are exported from an application.

To export Industrial graphics from an application

1. Open the application in WindowMaker containing the Industrial graphics that you want to export.
2. On the **File** menu, click **Export**, and then click **All Industrial Graphics**.

   The **Export Industrial Graphics** dialog box appears to specify the destination folder and the name of the export file.



3. Select the destination folder to export the aaPKG file.
4. If you want, enter the name of the export file in the **File name** field.

   The default export file name is IndustrialGraphics.aaPKG.
5. Click **Save**.

   A horizontal bar shows the progress of the Industrial Graphics being loaded into the export file.
6. Once the export process is finished, navigate to the destination folder in Windows Explorer, and verify that the export file has been created.

Importing Industrial Graphics to an Application

You can import Industrial Graphics created in another application to the active application running in WindowMaker.

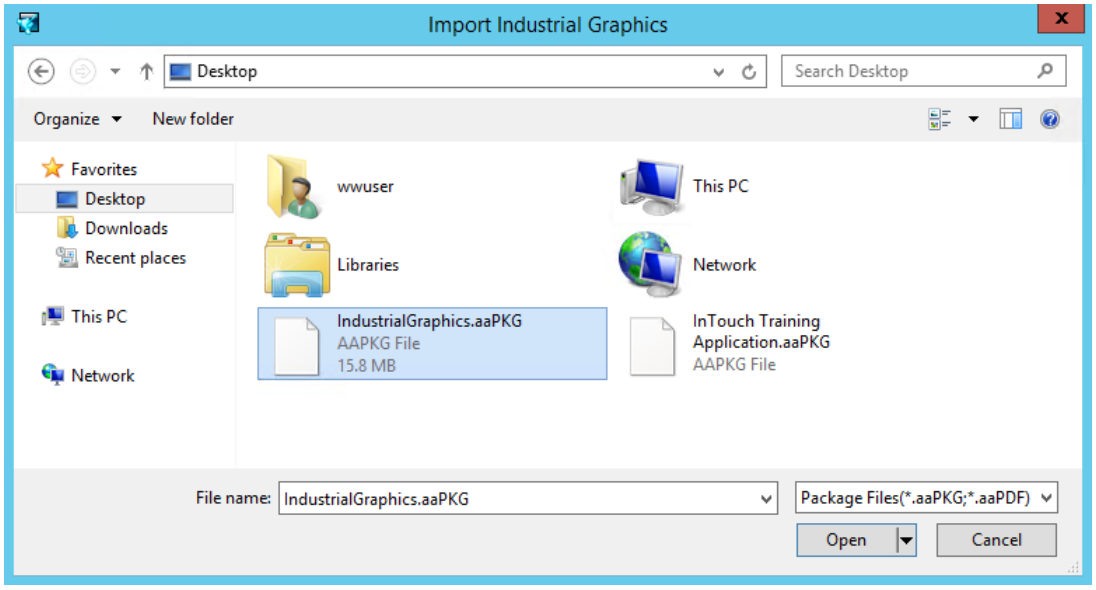
Only Industrial Graphics from the aaPKG file are imported. The imported graphics overwrite any graphics of the application open for editing in WindowMaker. If the aaPKG file contains non-supported components, the import fails and a dialog box with an error is shown.

To import Industrial Graphics to an application

1. Open the application in WindowMaker that you want to import Industrial Graphics.

2. On the **File** menu, click **Import**, and then click **Industrial Graphics**.

   The **Import Industrial Graphics** dialog box appears to specify the folder containing an export file of Industrial Graphics.



3. Using Windows Explorer, go to the folder containing an aaPKG file of exported Industrial Graphics.
4. Select the aaPKG file to import.

   The **File name** field shows the name of the file you selected.

5. Click **Open**.

   The Import Industrial Graphics dialog box appears with the following options for overwriting graphics:

   - Skip: Do not Import - The graphics will not be imported
   - Overwrite if the importing graphic change version is higher - Will import the graphics only if the version of the file imported is higher than the installed version.
   - Overwrite regardless of graphic change version - The graphics will be imported.

6. Click **OK**.

   A horizontal bar shows the progress of the Industrial Graphics being imported into the active application. When finished, the progress indicator disappears.

Exporting Selected Symbols from the Industrial Graphic Toolbox

You can export selected Industrial Graphics from the Industrial Graphic Toolbox of an application to an aaPKG file. You can then import these graphics from the file to another application on the same or different computer.

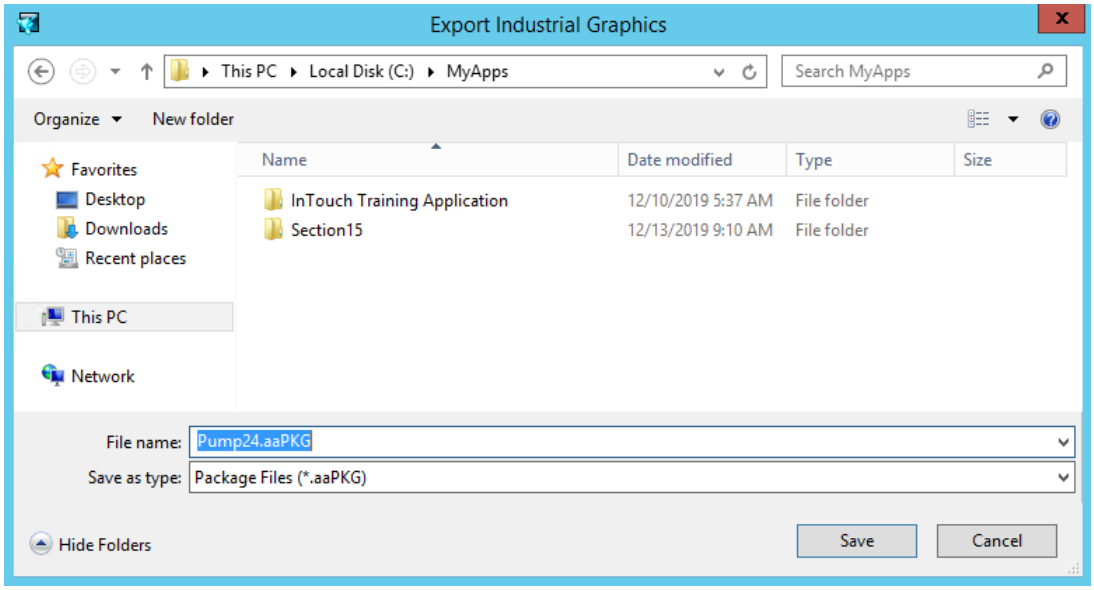
Note: This procedure explains how to export selected Industrial Graphics. See *Exporting Industrial Graphics from an Application* on page 155 for instructions to export all Industrial Graphics.

To export selected Industrial Graphics from an Application

1. Open the application in WindowMaker containing the Industrial Graphics that you want to select to export.

2. Select the symbols you want to export in the Industrial Graphic Toolbox.
3. Right-click on a selected symbol to show the shortcut menu.
4. Select **Export** and then **Symbol(s)...** from the shortcut menu.

   The **Export Industrial Graphics** dialog box appears to specify the destination folder and the name of the export file.



5. Select the destination folder to export the aaPKG file.
6. If you want, enter the name of the export file in the **File name** field.

   The default export file name is the name of the first selected symbol from the Industrial Graphic Toolbox.
7. Click **Save**.

   A horizontal bar shows the progress of the Industrial Graphics being loaded into the export file.

Importing and Embedding Custom Client Controls

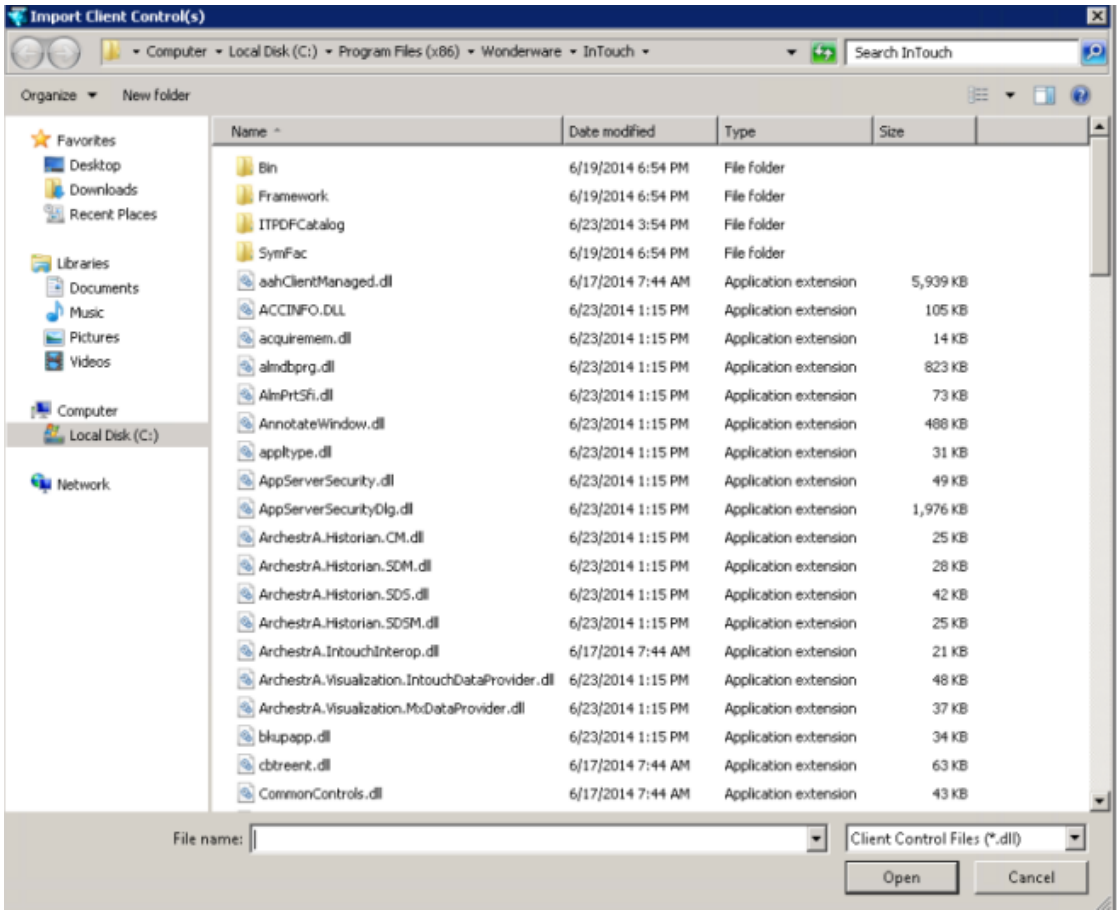
You can create a custom Windows client control and embed it in an Industrial Graphic in your application. First, you must import the client control to WindowMaker's Industrial Graphic Toolbox. This section describes the steps to import and then embed a custom client control in separate procedures.

To import a custom client control

1. Create a custom client control for your application.
2. Place the client control in a folder accessible to the computer where InTouch WindowMaker is installed.
3. On the **File** menu, click **Import**, and then click **Client Control**.

Important: Only standalone applications can import custom client controls. You cannot import custom client controls to legacy or published InTouch HMI applications.

The **Import Client Control(s)** dialog box appears with a field to enter the name of a custom client control you created.



4. Using Windows Explorer, go to the folder where you placed the client control .dll file.
5. Select the client control .dll file and click **Open**.

   WindowMaker updates and shows the custom client control you imported in the Industrial Graphic Toolbox.

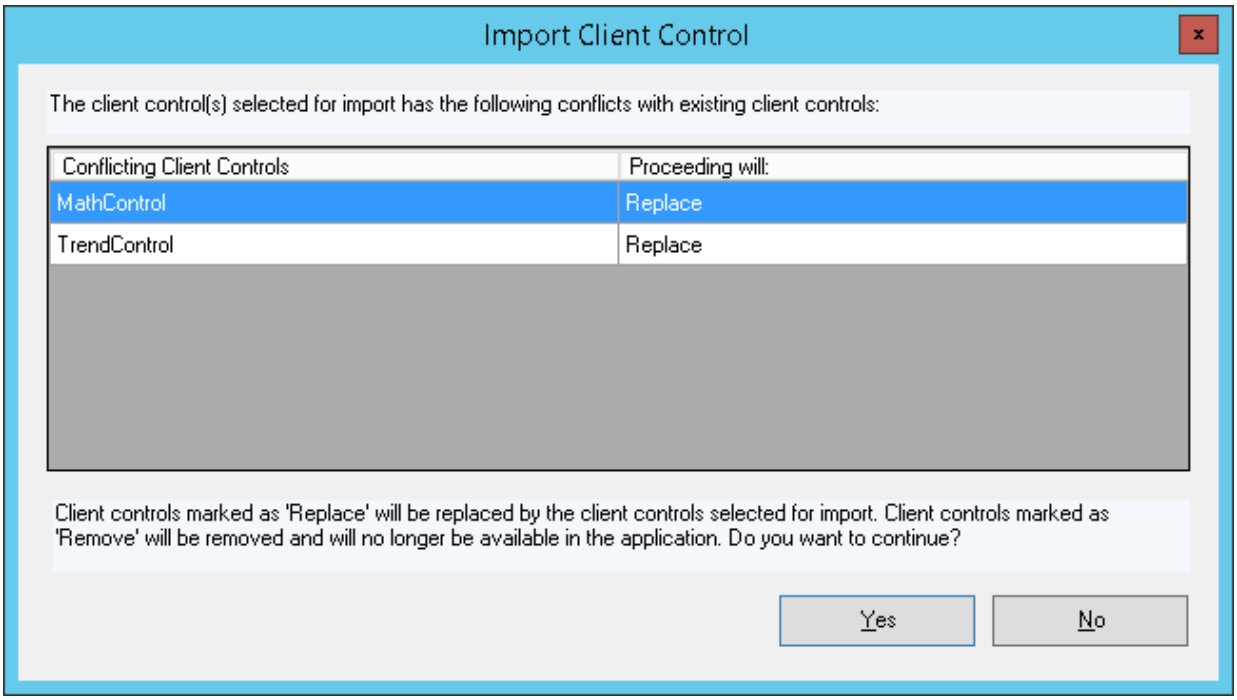
You can also remove an imported client control from the Industrial Graphic Toolbox. First, select the client control within the Industrial Graphic Toolbox. Then, right-click to show the shortcut menu and select **Delete**.

Resolving Conflicts When Importing Duplicate Client Controls

You can import a different version of a client control and overwrite the existing control. The .dll hosting the existing control will be replaced by the importing library. Conflicting client controls will be detected upon import of the new client control .dll.

Note: Conflict detection is based solely on the name of the control. Library filenames or versions have no affect on conflict detection.

For example, if you import a client control .dll containing the two controls MathControl and TrendControl and the current library contains controls of the same name, the Import Client Control dialog box will display:

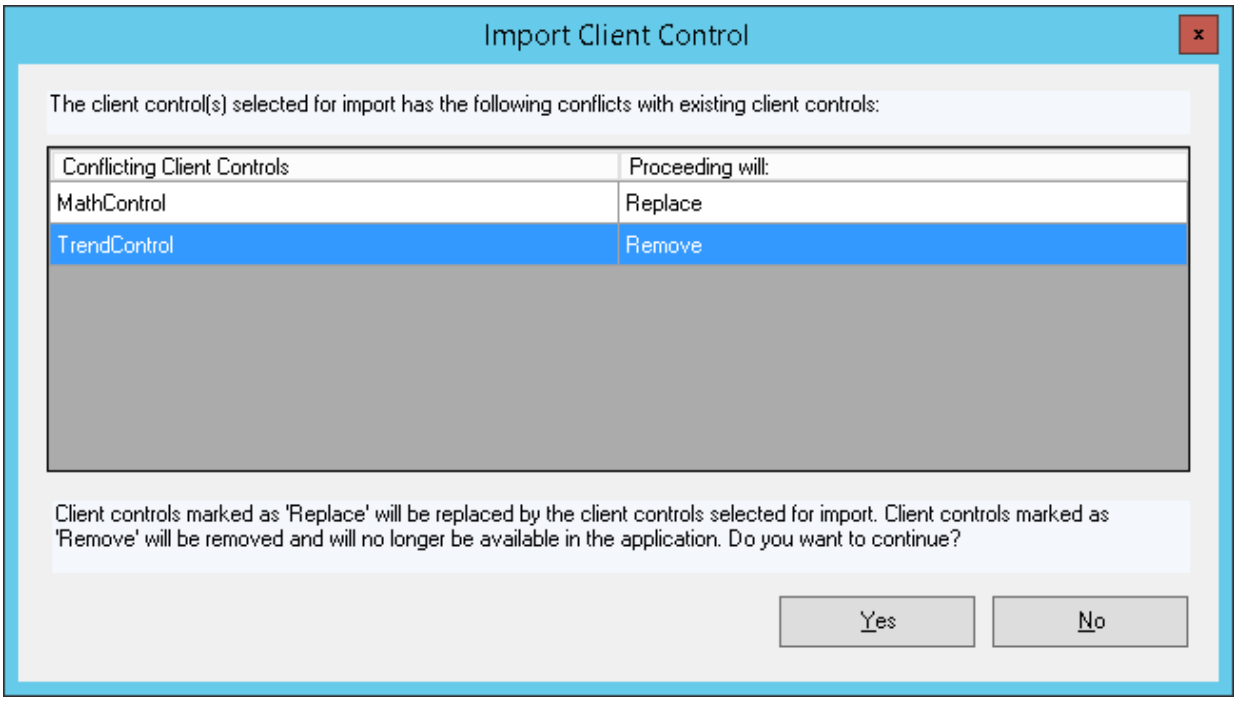


The existing client control .dll will be replaced, and the new control will now be available in the library.

If you see "Remove" in the "Proceeding will" column, it means there are controls in the current library that are not in the importing library. Because the hosting .dll must be replaced to resolve the conflicting controls, any controls that are in the current.dll but not in the importing .dll will be removed upon proceeing with the import.

For example, importing a client control .dll containing the controls MathControl and DatabaseControl and the current library contains MathControl and TrendControl, TrendControl will be removed from the library upon import.

The Import Client Control dialog box will prompt you to acknowledge the removal:



The library will be replaced and TrendControl will be removed upon completion of the import.

Restart WindowMaker to update the controls in the Graphic Toolbox.

Note: If you have imported a newer version of client control already embedded in a symbol, restarting WindowMaker and refreshing the graphic thumbnail will not update the contents of the control. You must edit and save the symbol for the new client control to be reflected in the thumbnail.

Embedding Client Controls in Industrial Graphics

Client controls are embedded from the Industrial Graphic Toolbox. The Graphic Toolbox already contains several client controls. You can embed these existing controls into Industrial Graphics, or you can import custom controls and embed those.

To embed a client control into an Industrial graphic:

1. Open the application in WindowMaker that you intend to embed a custom client control.
2. Open the window containing the Industrial Graphic that you intend to embed a custom client control.
3. Select the Industrial Graphic.
4. From the menu bar, click the **Embed Industrial Graphic** icon.

Important: You cannot drag and drop the custom client control from the Industrial Graphic Toolbox onto the Industrial Graphic. You must always embed the custom client control.

5. Configure your custom client control as needed for the application.

Importing HTML5 Widgets

Widgets are small web components that can extend the functionality of a webpage or website. Custom-built websites can also incorporate widgets, by using open-source code or frameworks to provide certain functionality in whole or in part. A widget is a self-contained code block that slots into a website without changing any of its features. Widgets are most frequently used to provide on-screen user interface elements that ingrate with other platforms and data sources. A widget can be run on any web page on a website, with consistent placement and user interface. For example, social media, weather, RSS or podcast widgets.

By default, the following widgets are available under the Widgets folder in the Graphic Toolbox:

- Carousel
- Web Browser
- QR Code Scanner

You can import a widget for a standalone and managed application. The file format is Custom Widget Package (.cwp), which includes HTML5, CSS, and Javascript files.

Importing HTML Widgets

1. Launch WindowMaker.
2. From the **File** menu, click **Import**, then click **HTML5 Widget**.
3. Select a .cwp file.
4. Click **OK**.

   The widget will appear in the toolbox.

After importing the widgets

1. Create a Symbol.
2. Edit the symbol and embed the widget.
3. Set the properties under the Widget Properties section. Each widget will have its own set of properties.
4. Insert the widget on a window

The widget can now be viewed on WindowViewer and any web browser. Depending on the properties set in the design time you can manipulate the widget in runtime. Scripts using Custom Properties under 'Widget Properties' to modify widgets are not supported.

Carousel Widget

A carousel widget allows you to cycle through elements—images or slides of text—like a carousel, without any input. This widget can be used to display dashboards, alerts or alarm information on large monitors on the plant floor.

Properties

In addition to the standard graphics properties, you can also configure properties specific to the widget, under **Widget Properties**.

Name	Description	Default
Autoplay	If the Autoplay property is set to true, the carousel widget will automatically start on load. If it is set to false, the user must select the next item to start the carousel.	True
BackgroundColor	Sets a background color for the widget. Specify the color value in RGB, HTML Code (#FF0000) or valid HTML color name.	White
GraphicNames	A comma separated list of graphics the carousel will display in runtime.	Empty
Interval	The amount of time delay (in milliseconds) between automatically cycling an item.	5000
Keyboard	If the Keyboard property is set to true, the carousel will respond to keyboard inputs.	True
Loop	If the Loop property is set to true, the carousel will cycle through the graphics continuously, else it will stop after a single cycle.	True
Pause	If the Pause property is set to true, the carousel will pause the cycling of the graphics, when it detects the mouse hovering or a touch down event. The graphics will resume cycling when the mouse is moved away.	True

The carousel widget is based on the Bootstrap 4.0 Carousel component, for more information on bootstrap, go here: https://getbootstrap.com/docs/4.0/components/carousel/

Web Browser Widget

Using the web browser widget, users can display a web site in WindowViewer and the Web Client.

If Web Client is running in HTTPS, then only HTTPS URL page can be loaded. If Web Client is running in HTTP, then HTTP and HTTPS can both be loaded. If the policy of the web site blocks cross domain access, then this widget will not work. The URL need not be in double quotes, but must be a valid URL.

Properties

URL: The address of the website.

Limitations

- If no protocol is specified, by default the https protocol will be used.
- If the Web Client is configured to use the HTTPS protocol, then only the HTTPS URL page will be loaded. If the HTTP URL is used, the web browser widget will display a message "`Mixed Content: The page at 'https://localhost/intouchweb' was loaded over HTTPS, but requested an insecure frame 'http://**********'. this request has been blocked: the content must be served over HTTPS.`"

- The web browser widget will not function, if the web site policy blocks cross domain (cross origin) access. A link will be provided to open the web page in a separate tab.

QR Code Scanner

The QRCode_Scanner widget connects to a camera to scan for a QR code and returns the resulting string.

Properties

Property Name	Description	Default Value
QRCode	The resulting string of the scanned QR code. The default value is empty.	Empty
AutoStart	If set to true, the camera will start automatically.	True
AutoStop	If set to true, the camera will stop after scanning a QR Code.	True
Start	If set to true, the camera will start.	False
Stop	If set to true, the camera will stop	False
BackgroundColor	Sets the background color of the widget. Specify the color value in RGB, HTML Code (#FF0000) or valid HTML color name.	Black

Limitation

- The device must have a camera.
- Using the QR Code on a physical machine instead of a virtual machine is recommended.
- Access the web client using the secure URL (https://) when using the web client remotely.

Usage

You can configure a script to read the QR code and display a graphic based on the scanned value.

In RunTime, the QR Code Scanner widget will appear with a floating toolbar with the following buttons - AutoStart, AutoStop and StartStop

When the widget is loaded, the camera will start automatically if AutoStart is set to True. To leave the camera on, click **AutoStop**.

To manually start the camera, click **StartStop** and scan the QR Code.

The camera will stay on after you scan the QR code, allowing the user to scan additional QR codes. To stop the camera, click **StartStop**.

The floating toolbar will display the QRCode derived from the QR Code scanned by the camera.

The user can script an action based on the QRCode returned.

Importing Script Function Libraries to an InTouch Application

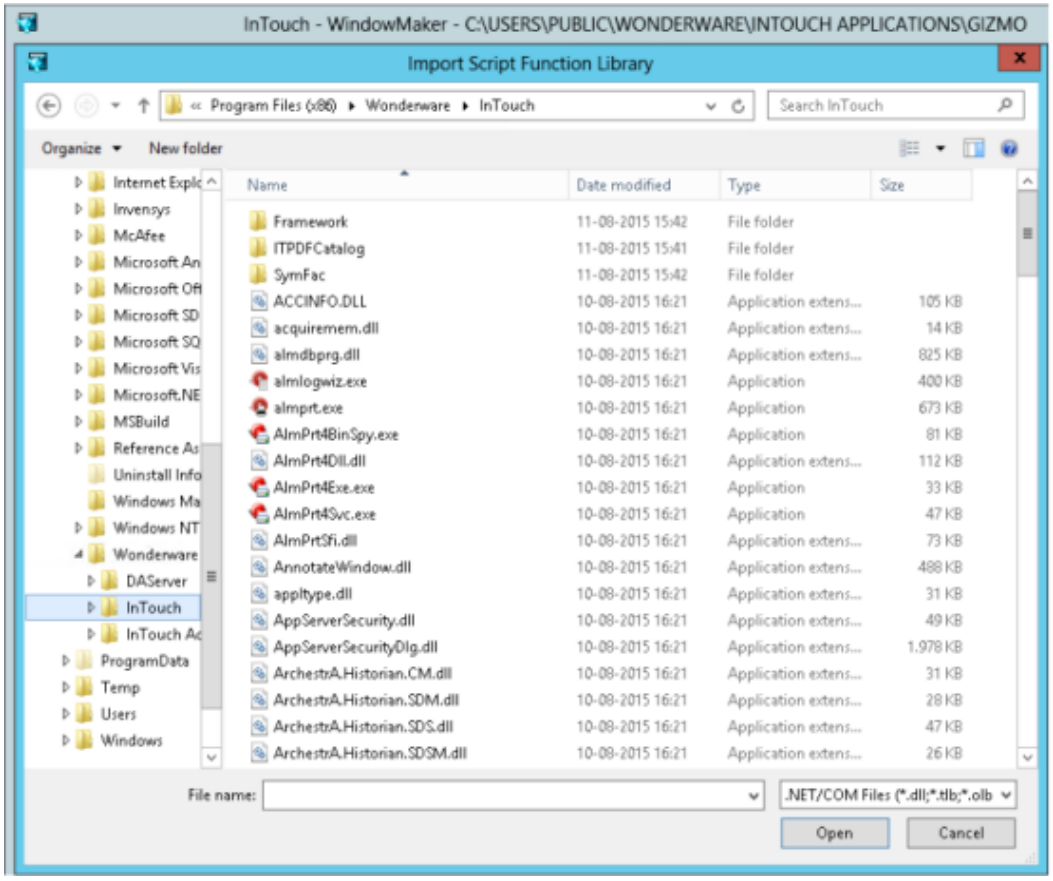
You can import script function libraries to an InTouch Application. Different types of script function libraries can be imported, including .NET (*.dll and other .NET file extensions), script library files (*.aaSLIB), and InTouch script extension files (*.wdf).

The script function library you imported to one application is automatically included when exporting the application to create another Application. The script function library also is available when publishing the application to which it was imported.

To import a script function library to an application

1. Open the InTouch application to which you want to import a script function library.
2. Click **File** on the WindowMaker main menu, then click **Import**, then click the **Script Function Library** option.

   The **Import Script Function Library** dialog opens.



3. Browse to the function library you want to import.
4. Select the file to import and click **Open** to start importing the script function library.

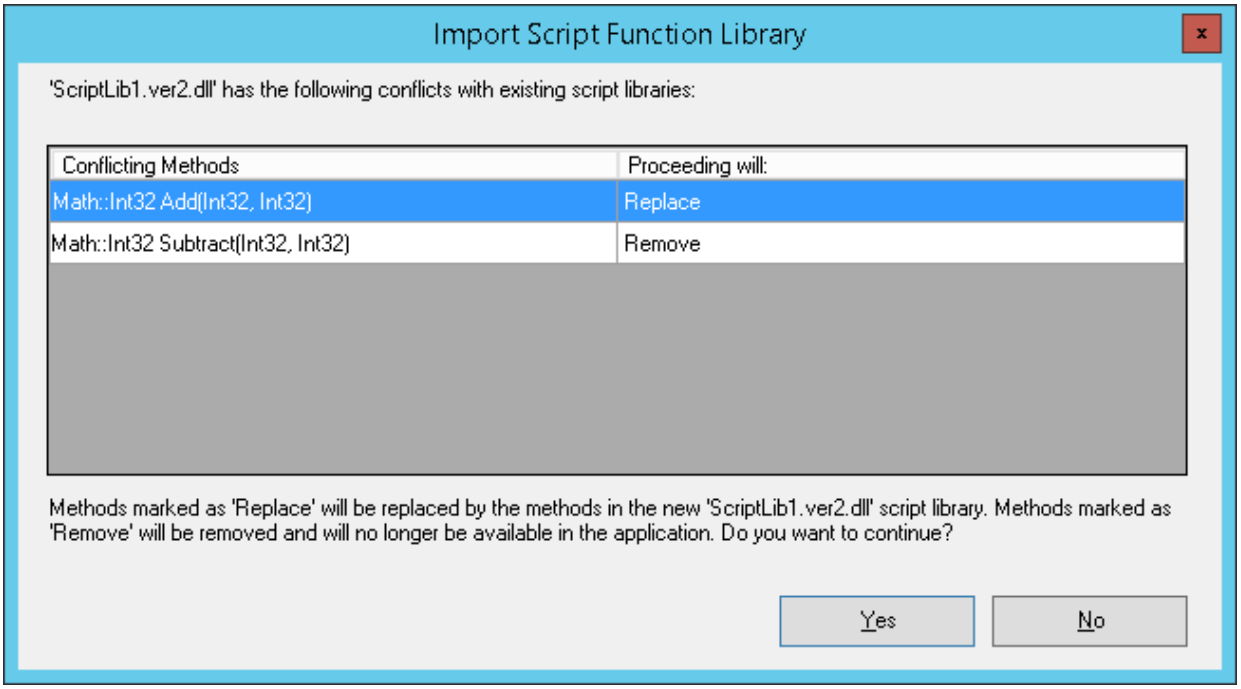
Note: No progress bar or progress information window appears during the import. An information window opens when the import successfully completes.

Resolving Imports of Conflicting Methods in .NET Script Libraries

When importing a .NET class script library into an application, the existing script library will be replaced by the importing library. Conflicting script methods will be detected at this time. Conflict detection is based on name space, class name, method name and parameter declaration.

Note: The version or filename or either .dll have no affect on method conflict detection.

Upon import, conflicting methods will be displayed in the Import Sript Function Library dialog box:



In this example, the `Math::Int32 Add(Int32, Int32)` exists in the current library and contains the same class, method name and parameters as a method in the importing library. It is marked "Replace" in the "Proceeding will" column. Proceeding with the import will replace the entire script library in the application with the importing library.

The `Math::132 Subtract(Int32, Int32)` is marked "Remove" because the importing library does not contain the `subtract` method. Script method conflict resolution requires replacing the entire script library, which will also result in the removal of this method if it is not in the importing library.

You cannot cancel the import of an individual method that would remove an existing method from the library, as in the example above. You must proceed with all the conflicting methods or cancel the entire import.

Important: Only .NET class library files can be detected as duplicates at time of import. .aaSLIB library and .wdf script extension files will not import if they conflict with methods in the existing library. In this case, no notification of the conflict will be given.

Configuring the Application Style Library for Applications

You can configure style libraries for graphics in a InTouch application. You can configure application styles for Quality and Status, Element Styles, and numeric Format Styles. Your configuration changes are saved to the application's repository.

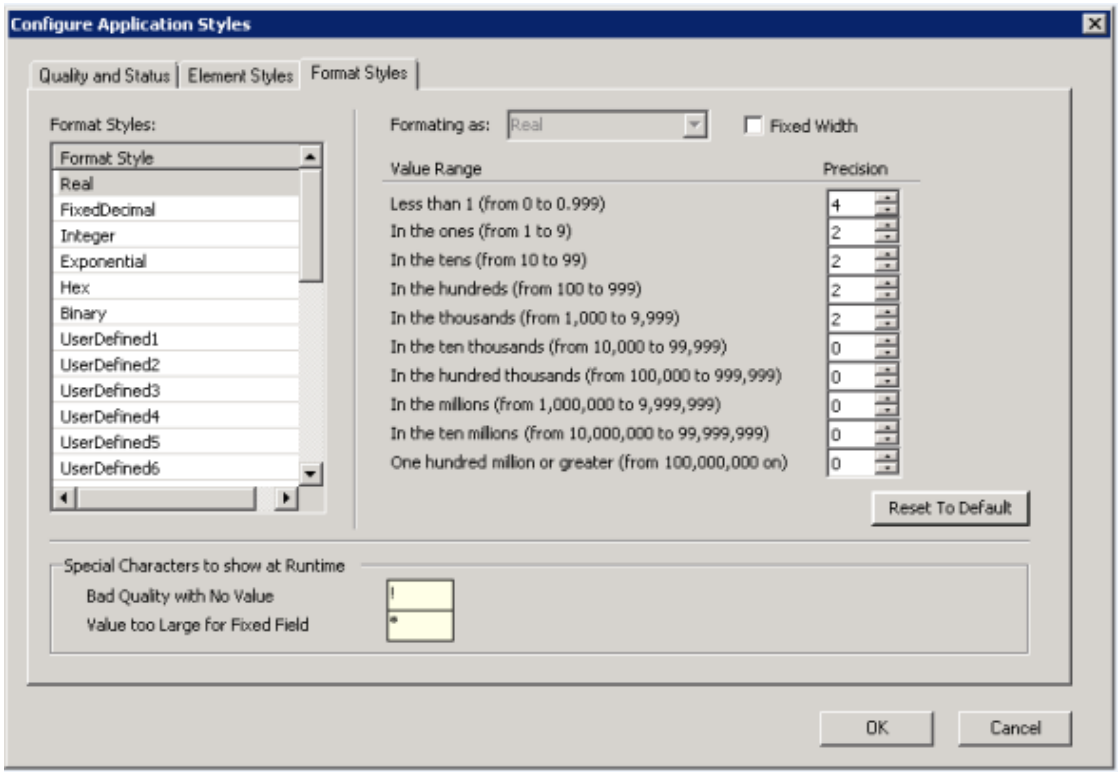
- Quality and Status indicators are graphic icons that represent the current quality of application data and the state of equipment shown by application symbols.
- Element Styles define a set of visual properties that determine the appearance of text, lines, graphic outlines, and interior fill shown in Industrial graphics.
- Format Styles provide options to individually configure application-wide styles for common types of numbers used in industrial graphics.

Important: This section describes the workflow within WindowMaker to access a application's style libraries. For more information about editing application styles, see WindowMaker online help or the *Industrial Graphic Editor User Guide*.

To configure the Application Style Library

1. Open an application in WindowMaker.
2. On the **Special** menu, click **Configure**, and then click **Application Style Library**.

   The **Configure Application Styles** dialog box appears with tabs to configure quality and status indicators, graphic Element styles, and number format styles.



3. Select a tab for the application style you want to edit.

Note: WindowViewer can run only one application at a time. If a platform is deployed on a local node, the configured styles of the Galaxy will take precedence over any configured styles in any other standalone or managed applications.

Exporting and Importing the Application Style Library

You can export an Application Style Library from an application and then import it to another application. The settings for quality, Element Styles, and numeric formats are exported to an XML file.

To export an Application Style Library from an Application

1. Open WindowMaker.
2. From the **File** menu, select **Expor**t and then **Application Style Library**.

   The **Export Application Style Library** dialog box appears with fields to specify a file name.
3. Select the folder to place the exported XML file and the name for the file.
4. Click **Save**.

   A dialog box confirms that the Application Style Library was exported successfully.

To import an Application Style Library into an Application

1. Open WindowMaker.
2. From the **File** menu, select **Impor**t and then **Application Style Library**.

   The **Import Application Style Library** dialog box appears with fields to specify a file name.
3. Select the folder where the exported XML file is located and select it to show the name of the export file in the **File Name** field.
4. Click **Open**.

   A dialog box confirms that the Application Style Library was imported successfully.

Configuring Alarm Priority Mapping for Applications

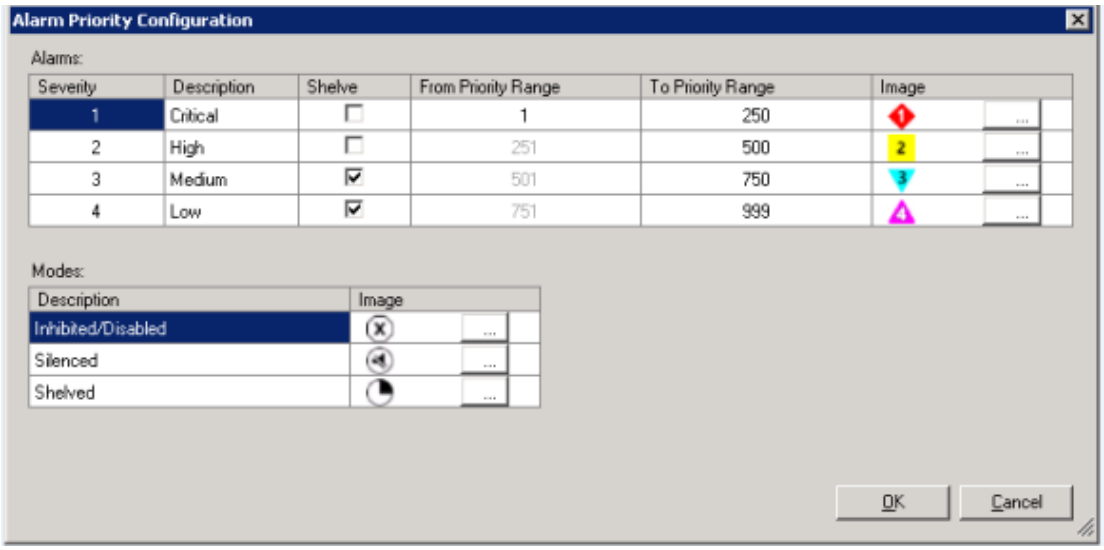
You can configure the alarm priority mapping of an InTouch application to set a priority range for each alarm severity level.

Important: This section describes the workflow within WindowMaker to map alarm priority ranges to alarm severities. While InTouch does not have built-in Alarm Severity management as does Application Server, users can make use of InTouch tags to implement Alarm Border animation. In this case, the priority to severity mapping in the dialog box is used only as a visual aid to associate priorities to alarm border colors and alarm indicator icons. For more information about configuring alarm priority mapping and alarm shelving, see WindowMaker online help or the *Industrial Graphic Editor User Guide*.

To configure Alarm Priority Mappings for Applications

1. Open an application in WindowMaker.
2. On the **Special** menu, click **Configure**, and then click **Alarm Priority Mapping**.

The **Alarm Priority Configuration** dialog box appears with fields to map a priority range to each alarm severity. The **Alarms Priority Configuration** dialog box also contains fields to enable alarm shelving based on alarm severity.



3. In the **From Priority** and **To Priority Range** fields, click and enter numbers from 1 to 999 to set the lower and upper boundaries of an alarm priority range for each alarm severity.

   Each priority range should be contiguous without overlap between priority ranges. Alarm severity 1 starts at priority 1 by default.

4. In the **Shelve** column, select or clear the check box to enable alarm shelving for each alarm severity.

5. Click **OK** to save your changes.

   Your changes are saved to the application's application folder.

Exporting Industrial Graphic Text Strings from an Application

If your application is intended to support run time language switching, you can export the text strings of its Industrial graphics to a dictionary file. You can then translate the strings within the dictionary file to other languages using a text editor, an XML editor, or a spreadsheet program like Microsoft Excel or the Language Assistant.

When you export the graphic text strings, you must specify an output folder for the dictionary file. A best practice is to create a separate folder for each dictionary file whose strings will be translated into another language.

All exported dictionary files follow a naming convention: <AppFolderName>AA_<LanguageID>.xml. For example, if an application folder name is PumpStation and the language being exported is French (Language ID = 1036), then the file name is PumpStationAA_1036.xml.

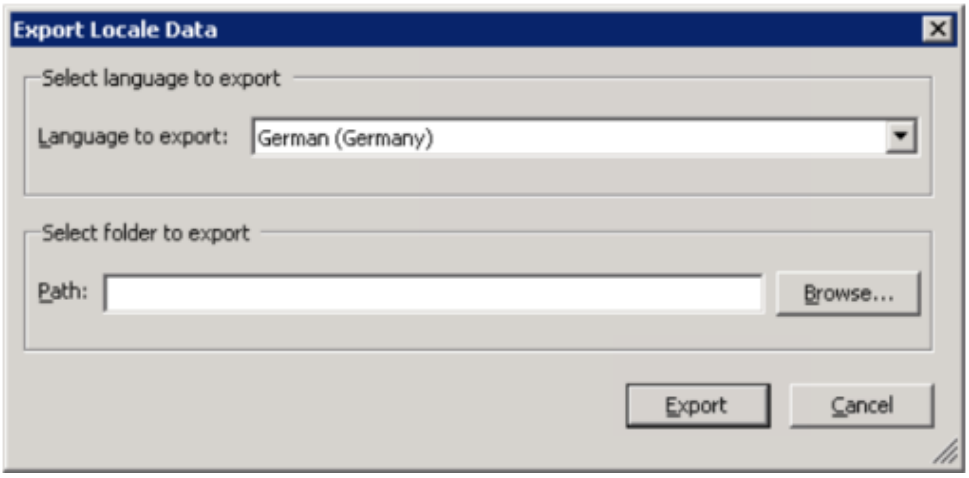
If you will be exporting language strings for different objects at different times, use separate target folders to prevent subsequent exports from overwriting the first export.

To export Industrial graphic text strings

1. Open the application in WindowMaker.

2. On the **Special** menu, click **Language**, and then select **Export Industrial Graphics Localization**.

   The **Export Locale Data** dialog box appears with fields to select the exported language and a folder to place the exported dictionary file.



3. Configure the symbol text strings to export.
   - In the **Languages to export** list, select the language dictionary to export. The default language is not listed.
   - In the **Path** field, type the folder to which you want to export the dictionary file. Click **Browse** to select an existing folder or create a new folder.
4. Click **Export**. A bar shows the progress of the export operation.

Importing Text Strings of Industrial Graphics to an Application

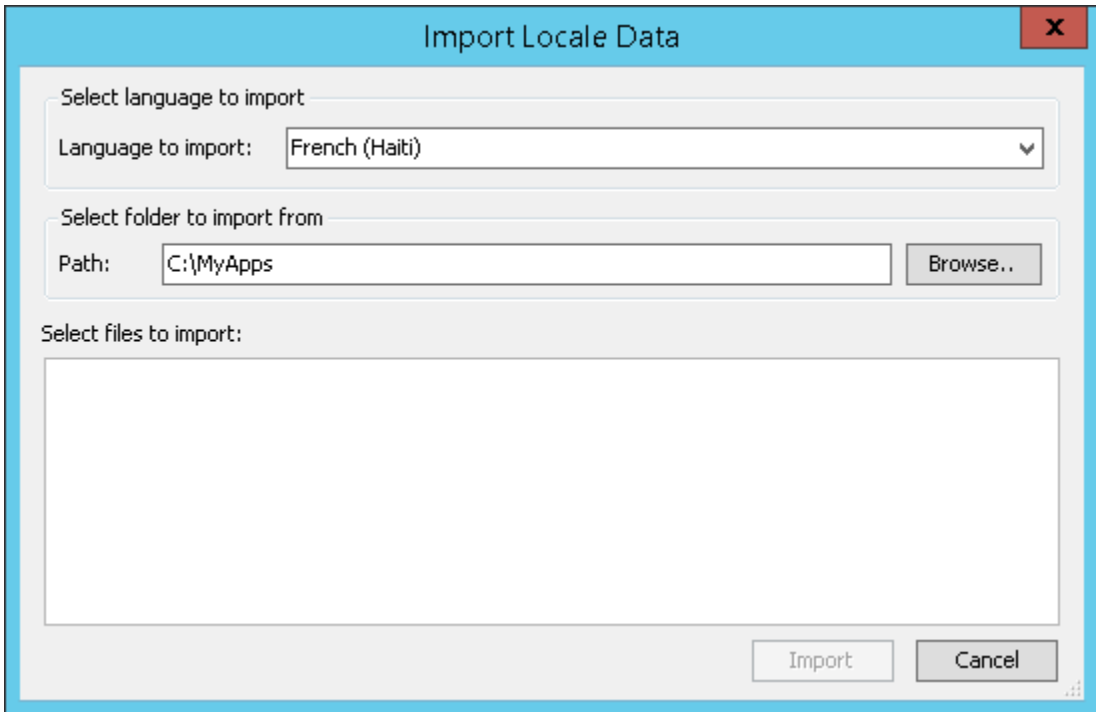
For symbol text, you must import the translated dictionary files for each language to enable run-time language switching for those languages. All dictionary files for a given language should be placed in the same folder.

You can import files for only one language at a time. When you import, you select the desired language and specify the dictionary files to import.

To import a translated dictionary file

1. Open the application in which you want to import Industrial graphic text.

2. On the **Special** menu, click **Language**, and then select **Import Industrial Graphics Localization**.



3. Configure the import settings.
   - In the **Language to import** list, select the language dictionary to import.
   - In the **Path** box, specify the folder that includes the dictionary file to import.
   - In the **Select files to Import** box, select the .xml files to import. Only files that include the current application folder name and the locale ID for the selected language are shown.
4. Click **Import**. The import progress is shown.
5. Click **Close**.

Exporting Localization Strings from a Symbol

If your application is intended to support run time language switching, you can export the text strings of one or more symbols selected from the Industrial Graphic Toolbox. You can then translate the exported strings within the file to other languages using a text editor, an XML editor, or a spreadsheet program like Microsoft Excel.

When you export the text strings from a symbol, you must specify an output folder. A best practice is to create a separate folder for each file whose strings will be translated into another language.

All exported localization files follow a naming convention: <AppFolderName>AA_<LanguageID>.xml. For example, if an application folder name is PumpStation1 and the language of the localization strings being exported is Mexican Spanish (Language ID = 2058), then the file name is PumpStation1AA_2058.xml.

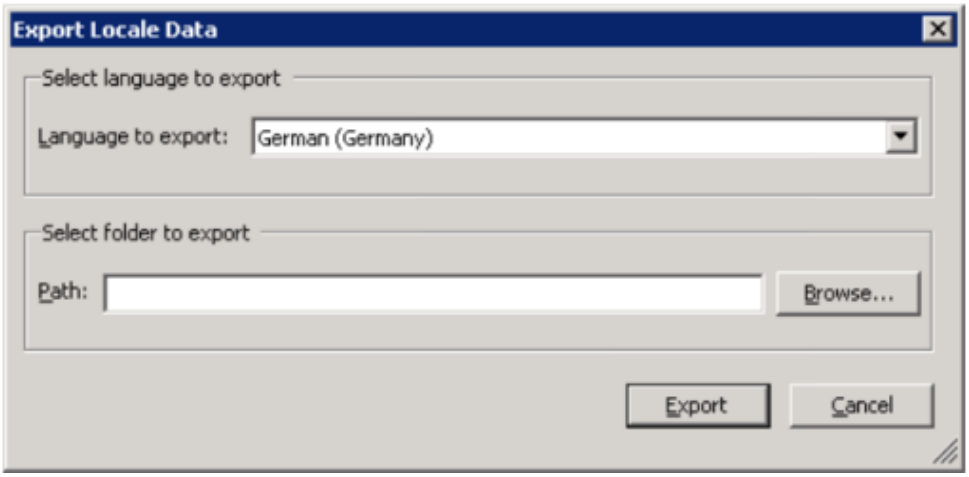
To export localization strings from a symbol

1. Open the application in WindowMaker.
2. Select the symbols from the Industrial Graphic Toolbox whose localization strings you want to export.

- Left-click on a symbol name to select a single symbol.
- Press the Ctrl key and left-click on symbol names to select two or more symbols.
- Left-click on a symbol name and then press the Shift key and left-click on another symbol name to select all symbols between the two selected symbols.

3. Right-click on a selected symbol to show the shortcut menu.
4. Select **Export**, then **Localization**, and finally **Selected Symbols(s)...**.

   The **Export Locale Data** dialog box appears.



5. Configure the symbol text strings to export.
   - In the **Languages to export** list, select the localization strings to export from the symbols. The default language is not listed.
   - In the **Path** field, type the folder to which you want to export the localization strings. Click **Browse** to select an existing folder or create a new folder.
6. Click **Export**. A bar shows the progress of the export operation.
7. Click **View Details** and verify the localization strings within each selected symbol were exported successfully.

Importing the Industrial Graphic Library

During application development you can import the Industrial Graphic Library and Situational Awareness Library into a standalone application, if

- The application was created with a blank template and does not contain the Industrial Graphic Library or Situational Awareness Library
- An older standalone application or modern application was migrated, but the libraries were not imported

To import the Industrial Graphic Library to an application:

- In the Industrial Graphic Toolbox, right-click the application name and select **Import Industrial Graphic Library**.

  The Import Industrial Graphics dialog appears. The Industrial Graphics Library is imported first followed by the Situational Awareness Library.

On completion, the Industrial Graphic Library and the Situational Awareness Library appear in the Industrial Graphic Toolbox.

Chapter 9

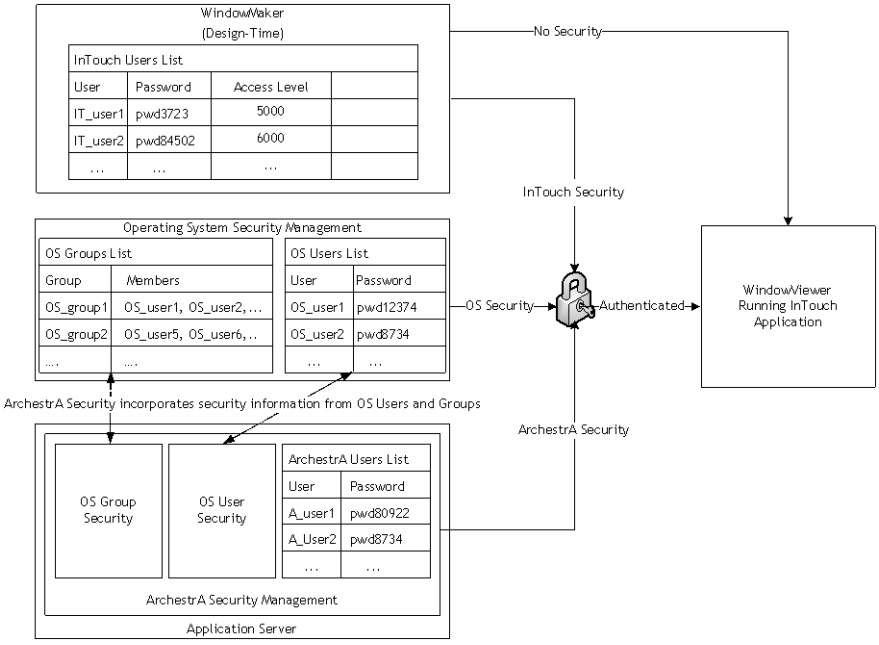
Securing InTouch

About Securing InTouch

You can protect your InTouch applications using:

- Traditional InTouch-based security
- Operating system-based security
- ArchestrA-based security

The following figure shows the relationship between the three types of security.



InTouch Security Features

To protect your InTouch application while it is running, you can:

- Set an inactivity time-out period
- Lock keys
- Hide menus

Configuring an Inactivity Time-Out

You can configure WindowViewer to automatically log off an inactive operator from an InTouch application. An operator must log on again after being logged off for inactivity. Setting an automatic inactivity log off period prevents unauthorized access to your InTouch application when operators leave their workstations unattended.

A timer measures the period the operator has not interacted with the running InTouch application. The timer resets each time the operator uses a mouse or any other input device to enter data. If the timer expires, the user is automatically logged off.

Note: The inactivity timer does not reset for Active-X controls, and OLE Automation controls.

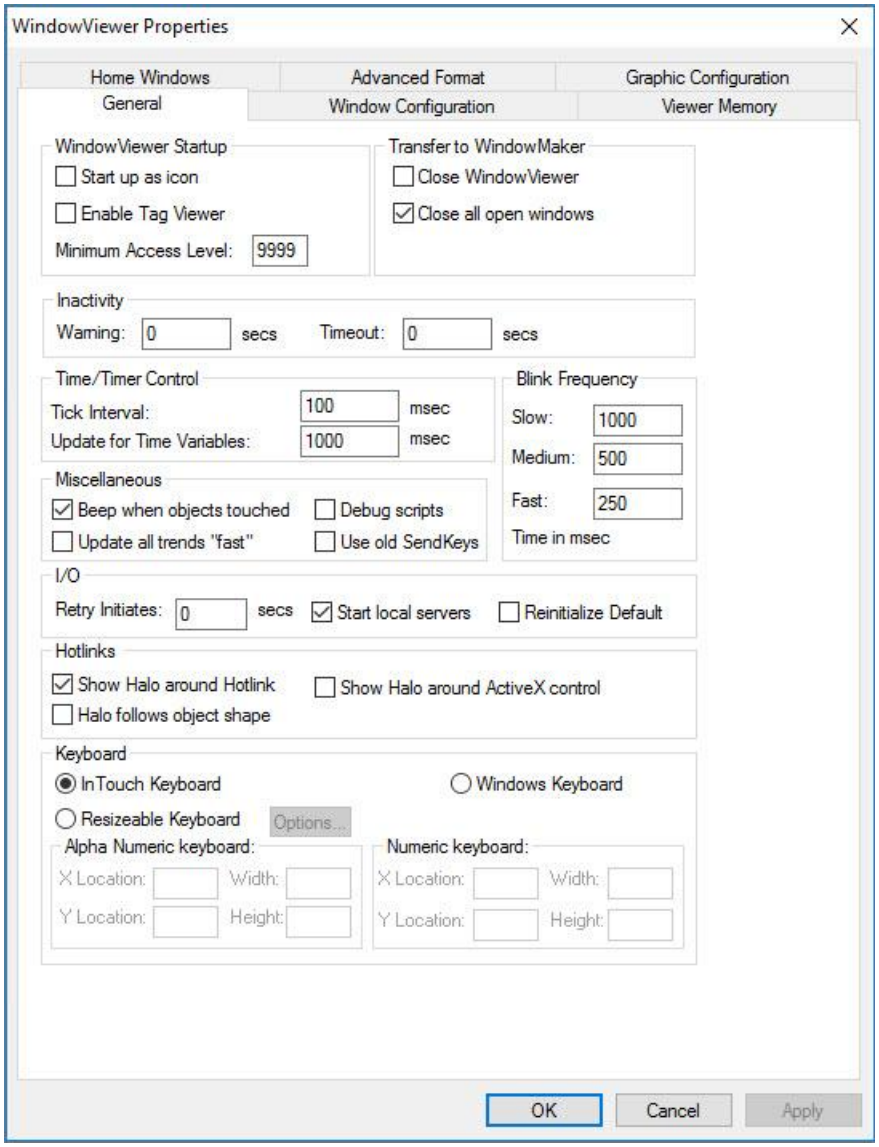
Automatically logging off an operator is a two-step process:

1. WindowViewer sets the $InactivityWarning system tag to 1 when the operator's inactivity period exceeds a specified warning period. You can use the $InactivityWarning tag in a condition QuickScript to show a window that warns the operator about the pending log off for inactivity. The operator stays logged on by responding before the specified time-out period occurs. When the operator takes some action, the $InactivityWarning tag and inactivity timer are reset to zero.
2. If the operator fails to respond after the inactivity warning, the $InactivityTimeout system tag is set to 1 when the time-out period has been reached. When $InactivityTimeout is 1, WindowViewer equates the logged on operator name to the reserved name None and sets the $AccessLevel security tag to 0. The user is automatically logged off.

You can use the time-out feature independently of the warning feature.

To configure an inactivity time-out

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



2. In the Inactivity area, configure the warning and time-out values. Do the following:
   - In the **Warning** box, type the number of seconds that can elapse before the $InactivityWarning tag is set to 1.
   - In the Timeout box, type the number of seconds that can elapse before the $InactivityTimeout tag is set to 1 and the user is automatically logged off.
3. Click **OK**.
4. To show a window named "Warning - Logoff Pending" after the inactivity warning time elapses, create a condition script with "$InactivityWarning" as the condition and the following script body:

```
Show "Logoff Pending";
```

5. To show a window named "Logged Off" after the inactivity timeout elapses, create a condition script with "$InactivityTimeout" as the condition and the following script body:
```
Show "Logged Off";
```

$InactivityTimeout System Tag

Indicates that the time configured for inactivity elapsed.

Category

security

Usage

$InactivityTimeout

Remarks

Set to 1 when the inactivity timer elapses. For more information on setting the log off time, see *Configuring an Inactivity Time-Out* on page 175.

Note: The inactivity timer does not reset for ActiveX controls, OLE and automation controls.

Data Type

Discrete (read only)

See Also

$InactivityWarning

Example(s)

The following example is an "on true" condition script:
```
If $InactivityTimeout == 1 THEN
    Show "Logged Off";
ENDIF
```

See Also

$InactivityWarning

$InactivityWarning System Tag

Indicates that the time configured for warning the user that log off is about to occur elapsed.

Category

security

Usage

$InactivityWarning

Remarks

Set to 1 when the inactivity warning time elapses. The inactivity timer is reset by mouse clicks or keyboard activity only. For more information on setting the log off warning, see *Configuring an Inactivity Time-Out* on page 175.

Note: The inactivity timer does not reset for ActiveX controls, OLE automation controls, and SPC wizards.

Data Type

Discrete (read only)

Example(s)

The following example is an "on true" condition script.

```
If $InactivityWarning == 1 THEN
    Show "Logoff Pending";
ENDIF;
```

See Also

$InactivityTimeOut

Locking System Keys

You can restrict operator access to standard Windows functions by disabling system keys on the computer running an InTouch application. For example, you can prevent an operator from using the Windows CTRL+ALT+DEL key combination to show the **Task Manager** dialog box. Disabling system keys prevents operators from switching from the InTouch HMI to another Windows application.

WindowViewer has key filter options that set the default state of system keys when an InTouch application starts. A key filter disables a system key when it is active.

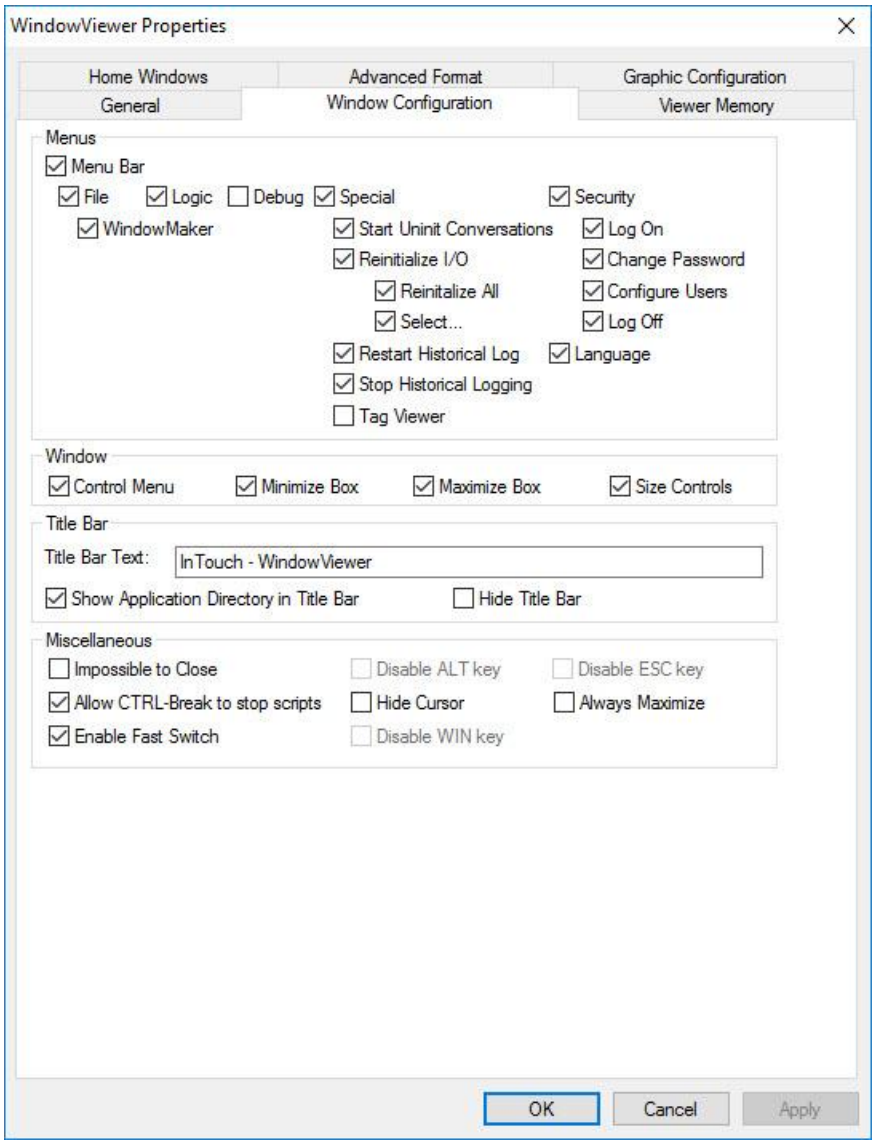
Disable system keys based on what tasks you expect your various InTouch users to complete. Most function keys should be disabled for operators. Administrators still need function keys for their InTouch tasks.

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.

To enable key filters

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

2. Click the **Window Configuration** tab.



3. In the **Miscellaneous** area, disable WindowViewer system keys. Do the following:
   - Clear the **Enable Fast Switch** check box to remove the **Development** button from WindowViewer that switches the user to WindowMaker.
   - Select the **Disable ALT key** check box to disable the ALT key on the computer running the InTouch application.
   - Select the **Disable WIN key** check box to disable the WIN key on the computer running the InTouch application.
   - Select the **Disable ESC key** check box to disable the ESC key on the computer running the InTouch application.
4. Click **OK**.
5. Write a script that runs when WindowViewer starts running the InTouch application.

The script should include statements to dynamically lock or unlock key based on the access level of the person who logged on to WindowViewer.

Include the EnableDisableKeys() function within the script to enable/disable the ALT, ESC, and WIN keys. The EnableDisableKeys() function enables or disables system keys based on the discrete values of its arguments:

```
EnableDisableKeys(AltKey,EscKey,WinKey);
```

An argument value of 1 enables the key filter to disable the key.

EnableDisableKeys() Function

Enables/disables key filters for the Alt, Escape, and Windows keys.

Category

View

Syntax

```
EnableDisableKeys(AltKey, EscKey, WinKey);
```

Parameters

AltKey
Integer to enable or disable key filters for the Alt key:

1 = enable filter (disable Alt key)

0 = disable filter (enable Alt key)

EscKey
Integer to enable or disable key filters for the Escape key:

1 = enable filter (disable Esc key)

0 = disable filter (enable Esc key)

WinKey
Integer to enable or disable key filters for the Windows key:

1 = enable filter (disable Win key)

0 = disable filter (enable Win key)

Remarks

Disabling the Alt key also disables the Win+L key combination (for locking the Windows desktop). Win+L is the shortcut for another combination of keys that involves the Alt key. Thus, disabling the Alt key also disables the shortcut for locking the Windows desktop.

Disabling the Esc key disables it for all actions.

Example(s)

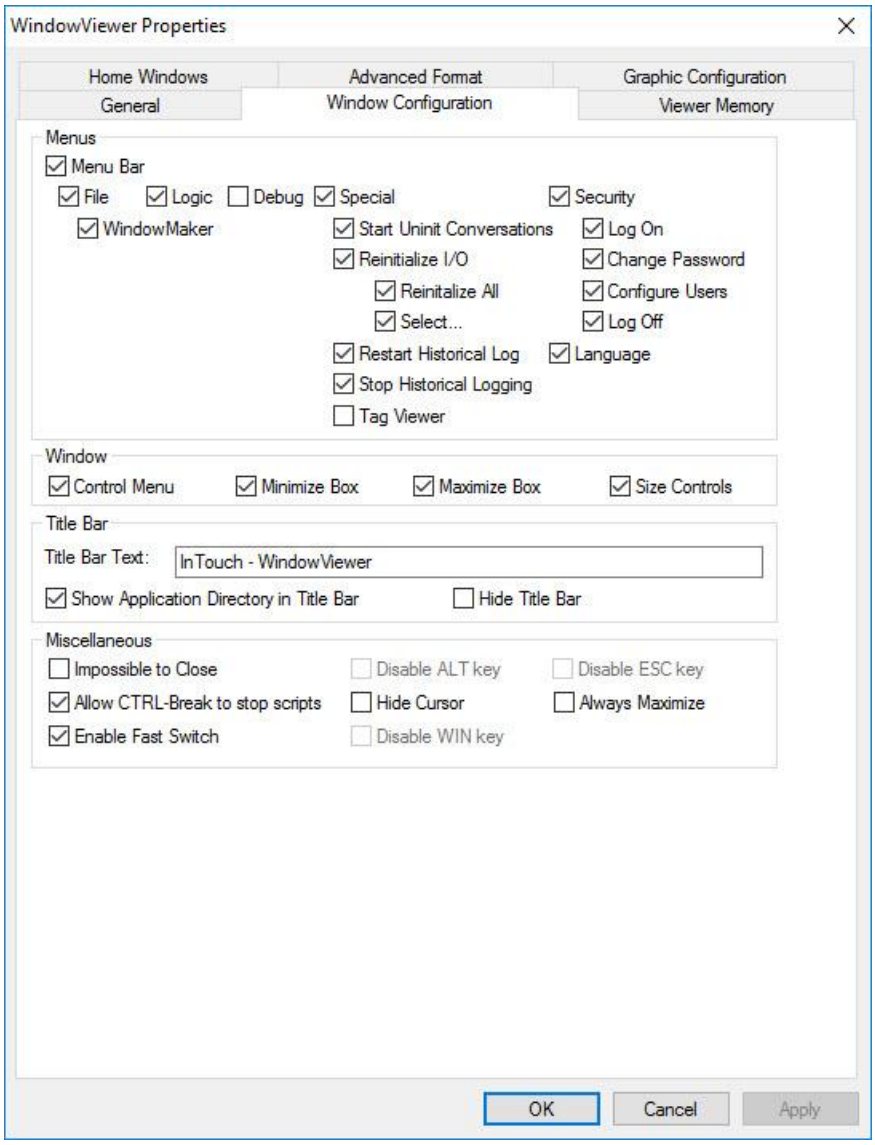
```
EnableDisableKeys(0,0,0); // enable all three keys
EnableDisableKeys(1,1,1); // disable all three keys
EnableDisableKeys(0,0,1); // enable Alt and Escape keys, disable Windows key.
```

Hiding Menu Items at Run Time

You restrict operator access to WindowViewer menus and commands by hiding them while an InTouch application is running.

To hide menu items at run time

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



3. In the **Menus** area, select the WindowViewer menus and commands that you want to be visible to an operator. Do the following:
   - Clear the WindowMaker check box to make the WindowMaker command unavailable from the WindowViewer **File** menu. Clearing this option does not affect the fast switch to WindowMaker.

- Clear the Logic check box to hide the WindowViewer **Logic** menu that contains commands to start and stop QuickScripts.

Note: You can use the $LogicRunning system tag to enable the operator to start and stop all QuickScripts.

If you select the **Allow CTRL-Break to stop scripts** option, the operator can stop all QuickScripts from running regardless of whether the **Logic** menu appears or not.

Currently executing asynchronous QuickFunctions cannot be stopped. However, you can prevent operators from starting new asynchronous QuickFunctions.

- Select the Debug check box if you are testing your application. Otherwise, clear the **Debug** check box to hide the **Debug** menu during run time.
- Clear the **Special** menu items to prevent operators from stopping ongoing InTouch functions like logging and I/O connections.
- Clear the **Security** check box to prevent operators from changing security related options.

4. In the **Window** area, select the window controls that you want to make available to an operator from WindowViewer. These options affect the window that is running the InTouch application. Do the following:
   - Clear the **Control Menu** check box to hide the controls that close, minimize, maximize, and resize the window.
   - Clear the **Minimize Box** check box to prevent an operator from minimizing the window.
   - Clear the **Maximize Box** check box to prevent an operator from maximizing the window.
   - Clear the **Size Controls** check box to prevent an operator from resizing the window.
5. In the **Title** bar area, configure the title bar of the window running the InTouch application. Do the following:
   - In the **Title Bar Text** box, type a title to be shown in the WindowViewer title bar.
   - Select the **Show Application Directory** check box to include the path to the InTouch application's folder in the title bar.
   - Select the **Hide Title Bar** check box to hide the window's title bar.
6. In the **Miscellaneous** area, do the following:
   - Select the **Impossible to Close** check box to prevent an operator from closing the WindowViewer window running the InTouch application. Selecting this option disables the window's **Close** button.

     If you want to hide the **Close** button, clear the **Control Menu** check box in the **Window** area.
   - Clear the **Allow CTRL-Break to stop scripts** check box to disable the CTRL + BREAK key combination that enables operators to stop QuickScripts.

Note: Currently executing asynchronous QuickFunctions cannot be stopped. However, you can prevent new asynchronous QuickFunctions from executing.

- Select the Hide Cursor check box to hide the mouse pointer during run time. This is useful if you are designing the application for a touch-screen.
- Select the Always Maximize check box to keep the window running the InTouch application fully maximized on the operator's screen.

7. Click **OK**.

8. Restart WindowViewer to apply your changes.

Authentication and Authorization Based 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 are granted to an authenticated user.

Comparing Authentication and Authorization

Authentication is the process of verifying the identity of the user. Typically, operators enter a user name and password to authenticate themselves before using an InTouch application. All three types of security verify the user's credentials during the logon process as part of the authentication process.

Authorization is the process of determining if an authenticated user has access to the requested resources. Typically, access to InTouch functions is granted based upon the user's membership in a group or assigned access level.

Different Authentication Security Modes

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.

Using InTouch-Based Security

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.

When you set the InTouch Security Type to InTouch, a pop-up dialog appears, suggesting to switch to OS security mode. The dialog also appears on the launch of WindowMaker, if InTouch security is used for the application. You can choose to turn off the notification.

Security is based on operators authenticating themselves by entering a user name and password to log on to an InTouch application. You must assign user name, password, and access level for each operator.

When you create a new application, by default, the user name is set to `Administrator` with an access level of 9999, which allows access to all security commands. The default administrator password is `wonderware`. The maximum number of characters for a password is 29.

After you add a new user name to the security list and restart WindowMaker or WindowViewer, the default user name is automatically reset to None with an access level of 0, which prevents access to the Configure Users command in both WindowMaker and WindowViewer. However, the Administrator account and password remain and can still be used.

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.

For Standalone applications, only users with Administrator privileges are allowed to open and edit applications in InTouch WindowMaker. If a user without administrator privileges attempts to launch InTouch WindowMaker, an error dialog box appears, informing users that they need administrative privileges to proceed. Users without administrative privileges can launch WindowMaker via ArchestrA IDE for Managed applications.

Using Operating System-Based Security

An operating system-based authentication method inherits enforcement of some account policies from the Windows operating system, while other policies are enforced from the InTouch HMI. Password policies such as maximum and minimum password age and minimum password length are enforced by the operating system.

User names used during installation act as a part of the operating system. The Windows domain must be set up with the desired account policies to enforce these standards. The InTouch HMI enforces the inactivity time-out period.

In the operating system-based authentication method, user names can be chosen from the list of users associated with a Windows Network Domain or Workgroup. Each user name has an assigned access level that determines the user's authorization for a given activity. Because the operating system manages passwords internally, the InTouch HMI does not store passwords on the node hosting the application.

Operating system-based security uses the InTouch AddPermission() script function to define and maintain a list of users and their corresponding access levels. This list, created after the execution of the AddPermission() call, is written to disk. The file containing the authentication details of users is not copied to NAD client nodes.

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 shown), or you can create a custom log on window with touch-sensitive input objects that are linked to internal security tags.

The commands used to establish security on an application are located under **Security** on the **Special** menu in both WindowMaker and WindowViewer. The security commands are used to log on and off the application, change passwords, and to configure the list of valid user names, passwords, and access levels.

For example, you can control access to a window, the visibility of an object, and so on, by specifying the logged on operator's access level must be greater than 2000.

Using ArchestrA-based Security

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 Galaxy Repository node. For more information, see the Application Server documentation.

ArchestrA security enables you to easily define users and assign the operations they are allowed to perform. Define security permissions in terms of the operations the users can perform using automation objects. The basic approach consists of the following steps:

1. Define the security model.
2. Organize the automation objects according to the security model for protection.
3. Define the users according to the security model.

The system administrator defines the system users by creating corresponding user profiles. The system administrator then assigns one or more roles to each user by selecting from a list of user roles predefined in the security model.

If you are using InTouch with ArchestrA-based security, the maximum number of characters for a password is 31.

InTouchView users are normally authenticated by means of a password-based log-on.

Using Smart Cards for Authentication

A Smart Card is a pocket-sized card with embedded integrated circuits. The card has secure storage for data, including private keys and public key certificates. The card holder is authenticated through a Personal Identification Number (PIN) and can be authorized to access only particular data on the card.

You can configure an InTouch application to support Smart Cards for user authentication. Instead of the application requiring a username, password, and domain to be provided, the Smart Card certificate and associated PIN number can be used for authentication. You can also choose to log on with your name, password, and domain instead of the Smart Card.

Operations that require user authentication, such as logging on or secured/verified writes, can also take advantage of Smart Card authentication. For more information, see *Using Secured and Verified Writes* on page 187.

Setting up Smart Card Authentication

You must do the following to set up Smart Card authentication:

- Configure the InTouch application to use either InTouch OS or ArchestrA OS security. The ArchestrA security can be either user-based or group-based. You configure ArchestrA security using the ArchestrA IDE. For more information, see the ArchestrA IDE documentation.
- Join the WindowViewer computer to the correct domain for your network.
- Within WindowMaker, enable Smart Card authentication for the InTouch application. For more information, see *Enabling Smart Card Authentication in WindowMaker* on page 185.
- Configure the Smart Cards for the domain where you will use them.
- Install card drivers on the WindowViewer computer. Smart Card and their drivers are hardware-specific. For information on installing and setting up your Smart Card reader, refer to the documentation for your specific reader.
- Connect the Smart Card reader to the appropriate port of the WindowViewer computer. For instructions, see the documentation that comes with the Smart Card.
    - More than one Smart Card reader is required to perform verified write functions, which involve more than one user.
    - To use Smart Card with Terminal server and RDP clients, a Smart Card reader must be attached to the client systems to enable Smart Card authentication. To connect a Smart Card reader to a Terminal Server using RDP, you need to make sure that the RDP client connection settings have the **Smart Card** option enabled under **Local Devices and Resources**.

Enabling Smart Card Authentication in WindowMaker

You must enable Smart Card authentication in WindowMaker before you can use the Smart Card reader for authentication.

To configure the Smart Card Option

1. Open WindowMaker.
2. On the Special menu, point to Security, point to Select Security Type, and then click ArchestrA or OS.

Note: If you click **ArchestrA**, be sure that you have configured ArchestrA OS security (OS user-based or OS group-based) using the ArchestrA IDE.

3. On the **Special** menu, click **Smart Card Authentication** so that a check mark appears. By default, this is not checked.

Logging on with Your Smart Card

You can use a smart card to log on to InTouch WindowViewer. You must have an application with smart card authentication enabled to use it to log on to the InTouch application.

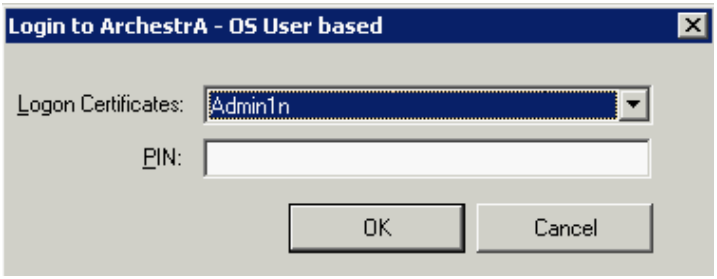
Your smart card must contain at least one certificate that is configured in your domain. A smart card reader must be attached to the computer running WindowViewer. You will be required to enter the PIN of the smart card you are using.

If a smart card is not detected in the reader when you try to log on, you are prompted to authenticate using your user name and password instead.

You can use smart card for authentication for secured and verified data writes. For more information, see *Using Secured and Verified Writes* on page 187.

To log on with your smart card

1. Run WindowViewer.
2. Insert your smart card if not already inserted.
3. On the **Special** menu, point to **Security**, and then click **Log On**. The Login dialog box appears.



If you have inserted your smart card, your log on certificate—the domain and the user name—is shown in the dialog box.

The smart card log on dialog box also appears if the LogonCurrentUser() or PostLogonDialog() functions execute from scripts in WindowViewer. These functions are available only in InTouch scripting, not in ArchestrA client scripting.

4. In the **PIN** box, enter the PIN for the smart card being used

   If a smart card is not available, the system will prompt you to log on with your user ID and password.

5. Click **OK**. You are logged on to WindowViewer.

Note: After you log on as a smart card user, you must keep the card in the smart card reader. If you remove it, the system logs you off.

Using Secured and Verified Writes

You can configure an InTouch application so that operators can write data to Galaxy attributes that are configured with certain security classifications:

- A "secured write" classification requires the run-time operator to enter his or her credentials to complete the write operation.
- A "verified write" classification requires two signatures. An operator can write data if the appropriate credentials are provided, but authorization is also required by an additional verifier to complete the write operation.

Secured and verified writes require the following:

- Security must be enabled for the Galaxy.
- ArchestrA security must be enabled for the InTouch application.
- Run-time operators must have the appropriate operational permissions configured within the Galaxy:
  - An operator must have the "Can Modify Operate Attributes" operational permission to perform either a Secured Write or a Verified Write.
  - A verifier must have the "Can Verify Writes" operational permission to confirm the verified write.

Regardless of who is currently logged on as the run-time user for the InTouch application, Secured or Verified Writes always require user authentication. You can modify attributes configured with Secured or Verified Write security classification even if you are not the logged-on user. This does not affect the session of the logged-on user.

Important: For Galaxies that have security enabled and are migrated to Application Server version 3.5, the "Can Modify Operate Attributes" operational permission setting will be copied to the "Can Verify Writes" attribute. Starting with Application Server 3.5, Galaxies have the "Can Verify Writes" operational permission setting disabled by default.

Within InTouch Tag Viewer, a run-time user can only write to an indirect tag with a reference to an ArchestrA attribute.

You can use smart cards for authentication for secured and verified data writes. For more information, see *Using Secured and Verified Writes* on page 187.

Performing a Secured Write

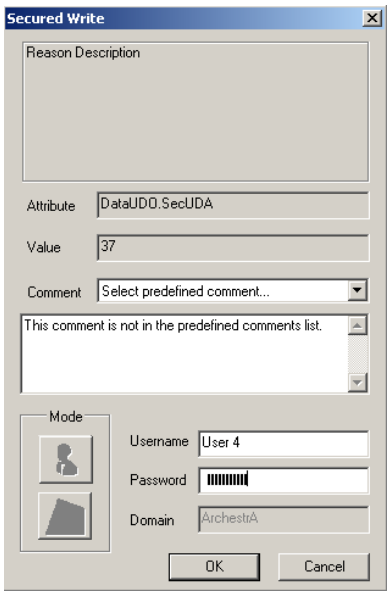
If you attempt to modify the value of an ArchestrA Galaxy attribute that has been configured with the Secured Write security classification, you must authenticate yourself using either a valid security account (domain name, username, and password) or a smart card. The smart card option is only available to you if a smart card reader is attached to the WindowViewer computer.

You must have the "Can Modify Operate Attributes" operational permission within the Galaxy to perform a secured write.

Your authentication for a secured write does not affect the session of the currently logged-on user. If you previously logged on with your smart card, you must re-authenticate yourself.

To perform a secured write

1. Attempt to modify the value of an attribute configured with the **Secured Write** security classification. The **Secured Write** dialog box appears. The **Mode** buttons are disabled if no smart card is available.



2. Add a comment for the write action by selecting from the predefined **Comment** list or by entering a comment in the **Comment** text box. The comment is limited to 200 characters.

   You can predefine a list of comments using the SignedWrite() script function, or you can enter a new comment in the **Comment** text box. The predefined comments list is only accessible when using the SignedWrite() script function.

3. If you are authenticating using a network user account, the user account options are shown.

   Do the following.

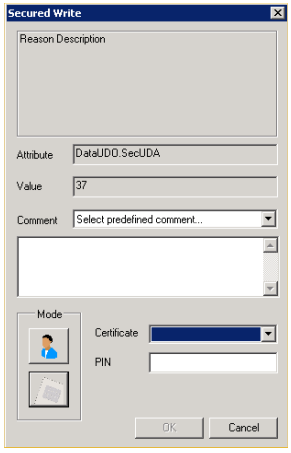
   a. In the **Username** box, type your user name. The name of the currently logged-on user is shown by default. If no user is currently logged on, the box is blank.

   b. In the **Password** box, type the password associated with the user name.

   c. In the **Domain** box, type the domain name.

   d. Click **OK**.

4. If you are authenticating using a smart card, the smart card options appear.



Do the following to authenticate using a smart card.

a. In the **Certificate** list, select your smart card certificate. The certificate list appears as domain_name/user_name. For a certificate to appear in the list, smart card must be currently inserted into a reader attached to the computer. The certificate of the currently logged-on user is shown by default. If you insert or remove a card while the **Secured Write** dialog box is open, the certificate list automatically updates.

b. In the **PIN** box, the PIN for the smart card being used.

c. Click **OK**.

When a smart card is present, if you want to authenticate using your name, password, and domain instead, click the Mode button. Go to Step 3.

Performing a Verified Write

If you attempt to modify the value for an ArchestrA Galaxy attribute that has been configured with the Verified Write security classification, you must authenticate yourself using either a valid security account (domain name, username, and password) or a smart card. The write must also be verified by another person.

- The operator must have the "Can Modify Operate Attributes" operational permission to perform the Verified Write.
- The verifier must also have the "Can Verify Writes" operational permission to confirm the Verified Write.

Your authentication for a verified write does not affect the session of the currently logged-on user.

The smart card option is only available if a smart card reader is attached to the WindowViewer computer. You can use smart cards for logging in as either an operator, a verifier, or both, but the operator and the verifier must be two different people. If you previously logged on with your smart card, you must re-authenticate yourself.

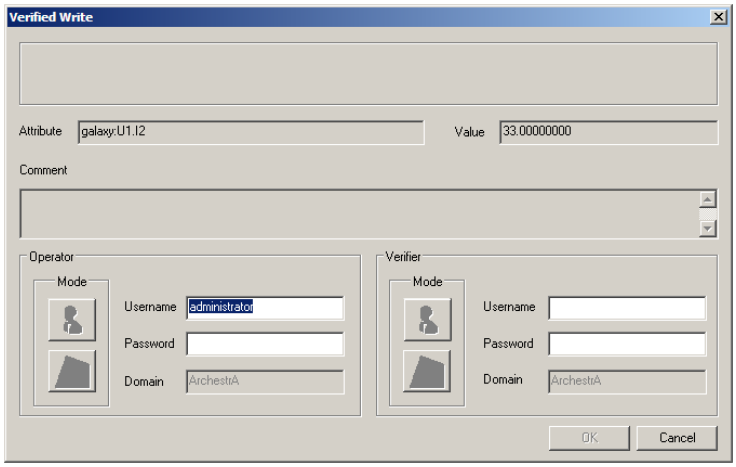
You have the following options:

- You can use two smart card readers and two smart cards.
- If you have only one smart card reader available, you can use one smart card reader with one smart card for the Operator or for the Verifier. When the Operator logs on using certificate number and PIN, the Verifier needs to log on using the username and password or vice versa.

- You can use username and password authentication for both the operator and the verifier.

To perform a verified write

1. Attempt to Modify the value of an attribute configured with the **Verified Write** security classification. The **Verified Write** dialog box appears. The **Mode** buttons are disabled if no smart card is available.



2. Add a comment for the write action by selecting from the predefined **Comment** list or by entering your own comment in the **Comment** box. The comment is limited to 200 characters.

   The predefined comments list is only accessible when using the SignedWrite() script function.

3. If you are authenticating using a network user account, the user account options are shown.

   Do the following.

   a. In the **Username** box, type your user name. The name of the currently logged-on user is shown by default. If no user is currently logged on, the box is blank.

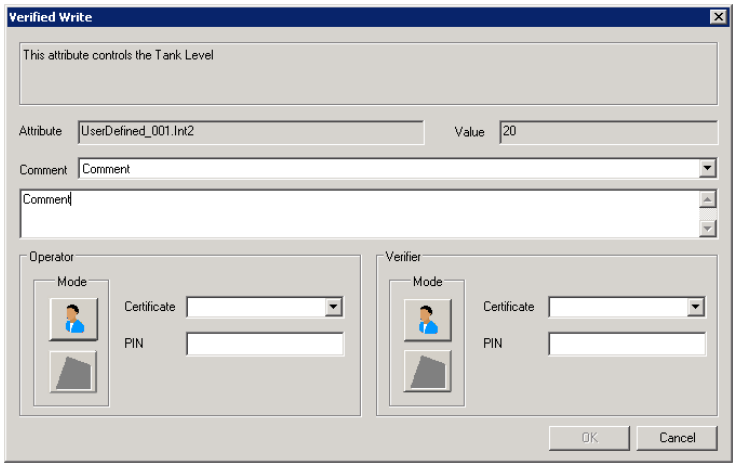
   b. In the **Password** box, type the password associated with the user name.

   c. In the **Domain** box, type the domain name.

   d. Click **OK**.

   e. To Authenticate using a smart card instead, click the Certificate button. Go to Step 4.

4. If you are authenticating using a smart card, the smart card options are shown.



Do the following to authenticate using a smart card.

a. In the **Certificate** list, select your smart card certificate. The certificate list appears as domain_name/user_name. For a certificate to appear in the list, smart card must be currently inserted into a reader attached to the computer. The certificate of the currently logged-on user is shown by default, if the user logged on using a smart card. If you insert or remove a card while the **Secured Write** dialog box is open, the certificate list automatically updates.

b. In the **PIN** box, type the PIN for the smart card being used.

c. Click **OK**.

d. When a smart card is present, if you want to authenticate using your name, password, and domain instead, click the Mode button. Go to step 3.

Customizing the Secured/Verified Write Dialog Box

You can use the SignedWrite() script function to configure the following in the **Secured Write** or **Verified Write** dialog box:

- Show a reason message
- Populate the predefined **Comment** list

- Allow editing in the **Comment** text box

For information about the SignedWrite() function and how to use it, including syntax, parameters, and detailed examples, see the Industrial Graphic Editor User Guide.

Working with the SignedWrite() Function at Run Time

It is possible to use the SignedWrite() function to directly assign a value to an attribute that requires a Secured or Verified Write signature.

When you configure a value with Secured or Verified Write security classifications and modify it, the **Secured or Verified Writes** dialog box appears. Depending on how the value is modified, the content appearing in the **Secured or Verified Writes** dialog box differs.

- If the value is modified using the SignedWrite() function, then the **Secured or Verified Writes** dialog box shows options based on the parameter settings from the function.
- If the value is modified by a user operation, then the reason message area shows the field attribute description, if there is one. If the attribute is not a field attribute or does not have a description, then the reason message area shows the description of the ApplicationObject to which the attribute belongs. The predefined **Comment** list is not available.

You can view the reason message in the **Secured Write** or **Verified Write** dialog box when you try to modify the value of the attribute in InTouch WIndowViewer. The dialog box displays the name of the attribute and the new value that is written to the attribute.

Note: The reason description and the predefined **Comment** list and box are shown in the **Secured Write** or **Verified Write** dialog box only in InTouch WindowViewer and not in Tag Viewer.

Managing Users and Setting Their Authorization Levels

To implement security for the group of users who need to use the InTouch HMI, you must:

- Assign user name and password authentication credentials to each user.
- Assign an InTouch authorization level (access level) to each user.

Configuring InTouch Security Authentication and Authorization

For each of your operators, you need to assign a user name, password, and access level.

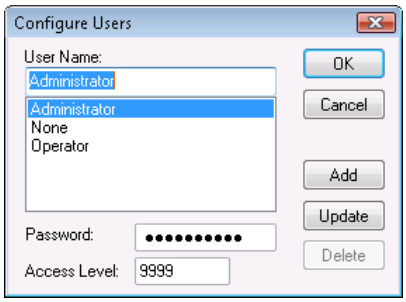
The **None** and **Administrator** names are reserved and only the password of the Administrator can be changed (the default is `wonderware`). After you configure user names for your application, change the Administrator password. The Administrator default access level (9999) is the highest and allows access to all InTouch functions including the **Configure Users** command.

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. When the operator clicks the button, the value of the $ConfigureUsers tag is set to 1 and the **Configure Users** dialog box appears. When the operator closes the dialog box, the system resets the value to 0. This is a system discrete tag intended for write operation only.

Note: The $ConfigureUsers tag only works if the security type is set to InTouch. It does not work for ArchestrA-based and operating system based security.

To configure security for operators of your application

1. On the WindowMaker **Special** menu, point to **Security**, and then click **Log On**.
2. Log on with your InTouch administrator account.
3. On the **Special** menu, point to **Security**, then click **Configure Users**. The **Configure Users** dialog box appears.



4. To add a security account, do the following:
   a. In the **User Name** box, type the name that you want to assign to the operator.
   b. In the **Password** box, type an operator password up to a maximum of 29 characters.
   c. In the **Access Level** box, type the operator's access level (lowest = 0 to highest = 9999).

d. Click **Add** to add the user name to the InTouch security list.

5. To change a user name, select the name, make any changes, and then click **Update**.
6. To delete a user name, select the name and then click **Delete**.
7. Click **OK**.

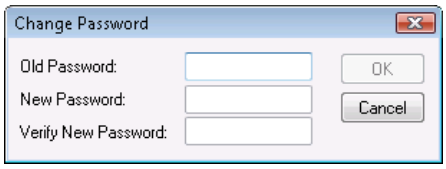
Changing an InTouch Operator Password at Run Time

Operators can change their passwords at run time using the **Special** menu in WindowViewer.

If you do not plan on showing the **Special** menu in WindowViewer, you can create a discrete button and link it to the $ChangePassword internal tag. When the value of $ChangePassword tag is set to 1, the **Change Password** dialog box appears. Operators can then change their passwords. When the operator closes the dialog box, the system resets the $ChangePassword value to 0. This is a system discrete tag intended for write operation only.

To change an operator password

1. On the **Special** menu, point to **Security** and then click **Change Password**. The **Change Password** dialog box appears.



2. Configure the password. Do the following:
   - In the **Old Password** box, type the old password.
   - In the **New Password** box, type the new password.
   - In the **Verify Password** box, type the new password again.
3. Click OK.

Setting Up Operating System-Based Authentication and Authorization

Operating system-based security authenticates InTouch users from a list of authorized Windows user groups. You create Windows user groups either on the local computer or an Active Directory server. You must associate Windows users to groups by adding them to specific groups. For more information on creating user groups, see your Windows operating system documentation.

You then 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 a user is ready to log on.

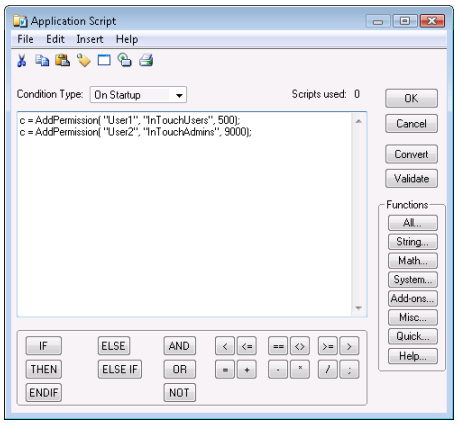
You typically specify operating system-based security immediately after you create an InTouch application.

After you configure the InTouch application to use the operating system authentication and internal InTouch authorization, the **Change Password, LogOn, Configure Users** and **LogOff** commands on the **Special...Security** menu are unavailable.

To set operating system-based security and configure access levels

1. On the WindowMaker **Special** menu, point to **Security**, point to **Select Security Type**,and then click OS.

2. On the **Special** menu, point to **Scripts**, and then click **Application Scripts**. The **Application Script** dialog box appears.



3. In the **Condition Type** list, click **On Startup**.
4. Use the AddPermission() function to specify the group names and corresponding access levels. The arguments for AddPermission() are operating system (or domain), group name, and access level.
5. Click **OK**.

Setting Up 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 system between users and the objects and functions of the Galaxy. This system 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. Configuration of the security system is done in the Integrated Development Environment (IDE) and applied to every object through its own editor.

After you configure the InTouch application to use ArchestrA authentication, the **Change Password**, **LogOn**, **Configure Users** and **LogOff** commands on the **Special...Security** menu are unavailable in WindowMaker.

To set ArchestrA-based security

1. Open a window in WindowMaker.
2. On the **Special** menu, point to **Security**, point to **Select Security Type,** and then click ArchestrA.

AddPermission() Function

Assigns a certain InTouch access level to a given user group on the local system or on the domain. When a user belonging to that group logs on to the InTouch HMI after the AddPermission() function is called, he or she receives the specified access level.

Category

security

Syntax

```
DiscreteTag=AddPermission( "Domain", "Group", AccessLevel);
```

Arguments

Domain
Name of the domain or local computer in which the group is located.

Group
Windows user group.

AccessLevel
InTouch access level that you want to associate with the given group.

Remarks

Valid for operating system security only. When this function is called, it checks for the presence of the specified group in the specified domain or workgroup. If successful, TRUE is returned, and the specified Access Level is associated with the group for subsequent user log ons. In all other cases, (that is, if an invalid value is specified for any of the arguments) FALSE is returned.

This function is typically configured to run on application startup. It does not affect users that are currently logged on. Only users that log on after AddPermission() is successfully called receive the access level associated with their group.

Examples

```
DiscreteTag=AddPermission( "corporate_hq", "InTouchAdmins", 9000);
DiscreteTag=AddPermission( "johns01", "InTouchUsers", 5000);
```

See Also

PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership()

ChangePassword() Function

Shows the **Change Password** dialog box, allowing the logged on operator to change his/her password.

Category

security

Syntax

```
[Result=]ChangePassword();
```

Return Value

Returns one of the following integer values:

0 = Cancel was pressed.

1 = OK was pressed.

Remarks

If the operator uses a touch screen, the operator can use the alphanumeric keyboard to enter the new password.

Example

The following script can be placed on a button or called from a condition script or data change script.

```
Errmsg=ChangePassword();
```

$AccessLevel System Tag

Defines the access level of the currently logged-in user .

Category

security

Usage

$AccessLevel

Remarks

The value for this tag is determined by the access level assigned to the currently logged-in user's security profile within the InTouch HMI. This profile can be accessed using the **Configure Users** menu in WindowViewer.

The actual numeric value of $AccessLevel has no meaning to WindowViewer, except that a value of 9000 or greater indicates administrative privileges and enables the **Security** menu within WindowViewer. Use the $AccessLevel system tag to further customize security within the system.

Data Type

Integer (read only)

Valid Values

0 through 9999

Example(s)

The following statement is used for the visibility link to make an object, such as a button, visible based on the logged on user's access level:

```
$AccessLevel >= 2000;
{Objects can have a "disable" link associated with them, with the expression based on
$AccessLevel.}
$AccessLevel < 5411;
IF $AccessLevel <=500 THEN
Show "Access Denied"; {popup window denying access}
ELSE
Show "Access Granted"; {popup window granting access}
ENDIF;
```

See Also

$Operator, $OperatorEntered, $PasswordEntered; $ConfigureUsers

$ChangePassword System Tag

Shows the **Change Password** dialog box.

Category

security

Usage

```
$ChangePassword
```

Remarks

Set this value to 1 to show the **Change Password** dialog box. The value of the $ChangePassword system tag resets to 0 when the dialog box closes. If you set this system tag to a value other than 1, the results are undefined.

Data Type

Discrete (write only)

Valid Values

1

Example(s)

You can create a discrete push button that opens the **Change Password** dialog box. Assign a single discrete push button link, with the Set option selected, to the push button. When the button is pressed, the $ChangePassword system tag is set to 1, and the **Change Password** dialog box opens.

See Also

$AccessLevel, $OperatorEntered, $PasswordEntered, $Operator, $ConfigureUsers

$ConfigureUsers System Tag

Shows the **Configure Users** dialog box.

Category

security

Usage

```
$ConfigureUsers
```

Remarks

This function only works with InTouch security.

Set the value to 1 to open **Configure Users** dialog box.

The value of this system tag resets to 0 when the dialog box closes. If you set this system tag to a value other than 1, the results are undefined.

The user must have an $AccessLevel of >9000 to show this dialog box.

Data Type

Discrete (write only)

Valid Values

1

Example(s)

You can create a discrete push button that opens the **Configure Users** dialog box. Assign a single discrete push button link, with the Set option selected, to the push button. When the button is pressed, the $ConfigureUsers system tag is set to 1, and the **Configure Users** dialog box opens.

See Also

$Operator, $OperatorEntered, $ChangePassword, $PasswordEntered, $AccessLevel

Logging On and Off

Logging on to and logging off from an InTouch application varies by the type of security used to protect an application.

Logging on to an InTouch-Secured Application

If the logon information is entered incorrectly or is invalid, a message indicates the log on attempt failed.

If the log on is successful, the $AccessLevel system tag is set to the predefined value associated with the user in the InTouch security user list.

Note: You can also show the **Log On** dialog box using the PostLogonDialog() function. For more information, see *PostLogonDialog() Function* on page 200.

To log on to an application

1. On the **Special** menu, point to **Security**, and then click **Log On**. The **Log On** dialog box appears.
2. In the **Name** box, type your user name.
3. In the **Password** box, type your password.
4. Click **OK**.

Logging On to an Operating System-Secured Application

When a user logs on to an InTouch application, a dialog box appears requiring the following:

- User name
- Password
- Domain or local computer name

The domain/user name combination is passed to the operating system to authenticate the user's credentials. An attempt is made to log on with or without enabling the operating system cache. If the user cannot be logged on without the cache (due to a network outage, for example), but the user was previously authenticated with the cache enabled, then the user's full name and access level is obtained from the local InTouch cache.

If all of the security checks are cleared successfully, the user is considered to be logged on to the InTouch HMI and the relevant data structures (for example, $Operator) are updated. Otherwise, an error message is shown.

If the operator has never logged on successfully before and the domain is unavailable, the logon attempt fails. The InTouch HMI logs a system event to the error log.

If the password is expired, an error message is shown. After the operator clicks **OK**, the **Change Expired Password** dialog box appears, so that the operator can change the password and attempt to log on again with the new password.

Logging On to an ArchestrA-Secured Application

Users typically log on and log off from an ArchestrA-secured InTouch application by entering a valid user name and password.

If your InTouch application has been configured for the ArchestrA security "None", the log on credentials of the default user are used and the operator is not prompted to log on. The following procedure assumes your system has been configured for ArchestrA authentication modes, such as "Galaxy", "OS User based", or "OS Group based".

To log on

1. Start the ArchestrA-secured InTouch application. A log in dialog box appears.
2. Type a valid user name and password. If the system cannot authenticate you, you are prompted again to log on.

After the system authenticates your logon credentials, access to all future operations is granted based on your associated roles/permissions in the security model.

Logging Off from an InTouch Application

Operators log off from an InTouch application after completing their work. You can also configure an application to automatically log off an operator after a specified amount of time has elapsed without any activity by the operator. For more information, see *Configuring an Inactivity Time-Out* on page 175.

To log off from an application

- On the **Special** menu, point to **Security** and then click **Log Off**.

Creating a Custom Logon Window

If the **Special** menu is not shown in WindowViewer, you can create a custom logon window for operator to log on to the application.

To create a custom log on window

- Link the $OperatorEntered, $PasswordEntered and $OperatorDomainEntered system tags to user input objects or use them in a script to set the user name, password, and domain name. These tags are internal message type tags that are intended for write operation only.

The $OperatorDomainEntered tag is required only if the security mode is operating system-based. Otherwise, this tag is ignored. If the security mode is operating system-based and the $OperatorDomainEntered is null, it is treated as pointing to the local computer.

When a value is written to the $PasswordEntered system tag, a logon attempt occurs using the $OperatorEntered, $PasswordEntered, and $OperatorDomainEntered system tag values. No logon occurs if values are written to only the $OperatorEntered or $OperatorDomainEntered system tags.

If the entries are valid, the $AccessLevel and $Operator internal tags are set to their predefined values (configured in the security user list).

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. When the operator clicks the button, the value of the $ConfigureUsers tag is set to 1 and the **Configure Users** dialog box appears. When the operator closes the dialog box, the system resets the value to 0. (This is a system discrete tag intended for write operation only.)

Note: The $ConfigureUsers tag only works if the security type is set to InTouch. It does not work for ArchestrA-based security.

PostLogonDialog() Function

Shows the InTouch **Logon** dialog box and returns TRUE.

Category

security

Syntax

```
DiscreteTag=PostLogonDialog();
```

Examples

```
DiscreteTag=PostLogonDialog();
```

See Also

InvisibleVerifyCredentials(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission()

LogonCurrentUser() Function

Logs on to InTouch with a user account that is currently logged on to the Windows operating system.

- InTouch configured with OS security: the user is logged on to WindowViewer.
- InTouch configured with ArchestrA security: the user must be a member of ArchestrA OS user-based or OS group-based security.
- InTouch configured with ArchestrA OS user-based or OS group-based security and the user account is configured with smart card credentials: user is logged on using the smart card credentials. The user is logged off if the smart card is removed from the reader.

Category

security

Syntax

```
IntegerResult = LogonCurrentUser();
```

Return Value

Returns -1 and no change to the values assigned to $Operator, $OperatorName, $OperatorDomain, and $AccessLevel if the logon fails.

Remarks

This function is available only in InTouch scripting, not in ArchestrA client scripting.

Example

```
IntegerResult = LogonCurrentUser();
```

See Also

PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership(), AddPermission()

Logoff() Function

Logs the user off from an InTouch application.

Category

security (write only)

Syntax

```
DiscreteTag = LogOff();
```

Remarks

Logs off the currently logged on user and sets the current user status to the default none operator.

Example

```
DiscreteTag = LogOff();
```

See Also

PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership(), AddPermission()

AttemptInvisibleLogon() Function

The AttemptInvisibleLogon() function can be used in a script to log on a user to InTouch using the supplied credentials. The user is not required to enter a password or user ID.

Category

security

Syntax

```
 DiscreteTag=AttemptInvisibleLogon( "UserId", "Password", "Domain" );
```

Arguments

UserId
A valid user account name.

Password
Password of the user.

Domain
Name of the local computer, workgroup, or domain to which the user belongs. This argument applies only if the current security type is operating system-based.

Return Value

Returns TRUE if authentication is successful. Otherwise, it returns FALSE.

Remarks

An attempt is made to log on to the InTouch HMI using the supplied credentials.

- If the logon attempt succeeds, then TRUE is returned and the $OperatorDomain, $OperatorName, $AccessLevel, and $Operator system tags are updated accordingly.
- If the log on attempt fails, then FALSE is returned, and the currently logged on user (if any) continues to be the current user.

The *Domain* argument is only valid for operating system-based security. If ArchestrA security mode is in use and if ArchestrA security is in turn using operating system-based security, the *UserId* argument should contain the fully qualified user name with domain name or computer name.

Examples

When security is operating system-based:

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "Domain" );
```

When security is either InTouch-based or ArchestrA-based:

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "" );
```

See Also

PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), QueryGroupMembership(), AddPermission()

$OperatorEntered System Tag

Used to enter a valid user name.

Category

security

Usage

```
$OperatorEntered
```

Remarks

You can use this tagname to create a custom log-on window. You can link touch-sensitive input objects and/or QuickScripts to this tag to set the user name for the logon.

Note: When the $OperatorEntered system tag is valid, $AccessLevel and $Operator system tags are set to their pre-defined values.

Data Type

Message (write only)

See Also

$AccessLevel, $Operator, $PasswordEntered, $ChangePassword, $ConfigureUsers

$PasswordEntered System Tag

Used to enter a valid password.

Category

security

Usage

```
$PasswordEntered
```

Remarks

The $PasswordEntered system tag always reads as an empty string. Display links tied to this system tag are always blank. Because the tag always returns an empty string, data change scripts never trigger off of this tag. You can use this tagname to create a custom log-on window. You can link touch-sensitive input objects and/or scripts to this tag to set the password for the user.

Note: When the $PasswordEntered is valid, the $AccessLevel and $Operator system tags are set to their pre-defined values.

Data Type

Message (write only)

See Also

$AccessLevel, $Operator, $OperatorEntered, $ChangePassword, $ConfigureUsers

$OperatorDomainEntered System Tag

The domain name as entered by the operator.

Category

Security

Remarks

Whenever the $PasswordEntered tag changes, a log on is attempted without showing a dialog box. The log on attempt uses the $*Entered tags as input user name and the string value of $OperatorDomainEntered as the domain name (used only if the current mode is operating system-based security). If the mode is not operating system-based, this tag is ignored.

Data Type

String

Examples

```
$OperatorEntered == "john";
$OperatorDomainEntered == "Corporate_HQ";
$PasswordEntered == "password";
```

See Also

$Operator

Enabling and Disabling Functionality Based Upon Operator or Access Levels

After you implement security for your application, you can use the $AccessLevel and $Operator security tags on buttons, in animation link expressions, or in QuickScripts to control whether or not the logged on operator is allowed to perform specific application functions.

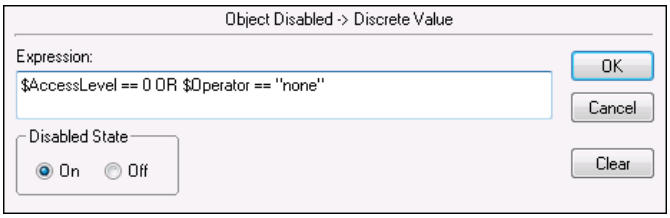
For example, to make an object become visible based on the access level of the logged on user, use the following statement in a visibility animation link expression:

```
$AccessLevel >= 2000;
```

Or, a script can be bounded by an IF statement:

```
IF $Operator == "DayShift" THEN
    Show "Control Panel Window";
    {and other lines that only execute for the DayShift Operator}
ENDIF;
```

You can also control an object's touch functionality based upon the value of an internal security tag by using the Disable animation link. For example:



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

InvisibleVerifyCredentials() Function

The InvisibleVerifyCredentials() function can be used in a synchronous QuickScript to verify the credentials of the given user without logging the user on to the InTouch HMI.

Category

security

Syntax

```
AnalogTag=InvisibleVerifyCredentials( "UserId", "Password", "Domain" );
```

Arguments

UserId
Windows operating system user account name that is part of local computer, workgroup, or domain.

Password
Password for the account.

Domain
The Windows domain for the account.

Remarks

If the supplied combination of user, password, and domain are valid then the corresponding access level associated with the user is returned as an integer. Otherwise, -1 is returned.

Note: The InvisibleVerifyCredentials() function must be run from a synchronous QuickScript. The function always returns -1 if run from an asynchronous QuickScript.

This function does not change the currently logged on user. The Domain argument is only valid for operating system-based security. If ArchestrA security is in use and if ArchestrA security is in turn using operating system-based security, the UserId argument should contain the fully qualified user name with domain name or computer name.

Example

```
AnalogTag=InvisibleVerifyCredentials( "john", "Password", "corporate_hq" );
```

See Also

PostLogonDialog(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission()

Retrieving Information About the Currently Logged-on Operator

Auditing is a primary function of any security system. You can use a set of security system tags to identify the users who logged on to an InTouch application, the domain from which the user logged on, and when the attempt was made.

GetAccountStatus() Function

Returns the number of days until the user's password expires.

Category

security

Syntax

```
Result=GetAccountStatus(Domain, UserID);
```

Arguments

Domain
Name of the domain or local computer in which the user account is located.

UserID
Windows user account name that is part of the local computer, workgroup, or domain.

Return Value

This function also returns the following values:

Result	Description
-1	Account password expired
-2	Account password never expires

Result	Description
-3	Account locked out
-4	Account disabled
-5	Account info failed

Remarks

Use this script function with operating system-based security. Do not use this function with the ArchestrA security mode.

If the GetAccountStatus() function is used with ArchestrA security, the script attempts to retrieve the account information directly from the domain controller. This works as long as the ArchestrA Galaxy Repository is using operating system security with the same domain.

Example(s)

```
Status = GetAccountStatus("Corporate_HQ","Operator");
```

IsAssignedRole() Function

Determines whether the currently logged on user is a member of the specified user role. Only applies to ArchestrA security.

Category

security

Syntax

```
DiscreteTag=IsAssignedRole( "RoleName" );
```

Arguments

RoleName
The role associated with an Application Server user.

Remarks

Valid for ArchestrA security mode only and applies to the currently logged on user. If a user is currently logged on and has the *RoleName* role assigned in the Galaxy IDE, then TRUE is returned. Otherwise, FALSE is returned.

Example

```
DiscreteTag=IsAssignedRole( "Administrators" );
```

See Also

AttemptInvisibleLogon(), PostLogonDialog(), InvisibleVerifyCredentials(), QueryGroupMembership(), AddPermission()

QueryGroupMembership() Function

Determines whether the currently logged on user is a member of the specified user group. Only applies to operating system security.

Category

security

Syntax

```
DiscreteTag=QueryGroupMembership( "Domain", "Group" );
```

Arguments

Domain
Name of the domain or local computer in which the group is located

Group
Name of the group.

Remarks

Valid for operating system security mode only and applies to the currently logged on user. If a user is currently logged on and if he or she is part of the group located on the domain, then TRUE is returned. Otherwise, FALSE is returned.

The QueryGroupMembership() function works with operating system-based security and with ArchestrA security only when the ArchestrA security is set to operating system-based security.

Examples

```
DiscreteTag=QueryGroupMembership( "corporate_hq", "InTouchAdmins" );
DiscreteTag=QueryGroupMembership( "JohnS01", "InTouchUsers" );
```

See Also

PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), AddPermission()

$OperatorName System Tag

Contains the full name of the operator if operating system-based or ArchestrA authentication is used and someone has logged on and has not logged off. Otherwise, the tag contains the name of the user logged on (same contents as the $Operator tag).

Category

Security

Data Type

String (read-only)

Examples

```
IF $OperatorName <> "" THEN
    {Configure some defaults}
ENDIF;
```

See Also

$Operator

$OperatorDomain System Tag

Contains a different value depending on the type of security used:

- If operating system-based security is selected and an operator has successfully logged on, the $OperatorDomain tag contains the domain or node name that was specified at log on.
- If ArchestrA security is selected and a user is logged on, the $OperatorDomain contains "ArchestrA."
- If InTouch security is selected, the $OperatorDomain tag contains the string "InTouch".
- If "None" is selected, it is a empty string ("").

Category

Security

Data Type

String

Examples

```
IF $OperatorDomain == "PRODUCTION" THEN
   {Allow change to setpoint}
ELSE
   {Change denied}
ENDIF;
```

See Also

$Operator

$Operator System Tag

Contains the logon name of the user logged on.

Category

Security

Data Type

String

$VerifiedUserName System Tag

Contains the verified user's full name if the call to the InvisibleVerifyCredentials() function is successful and if the security mode is set to operating system-based or ArchestrA Application Server-based security. If the call fails, then the system tag is set to null.

Category

security

Usage

```
$VerifiedUserName
```

Remarks

When the $VerifiedUserName system tag changes (when the InvisibleVerifyCredentials() function is called), an event is generated.

Data Type

Message (read only)

Valid Values

A user's full name.

Example(s)

```
Tag =  InvisibleVerifyCredentials( "john","password", "Plant_Floor");{ If the call is successful, the $VerifiedUserName is set to "John Smith" and an Operator Event is generated. If the above call is not successful, $VerifiedUserName is set to "".}
```

See Also

InvisibleVerifyCredentials(); $OperatorName, $Operator

Summary of Security System Tags and Functions

The following table shows which security system tags and functions you can use with the different security modes.

	InTouch Security	Operating System Security	ArchestrA Security
$AccessLevel	Yes	Yes	Yes
$ChangePassword	Yes	Yes	Yes
$ConfigureUsers	Yes	No	No
$InactivityTimeout	Yes	Yes	Yes
$InactivityWarning	Yes	Yes	Yes
$Operator	Yes	Yes	Yes
$OperatorDomain	No	Yes	Yes*
$OperatorDomainEntered	No	Yes	Yes*
$OperatorEntered	Yes	Yes	Yes
$OperatorName	Yes	Yes	Yes
$PasswordEntered	Yes	Yes	Yes
$VerifiedUserName	No	Yes	Yes
AddPermission()	No	Yes	No
AttemptInvisibleLogon()	Yes	Yes	Yes
ChangePassword()	Yes	No	No
EnableDisableKeys()	Yes	Yes	Yes

	InTouch Security	Operating System Security	ArchestrA Security
GetAccountStatus()	No	Yes	Yes*
InvisibleVerifyCredentials()	No	Yes	Yes*
IsAssignedRole()	No	No	Yes
Logoff()	Yes	Yes	Yes
LogonCurrentUser()	No	Yes	Yes*
PostLogonDialog()	Yes	Yes	Yes
QueryGroupMembership()	No	Yes	Yes*

* When ArchestrA security is OS user or group based

Application Manager Operations Allowed for a Non Administrator User

The operations allowed for a non-administrator users are a subset of the operations allowed for a administrator. Restricting access to essential operations for non-administrator users will prevent security threats. The following table lists the operations under each tab of the InTouch HMI Application Manager.

InTouch Tab

Operation	Non-Administrator
New Application	No
Launch WindowMaker	No
Launch WindowViewer	Yes
DBLoad	No
DBDump	No
Delete Application	No
Rename Application	No
Application Properties	No
Export as Template	No
Import Application	No
OPC UA Server Setting	Yes
Find Applications	Yes

Operation	Non-Administrator
Export for IoT	Yes
Upload to AVEVA Connect	Yes
Configure Users for Edge Device	No
Publish Tag data as AVEVA Insight data source	Yes
Node Properties	
App Development > Start following application in WindowViewer as a Service	No
App Development > Enable Network Application Development	Yes
Resolution	Yes
Memory Settings	No
Performance	No
Refresh	Yes
Views	Yes
AVEVA Connect Login	Yes
Change Thumbnail	Yes
Open Application Folder	Yes

Web Client Tab

Operation	Non-Administrator
Enable Web Client	No
Launch Web Client	Yes
Web Client Settings	
Graphic Refresh Rate	No
Alarm Refresh Rate	No
Web Client Site Name	No
Show Header	No
Enable NavBar	No
Allow Anonymous Access	No

Operation	**Non-Administrator**
AIM Registration Settings	
Use AIM Server as the authentication server	No
Identity Server	N/A
User Name	No
Password	No
Secure Gateway	No
Allow Industrial Graphics to be embedded in any website	No

AVEVA

Chapter 10

Switching a Language at Run Time

About Switching a Language at Run Time

You can develop applications that can be switched to another language at run time.

To enable run-time language switching, you must complete the following tasks:

- Configure multiple languages for the application.
- Export your application text for offline translation.
- Translate one or more exported dictionary files.
- Import one or more translated dictionary files.

As part of the setup for run-time language switching, you can also localize alarm comments and alarm fields. In addition to switching the run-time language of text strings, you can also configure run-time language switching of alarm comments, alarm states, alarm types, and alarm classes in the Alarm Viewer and Alarm DB View controls.

Configuring Languages for Run-time Language Switching

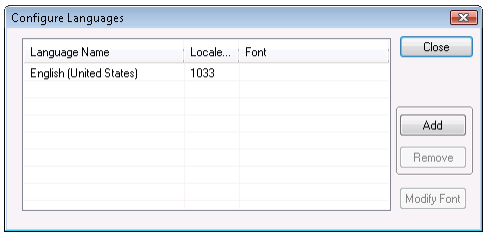
Every InTouch application is associated with a base language used to develop the application. You must configure any additional languages that you want to support.

Note: If you are using language switching in combination with Network Application Development (NAD), we recommended that you set the change mode to "Restart WindowViewer" or "Prompt user to restart WindowViewer" instead of "Load changes into WindowViewer" or "Prompt user to load changes into WindowViewer" for the NAD client node.

To configure languages for run-time language switching

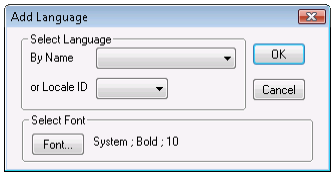
1. In WindowMaker, open the application for which you want to configure languages.

2. On the **Special** menu, point to **Language**, and then click **Configure Languages**. The **Configure Languages** dialog box appears.

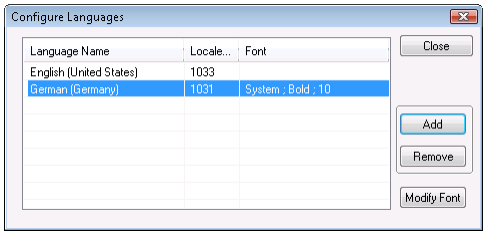


   The **Configure Languages** dialog box shows the base language of the application.

3. Click **Add**. The **Add Language** dialog box appears.



4. Specify the language and font settings. Configuring the font settings defines the default font properties of your translated text.
   - In the **By Name** or the **Locale ID** list, click the language to add. If you select the language by name, the corresponding locale ID appears in the **Locale ID** list, and vice versa.
   - Click **Font**. The **Font** dialog box opens. Configure the font and then click **OK**.
5. Click **OK** to close the **Add Language** dialog box. The language you configured is listed in the **Configure Languages** dialog box.



6. To add more languages, repeat steps 3 through 5.
7. When you are done, click **Close**.

To remove a language:

a. Select the language you want to remove from the **Configure Languages** dialog box.

b. Click **Remove**.

   The **Confirm Delete** dialog box appears to verify that you want to remove the language from the application.

c. Click **Yes**.

   The **Configure Languages** dialog box refreshes and shows the language has been deleted.

To set a default language:

a. Select the language you want to set as the default for the application on the **Configure Languages** dialog box.

b. Click **Set Default**.

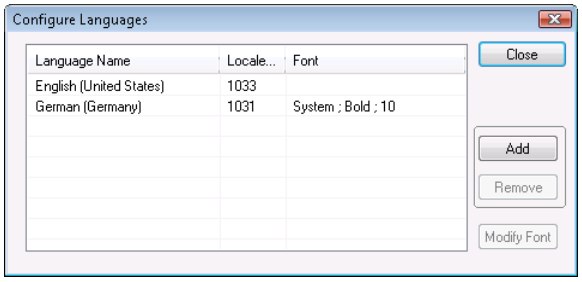
The **Configure Languages** dialog box refreshes and shows the default language of the application in the bottom left corner.

Changing the Font Settings for a Configured Language

The default font for all languages is Tahoma. The font style and size depends on the corresponding settings for the individual phrases in the base language. You can change the font setting for a language that you have already configured. Because of the differences between the textual display of different languages, you can specify an appropriate font to ensure that your translated text fits correctly on buttons and other objects.

To change the font settings for a configured language

1. In WindowMaker, open the application for which you want to change font settings for a configured language.
2. On the **Special** menu, point to **Language**, and then click **Configure Languages**. The **Configure Languages** dialog box appears.



3. In the list of languages, select the target language, and then click **Modify Font**. A standard Windows **Font** dialog box appears.
4. Make your changes and then click **OK**.
5. Click **OK** to close the **Configure Languages** dialog box.

Adding Run-Time Language Switching Functionality

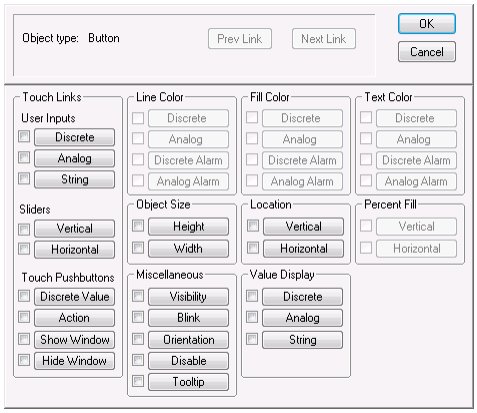
Run-time users can switch the language of an application interface by using the WindowViewer **Language** command on the **Special** menu.

You can also add a button to your application to allow run-time users to switch the language. Before you start, make sure that you have configured the additional language for the application and that you know the locale ID for the language. For more information on configuring languages for an application, see *Configuring Languages for Run-time Language Switching* on page 213.

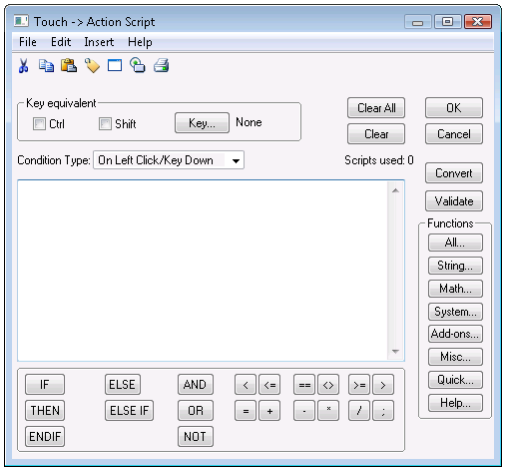
To add a button for switching languages at run time

1. In WindowMaker, open the application window that you want to add the language switching button.
2. In the window, draw a button.

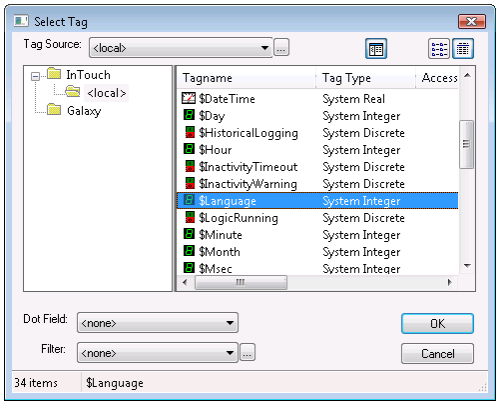
3. Assign a text label to the button that indicates the language to be switched to when selected.
4. Double-click the button. The animation selection dialog box appears.



5. In the **Touch Pushbuttons** area, click **Action**. The **Touch -> Action Script** dialog box appears.

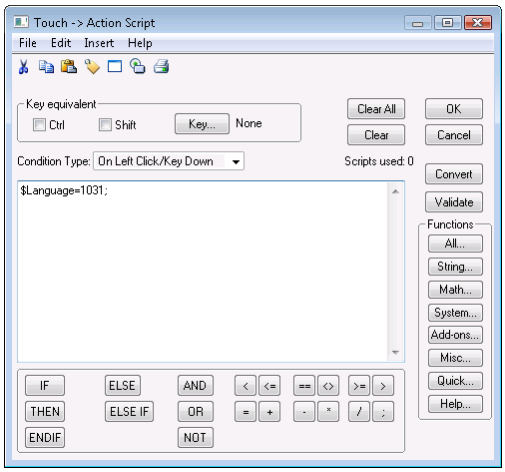


6. Double-click anywhere in the script area of the **Touch -> Action Script** dialog box. The **Select Tag** dialog box appears.



7. Click the **$Language** system tag and then click **OK**.

Set the **$Language** system tag equal to the locale ID of the language you are assigning to the button and click **OK**.



Note: You can also use the script function SwitchDisplayLanguage(LocaleID) instead of the $Language tag.

8. Click **OK** to close the dialog box.

SwitchDisplayLanguage() Function

Switches the display of visible, static texts and alarm fields in a desired language for which translated strings are provided.

Category

misc

Syntax

```
SwitchDisplayLanguage(LocaleID);
```

Parameter

LocaleID
The language in which static text strings and alarm fields are to be shown at run time.

Example(s)

In this example, German is the language to be shown at run time.

```
SwitchDisplayLanguage(1031);
```

See Also

$Language system tag

$Language System Tag

If multiple languages are defined for an InTouch application, the $Language system tag reflects the value of the Language ID for the currently shown language. By default this is the language ID (locale ID) of the base InTouch system (E/F/G/J/SC). Setting this to another ID switches strings and alarm fields with defined values in the new language.

Note: The $Language tag is configured as a local tag to allow independent language switching in the Web Client. Multiple user sessions can view the web client in different languages. The change in language in the web client will not affect the language selected in WindowViewer and vice versa.

Category

system

Data Type

Integer (read / write)

Exporting Application Text for Offline Translation

If your InTouch application has many strings, you typically send the text strings out for bulk translation. You can export your application's strings for translation and keep them organized using a text editor, an XML editor, or a spreadsheet program like Microsoft Excel.

You can export static text from the following:

- Text objects.
- Button text.
- Text inside SmartSymbols.
- Tooltip static text.
- User messages.
- On/off messages inside input links.
- On/off messages in output links.
- Text on wizards.

You cannot export the dictionary until you close all windows in WindowMaker. If you make changes to your application after you export your dictionary files, you must export the dictionary file again. For more information, see *Exporting Text to an Existing Dictionary File* on page 219.

You can only export the text strings for one language at a time. By default, the InTouch HMI opens the My InTouch Applications folder. If you choose any other folder, the InTouch HMI then defaults to that path. Creating a new folder to export phrases for each language makes it easy to manage dictionary files. For example, ...\My InTouch Applications\My German Files\.

The InTouch HMI creates a dictionary file for your application and a separate dictionary file for each SmartSymbol within the application. The application dictionary name has a format of application name_localeID whereas SmartSymbol dictionary files have a format of SSD_Name of the Symbol_localeID_GUID.

When you export the dictionary for an application, the file is an .xml file that you can edit using Microsoft Excel 2003 or later.

To export application text for offline translation

1. Start WindowMaker and open the application for which you want export text strings for offline translation.

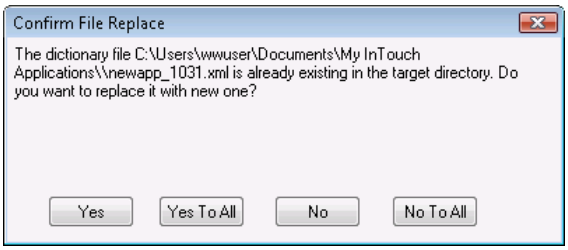
2. On the **Special** menu, point to **Language**, and then click **Export Dictionary**. The **Export Dictionary** dialog box appears.

3. Configure the export settings.
   - In the **Defined Languages** list, click the language dictionary to export.
   - In the **Path** box, type the folder to which you want to export the dictionary. Click **Browse** to select an existing folder or create a new folder.
4. Click **Export**. The export progress is shown. If the export is successful, the **Export Successful** dialog box appears.
5. Click **Close** to return to the WindowMaker window or click **Close and Launch Explorer** to open the folder containing the dictionary files.

Exporting Text to an Existing Dictionary File

After you export your application text for offline translation, you might need to make changes to your application. If you change the application, you need to export the text again. For more information, see *Exporting Application Text for Offline Translation* on page 218.

If you export more than one time to the same directory, the **Confirm File Replace** dialog box appears.



If you click **Yes**, the existing .xml files are updated with any new strings and language information added since you exported last. If the existing dictionary file contains translations for any phrases and you imported it to the InTouch HMI previously, those translations are preserved. If you deleted any phrases from the application since the last export, they are removed from the dictionary file.

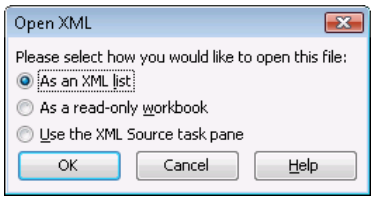
Translating an Exported Dictionary File

After you export the dictionary file containing your application text, use Microsoft Excel 2003 or later to edit the text.

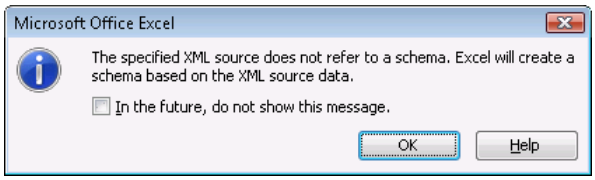
The InTouch HMI creates a separate dictionary file for each language that you export. The InTouch HMI also creates separate dictionary files for each SmartSymbol in your application. Be sure to translate all dictionary files for all languages and SmartSymbols.

To translate an exported dictionary file

1. Open the XML file in Excel. The **Open XML** dialog box appears.



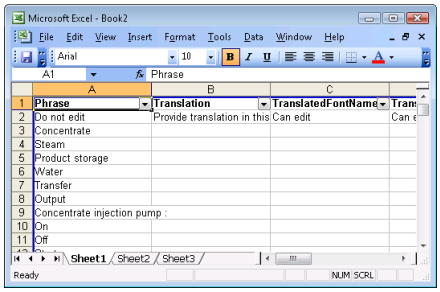
2. Click **As an XML list**, then click **OK**. A message may appear.



3. Click **OK**.

   The XML file opens in Excel with columns for the:

   - Phrases in your application.
   - Translated phrases from the translator.
   - Translated font name.
   - Translated font properties.
   - Translated font size.
   - Base font properties.
   - Base font size.
   - Context, phrase ID, language ID and foreign language ID.



Important: Only modify data in the **Translation**, **TranslatedFontSize**, **TranslatedFontName**, and **TranslatedFontProperty** columns. Do not change any column header. Do not insert or delete rows.

4. Type the language-specific text in the **Translation** column in the row that corresponds with the base language string in the **Phrase** column.

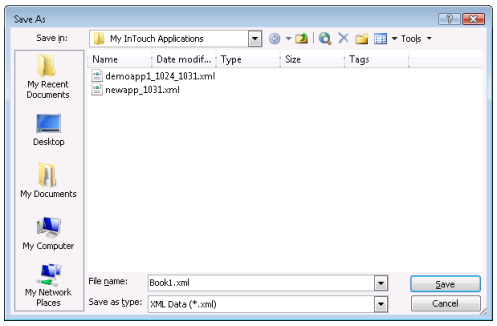
5. If necessary, change the font parameters for the translated strings to fit the text in the space allowed in WindowViewer.
   - In **TranslatedFontName** column, type the font name.
   - In the **TranslatedFontProperty** column, type the notation for the font properties:

     B = **bold**

     I = *italic*

     U = underline

     For example, if you want to the text to be bold, type **B** in the **TranslatedFontProperty** column. If you want the text to be bold and underlined, type **BU** in the **TranslatedFontProperty** column.
6. Save the file using XML Data as the file type.



Important: If you save as another file type, such as XML Spreadsheet, Excel changes the schema and the InTouch HMI cannot load the file. If you change the name of the XML file, run-time language switching does not work.

Importing Translated Dictionary Files

The InTouch HMI creates a dictionary file for each language that you export. The InTouch HMI also creates separate dictionary files for each SmartSymbol in your application. After translation, you must import the dictionary files for each language to enable run-time language switching for those languages. All dictionary files for a given language should be placed in the same folder.

To import a translated dictionary file

1. Start WindowMaker and open the application to import translated dictionary files into.
2. On the **Special** menu, point to **Language**, and then click **Import Dictionary**. The **Import Dictionary** dialog box opens.

3. Configure the import settings.

- In the **Defined Languages** list, click the language dictionary to import.
- In the **Path** box, type the path to the dictionary file to import. Click **Browse** to browse and select the file.

4. Click **Import**.
5. If you are re-importing a SmartSymbol dictionary file, you are prompted to replace the existing file.

   If the import is successful, the **Import Successful** dialog box appears.

Exporting Alarm Comments for Translation

You can export alarm comments for translation.

You export the Alarm State, Alarm Type, and Alarm Class fields for:

- All tags with an alarm comment.
- All tags with a tag comment.
- System tags so you can localize comments shown in clients when events are raised by system tags.

Understanding Two-Character Application IDs

When you export alarm and tag comments for localization, you must specify a two-character application ID. The ID is used internally by the system to distinguish between alarms generated by applications having the same name.

Because a tag can contain both a tag comment and an alarm comment, 1 and 2 are added after the two-character application ID to differentiate between these two fields. Tag comments have a 1 between the ID and the tag name. Alarm comments have a 2 between the ID and the tag name. For example, AA1TankLevel is a tag comment, and AA2TankLevel is an alarm comment.

If you export an application, the application ID information is removed.

If the alarm database contains old data without a two-character application ID and new records are prefixed with an ID, then alarm comment queries in the Alarm DB View control do not work with the following operators: <, <=, >, and >=.

Exporting Alarm Comments

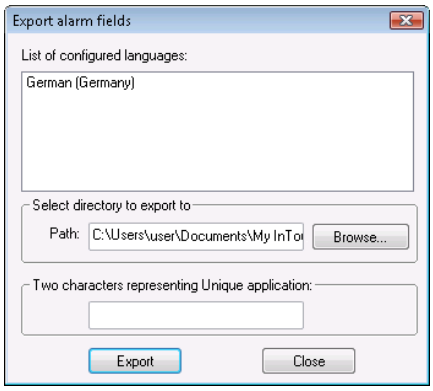
You can export alarm comments for translation.

Caution: Before exporting alarm and tag comments, back up any files in your target directory in case of possible data corruption or errors.

To export alarm comments for offline translation

1. Start WindowMaker and open the application for which you want export alarm comments for offline translation.

2. On the **Special** menu, point to **Language**, and then click **Export Alarm Fields**. The **Export Alarm Fields** dialog box appears.



3. In the **Path** box, type the folder to which you want to export the dictionary. Click **Browse** to select an existing folder or create a new folder.
4. In the **Two characters representing Unique application** box, type the two characters. The ID can only contain alphanumeric characters and it is case-sensitive.

Caution: If you previously exported alarm or tag comments from this application, you must use the same two-character application ID when you export them the next time. If you enter a new two-character application ID, the InTouch HMI regenerates the IDs for all the alarms and tags, which causes all existing translations to be lost.

5. Click **Export** to export the information to an XML dictionary file.

   The InTouch HMI creates an individual export file for each configured language. All the dictionary files for different languages are exported to the single directory you specify.

   If a duplicate file exists for any language being exported, you are prompted with the name of the file. You can cancel the export or continue the export operation.

   If the export is successful, the **Export Successful** dialog box appears.

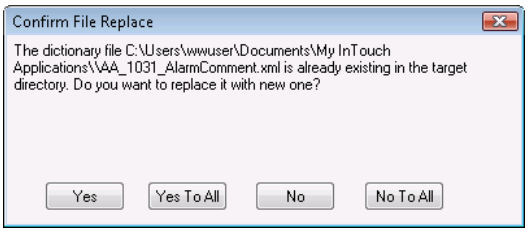
   Note: If the size of the alarm comment configured in the tag dictionary is greater than 127 characters or the tag comment is greater than 46 characters, the alarm or tag comment is not exported. You are notified that the comment was not exported to the dictionary file at the end of the export process and an AlarmComment.log or TagComment.log file is created in the export directory.

6. Click **Close** to return to the WindowMaker window or click **Close and Launch Explorer** to open the folder containing the dictionary file.

Exporting to an Existing Alarm Comment File

After you export your alarm and tag comments for offline translation, you may need to make changes to your application that require you to export alarm and tag comments again. For more information, see *Exporting Alarm Comments for Translation* on page 222.

If you export more than one time to the same directory, the **Confirm File Replace** dialog box appears.



Click **Yes** to update the existing dictionary files with any new strings and language information added since you exported last. If the existing dictionary file contains translations for any phrases and you imported it to InTouch previously, those translations are preserved. If you deleted any phrases from the application since the last export, they are removed from the dictionary file.

Click **Yes to All** to update existing dictionary files for all languages configured in the InTouch HMI.

Click **No** or **No to All** to prevent overwriting the existing file or the existing files for all languages, respectively.

The existing translations for any alarm comments, alarm fields and tag comments are preserved if they are exported again.

Editing the Dictionary File

After creating the dictionary file, you need to edit the strings.

The name of the dictionary file is created from the two-character application ID and the language being exported. For example, if the configured language is Chinese(PRC)-2052 and two-character application ID is **AA**, the resulting file name is **AA_2052_ AlarmComment.xml**. The file is written using the same XML schema used by the run-time language switching files.

The general structure of the dictionary file is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfAlarmCommentPhraseItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <AlarmCommentPhraseItem Phrase="Do not edit">
      <Translation>Provide translation in this column</Translation>
      <Context>Do not edit</Context>
      <PhraseID>0</PhraseID>
      <LanguageID>0</LanguageID>
      <ForeignLanguageID>0</ForeignLanguageID>
    </AlarmCommentPhraseItem>
  - <AlarmCommentPhraseItem Phrase="System">
      <Translation />
      <Context>$System : System Tag Comment</Context>
      <PhraseID>AA1$System</PhraseID>
      <LanguageID>1033</LanguageID>
      <ForeignLanguageID>2052</ForeignLanguageID>
    </AlarmCommentPhraseItem>
</ArrayOfAlarmCommentPhraseItem>
```

Enter the translation for the translation strings. Do not change any of the other information.

You can override some of the Alarm State, Alarm Type, and Alarm Class values. The maximum allowed length for these values is 50 characters.

The following Alarm State values can be overridden for InTouch generated alarms:

Value to override	Default string to be shown
UNACK_RTN	UNACK_RTN
ACK_RTN	ACK_RTN

Value to override	Default string to be shown
UNACK_ALM	UNACK_ALM
ACK_ALM	ACK_ALM

The following Alarm Type values can be overridden for InTouch generated alarms:

Value to override	Default string to be shown
SPC	SPC
HIHI	HIHI
HI	HI
LO	LO
LOLO	LOLO
MINDEV	MINDEV
MAJDEV	MAJDEV
ROC	ROC
DSC	DSC
OPR	OPR
LGC	LGC
DDE	DDE
SYST	SYST
USER	USER
PRO	PRO
LOGON_FAILED	LOGON_FAILED

The following Alarm Class values can be overridden for InTouch generated alarms:

Value to override	Default string to be shown
DEV	DEV
ROC	ROC
DSC	DSC
EVENT	EVENT
VALUE	VALUE

Importing Translated Alarm Comments

After translating the strings, you must import the dictionary files for each language to enable run-time language switching for those languages.

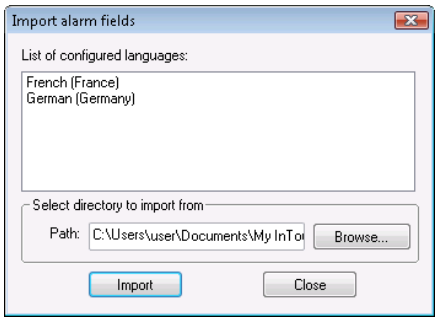
After you import the translated alarm comment dictionary files, they are copied into the respective language folders inside the application directory.

Any translated alarm comments for an existing application are overwritten with the contents of the imported files and the application version ($AppVersion) increments by 1.

To import multiple dictionary files from other nodes to support localization of alarm fields from multiple nodes, copy the translated dictionary files from the other nodes into a single directory. Select this directory as the import path. Multiple dictionary files are imported on a single import operation. The InTouch HMI automatically creates the file path based on the language being imported.

To import a translated alarm comment file

1. Start WindowMaker and open the application to import translated dictionary files into.
2. On the **Special** menu, point to **Language**, and then click **Import Alarm Fields**. The **Import Alarm Fields** dialog box appears.



3. In the **Path** box, type the path to the dictionary file to import or click **Browse** to browse and select the file.
4. Click **Import**. If there is no translation done in the dictionary file, a dialog box appears saying that there are no translated dictionary files to import.
5. Click **OK**. If the import is successful, the **Import Successful** dialog box appears.

Testing the Language Switching Functionality at Run Time

After you enable run-time language switching in your application, test the language switching functionality. Language switching of alarm and tag comments and alarm fields can be viewed only in the Alarm Viewer and Alarm DB View controls.

As you work with localized alarm and tag comments, be aware of the following:

- If the alarm or tag comment hasn't been translated to the language specified by $Language, the default comment appears in the alarm client.

- If an Alarm Viewer control is querying from multiple providers, the alarm comment, tag comment, and alarm fields from remote nodes also appear translated if the application has the translated dictionary files of the remote node applications.
- If you acknowledge an alarm and provide a comment, this comment appears in the alarm client instead of the localized alarm comment.
- When an Alarm Viewer control is in freeze mode, then the language does not appear switched even though you switched the language. The moment you unfreeze the control, the control is updated with the translated strings.
- The localization of the Alarm Viewer control is only for the display of the control. All script functions still return the default strings even though the language is switched.
- The Alarm DB Logger only stores the data default language strings in the database. The localized strings are not stored in the database.
- The unique IDs for the alarm fields such as EVENT and ACK, are predefined and have the same ID across multiple dictionary files in different nodes. Alarm clients pick the translation from the first loaded dictionary file and the translations from other dictionary files are ignored. Ideally, the alarm fields in all dictionary files should have the same translation in a language. Multiple alarm clients (Alarm DB View and Alarm Viewer controls) use the same translation for the same alarm state for a given language
- Translated text is truncated to 131 characters for alarm comments and to 50 characters for tag comments.

To test the language switching functionality

1. Open the application in WindowViewer.
2. On the **Special** menu, point to **Language**, and then click the name of the language to switch to.

   The information from the corresponding translated dictionary file (if one exists) loads and appears.
3. If you added a button to switch the language, click the button to test the script.

Distributing Localized Files to Network Application Development Clients

The files containing the localized alarm comments, tag comments, and alarm fields are distributed to Network Application Development (NAD) clients as part of the InTouch application. When you receive updated files containing alarm comments, you must restart WindowViewer before the translated alarm comments can be seen in the supported alarm clients.

If you are using language switching in combination with Network Application Development (NAD), set the change mode to "Restart WindowViewer" or "Prompt user to restart WindowViewer" for the NAD client node.

Chapter 11

Viewing Applications at Run Time

About Viewing Applications at Runtime

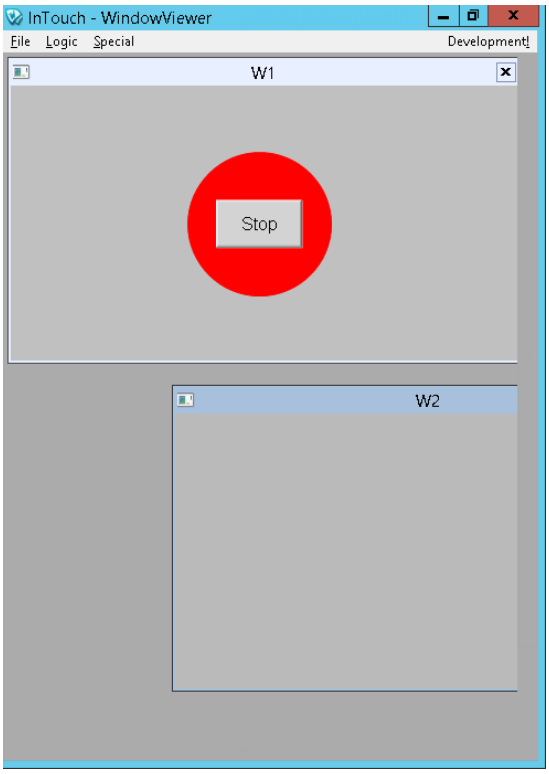
You use WindowViewer to run your InTouch applications. Applications that are designed specifically for use in an ArchestrA Application Server environment are called InTouchView applications. You can use the InTouch Web Client to view Industrial graphics in any HTML5 supported web browser. These applications run in WindowViewer, but the Application Server provides most of the HMI functionality.

Viewing Applications at Run Time in a Different Target Resolution Size

If you have specified an application target resolution that is different than your screen resolution, WindowViewer displays the application at the specified target resolution. Only windows, window controls and graphics developed within the canvas boundary that outlines the target resolution size will display at run time. The target resolution size is automatically adjusted at run time to account for WindowViewer's menu and title bar controls.

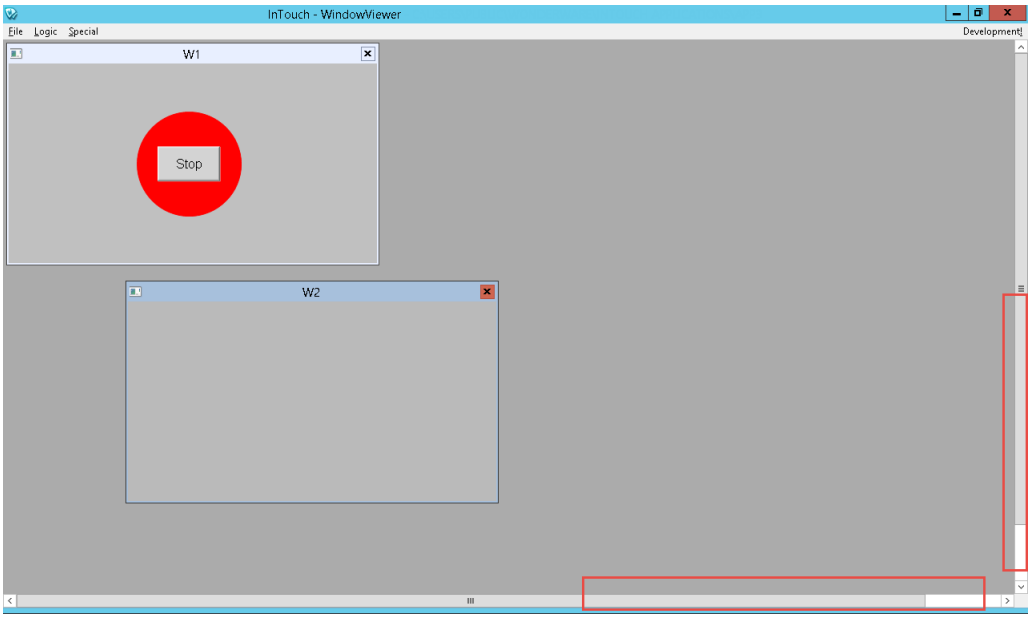
The aspect ratio of embedded controls will be maintained at run time.

For example:



Note: If the specified target resolution is less than the screen resolution, the width and height of the runtime window cannot be adjusted beyond the specified target resolution. Maximizing the WindowViewer window will enlarge it only to the maximum of the target resolution.

If the specified target resolution is greater than the screen resolution, vertical and horizontal scroll bars will display at run time. Windows will scroll accordingly. Popup windows and popup graphics from ShowSymbol animations and ShowGraphic scripts will not be scrolled.



Error messages and popup dialogs will display in the center of the application at its target resolution, not the center of the screen. The same applies to keyboards and keypads.

Original Application Resolution

The original application resolution is the screen resolution when the application was created regardless of the target resolution settings.

The original application resolution is updated only under the following conditions:

- At the time of application creation
- When conversion is applied on the application

If the application is later switched back to screen resolution, then application conversion will occur if, the current screen resolution is different from the original application resolution. Else, no conversion will occur. If a target resolution is used when the application is created it will impact the behavior of the Dynamic Resolution Conversion.

About the InTouch Web Client

The InTouch Web Client feature allows you to view Industrial graphics used within an InTouch HMI application on any HTML5 supported web browser. A built-in Web Server provides web browsers access to application graphics, from any Microsoft Windows client or server operating system without the use of Remote Desktop Protocol (RDP) or Internet Information Services (IIS) for Microsoft Windows® Server. You can view application graphics in a web browser for both standalone and managed applications. Standalone application windows must be converted to Industrial Graphics before they can be viewed on the Web Client.

For more information on the InTouch Web Client, refer to the *Viewing InTouch Application Graphics in a Web Browser* guide. The Web Client is installed as part of the System Platform installation. For information on configuring the Web Client, refer to the System Platform Installation Guide.

About WindowViewer

WindowViewer provides the run-time environment for InTouch applications. Based upon your application's operational requirements, you can configure how WindowViewer supports an application. For example, depending on your application's security requirements, you can configure the menus and commands available to operators from WindowViewer.

Customizing Your Run time Environment

You can set properties to customize your run time WindowViewer environment.

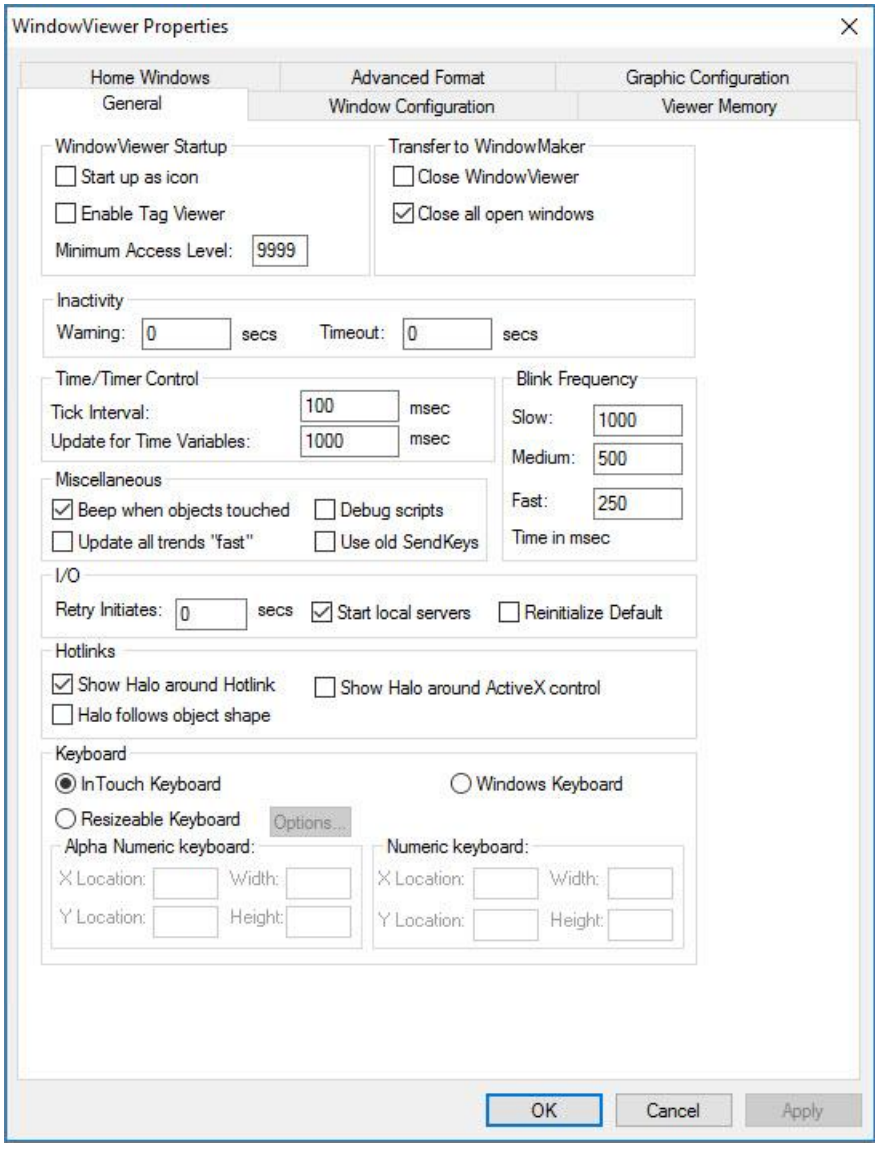
- General properties determine the environmental conditions to run an InTouch application.
- Window configuration properties determine the menus, commands, and window components accessible to users as they interact with an InTouch application running in WindowViewer.

Configuring General WindowViewer Properties

You can configure a set of general properties that determine the characteristics of WindowViewer as it runs an InTouch application. After you modify these general properties, you must restart WindowViewer for your changes to become effective.

Note: Close WindowViewer before changing the Regional Formats of the Operating System.

1. Open WindowMaker.
2. On the **Special** menu, point to **Configure**, and then click WindowViewer. The WindowViewer Properties dialog box appears.
3. Click the **General** tab.



4. In the **WindowViewer Startup** area, do the following:
   - Select the **Start up as Icon** check box if you want WindowViewer to start up minimized.

- Select the **Enable Tag Viewer** check box to allow the Tag Viewer application to be started at run time. The Tag Viewer allows you to watch and monitor tags and modify tag values at run time. For details, see the *Enabling Tag Viewer* section in the *InTouch® HMI Tag Viewer Guide*.
- In the **Minimum Access Level** box, type the minimum security access level required to run the Tag Viewer application. The value must be between 0 and 9999. For details, see the *Enabling Tag Viewer* section in the *InTouch® HMI Tag Viewer Guide*.

5. In the **Transfer to WindowMaker** area, do the following:
   - Select the **Close WindowViewer** check box if you want WindowViewer to automatically close when you start WindowMaker.

     When you select this option, the Close on Transfer to WindowViewer option located on the WindowMaker Properties/General tab is automatically selected too. If memory is not an issue, and you are using the fast switch to move between WindowViewer and WindowMaker, this option should be cleared.
   - Select the **Close all open windows** check box if you want all open windows to close automatically when you transfer from WindowViewer to WindowMaker.
6. In the **Inactivity** area, set warning and time-out periods for operator inactivity.

   For more information about setting warning and time-out periods, see *Configuring an Inactivity Time-Out* on page 175.
7. In the **Time/Timer Control** area, do the following:
   - In the Tick Interval box, type the interval that the InTouch HMI uses to check its internal timers.

     This interval determines when Application While Running, Window While Showing, Condition While On True/On False, Key and Touch Pushbutton Action While Down QuickScripts can be started.

     This option sets the value for TimerTickInterval parameter in the INTOUCH.ini file. You should set the Tick Interval no longer than 50 msec for a script scheduled to run every 100 msec. On computers running Windows XP or Windows 2003, the lower tick interval is 10 msec.
   - In the **Update for Time Variables** box, type the interval in milliseconds that time is updated for system tags like $Msec, $Second, or $Minute.

     We recommend that you use the default setting of 1000 milliseconds. Setting this option to zero prevents time variables from being updated.
8. In the **Miscellaneous** area, do the following:
   - Select Beep when objects touched if you want the InTouch application to emit a beep sound when operators select touch-sensitive objects on a window.
   - Select **Update all trends "Fast"** if you want your trend objects to be updated more quickly.

     Select this option only if no objects overlap run-time trends on the application window. If you select this option and any object overlaps a trend, the object can be drawn incorrectly.
   - Select **Debug Scripts** if you want a message to be written to the Logger each time a QuickScript runs.

     If you select **Debug** from the **Window Configuration** property sheet, you can switch QuickScript logging on and off from WindowViewer's **Special** menu.

- Select the **Use old SendKeys** check box if you are using an international application developed with InTouch version 3.26 or earlier.

9. In the **Blink Frequency** area, type the interval length in milliseconds for your **Slow, Medium**, and **Fast** blink animation links.
10. In the **I/O** area, do the following:
    - In the **Retry Initiates** box, type the number of seconds to wait before the InTouch application retries connecting to an I/O Server after a failed connection attempt. The **I/O Retry Initiates** box has no effect when InTouch can successfully connect to the I/O server the first time.
    - Select the **Start local servers** check box if you want a dialog box to appear when you start WindowViewer and the I/O Server you are trying to communicate with is not running.
    - Select the **Reinitialize Default** check box if you want to reinitialize Access Names with their default settings. Current values assigned to Access Names are ignored and they are reinitialized with their original settings.
11. In the **Hotlinks** area, do the following:
    - Select the **Show Halo around Hotlink** check box if you want an object on the run time screen to be highlighted when the user moves the cursor over it.
    - Select the **Halo follows object shape** check box if you want a highlight halo around the contours of an object when the user moves the cursor over it.
    - If you want a halo around Active X controls, select the **Show halo around Active X control** check box.
12. In the **Keyboard** area, select the type of keyboard you want to use, if any.

    For more information about setting keyboard options in WindowViewer, see Animating Objectsin the InTouch® HMI Visualization Guide.
13. Click **OK**.

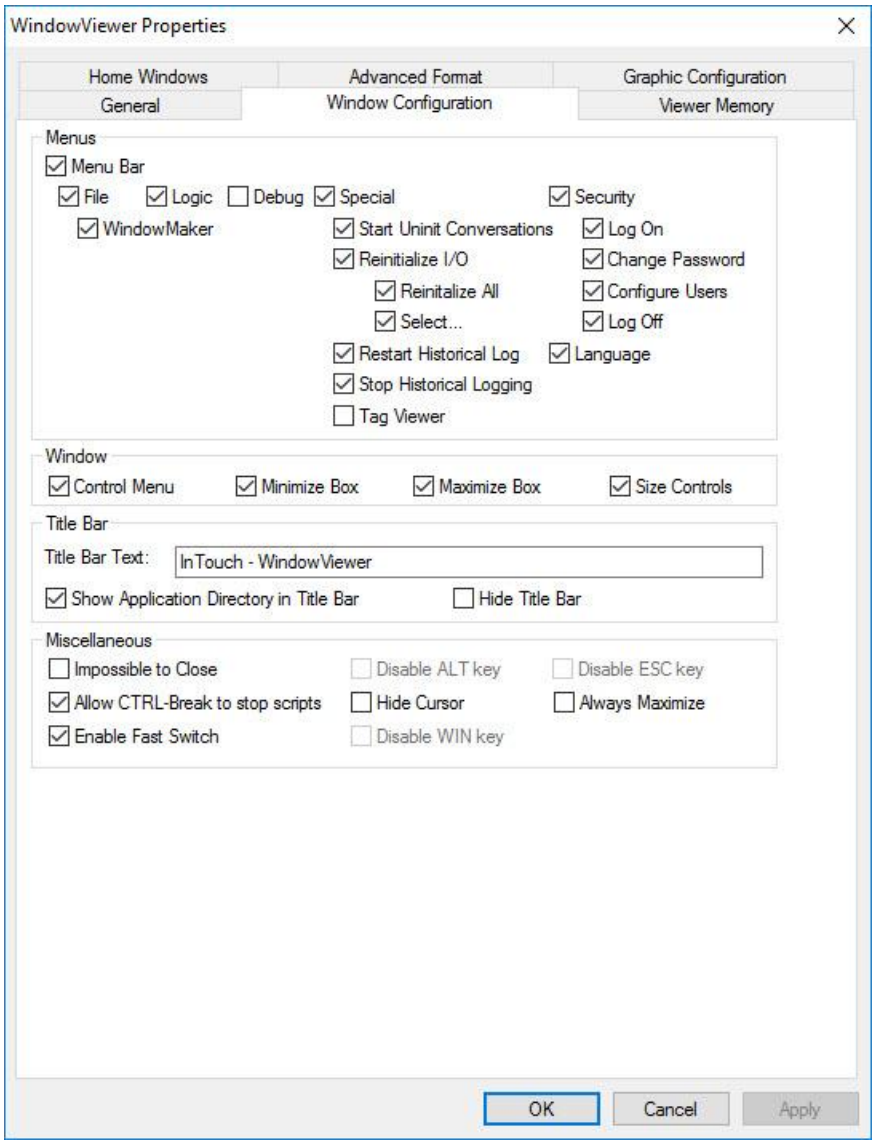
Configuring Visual Characteristics of WindowViewer

You can configure properties that determine the visual characteristics of WindowViewer as it runs an InTouch application. These properties determine the menus, commands, and standard controls that appear on a WindowViewer window.

To configure WindowViewer visual characteristics

1. Open WindowMaker.
2. On the **Special** menu, point to **Configure**, and then click WindowViewer. The WindowViewer **Properties** dialog box appears.
3. Click the **Window Configuration** tab.

Select the check boxes for the visual characteristics.



4. Restart WindowViewer.

Configuring User Access to Applications Running in Remote Sessions

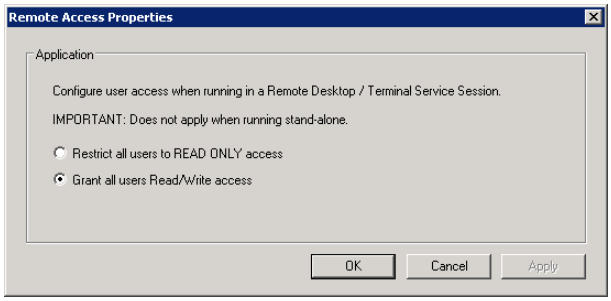
You can assign Read Only access to a distributed InTouch application running in a Remote Desktop Connection or Terminal Services session. Read Only access is appropriate for non-operators who need to view an application, but who should not have write permission. For example, managers need the capability to view an application on a mobile device with InTouch Access Anywhere over an unsecured public network. Using Read Only access provides better protection for distributed InTouch applications accessible from public networks.

To configure user access to applications running on remote nodes

1. Open the application in WindowMaker.
2. Click **Special**, and then click **Configure** to show a list of configuration options.

3. Click **Remote Access** from the list of configuration options.

   The **Remote Access Properties** dialog box appears with options to grant Read/Write or Read Only access to the application open in WindowMaker. Read/Write access is the default.



4. Make your selection and click **OK** to close the dialog box.

During initialization, WindowViewer verifies if the application is running in a remote session and is specified as Read Only. Also, a check is made that the remote node has a Read Only license. If all of these conditions are true, the application's InTouch Links and User Input animations are viewable only in Read Only mode.

About Managing Memory for WindowViewer

You can configure how WindowViewer uses memory for windows and popup symbols to improve performance at run time. Windows and popup symbols displayed using ShowSymbol animation and the ShowGraphic script function can be kept in memory at run time in certain conditions to allow for fast retrieval.

In-memory caching of Industrial graphics is available only in Managed or Published InTouch applications. This capability is disabled in Standalone InTouch applications.

You can also specify the interval for a periodic memory health check and settings for the heap memory segment. Each time a new popup symbol opens, it triggers a memory health check regardless of the pre-set interval.

Fast switching to WindowViewer or from WindowViewer to WindowMaker clears both the graphic cache and the window memory cache.

For information on memory management for symbols displayed by the ShowGraphic function, see the Industrial Graphic Editor User Guide.

Configuring Memory Usage for WindowViewer Windows

You can configure how WindowViewer uses memory for application windows to improve performance at run time.

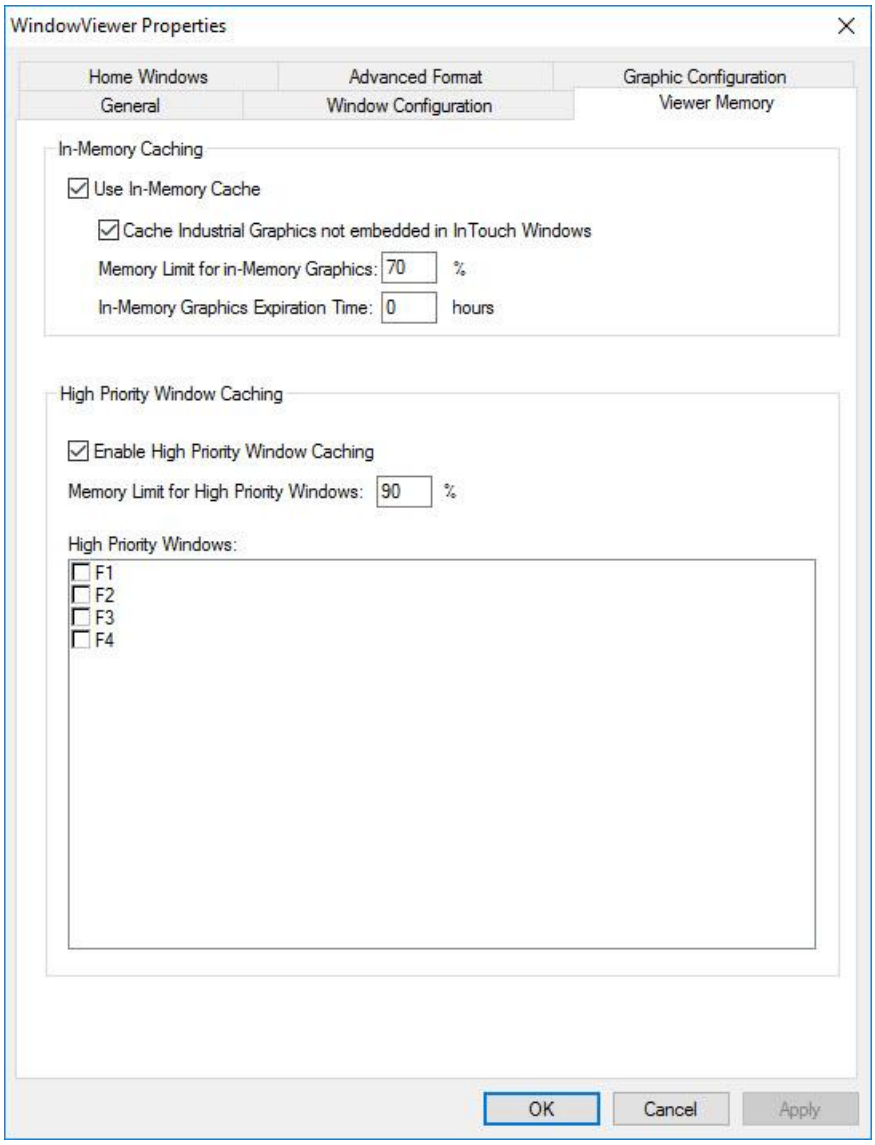
Reopening closed windows that have been cached retrieves them from memory rather than loading them from disk in certain conditions.

You can designate certain windows to have a higher priority for memory usage and configure separate memory settings just for those windows.

After you modify any of the WindowViewer memory options, you must restart WindowViewer to apply your changes.

To set the memory properties

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



3. In the **In-Memory Caching** area, do the following:
   a. Select the **Use In-Memory Cache** check box if you want to save all closed windows to be cached in memory at run time.
   b. Select the **Cache Industrial Graphics not embedded in InTouch Windows** check box to enable Industrial graphics symbol caching.

      InTouch windows and Industrial graphics will share the in-memory cache.

      If the memory limit set in step 3c is exceeded, the system automatically removes the oldest in-memory graphic from the cache.

c. In the **Memory Limit for In-Memory Windows** box, enter the limit for keeping closed in-memory windows in cache memory at run time. The default memory limit is 70% of process memory.

   If the memory limit is exceeded, the system automatically removes the oldest closed in-memory window from the cache at run time, unless it is marked as a high-priority window.

   The memory limit for in-memory windows will always be less than the memory limit for high-priority windows.

d. In the **In-Memory Window Expiration Time** box, enter the maximum duration for which the closed in-memory windows will remain in cache memory at run time. You can enter a value between 0 and 8760 hours. The default value is 0 hours, which designates no time limit.

The memory limit or the expiration time limit is applied depending on which limit is reached first.

4. In the **High Priority Window Caching** area, do the following:

   a. Select the **Enable High Priority Window Caching** check box to allow some windows to be marked as high priority. These windows will always be kept in cached memory after they are closed at run time.

   b. In the **Memory Limit for High Priority Windows** box, enter the limit for keeping closed high-priority windows in cache memory at run time. The default memory limit is 90%. The system removes the oldest in-memory window first, and then removes the oldest high-priority window when the percentage of used memory exceeds this limit at run time.

   c. In the **High Priority Windows** box, select the windows you want to mark as high priority.

5. Click **OK**.

Configuring the Memory Health Check Interval

The system checks the memory and Graphical Device Interface (GDI) count at regular intervals. If the memory or GDI count limit is exceeded, the system starts removing windows. By default, this interval is set at 5 minutes.

If you want to change the interval, you can add a new entry in the InTouch.ini file and then specify a new interval value.

If you want to turn off the check, you can add the new entry and set the value to 0.

After modifying the interval, you must restart WindowViewer to apply the changes.

For more information on how windows will be removed, see *Configuring Memory Usage for WindowViewer Windows* on page 235.

To configure the memory health check interval

1. In the application folder, open the InTouch.ini file.
2. Under the [Intouch] section, add the following script, where *<interval>* is the desired interval, in minutes:
   ```
   MemoryHealthCheckInterval = <interval>
   ```

Opening a new popup symbol or a new window will trigger a memory health check regardless of the pre-set interval.

Configuring wwHeap Memory Settings

wwHeap is a memory manager that manages the heap memory segment. The memory manager provides a mechanism for one or more programs to share virtual memory.

Caution: Modify the wwHeap memory settings only if you are experiencing memory conflicts reported in the SMC Logger.

You can configure the wwHeap Memory settings by specifying the wwHeap size and start location. The default size, default start location, and allowable location range vary by operating system.

The default sizes are described in the following table:

Operating System	Default Size
32-bit	1519 MB
32-bit with the /3GB switch enabled	2048 MB
64-bit	2048 MB

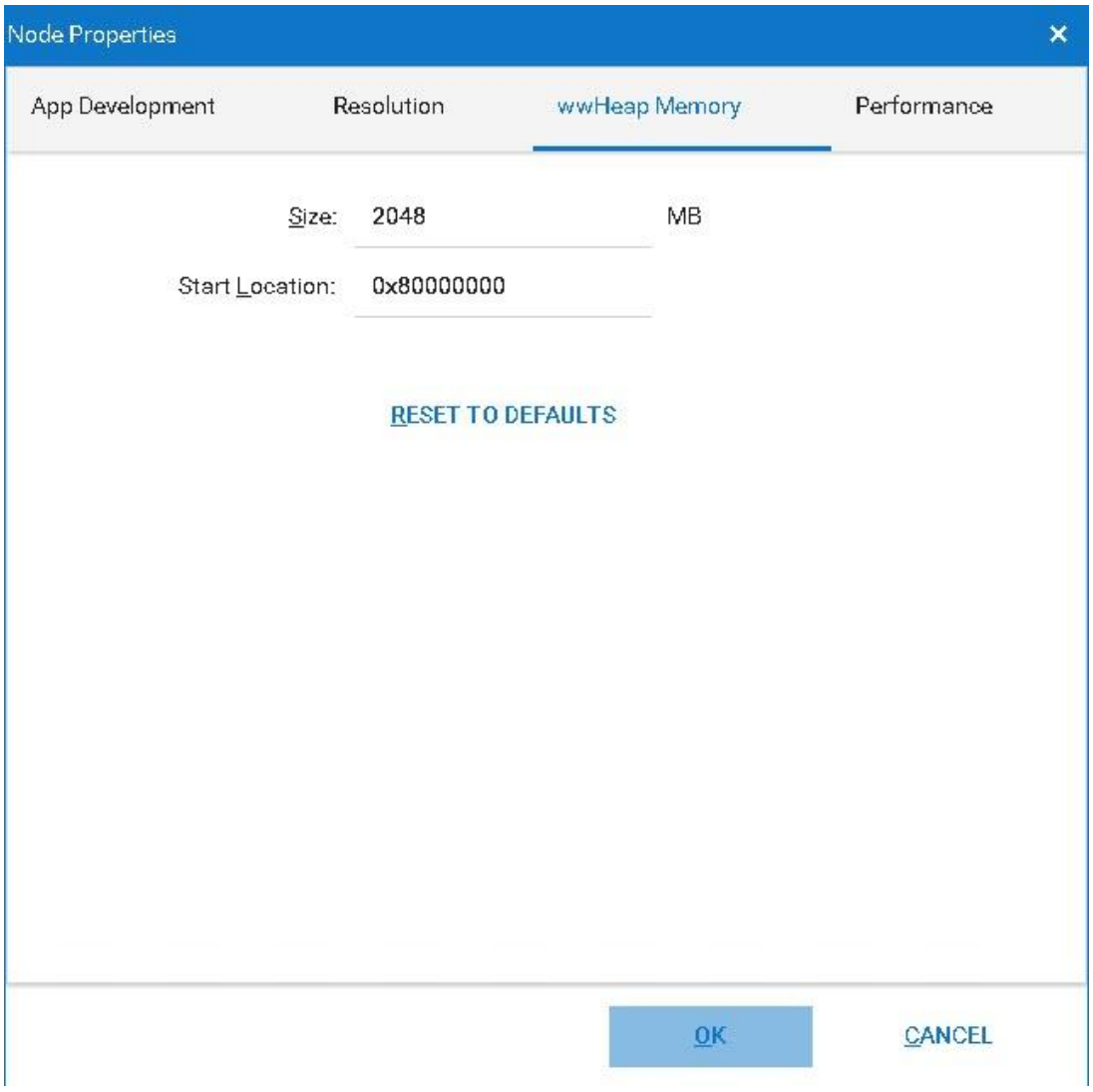
The default locations and allowable location ranges are described in the following table:

Operating System	Default Start Location	Allowable Range
32-bit	0x21000000	0x00010000 to 0x7FFEFFFF
32-bit with the /3GB switch enabled	0x40000000	0x00010000 to 0xBFFEFFFF
64-bit	0x80000000	0x00010000 to 0xFFFEFFFF

To configure wwHeap Memory settings

1. Start **Application Manager**.
2. On the **Tools** menu, click **Node Properties**. The **Node Properties** dialog box appears.

3. Click the **wwHeap Memory** tab.



4. Do the following:
   - In the **Size** box, enter the size of the wwHeap memory. You can enter any value between 1 MB and 2048 MB.
   - In the **Start Location** box, enter the start address.
5. To restore the default values, click **Restore Default**.
6. Click **OK**.

Setting Advanced Formatting Properties

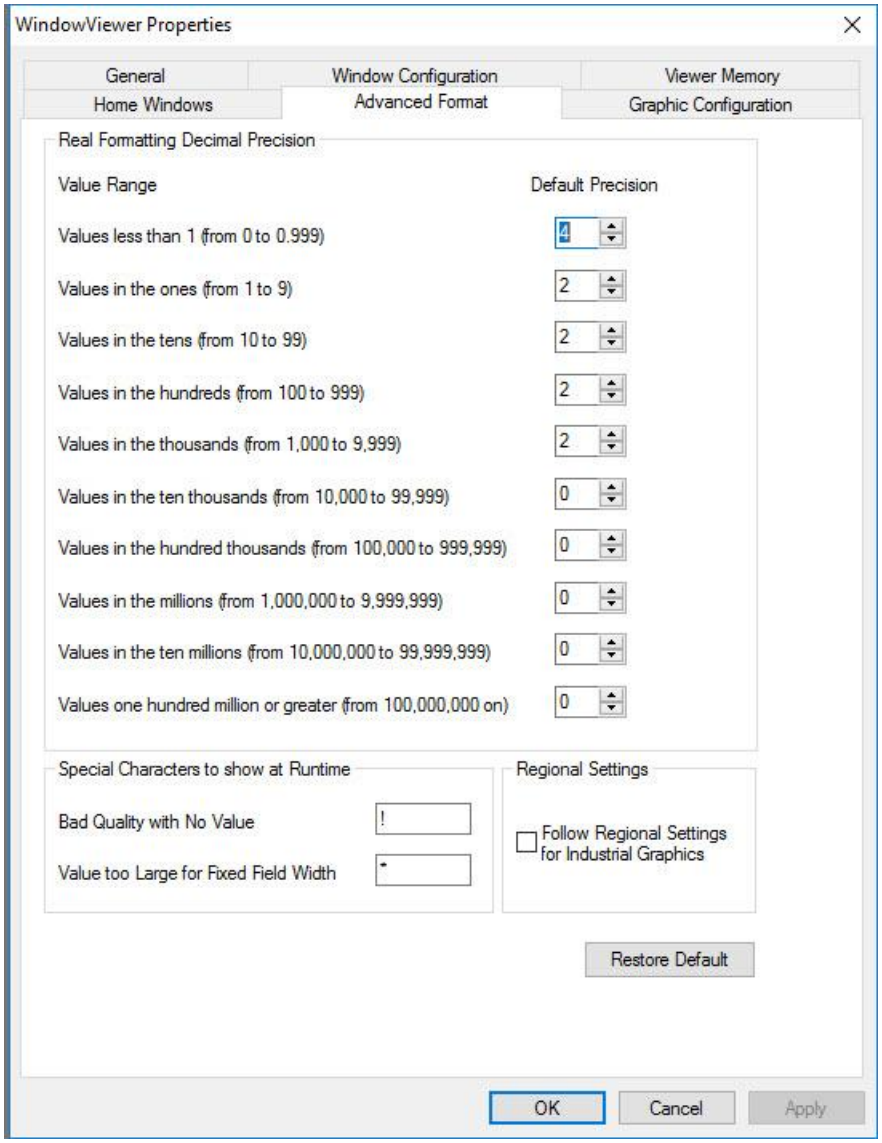
You can configure the number of decimal places to be shown at run time in WindowViewer, based on the size of the value. This feature is available only if you select Real in Value Display or User Input animation.

You can specify the special characters to be shown at run time if the data collected from the field devices is of bad quality or too large to be shown.

Numbers that appear in Industrial Graphics can be shown in the format of the country set as the home location with the Windows Control Panel's **Region** setting. During run time, Industrial Graphic numeric values can be displayed with thousands and decimal separators that match the numeric format of the country specified in the OS Regional Settings.

To set the advanced formatting properties

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



3. In the **Real Formatting Decimal Precision** area, enter the number of decimal places that you want to be shown at run time for each real type number range.
4. In the **Special Characters to Show at Runtime** area, do the following:

- In the **Bad Quality with No Value** box, enter the character you want to be shown at run time when the quality of the data point is bad and there is no last known good value. The default character is **!**. You can enter any ASCII character, except a space.
- In the **Value Too Large for Fixed Width** box, enter the character you want to be displayed at run time when the value is too large to be displayed. The default character is *****. You can enter any ASCII character, except space.

5. To show numbers within Industrial Graphics in the format of the country specified by computer's **Region** setting, select the checkbox in the **Regional Settings** area.

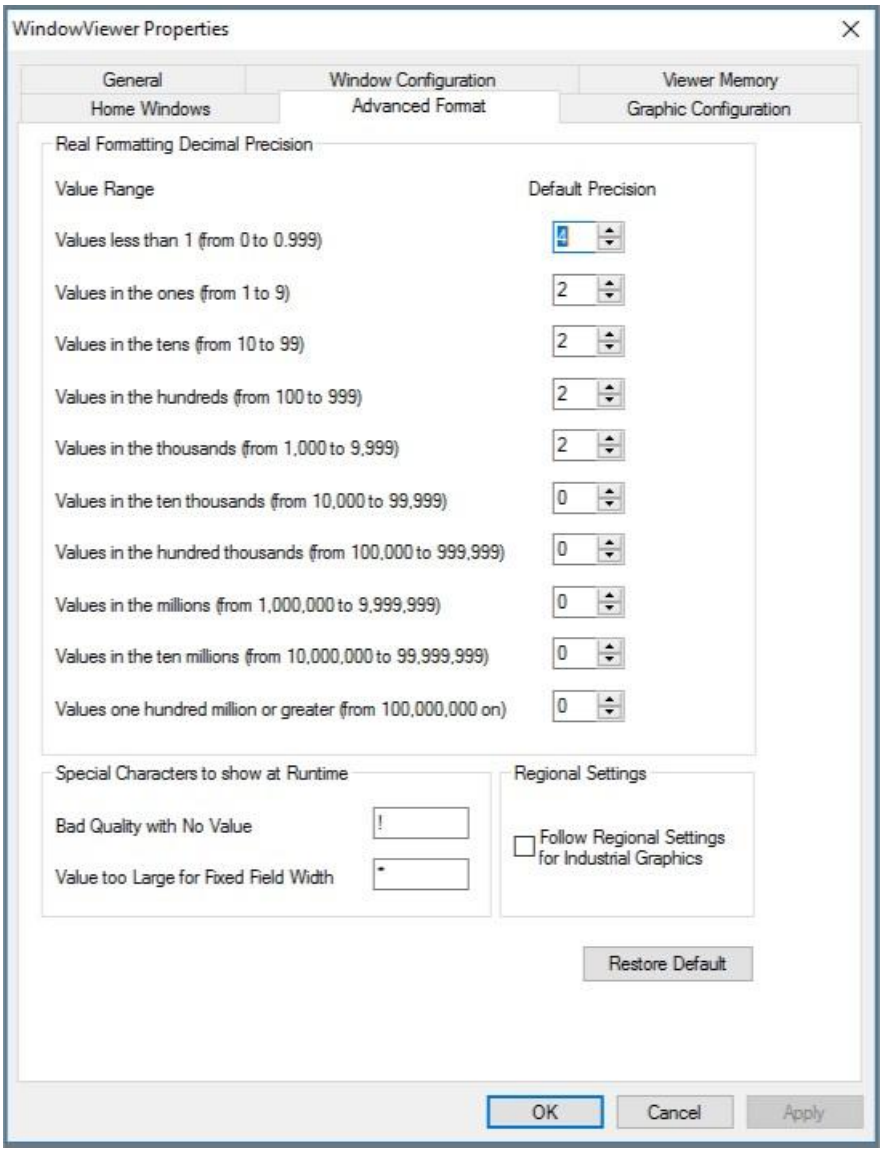
   By default, the **Regional Settings** option is disabled.

   Note: WindowViewer checks the OS Regional Settings only on startup. This means that you may need to restart WindowViewer if you do either of the following:
   1) Select the **Regional Settings** option while WindowViewer is running.
   2) Change the OS Regional Settings while WindowViewer is running with the **Regional Settings** option selected.

6. To restore the default values, click **Restore Default**

Select the Regional Settings WindowViewer Option

InTouch WindowMaker includes a WindowViewer Advanced Format **Regional Settings** configuration option.

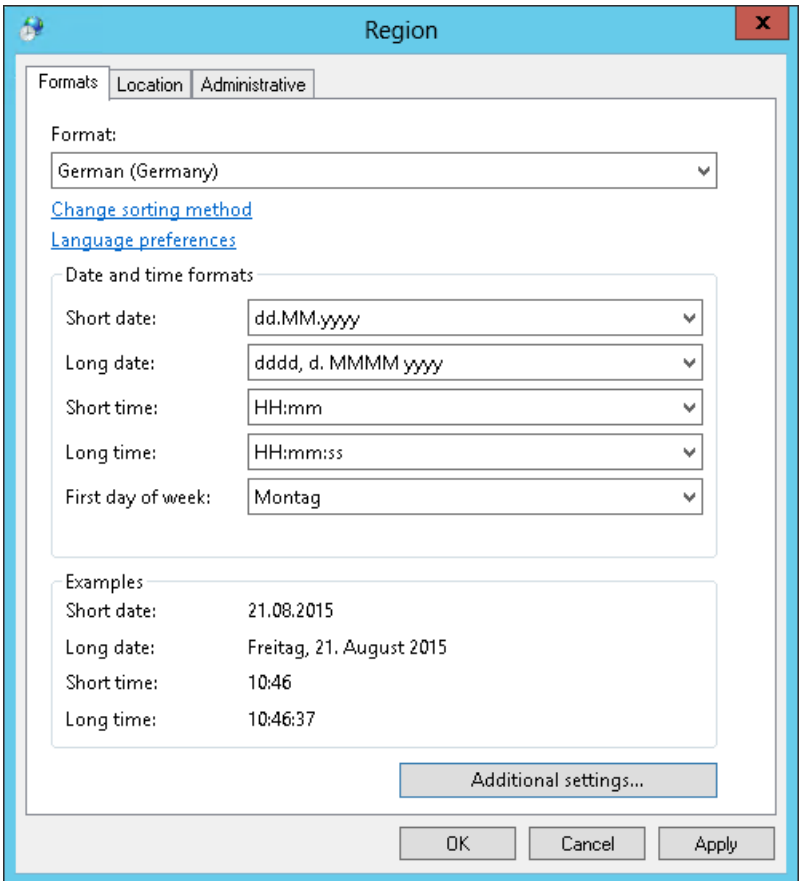


To enable numeric formatting by regional locale, the **Regional Settings** option must be selected during design time to format Industrial graphic numbers to the country selected in the **Region** setting. By default, the **Regional Settings** option is disabled.

Note: WindowViewer checks the OS Regional Settings only on startup. This means that you may need to restart WindowViewer if you do either of the following:
1) Select the **Regional Settings** option while WindowViewer s running.
2) Change the OS Regional Settings while WindowViewer is running with the **Regional Settings** option selected.

Set the Regional Locale of the Computer Hosting the HMI/SCADA Application

To enable numeric formatting by regional locale, the computer running an HMI/SCADA application must have its region set to the country in which you want Industrial graphic numbers to be formatted.



The **Region** setting is accessible from the Windows Control Panel. If you want to display Industrial graphic numbers in a non-U.S. format, select the **Formats** tab and select a country in the **Format** field.

Configuring Core Affinity for WindowViewer in a Terminal Server Environment

When running on a Terminal Services client, WindowViewer can use a CPU (core) other than CPU 0 for its execution, if the computer has multiple processors. This is so that InTouch applications that run in a Terminal Server environment can take advantage of the multi-core capabilities of the Terminal Server. When WindowViewer runs on a Terminal Server console, however, this option is not available. An InTouch application always runs on a single processor, regardless of the number of processors available.

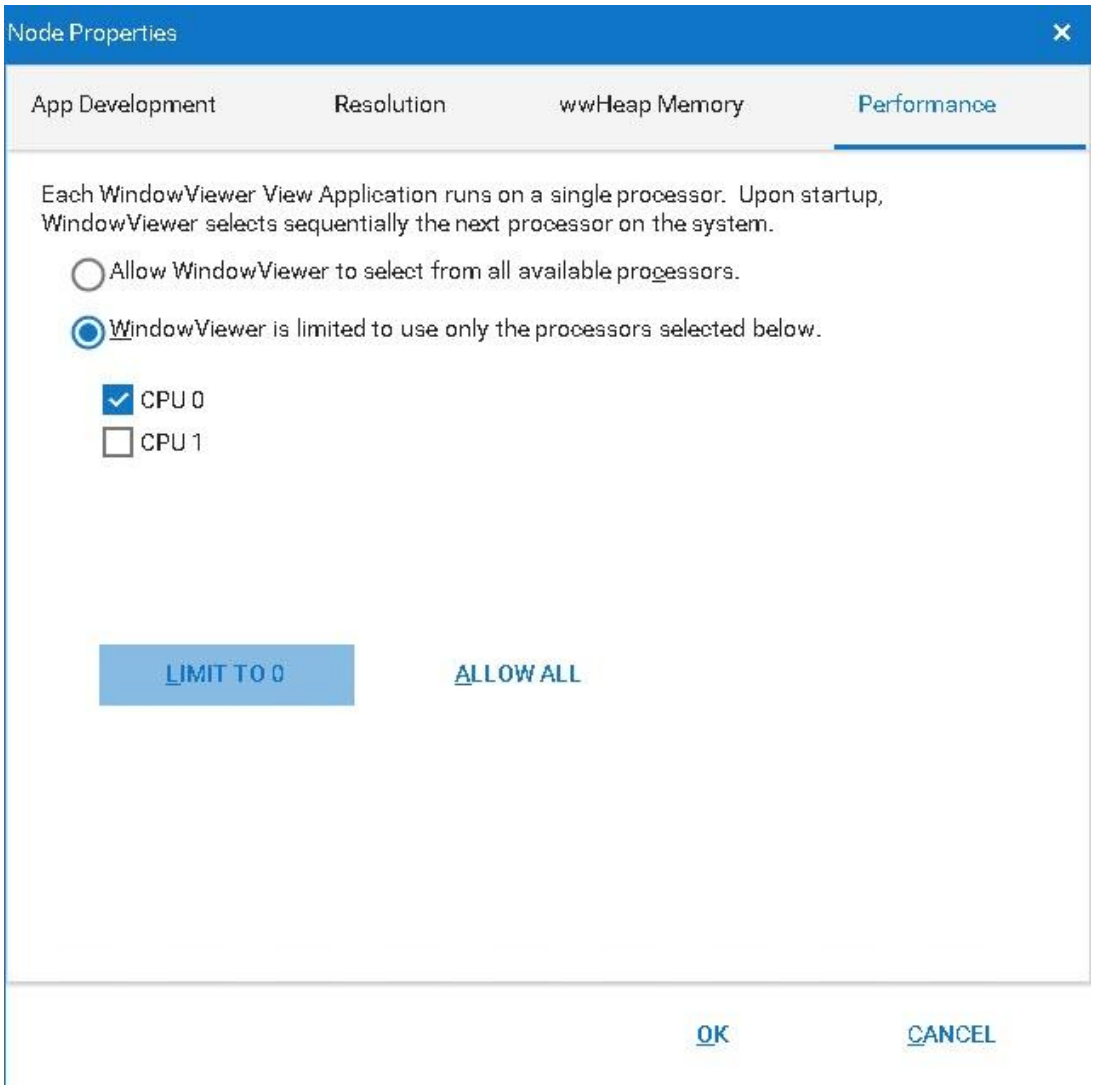
When WindowViewer starts, the system checks if the computer has multiple processors and which processors are allowed to run WindowViewer instances. WindowViewer then checks the processors sequentially and starts running on the processor that has the least number of View instances running on it.

If you have administrative privileges to access the **Performance** tab, you can configure the "pool" of processors from which WindowViewer selects the processor to run on.

You set the core affinity for WindowViewer within Application Manager. Avoid using Task Manager to manually adjust the core affinity for WindowViewer, as the WindowViewer core selection process does not take into consideration the core affinity settings configured in Task Manager.

To configure the processor "pool"

1. Start **Application Manager**.
2. On the **Tools** menu, click **Node Properties**. The **Node Properties** dialog box appears.
3. Click the **Performance** tab.



4. To allow WindowViewer to use any available processor, click **Allow WindowViewer to select from all available processors**.
5. To restrict the processors that WindowViewer can use, click **WindowViewer is limited to use only the processors selected below** and then do any of the following:
   - Make sure the **CPU** check box is selected for each processor you want WindowViewer to be able to run on.

- Click **Limit to 0** to only allow WindowViewer to run on processor 0. When you click this button, the **CPU 0** check box is automatically selected.
- Click **Allow All** to select all check boxes.

You can clear a selected processor at any time and select a new processor from the list. You can also select multiple processors at a time. If you clear a processor check box, the WindowViewer instance continues to run on that processor.

6. Click **OK**. WindowViewer starts on the next CPU based on the other View sessions.

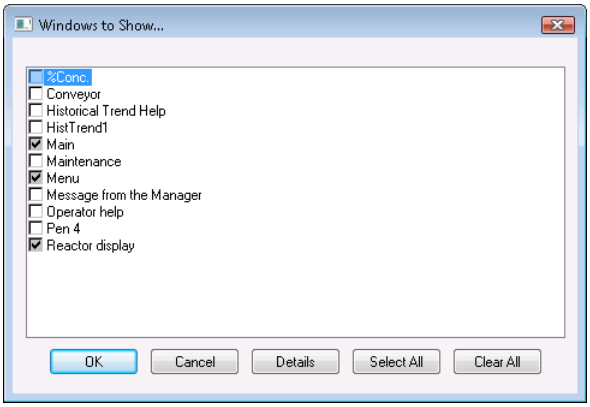
Working with WindowViewer Windows

A typical InTouch application includes at least several windows that operators interact with to manage their industrial processes. Based on the properties you set from the **Window Configuration** tab on the WindowViewer **Properties** dialog box, operators can run standard commands from the WindowViewer **File** menu to open and close windows.

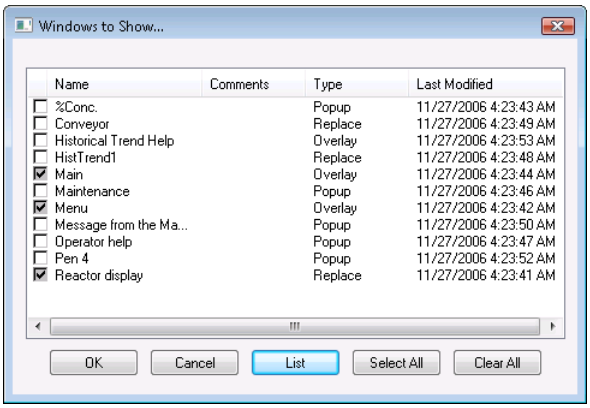
Common Dialog Box Features

If you configured WindowViewer to show the **File** menu, operators can open or close InTouch application windows. When operators click either the **Open Window** or **Close Window** command from the File menu, the respective dialog box for the selected command appears.

The names of all the windows that are applicable for the selected command appear in the list. For example, the **Windows to Show** dialog box appears after clicking on the **Open Window** command.



Click **Details** to change from the list view to the details view. The details show the window's type and the date and time when a window was last modified.



In the details view, you can select and deselect any unopened window by clicking on any portion of its row, not just the check box. The entire row is highlighted when selected.

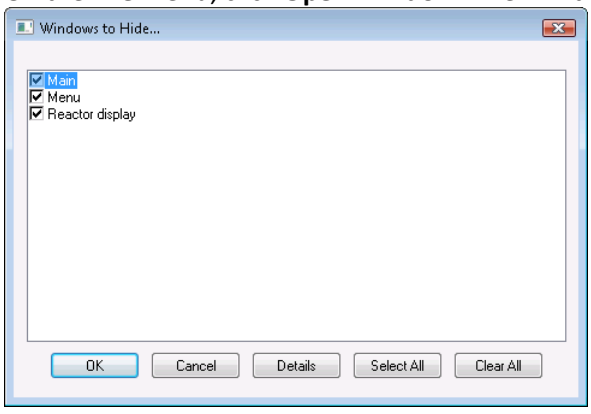
- To open selected windows click **OK**.
- To cancel your selections and close the dialog box, click **Cancel**.
- To return the dialog box to list view, click **List**.
- To select all listed windows, click **Select All**.
- To clear all selected windows, click **Clear All**.
- To sort the list in ascending or descending order, click the column header.

Opening Windows from WindowViewer

Operators can open InTouch application windows if WindowViewer is configured to show the **File** menu.

To open windows from WindowViewer

1. On the **File** menu, click **Open Window**. The **Windows to Show** dialog box appears.



2. Click the check box next to the name of each window that you want to open.
3. Click **OK** to close the dialog box and open windows you selected.

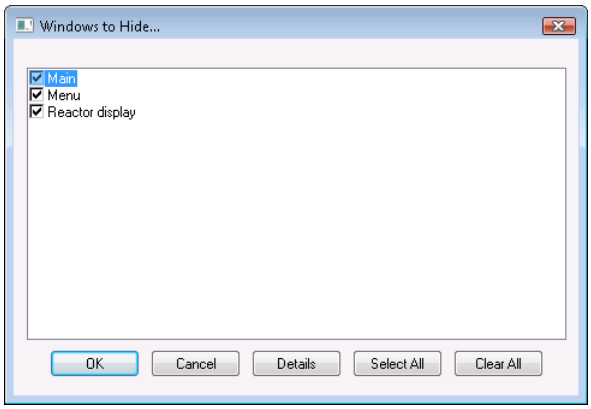
Note: If a "Replace" type window is selected, it closes any windows that it intersects.

Closing Windows from WindowViewer

Operators can close InTouch application windows if WindowViewer is configured to show the **File** menu.

To close open windows

1. On the **File** menu, click **Close Window**. The **Windows to Hide** dialog box appears.



2. Click the check box next to the name of one or more windows that you want to close.
3. Click **OK** to close the windows you selected.

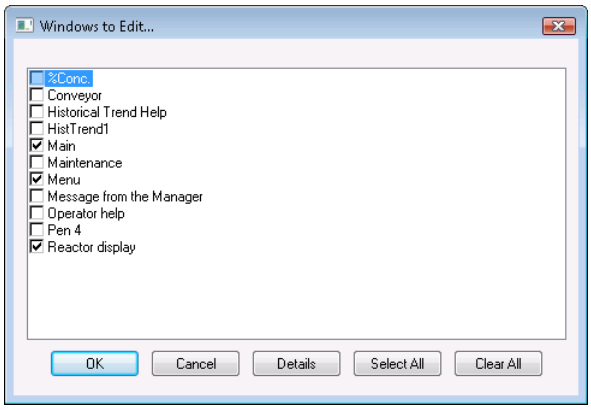
Transferring from WindowViewer to WindowMaker

When you develop an InTouch application, you can easily transfer between WindowMaker and WindowViewer by clicking the WindowMaker command in the **File** menu, or the **Development** command in the toolbar. This is called fast-switching.

Fast-switching is for rapid development testing only. Do not use it in a production environment. You can hide the command for switching to WindowMaker.

To transfer from WindowViewer to WindowMaker

1. On the File menu, click WindowMaker. The **Windows to Edit** dialog box appears.



2. Click the check box next to the name of each window that you want to open when you transfer to WindowMaker.
3. Click OK to close the dialog box and transfer to WindowMaker.

Note: If the application developer selected the **Close WindowViewer** option when WindowViewer's properties were configured during development, WindowViewer automatically closes when you transfer to WindowMaker.

Working with Keyboard, Mouse and Touch Gestures to Pan and Zoom at Run Time

Frame windows allow you to pan and zoom on Industrial Graphics at run time. This functionality is enabled by the **InteractionMode** property in WindowMaker.

See the *InTouch HMI Visualization Guide* for additional information on configuring frames in WindowMaker.

Zooming at Run Time

You can zoom in and out on the frame contents at run time. Be sure the frame has the correct pan and zoom property enabled. You cannot zoom in above 5000% or below 100%.

You can edit a symbol's **ZoomPercent** property to change the visibility of an symbol or element at run time. For example, add the following to a Visibility animation to allow you to dynamically change the symbol according to the zoom percent level.

```
ZoomPercent => 200
```

Note: You can write to this property at run time.

When **ZoomPercent** is set for a symbol, the symbol will be zoomed to the set percent at the center of the viewable area.

When **ZoomPercent** is set for a symbol’s element, the symbol will be zoomed to the set percent but will center on the element.

The following script is an example of **ZoomPercent** set for an element:

```
TextBox1.ZoomPercent = 500
```

To zoom with mouse gestures:

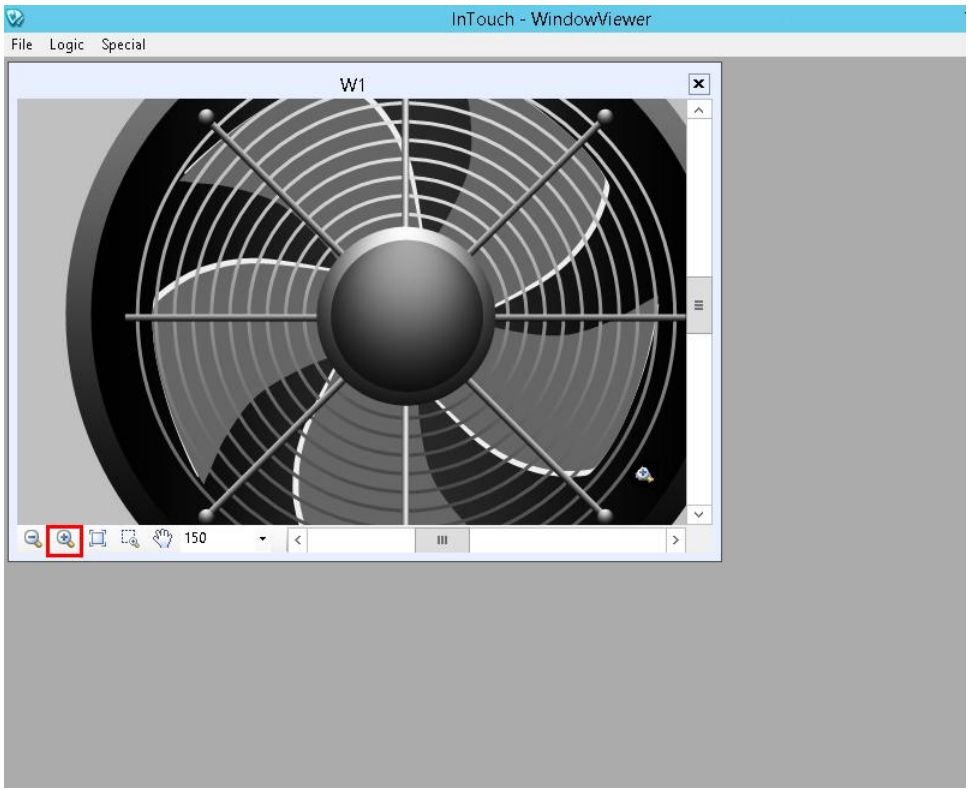
Do the following

- Press the "Ctrl" key and scroll up with the mouse wheel to zoom in on the frame contents. The contents will zoom in from the current position of the mouse pointer.

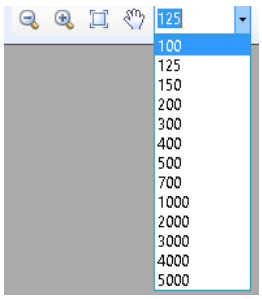
  Note: You cannot zoom in on the frame contents if your mouse pointer is outside the frame.

- Scroll down with the mouse wheel to zoom out on the frame contents.

- Select the zoom in and zoom out icons from the Pan and Zoom Control Toolbar. You must then left click on the contents of the frame to zoom in.



- Double left mouse click on the frame contents to restore the zoom level to 100%
- Use the Zoom Level combobox to select a predefined zoom level:



- Select the Rubber Band Zoom icon from the Pan and Zoom Control Toolbar to select a specific area to zoom in on.

To zoom with keyboard gestures:

Do the following:

- Press "Ctrl" and "+" keys together to zoom in
- Press "Ctrl" and "-" keys together to zoom out

To zoom with touch gestures:

Do the following:

- Place two fingers on the screen and expand them to zoom in
- Place two fingers on the screen and contract them to zoom out
- A double-tap will restore the zoom level back to 100%

Zoom Limitations

The following limitations apply to run time zoom functionality:

- Zooming for the Windows Common Controls and client controls is only supported within 500%. These controls include:
  - Customized controls: radio button group, checkbox, edit box, combo box, calendar, datetime picker, and listbox
  - Embedded Alarm Client and Trend Client Controls
  - Third party client controls
- Windows Controls can override mouse, keyboard and touch input
- InTouch will not override custom fonts in Windows Controls

Panning at Run Time

Configure the applicable symbol properties to enable pan functionality within a frame at run time. You can pan at run time using mouse and touch gestures. Panning with keyboard gestures is not supported.The zoom level on the graphic must be greater than 100% to pan at run time.

To pan using mouse gestures

Do either of the following:

- Hold down the center mouse wheel.

  The pan hand will display
- Select the pan hand from the Pan and Zoom Control toolbar. Hold down the left mouse button and move the pan hand to pan the display.

  The display will pan until the mouse button is released.

To pan using touch gestures:

1. Press one finger on the frame content.
2. Move your finger across the screen to pan as needed.

Note: For both panning methods, the horizontal and vertical scroll bars will adjust in accordance with the pan directions.

To pan and zoom simultaneously using touch gestures:

1. Place two fingers on the screen and move your fingers downwards, to the left and right to adjust the center of the zoomed content.
2. Use one finger to pan over the frame content.
3. Place your second finger on the frame content to zoom at the same time.

Panning Limitations

The following limitations apply:

- panning is not supported using keyboard gestures
- Window Controls can override mouse, keyboard and touch input. As a result, panning may be disabled over areas with Window Controls.

Animation Support for Touch Gestures

All action scripts configured for touch support should function the same no matter the zoom level. All interaction and visualization animations behave the same while frame content is zoomed in as when frame content is at its standard view. Interaction animations will function properly for touch.

Note: Industrial Graphic pop ups shown by the **ShowSymbol** animation or **ShowSymbol** script function will have pan and zoom enabled by default. However, you cannot disable this configuration.

The following table lists action scripts commonly configured for touch support.

Action Script	Touch Triggered
On Left Down	Touch down
While Left Down	Touch down and slide
On Left Up	Touch up
On Left Double Click	Double tap
On Right Click	Touch down and hold
On Right Up	Touch down and hold for square and execute on release
On Right Double Click	Not supported
While Right Down	Touch down and hold square and slide a bit while pressing
On Center Click	Not supported
While Center Down	Not supported
On Center Up	Not supported
On Center Double Click	Not supported

Note: When touch interaction is used on an area with animation, the animation will take precedence over panning actions and panning actions will be ignored. If one finger retains touch interaction, any subsequent touch points will be ignored.

To enable panning in this scenario,select the **Pan** icon in the **Pan and Zoom Toolbar**.

Touch Gesture Limitations

Some limitations apply to touch gesture functionality for run time panning and zooming. The below functionality is not supported:

- Scaling fonts for Windows Common Controls

Note: Additionally, a symbol with an embedded windows control uses different mechanisms for scaling than a symbol without a control. A symbol with an embedded control has a maximum zoom limitation of 500%, while a symbol without a control can zoom up to 5000%. The advantage of a symbol without a control is smoother scaling. An additional limitation of using symbols with embedded controls is, while zooming is in progress, the control will flicker. This is particularly visible while zooming with touch gestures.

Using the ShowGraphic() Function with Frame Windows

You can run the ShowGraphic() script function to change the symbol associated with a frame window at run time. Specify the frame window name and associated symbol as in the following example:

```
Dim graphicInfo as aaGraphic.GraphicInfo;
graphicInfo.Identity = "InTouch:FrameWindow01";
graphicInfo.GraphicName = "Symbol_002";
ShowGraphic( graphicInfo );
```

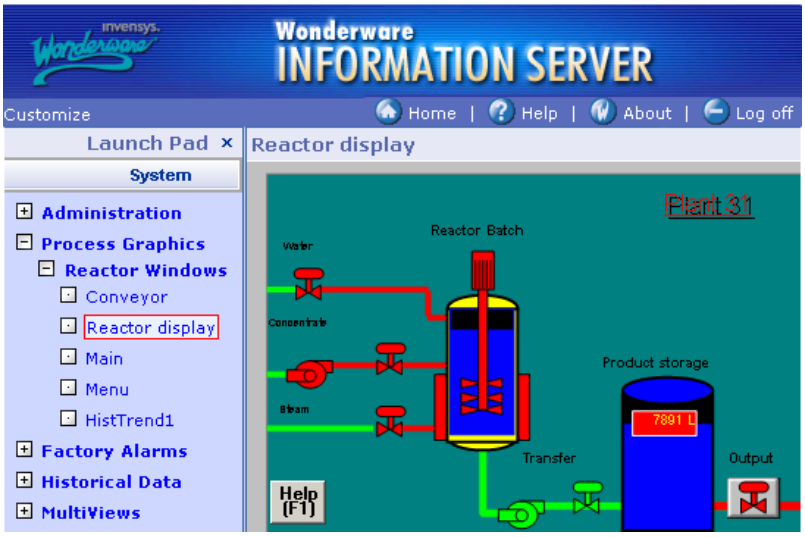
"Symbol_002" will display in "FrameWindow01" at run time regardless of the frame window's design time configuration. You can use a ShowGraphic script to host different symbols in the same window.

Limitation:

Graphic caching occurs only for the symbol currently shown in the frame window at the time the window is closed and cached. Replaced symbols will not be cached.

Running InTouch Windows over the Internet

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



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

- Process visualization

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

- Data interaction

  You can use the 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 Information Server portal to show InTouch real time and historical alarm data.

- Historical Data Display

  You can use the Information Server portal to show InTouch historical data saved in a Historian database.

- Table Weaver display

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

For more information, see the Information Server documentation.

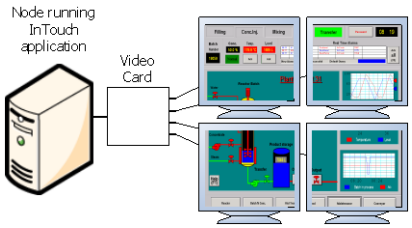
Chapter 12

Setting Up a Multi-Monitor System

About Setting Up a Multi-Monitor System

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.

While running an InTouch application, you can move the mouse between monitors and drag windows from one monitor to another. Also, in some multi-monitor configurations you can show an entire InTouch application window across all monitors, as shown in the following figure.



Multi-Monitor Configurations

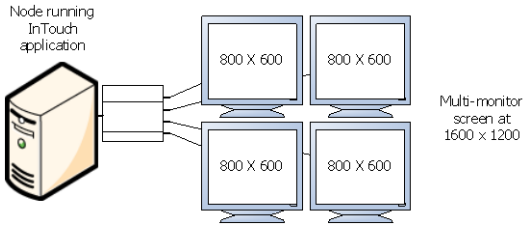
You can use two basic multi-monitor configurations.

- Single video card
- Multiple video card

Each configuration has unique hardware, software, and configuration requirements. Also, each configuration supports a different set of multi-monitor features.

Single Video Card Configuration

In the single video card configuration, the computer has a single video card installed with multiple output ports connected to monitors.



The composite screen resolution is the sum of the individual horizontal and vertical resolution of each monitor. For example, a popular video card connects four 17 inch monitors stacked as a cube: two on the bottom and two on the top. In the previous figure, each monitor runs at a screen resolution of 800 x 600 pixels. The composite virtual screen resolution is 1600 x 1200 pixels.

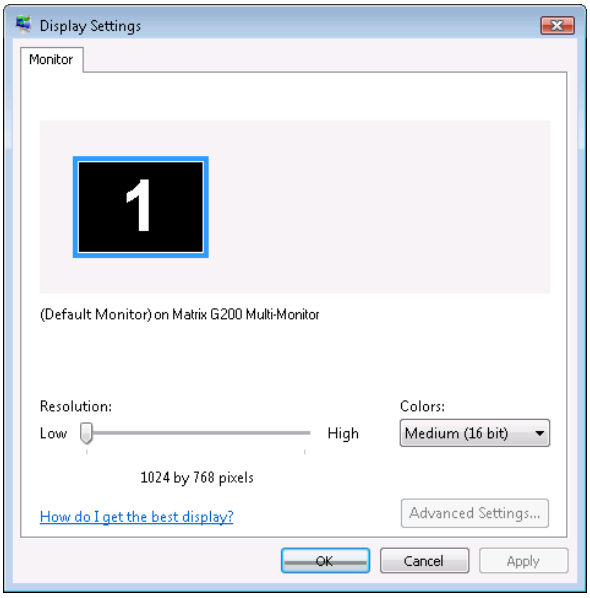
Characteristics of a Single Card Configuration

Single video card drivers have the following characteristics:

- The single video card drives all monitors simultaneously to create a single, large screen.
- The properties of all attached monitors can be configured using a single set of screen values.
- The composite screen shows the Windows taskbar across all of the monitors in the bottom row of the configuration.
- Windows applications can be maximized to fit all monitors.

Characteristics of Single Card Drivers

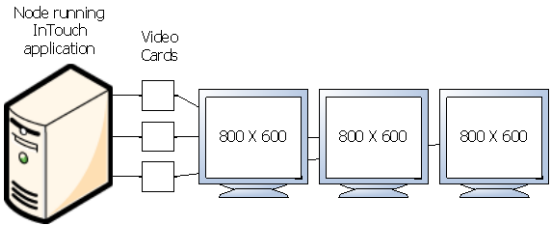
The following figure shows the Windows **Display Properties** dialog box to configure the driver for all monitors connected to a single video card with multiple output ports.



In this figure, the resolution setting is for four monitors arranged side by side in a single row. The resolution for each monitor is 1024 x 768. Added together, the composite screen resolution is 4096 x 768. You only need to configure a single monitor's resolution, color depth, and refresh rate. The resolution setting applies to all monitors connected to the single video card.

Multiple Video Card Configuration

In the multiple video card configuration, the computer has multiple video cards installed. Each video card connects a single monitor to the computer running an InTouch application.



Characteristics of a Multiple Card Configuration

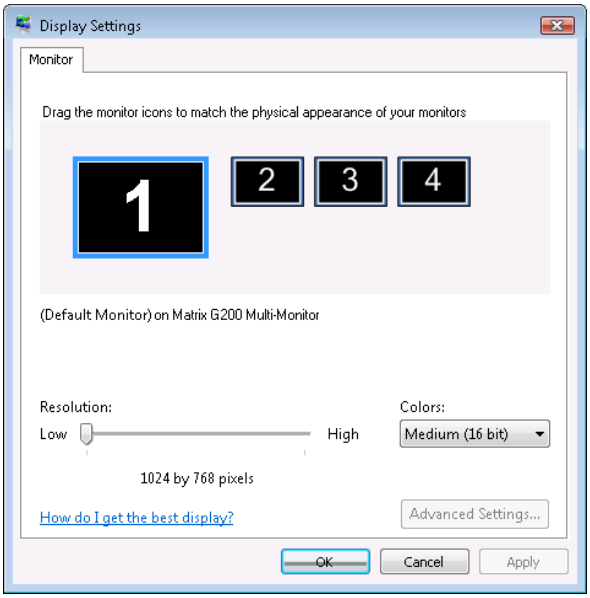
Dynamic Resolution Conversion (DRC) works with other distributed features to provide independence from screen resolution restrictions. In a NAD architecture, you create and maintain an InTouch application on a development node and then copy it to several View nodes. DRC allows all view nodes to show the application, even if the nodes are running at different screen resolutions.

DRC enables each View node to scale the application to a number of user-defined options, including a custom resolution. This scaling takes place while WindowViewer compiles the application and does not require WindowMaker. Because each View node can use a different DRC setting, you must configure each individual View node.

DRC makes it easy to support multi-monitor systems. Simply select from the DRC resolution conversion options to show an InTouch application over the entire composite screen or just a portion of it.

Characteristics of Multiple Card Drivers

The following figure shows the Windows **Display Properties** dialog box to configure the drivers for all monitors connected to individual video cards installed on the computer running an InTouch application.



You click a numbered rectangle in the **Display Properties** dialog box to select the monitor you want to configure. You arrange the numbered rectangles to match the physical placement of the monitors. Screen resolution, color depth and refresh rate apply only to the monitor you select.

Planning a Multi-Monitor Application

To set up multiple monitors for your application, you must:

- Choose a multi-monitor video card
- Determine the application screen resolution
- Determine the number of monitors to display the application
- Determine the placement of application windows

Choosing a Multi-Monitor Video Card

Technical Support can provide you with a list of recommended video cards that support multi-monitor InTouch applications.

Before you select a video card, get more information from Technical Support to answer the following questions:

- What versions of InTouch does the video card support?
- Does the card support a single or a multiple card configuration?
- What are the recommended drivers for the video card?
- What are the recommended configuration settings for the video card?

Determining the Application Screen Resolution

Determining your overall screen resolution and knowing the exact size of your viewing area simplifies the process of creating an application for a multi-monitor environment.

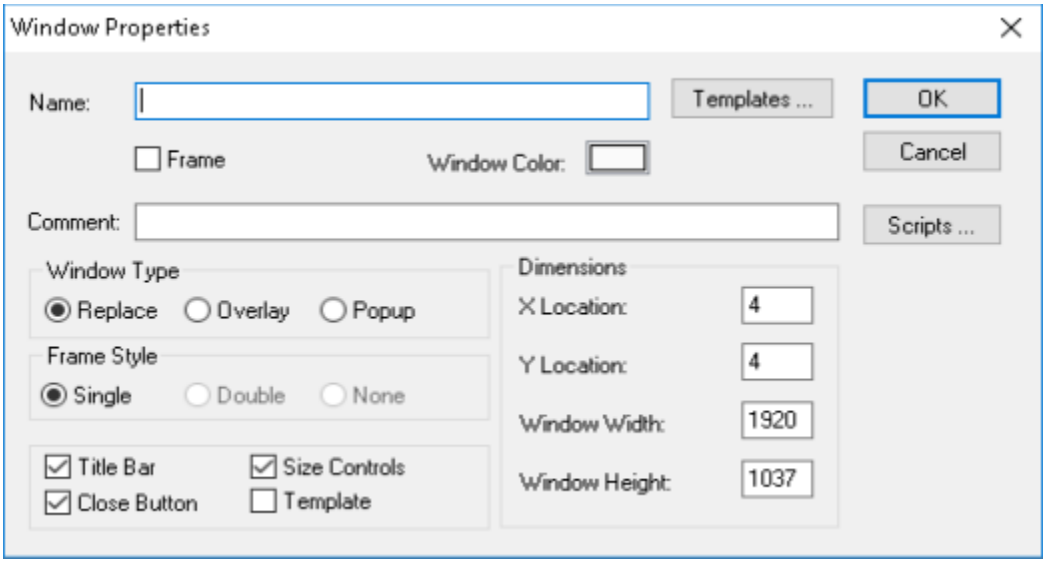
Create a drawing that shows the overall monitor configuration. The drawing should show the resolution of each monitor and the combined resolution of all the monitors together. This drawing helps you visualize the horizontal and vertical pixel range for each monitor.

For example, if you have a composite screen composed of two horizontal monitors with a screen resolution of 800 x 600, then the top left pixel location of the second monitor would be at pixel 800 x 0. The screen pixel count goes from 0 to 799 for the first monitor and 800 to 1599 for the second. Using the drawing as a guide, you can determine the placement of application windows on the composite multi-monitor screen.

Determining the Number of Monitors to Display the Application

You can simplify the effort to create a multi-monitor InTouch application by using a development environment similar to the production environment. Using a multi-monitor development environment may not be possible in all cases. When you only have a single monitor attached to the computer used to develop your InTouch application, you can still build a multi-monitor application by developing the windows and configuring the windows dimensions and locations to your estimated display needs.

Use the WindowMaker **Window Properties** dialog box to modify the characteristics of a window. You right-click on the name of the window listed in WindowMaker's **Windows &Scripts** pane, and then click **Properties** in the shortcut menu to show the **Window Properties** dialog box.



The **X Location** and **Y Location** values determine the horizontal and vertical pixel placement of a window's top left corner on a screen. The origin of horizontal and vertical pixel scales is at the top left corner of a screen.

The **Window Width** and **Window Height** settings determine the overall size of the window. For example, you can configure a window with the following settings:

- X location = 1024
- Y location = 0

- Window Width = 1024
- Window Height = 768

The multi-monitor configuration consists of four monitors arranged in a single horizontal row. Each monitor has a resolution of 1024 X 768. The overall composite screen resolution is 4096 X 768.

By setting the window's horizontal origin to 1024 and vertical origin to 0, you force this window to appear on the second monitor during run time. The window covers the entire screen surface of the second monitor.

Determining the Placement of Application Windows

You can use several different configurations when developing InTouch windows for a multi-monitor environment.

Windows Show in a Forced Location

One method is to simply develop and force windows to show where specified. Make sure that WindowViewer is maximized across the total viewing area of all monitors. This allows the InTouch application windows to show on specified monitors.

You can use InTouch security features to deny access to the Windows desktop.

Windows Are Manually Moved

Another option is to develop an application where windows are manually moved to the monitors of choice, allowing a single application to run on different monitor configurations. This involves the following:

- All windows in the application must be of type **Popup**.
- The main WindowViewer parent window can be small and not covering all monitors. However, you cannot use InTouch security for denying access to the Windows desktop in this configuration because InTouch is not maximized.

In this configuration popup windows are used which can be easily moved to any monitors, regardless of the main WindowViewer parent window location. Popup windows do not have to remain within the parent WindowViewer window. You can shrink the size of the main window and move it to a corner of a monitor, allowing all the popup windows to be moved freely to the monitors of choice.

Windows Are Placed Automatically Based on Environment

The final method includes an additional step added to the above method. The step allows an application to automatically place windows based on the environment used. This is the most complicated of configurations and requires extensive scripting and planning.

In this configuration, the ShowAt() and ShowTopLeftAt() script functions dynamically place windows based on a default set of coordinates and calculations. This can be configured many different ways depending on your application requirements.

Developing a Multi-Monitor InTouch Application

You must assign values to selected parameters in the InTouch.ini and Win.ini files to support multi-monitors. These parameters enable you to place InTouch system dialogs and keypads in the proper locations on the composite screen.

Configuring Multi-Monitor Parameters

To enable multi-monitor support, you add a set of InTouch parameters to the Windows Win.ini file. These parameters enable multi-monitor support for the node running the InTouch application and the resolution of each monitor.

To configure the multi-monitor settings on a node

1. Edit the Win.ini file located in the Windows folder of the computer running the InTouch HMI software.
2. Locate the [InTouch] section within the Win.ini file and add the following parameters:

Parameter	Description
MultiScreen=1	A value of 1 enables multi-monitor mode. A value of 0 disables multi-monitor mode.
MultiScreenWidth=*nnnn*	Width of a single screen in pixels.
MultiScreenHeight=*nnnn*	Height of a single screen in pixels.

For example, if you want to show your InTouch application with a screen resolution of 2560 x 1024 on two horizontal monitors, enter the following:

```
[InTouch]
MultiScreen=1
MultiScreenWidth=1280
MultiScreenHeight=1024
```

Configuring Screen Resolution Conversion

You can specify a parameter value to maintain the current resolution of InTouch application windows when migrating between nodes running different screen resolutions.

The ScaleForResolution parameter value determines whether application windows (*.win) are automatically scaled by WindowMaker after the display resolution changes on the computer running WindowViewer. The ScaleForResolution parameter does not affect the resolution of WindowViewer dialog boxes.

To configure screen resolution conversion on a node

1. Edit the InTouch.ini file of the computer running InTouch.
2. Add the ScaleForResolution parameter to the file.

```
ScaleForResolution=1
```

When set to 0, resolution conversion is disabled.

When set to 1, resolution conversion is enabled.

Note: If the ScaleForResolution parameter is not added to the InTouch.ini file, the default value is enabled (ScaleForResolution=1). When you disable the parameter (ScaleForResolution=0), you are still prompted to convert the resolution. But, the resolution conversion does not occur.

Deploying the Application and Verifying Multi-Monitor Settings

The ScaleForResolution parameter becomes particularly important when you develop an application on a single monitor system that is intended to run on a multi-monitor system. The value assigned to the ScaleForResolution parameter determines whether the application can be scaled when moved from one environment to the other.

Important: It is recommended that you make a backup copy of the application before moving it to an different environment.

For example, if an application is developed on a computer with a single monitor with a resolution of 1024 x 768 and is intended to run on a system with four monitors in a side-by-side configuration with a total resolution of 4096 x 768, this requires an application conversion.

When you deploy the application on the multi-monitor system, a message appears prompting you to convert the application.

If the ScaleForResolution .ini setting is configured, you still see this message but the application is not converted and can then be run as designed. Simply click **Yes** to continue startup.

If the .ini setting is not configured, the InTouch HMI converts and scale all of the graphics and windows in the application to the new resolution. Doing so stretches and enlarge all windows and graphic displays, thus creating some unwanted results.

Important: Make sure that the multi-monitor Win.ini parameter settings are also configured on the destination computer before running your application. Win.ini settings do not automatically transfer with an InTouch application.

Verifying Multi-Monitor Support During Run Time

You can download an optional script function from the Technical Support script library that verifies if the local node running the InTouch application provides multi-monitor support.

The WWMultiMonitorNode() function determines if the node supports multi-monitors and the number of monitors attached to the node.

Typically, you run the WWMultiMonitorNode() function from a QuickScript to determine the number of monitors assigned to the node running the InTouch application.

The following example shows an example of a QuickScript statement with the value of the WWMultiMonitorNode() function assigned to an InTouch integer tag. The QuickScript can be set to run when the application starts in WindowViewer.

```
{MultiMonitors defined as an integer tag}
MultiMonitors = WWMultiMonitorNode();
{After executing this function Result = 4}
```

WWMultiMonitorNode() reads the MultiScreen parameter specified in the node's Win.ini file. The WWMultiMonitorNode() function returns either a 0 or a positive integer.

- 0 return value

  WWMultiMonitorNode() returns a 0 if MultiScreen=0 or if the MultiScreenWidth or MultiScreenHeight parameters are set incorrectly to 0 in the [InTouch] section of the Win.ini file.

- Positive integer return value

  WWMultiMonitorNode() returns the number of monitors in the multi-monitor configuration if MultiScreen=1 and the MultiScreenWidth and MultiScreenHeight parameters have been assigned correct screen resolution values.

Chapter 13

Using InTouch on a Tablet PC

About Using InTouch on a Tablet PC

Windows XP Tablet PC Edition and InTouch comes pre-installed with a line of portable Tablet PCs. These rugged Tablet PCs are waterproof and vibration resistant, making them suitable for most industrial environments. Tablet PCs are also available from other computer manufacturers that can run InTouch applications.

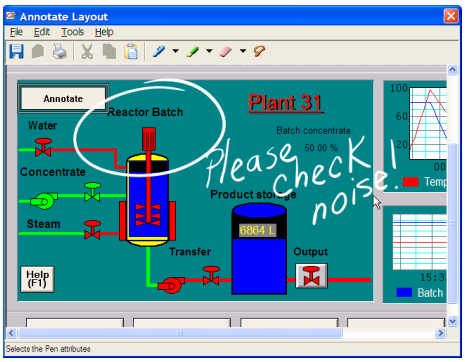
Operators carry a Tablet PC with them as they move around their plant. The Tablet PC runs an InTouch application that represents their actual plant processes. Using a pen that acts as a screen pointer or an input device, operators select InTouch objects on the screen or as a keyboard substitute to write notes directly on the screen.

Operators can write notes and annotate a running InTouch application with direct observations about their actual plant processes.

Annotating and Sending Visualization Screens as E-mail Messages

Use the AnnotateLayout() script function to capture screens shown on a Tablet PC. The AnnotateLayout() function is available only when InTouch runs on a Tablet PC using the Windows XP Tablet PC operating system.

The AnnotateLayout() function 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 contains a toolbar and menu options. The dialog box shows the screen capture in its client area. You can annotate the image using various drawing tools, and save, print, or send the screen capture in an e-mail message.

Making Window Annotations

To make annotations to the window, use the following tools:

- **Pen:** To draw and write comments.

- **Highlighter:** To highlight areas of the window using a semi-transparent color.

- **Eraser:** To delete parts of an annotation.

Each of these tools has certain options such as size, color, or transparency.

- To set these options, click the downward arrow next to each tool's icon and then click the command for the option.
- To restore these options to their default settings, on the **Tools** menu, click **Restore Defaults**.

Selecting, Copying, and Deleting Window Annotations

You can select, copy, and delete annotations that you make in the window.

To select annotations

1. Click the **Lasso** icon in the toolbar.
2. While holding down the stylus button, draw an area around the annotations that you want to select.

   You can now cut, copy or delete the selected annotations.

To cut, copy, and paste annotations

- Use the standard Windows Cut, Copy, and Paste commands.

To delete annotations

- Do any of the following:
  - To delete all annotations on a window, on the **Edit** menu, point to **Clear** and then click **All**.
  - To delete annotations that you selected using the lasso, on the **Edit** menu, point to **Clear** and then tap **Selection**.

Saving, Printing, and E-Mailing an Annotated Window

After you make annotations to a window, you can save it as an image file, print it, or send it as an e-mail attachment.

You only need to configure the e-mail server one time.

To save an annotated window

1. On the **File** menu, click **Save**. A standard Windows **Save As** dialog box appears.
2. Enter a name and format for the file and click **OK**.

To print an annotated window

1. On the **File** menu, click **Print**. A standard Windows Print dialog box appears.
2. Specify any printing options and click **OK**.

To send an annotated window as an e-mail attachment

1. On the **Edit** menu, click **E-Mail Configuration**. The **E-Mail Configuration** dialog box appears.
2. Enter the host name of the SMTP e-mail server to use for sending e-mail. If you are unsure, ask your administrator for assistance. Click **OK**.
3. On the **File** menu, click **E-Mail**. The **E-mail** dialog box appears.
4. Enter sender and recipient addresses and write a message. An image file of the annotated window is automatically added as an attachment.
5. Click **Send** to send the e-mail message.

AnnotateLayout() Function

Shows the **Annotate Layout** dialog box, where you can annotate the current view screen from where this script function is called. This function is only supported on the Windows XP Tablet PC Edition operating system.

Category

System

Syntax

```
AnnotateLayout()
```

Remarks

When **Annotate Layout** dialog box appears, the screen image of WindowViewer is captured. Use the dialog box to:

- Annotate the screen capture using the pen in conjunction with tool bar and menu item settings.
- Save the image and the annotation as a .gif or .jpeg file.
- Print the image and the annotation (if a printer is configured).
- Send the image and the annotation as an attachment of an e-mail message (if SMTP is configured).

Changing Screen Orientation

If the Tablet PC is running in tablet configuration, and WindowViewer is configured to dynamically change the application resolution to the screen resolution, an InTouch application developed in landscape mode is scaled to fit portrait mode.

If WindowViewer is not configured to dynamically change the application resolution, the landscape application is not scaled. In this case, some InTouch windows can be truncated on the Tablet PC.

When switching from one configuration to another, the screen resolution is switched by default. For example, if the tablet PC running in laptop configuration is switched to tablet configuration, the screen orientation switches from landscape (1024 x 768) to portrait (768 x 1024) mode.

Appendix A

Customizing Applications Settings from the INTOUCH.ini File

The first time you run an InTouch application, the INTOUCH.ini file is created in the application folder. When the INTOUCH.ini file is created, values are assigned to a set of parameters that determine the operating characteristics of an individual InTouch application.

As you continue configuring your application from WindowMaker or WindowViewer, new INTOUCH.ini parameters are created or existing parameters are modified. For example, when you configure logging from the WindowMaker **Historical Logging Properties** dialog box, logging parameters are added to the INTOUCH.ini file.

Other configuration parameters must be manually added to the INTOUCH.ini file.

After you customize your application, you can copy the INTOUCH.ini file to a different application's folder. This way, you can create consistent operating characteristics for your applications without having to repeat all customization steps.

Custom INTOUCH.ini Parameters

The following table lists a set of parameters that you can manually enter in the INTOUCH.ini file to provide additional custom properties to your InTouch applications.

INTOUCH.ini Parameter	Purpose
16PenTrendDrawMode	Determines whether a 16-Pen Trend shows data values in average mode or min-max mode.
ApplicationThumbnail	Sets the name of the application thumbnail file.
AllowPubAppEdit	Sets the application flag, so that it can edit a published application. If the value is 1, you can edit a published InTouch file.
CommentRetentive	Determines whether run-time changes to the Alarm Comment field are saved.

INTOUCH.ini Parameter	Purpose
ForceLogCurrentValue	Determines whether the current value of logged tags are written to the Historical Log file at an interval set by the ForceLogging parameter.
ForceLogging	Sets the length of the interval when tag values are periodically written to the Historical Log file regardless of their current values.
LoopTimeOut	Sets the time out period of FOR-NEXT loop processing in an InTouch script.
MarkAppReadOnlyNonRDS	On a non-RDS node, if this parameter is set to 1, it will consider this a read-only node and consume a read-only license for an InTouchView application.
NoKeyboardResize	Determines whether the numeric keyboard is resized to the resolution of the WindowViewer screen.
OldRightMouseBehavior	Determines whether the right mouse button is enabled in WindowMaker.
PrintScreenWait	Sets the wait period before printing a screen from WindowViewer.
PrintWindowWait	Sets the wait period before printing an InTouch window from WindowViewer.
RemoteTagsLogEvents	Determines whether an InTouch application logs remote referenced tag alarms and events.
RemoteTagsNoIOEvents	Determines whether an InTouch application logs remote referenced tag alarms.
ScaleForResolution	Determines whether InTouch application windows are automatically resized when changing nodes that have different screen resolutions.
ViewLicenseRetryCount	Determines the times WindowViewer will attempt to acquire the license in the background, during Startup and when no license is available.
WindowNameWithSpecialCharacters	If the parameter is set to 1, then new windows can be created with special characters in the window name.

Setting Custom Logging Properties

You can add a set of parameters to the INTOUCH.ini file that specify how tag values are saved to the InTouch historical log file. The values assigned to these parameters determine logging frequency and if the values of remote referenced tags are logged.

Setting Logging Frequency

The InTouch HMI writes entries to the historical log file based upon two conditions:

- The InTouch HMI writes an immediate log entry whenever a tag value changes by an engineering unit value greater than its log deadband value.
- The InTouch HMI writes the current values of all logged tags at a fixed interval. The default fixed interval is 60 minutes.

You add two parameters to the INTOUCH.ini file to change the interval.

- ForceLogging

  ForceLogging specifies the length of the fixed logging interval in minutes. ForceLogging can be set to a value from 5 to 120. The default is ForceLogging=60.

- ForceLogCurrentValue

  ForceLogCurrentValue forces the InTouch HMI to write log entries for all logged tags even if the current values are less than or equal to their log deadband ranges. The default is ForceLogCurrentValue=0.

In the following example, current tag values are written to the Historical Log file at 15 minute intervals or when the value of the tag changes:

```
ForceLogging=15
ForceLogCurrentValue=1
```

Logging Remote Referenced Tags

By default, remote referenced tags are not logged to the events log file. To log remote referenced tags, you must enable event logging and then add the RemoteTagsLogEvents parameter to the INTOUCH.ini file.

```
RemoteTagsLogEvents=1
```

To exclude I/O tags from being logged, add the RemoteTagsNoIOEvents parameter to the INTOUCH.ini file. The RemoteTagsNoIOEvents parameter applies only if the RemoteTagsLogEvents parameter is set to 1.

```
RemoteTagsNoIOEvents=1
```

Disabling WindowMaker Shortcut Menus

By default, WindowMaker shows an shortcut menu when you right-click with your mouse over the selected object. If you prefer to develop your application using the same mouse behavior as earlier versions of the InTouch HMI, you can turn off WindowMaker's right-click behavior by setting the oldrightmousebehavior parameter to 1 in the INTOUCH.ini file.

```
oldrightmousebehavior=1
```

Setting Custom WindowViewer Properties

You can add a set of INTOUCH.ini file parameters that set the behavior of WindowViewer to:

- Handle script looping.
- Scale InTouch windows for different screen resolutions.
- Set a waiting period to print windows or screens.
- Log run time changes to an alarm comment.
- Set the drawing mode of a 16-Pen Trend.
- Resize the numeric keypad.
- Resizing input fields of analog and string user input links.

Adding a Script Loop Timer

By default, a FOR-NEXT loop within an InTouch script must complete within five seconds. WindowViewer stops the script automatically if the FOR-NEXT loop processing has not finished by the time-out limit. This time-out limit prevents an infinite loop caused by a scripting error.

Occasionally, you may need to write a script in which the FOR-NEXT loop code processing exceeds the five second time-out limit. You can change the length of the time-out limit by adding the LoopTimeout parameter to your INTOUCH.ini file.

In this example, loop processing continues for a maximum of 20 seconds:

```
LoopTimeout=20
```

Scaling InTouch Windows to Different Screen Resolutions

You can add a parameter to the INTOUCH.ini file to maintain the current resolution of InTouch windows when you migrate the application to other nodes running different screen resolutions.

The ScaleForResolution parameter value determines if application windows are automatically scaled by WindowMaker after the display resolution changes on the computer running WindowViewer. The ScaleForResolution parameter does not affect the resolution of WindowViewer dialog boxes. Resolution conversion is enabled when the ScaleForResolution parameter is set to 1.

```
ScaleForResolution=1
```

Setting the Length of the Print Waiting Period

When you select a window or screen to print, WindowViewer loads the selected window or screen into memory. WindowViewer then waits 10 seconds to allow all DDE variables shown in the window or screen to be updated. After the waiting period ends, WindowViewer sends the window or screen to the printer.

The WindowViewer print waiting period can be changed by adding the PrintWindowWait or PrintScreenWait parameters to the INTOUCH.ini file. The wait period for either parameter is expressed in milliseconds.

```
PrintWindowWait=15000
PrintScreenWait=20000
```

Logging Alarm Comments

Operators can add a comment when acknowledging an alarm. To write run time changes to the Alarm Comment field in the tag database, add the following line to the INTOUCH.ini file for the current application.

```
CommentRetentive=1
```

Setting the Drawing Mode of a 16-Pen Trend

You can select the line drawing mode of a 16-Pen Trend based on the value of the 16PenTrendDrawMode parameter.

- Averaging mode: `16PenTrendDrawMode=0`

  Because of the time range and the buffer size of the 16-Pen Trend, each pixel on the trend can represent several seconds' worth of data. Each interval can contain several samples with different values. As a result, the trend's data point can appear as a vertical line between the maximum and the minimum values observed within the interval.

  After the minimum to maximum vertical line is drawn, the trend pen moves to the calculated average value for the interval. The next interval begins by drawing the line from the average value to the next interval on the trend. The vertical minimum to maximum line is drawn and the pen rests at the average value calculated for the interval. This process repeats for each sampling interval.

  Averaging is the default drawing mode of a 16-Pen Trend if the 16PenTrendDrawMode is not specified in the INTOUCH.ini file.

- Min-Max mode: `16PenTrendDrawMode=1`

  In the Min-Max drawing mode the trend line is drawn by directly connecting the endpoints of each data collection interval.

Resizing a Numeric Keypad

You can add a parameter to the INTOUCH.ini file that determines whether an InTouch application's numeric keypad can be resized or not. Increasing the size of the keypad at higher screen resolutions (1280 x 1024) keeps the text appearing on the keypad legible. But, you may have applications with limited screen space that set practical limits on the size of the keypad.

You can add the NoKeyboardResize parameter to the INTOUCH.ini file. By default the parameter is not included. Its default value is:

```
NoKeyboardResize=0
```

The default value permits the numeric keypad to be resized according to the screen resolution.

The alternative value you can assign to the parameter is:

```
NoKeyboardResize=1
```

In this case, the keypad does not resize based on screen resolution and the numeric keypad size remains fixed.

Resizing the Input Fields of Analog and String User Input Links

You can add the Resizable InputLink parameter to the INTOUCH.ini file to resize the input box of the Analog or String user input links with your mouse. The Resizable InputLink parameter must be set to a non-zero value.

After the Input field is resized the first time, WindowViewer adds the Resizable InputLink Width and Resizable InputLink Height parameters to the INTOUCH.ini file. These parameters specify the width and height of Input boxes in pixels.

Example:

```
Resizable InputLink = 1
Resizable InputLink Width=300
```

```
Resizable InputLink Height=50
```

Also, you can edit the INTOUCH.ini file to manually modify the values assigned to these parameters.

Resolving Stuck Application Button or Displayed Value Problems

A parameter can be added to the InTouch.ini file to resolve problems in which an InTouch application button is stuck in the down position or a displayed value does not change. The button and the value do not respond to repeated mouse clicks.

Possible causes of this problem can be an OnKeyUp script that does not run because a graphic element with an OnKeyDown script hides the window. Also, the stuck button problem can be caused when there are two scripts, OnKeyDown to set a bit and OnKeyUp to clear the bit. The operator clicks the button, but the window containing the button closes before the mouse is released.

To solve these problems, do the following:

- Insert the `UseLegacyOnKeyUp=1` parameter in the Intouch.ini file.
- Select the **Use In-Memory Window Cache** check box in the WindowViewer **Properties** dialog box.

AVEVA

Appendix B

Managing Security for InTouch HMI

General Considerations for Security

Before you review the information in this section, it is recommended that you go through the following checklist to ensure you plan to cover the security areas that apply to your ICS and organization.

Security Area	**Reference Section**
Physical and virtual access to the host	*General Guidelines for Securing the Host* on page 275
Latest Windows patches applied	*Windows Updates* on page 275
Protecting the host from viruses and malware	*Scanning the Host* on page 276
Access to content on the host	*Protecting the Applications and Content on the Host* on page 277
Securing your network	*Securing the Network* on page 278
Configuring services and ports	*Managing Network Services and Ports* on page 280
Securing client/server communication	*Securing Communication between the Client and Server* on page 280
User and group management	*Securing Systems through Authentication and Authorization* on page 282
Planning for emergencies	*Contingency Planning* on page 284

For a list of security feature help topics refer to the table at the end of this section.

Introduction

This appendix provides a general overview on how to securely deploy your AVEVA software product as an Industrial Control Systems (ICS) application.

This appendix is not meant to be comprehensive, and it does not provide any detailed instructions. It is only a collection of basic concepts and recommendations that you can use as a checklist to secure your own systems. If you need help with a specific item in this guide, see the official documentation for that item -- for example, if you need help with your anti-virus software, see the documentation for that software.

AVEVA's approach to securing site networks and ICS software is driven by the following principles:

- View security from both Management and Technical perspectives
- Ensure that security is addressed from both IT and ICS perspectives.
- Design and develop multiple network, system and software security layers.
- Ensure industry, regulatory and international standards are taken into account.
- Aim to prevent security breaches, supported by detection and mitigation.

These principles are realized by implementing the following security recommendations:

- Prevent security breaches using the following components:
  - Firewalls
  - Network-based intrusion prevention/detection
  - Host-based intrusion prevention/detection
- Segregate IT and Plant networks
- Include a clearly defined and clearly communicated change management policy. For example, firewall configuration changes.

Note: AVEVA strongly recommends following the guidelines prescribed by the U.S. Department of Commerce for securing ICS software. The document "Guide to Industrial Control Systems (ICS) Security" [NIST Special Publication 800-82 Revision 2] (*https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf*) provides detailed information about ICS, typical system topologies, security threats and vulnerabilities, and recommendations for implementing security measures.

Securing the Host

Given the sensitive nature of industrial control, it is important to secure not only the ICS software, but also:

- the host on which it runs
- the network to which it is connected
- the hardware used for the ICS software.

Note: The "host" is the Windows computer or Windows Embedded device on which your ICS software is installed and running.

There are several factors to consider for securing the host including:

- Access to the host
- Keeping track of and applying the latest Windows updates
- Keeping the host computer free of viruses and malware
- Protecting the applications and content on the host

Each of these factors is covered in the sections below.

General Guidelines for Securing the Host

Here are a few guidelines to secure the host:

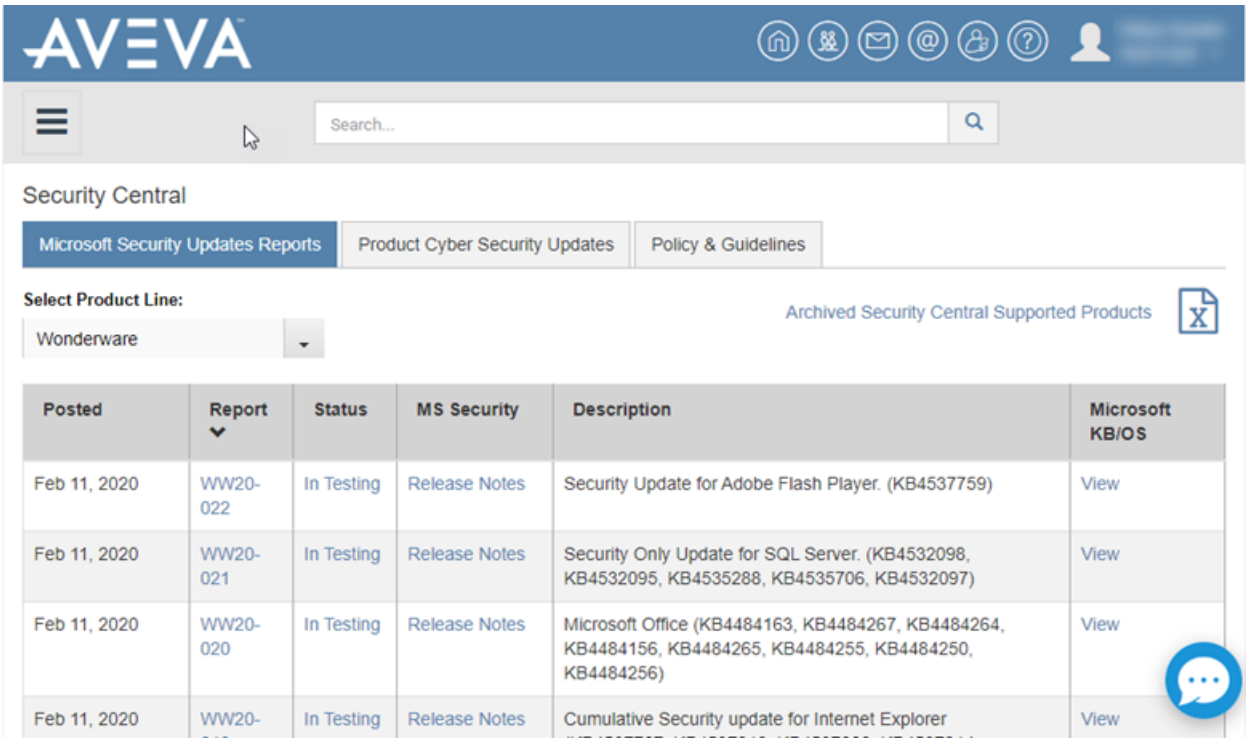
- Use an account with administrative privileges to install the ICS software, and one without administrative privileges to run the ICS software.
- Restrict configuration of ICS to a limited set of users.
- Consider running the ICS software as a Windows service, if that option is available. If the ICS software is run as a service, run it as a low privileged virtual service account.
- Once the host is fully configured and placed in its permanent location, restrict physical access and remote access to it so that only authorized personnel (for example, system administrators, application engineers, run-time operators) can use it.
- Consider disabling or removing physical ports (for example, USB, memory card) that might be used to connect external storage devices and then transfer data.

Windows Updates

Check that the Windows operating system on the host is a version that is under what Microsoft calls "mainstream support", which means Microsoft actively maintains and releases updates for it. Older versions of Windows are under Microsoft "extended support", which means they are not actively maintained and therefore might become vulnerable without notice. For more information about the different versions of Windows and the different levels of support, see [Windows lifecycle fact sheet](*https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet*).

Automate Microsoft product updates using Microsoft Windows Server Update Services (WSUS), which enables you to manage and distribute updates to computers on your network. For more information about WSUS, see [Windows Server Update Services](*https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus*). If the host does not or will not have a reliable connection to the WSUS server, perhaps because it is located on a private network, you can either develop a procedure to manually apply updates or consider changing the operating system to a Long-Term Servicing Channel (LTSC) version of Windows, which is updated less frequently.

In addition, AVEVA ICS software is tested for compatibility with Microsoft updates the results of which are published on the *Security Central site https://softwaresupportsp.aveva.com/#/securitycentral*. Security advisories and bulletins are also published on this site.



ICS Software Updates

Check that the ICS software on the host has all the recommended patches and hot fixes installed.

Some AVEVA applications release regular updates, which should be applied as soon as possible as they may contain security-related fixes.

Note: AVEVA's Global Customer Support (GCS) group publishes a *Technology Matrix https://gcsresource.aveva.com/TechnologyMatrix/Home/Index* for AVEVA software products. This matrix lists the Windows operating system versions against which a software product has been tested for compatibility. In addition, it lists compatible runtime, browser and virtualization environments for the software. It also includes a list of other products that can be installed on the same computer and lists other products with which this software can communicate.

Scanning the Host

Use both anti-virus and anti-malware software and file integrity checking software to regularly scan the host.

Windows includes Windows Defender by default, but you may choose to install and use additional software that scans for more types of malware or performs other functions. If you do that, make sure the software is provided by a reputable company. And, as with the operating system, if the host does not or will not have reliable access to the software's update service, develop a procedure to manually apply updates. If you develop a manual update procedure, it should account for all devices on a network or at a site, because a single outdated device can leave the entire network or site vulnerable.

Protecting the Applications and Content on the Host

To protect the applications and content on the host:

- Enable Windows Firewall, and configure it to close all ports that are not used by the ICS software. For more information about port usage, see Managing Network Services and Ports.
- Disable Windows features like remote desktop and file sharing, and remove unnecessary programs like games and social media.
- Restrict access to the files, databases, registry and other resources on the host.
- Use Windows BitLocker to encrypt the hard drive of computers that are either mobile or not located in a secure facility. However, BitLocker may impact the performance of computers.
- Consider using server-class storage (SANs) infrastructure to avoid storing sensitive data on mobile devices.

AVEVA leverages the security built into the Windows operating system to store and manage encryption keys. The encryption keys are stored in a local storage location called the encryption store. For more information about the Windows encryption store, refer to the Microsoft documentation, located at:

https://docs.microsoft.com/en-us/windows-hardware/drivers/install/certificate-stores

Phases of Data Protection

Data exists in three different phases, and protection must be provided for each phase:

- At rest
- In transit
- In use

Data at rest

Data at rest is data that is not currently being used or accessed, such as data stored on a hard drive, laptop, flash drive, storage area network, RAID array, network attached storage (NAS), storage area network (SAN), or archived/stored in some other way. Data protection at rest aims to secure inactive data stored on any device or network. For protecting data at rest, you can simply encrypt sensitive files prior to storing them and/or choose to encrypt the storage drive itself. BitLocker Drive Encryption, which you can invoke via the Windows Control Panel, can be used to invoke whole-drive encryption.

In the context of SCADA and ICS systems, data at rest includes stored configuration data, historical data, backups, and other static data. The duration of storage, that is, long term or short term, does not impact this classification of data at rest. Protection for data at rest is applicable for as long as the data exists in this condition; it is not a fixed condition.

Proper authorization rights need to be set in place to ensure the data is not being viewed by unauthorised users. Other steps can also help, such as storing individual data elements in separate locations, such as a corporate-approved offline backup to decrease the likelihood of attackers gaining enough information to commit fraud or other crimes. Offline backups are the best mitigation against the threat of ransomware.

Data in transit

Data in transit, or data in motion, is data that is actively moving from one location to another.

In the context of SCADA and ICS systems, this encompasses deploying a project to a run-time node, transmitting process variables, VTQ data, and other data that is sent between nodes in a running, production system. This includes alerts and alarms.

Data protection in transit is the protection of this data while the data traveling, including the following examples:

- From node to node within a network
- From network to network
- Accessed via internet
- Transferred from a local storage device to a cloud storage device

Wherever data is moving, effective data protection measures for in-transit data are critical as data is often considered less secure while in motion. Best security practice is to ensure TLS 1.2 encryption is used for all communications using the HTTPS protocol.

Data in use

Data in use refers to data that is being processed or accessed either locally or remotely. This generally involves placing data into memory (RAM) for access and processing by applications and users, potentially multiple users across different computers, mobile devices, remote terminals or other device. Data in use is particularly vulnerable to attack. To protect data in use, encryption, user authentication, and identity management is highly recommended.

In the context of SCADA and ICS systems, data in use can apply to databases, such as those used actively by a historian or deployed to a run-time node. This needs to be safeguarded by a secure transfer channel.

Securing the Network

Usually the host computer will have some sort of network access; it is increasingly rare for an ICS device to run as an entirely standalone device. The host may use the network to communicate with other ICS components such as controllers, sensors, databases, remote clients, and even other hosts in peer-to-peer relationships. You may also use the network to manage several ICS devices from a development or supervisory workstation.

Once you determine that the host will have network access, decide how it will connect to the network. In recent years there has been a shift from wired networks (that is, "Ethernet") to wireless networks ("Wi-Fi"), even for business and industrial uses. We recommend against using Wi-Fi for your ICS network because you do not have physical control over who or what might access the network. Any computer or device within range of the Wireless Access Point (WAP) can try to access the network, and even if the network is ostensibly secure, an intruder can intercept and analyze network traffic and potentially discover a vulnerability.

Nevertheless, if you decide to use Wi-Fi for your ICS network, enable all access control features on the WAP including encryption (for example, WPA/WPA2), a strong password and a list of authorized MAC addresses. Do not try to "hide" the Wi-Fi network by disabling broadcast of the Service Set Identifier (SSID), because doing so actually generates more network traffic that can be intercepted and analyzed.

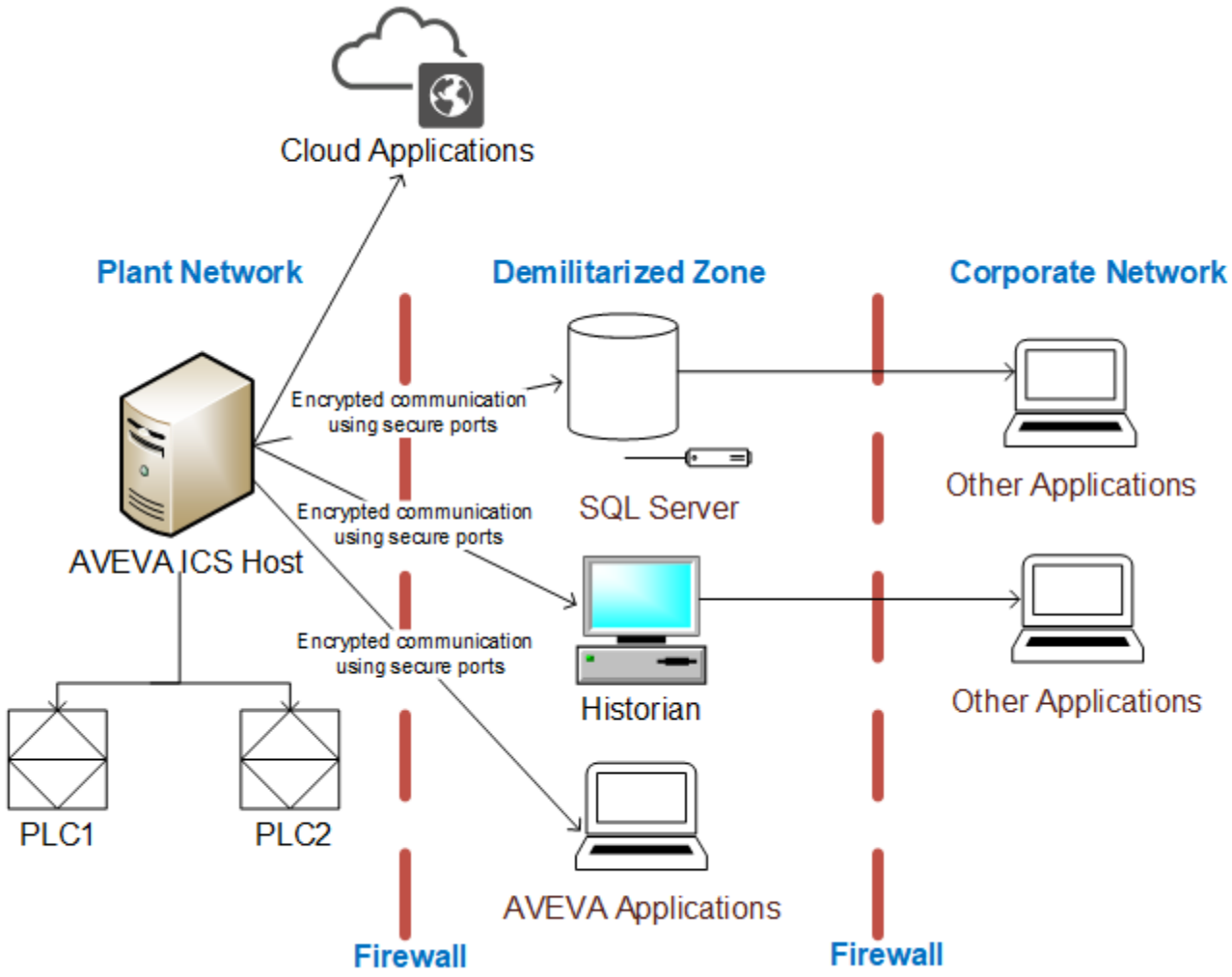
Segmenting the ICS Network

The ICS network itself can be either physically or logically segmented from your other corporate networks. A physically segmented network is by definition the most secure. The network hardware and all computers and devices connected to it form a single closed network with no physical connection to any other network, so an intruder cannot access the network unless they also have access to the physical location.

In contrast, a logically segmented network is physically connected to your other corporate networks and/or the public internet, but it uses various methods to segregate ICS network traffic from other network traffic. This may include:

- Using a unidirectional gateway
- Implementing a Demilitarized Zone (DMZ) network architecture with firewalls to prevent network traffic from passing directly between the corporate and ICS networks
- Having different authentication mechanisms and credentials for users of the corporate and ICS networks.
- The ICS should also use a network topology that has multiple layers, with the most critical communications occurring in the most secure and reliable layer.

Given below is a sample deployment topology.



In no case should your ICS network and devices be directly accessible from the public internet. If there is some part of your ICS that you want to be accessible, (for example, if you want be able to view web-enabled HMI screens from a browser or smart phone), your ICS software should include features that securely pass the necessary traffic between your ICS network and a public-facing server.

Managing Network Services and Ports

A network port is an endpoint of communication in an operating system. While the term is also used for hardware devices, in software it is a logical construct that identifies a specific process or a type of service. In other words, a network port is conceptually different from hardware ports like USB, memory card, and even the wired network connection.

Computers and devices can access many different network services at the same time by communicating on different network ports. Each network service or communication protocol has an associated port number. Some port numbers are specified by international standards, and therefore they are universally recognized. Other port numbers are claimed by proprietary software, and in most cases they can be changed in the software settings if there is a conflict with other software or services.

Firewalls control network traffic by either accepting or refusing communication on these network ports. If a port is open, it accepts communication, and if a port is closed, it refuses communication. Almost every layer of a network -- from the operating system on an individual computer or device, to the router that manages traffic within a network, to the gateway that manages traffic between networks -- has its own firewall.

The documentation for your ICS software should include a list of network ports that are commonly used by the software. Given the nature of ICS, the list typically includes services like web, email, file transfer, external databases, device drivers, and the ICS software itself for server-client communications. Configure the firewalls to open only those network ports that are actually used by your ICS. Disable all unused services and close all unused ports.

Securing Communication between the Client and Server

Like most server-client applications, your ICS software should support secure communication between the server and client in order to prevent the messages sent between those two stations from being read by any other stations on the same network. Note that this is different from securing the network itself in order to prevent unauthorized access to the network.

This sort of communication is also sometimes known as "Encrypted Channel" because it uses the Transport Layer Security (TLS) standard to encrypt the server-client messages. The latest version of the standard is TLS 1.3 (released August 2018), but it is not yet in common use. The latest version of the standard in common use is TLS 1.2 (released August 2008). TLS supersedes the earlier Secure Sockets Layer (SSL) standard, although SSL is still used in older applications.

Certificates

TLS and SSL use a system of certificates and keys to digitally "sign" the messages sent between the server and client. When the server establishes communication with the client (and vice versa), it presents its certificate which identifies its name, network address, organization, physical location, and so on. The client can then choose to either accept or refuse the certificate as presented. If it accepts the certificate, it agrees to accept messages encrypted with the same certificate, and it uses the associated key to decrypt those messages.

When you configure this sort of communication, you need to choose one of the following:

- Using self-signed certificate
- Using certificates signed by a Public Certificate Authority (CA)
- Using Domain-issued certificates or certificates signed by a Private Certificate Authority using systems like Microsoft Active Directory Certificate Service (AD CS)

A self-signed certificate is issued and signed by the same application that presents it. Self-signed certificates are easy to create and manage, but they are secure only if you control both the server and the client and therefore control which certificates are installed on each.

In contrast, CA-signed certificates are slightly difficult and expensive to acquire, but they are more flexible than self-signed certificates because you do not need to control both the server and the client. If you configure the server to present a CA-signed certificate, the client will accept the certificate because it recognizes the Certificate Authority.

Domain-issued certificates are internal certificates typically managed by your IT department. They are issued and validated by an Active Directory Certificate Authority. Domain-issued certificates are free and can be issued instantly.

You need to renew CA-signed and Domain-issued certificates at regular intervals.

For more information about how to enable Encrypted Channel features and manage self-signed certificates in your ICS software, see the documentation for that software. However, acquiring a CA-signed certificate and then using it to sign other certificates is typically beyond the scope of ICS software documentation.

Note: Encrypted and unencrypted communications typically use different network ports.

Cloud-based Systems

It is possible that your ICS software might access cloud-based solutions, or might itself be hosted on the Cloud.It is important to mitigate the risks associated with cloud-based access and hosting.

Accessing Cloud-Based Solutions

Several AVEVA applications are now being made available through the Cloud, and ICS software may need to connect to these applications. One of the main risks associated with accessing cloud-based applications is unauthorized access. Connecting ICS software to Cloud solutions must be done in a secure manner, and needs to use secure protocols such as Transport Layer Security (TLS).

It is important that data integrity is maintained at all times. Use data classification to identify data that is sensitive and data that can be made public. Secure machines,storage and networking in order to secure the data that is stored and transmitted. Work with your Cloud Service Provider (CSP) to configure users, assign access levels and monitor and control access. Ensure that the CSP's buildings are physically secure and protected from unauthorized access.

Cloud-based ICS Software

While hosting ICS software on the Cloud provides several benefits such as flexibility, scalability and availability, it is also fraught with security risks such as susceptibility to hacking resulting in damage to the organization's reputation . Therefore, it is important to implement a security strategy before you make your ICS software accessible on the Cloud. For securing ICS software on the Cloud, you need to consider the following:

- Securing access points by putting in place authentication, monitoring and support mechanisms.

- Implementing cloud-based, centralized security measures including encrypting communications using TLS.

Note: It is recommended that you review the *NIST Cybersecurity Framework https://www.nist.gov/cyberframework* for additional information.

Securing Systems through Authentication and Authorization

Typically, ICS software is comprised of a large number of systems, each accessed by a variety of users including engineers, operators and managers. The level of access that each type of user requires is different. So, it is necessary to manage user authentication and authorization to secure the system.

Authentication

Authentication is the process of verifying a user's/system's identity. Authentication can be managed in the following ways:

- Within the ICS software through application accounts
- Through Windows accounts, which can be local to a single computer
- Through Authentication systems (see the next section for details)

While ICS software allows for user and role management, it can become cumbersome and complicated to manage a large number of user accounts as employees and roles change. Because of this, use of Windows accounts is generally preferred.

Authentication Systems

Authentication systems such as Active Directory and Lightweight Directory Access Protocol (LDAP), referred to as authentication servers, are a repository of and provide centralized management for all system accounts and individual user accounts. An authentication protocol is used for all communication between authentication servers and the user or server requesting authentication.

Even though use of authentication systems provides improved scalability, the following factors must be considered depending upon the size and complexity of your operations:

- It is important that the authentication servers are highly secured.
- The authentication server system creates a single system for managing all system accounts. Therefore, it requires to be available at all times. To ensure minimal disruption during an emergency, redundancy must be considered.
- Permit caching of user credentials only for users who have authenticated their identity recently.
- Networks that support the authentication protocol must be reliable and secure to assist in trouble-free authentication.

It may also be worthwhile implementing two-factor authentication using additional applications such as PingID.

Authorization

Authorization is the process of providing the correct level of privileges to users by applying access rules to authenticated users, systems (HMIs, field devices and SCADA servers) and networks (remote sites' LANs).

Managing Users and Groups through Windows

When you configure security, you may choose to do one of the following:

- Keep the configuration local to a single application.
- Share the configuration between multiple applications.
- Manage the configuration as part of the network domain (for example, using Active Directory). This option typically allows users to have the same user account for the network, the host, and the ICS software. Using Active Directory gives you the following advantages:
    - A centralized repository for user and group data, enabling effective implementation of security policies and procedures.
    - Provides a single point of access to all network resources after the user is identified and authenticated.

To manage users and groups:

- First define a specific role for each group, and then configure the group privileges to fit that role.
- Groups may overlap, but it is often better to have clearly separate groups and then assign individual users to multiple groups, if necessary.
- Set or change the password for the ICS software's default user (e.g., "guest").
- Define stringent password policies to force users to create strong passwords. Enforce mandatory password updates on a regular basis.

Managing Users and Groups through ICS Software

Your ICS software should have a built-in security system that controls who may use the software and what privileges they have.

Users should be assigned permissions that determine what each user is authorized to do within the ICS system. Permissions can be managed either on a per-account basis or on a group basis by making use of roles. Group or role-based access control is preferred as it greatly simplifies management. Users can be moved from one role to another as the organization's needs change, and can also be members of multiple roles if required.

Each user should have their own user account with a unique user name and a strong password. The user account can then be assigned to one or more groups.

Accounts should always be assigned the least privileges necessary to perform their functions. Accounts with Windows Administrator permissions should be reduced to the minimum, and typically only used to install and configure the software. Likewise, accounts with SQL Server SysAdmin privileges should be reduced to the minimum, and typically only used to install and configure the software.

In most cases, the ICS software will allow associating Windows Groups with roles within the product. While defining and assigning roles, consider the following:

- Roles should be defined to have the least privileges necessary for their functionality.
- Roles should be limited to a single purpose in order to simplify the permissions assigned to them.
- Users can be members of multiple roles if necessary.

Contingency Planning

Incidents are inevitable. It is, therefore, important to develop a strategy to detect an incident quickly and respond to it in a timely manner in order to minimize loss and protect your system. An organization must consider contingencies arising from incidents such as fire, flood and so on, and those arising from failure of hardware or software components. Cyber attacks such as ransomware are becoming more common and must also be considered.

An organization should have contingency plans in place to cover the entire range of failures and eventualities. Employees should be trained and be familiar with the contents of the contingency plans.

As part of planning for contingencies, it is important to establish a site, physically separated from the central one, that has replication capability. Doing so will ensure the integrity of an operational system where the central site is at risk from fire, floods or other disasters. The replication capability includes having duplicated hardware, and requires software configuration and key state information to be periodically propagated from the central site to the recovery site. Each recovery scenario is unique, so it is important to consult with system integration experts regarding the design of communications equipment, hardware and the configuration of the software.

Protecting the data stored in your system is also of paramount importance. Full and incremental backups must be scheduled on a regular basis. Backups should be verified by running tests to restore from backed up data. Backups should be stored offline so that they are safe from cyber attacks such as ransomware.

Organizations should also have business continuity and disaster recovery plans that are similar to contingency plans. These plans are covered briefly in the sections to follow.

Auditing and Logging

As part of implementing security for ICS software, it is important to incorporate auditing and logging activities on various systems and networks.

Auditing and logging provide information on the current state of your ICS, and help to ensure that the system is functioning as expected. If an incident occurs, you can use the activity logs to trace the origin of the incident to a computer, user or network. Auditing and logging can also help with troubleshooting issues.

If you are connecting to cloud-based solutions, audit all virtual machines to ensure data integrity.

Business Continuity Planning

Business continuity planning addresses strategies to maintain or re-establish production in the event of any disruption. These disruptions may be caused by a natural disaster (flood, earthquake, etc), by an intentional or unintentional man-made event (arson, operator error, power outage, etc), or by system failure.

Depending upon the duration of the potential ICS application outage caused by a disruption, operational recovery plans for short-term outages and disaster recovery plans for long-term outages must be formulated. It is also important to employ physical security for areas of a production site that house data acquisition and control systems that might have higher-level risks. Your business continuity plan should specify system and data recovery procedures for your systems. Once the recovery procedures are documented, a schedule should be developed to test the recovery procedures. Particular attention must be paid to the verification of backups of system configuration data and product or production data. The procedures should be reviewed periodically.

If you are accessing cloud-based solutions, ensure that systems are available at all times. In case of a disaster, services should switch to a new physical location to provide continued service.

Disaster Recovery Planning

A disaster recovery plan (DRP) is a set of procedures to protect and recover an IT infrastructure in case of a disaster. It contains the procedures to follow before, during and after a disaster. Disasters can be natural, environmental or man-made (intentional or unintentional).

A DRP is essential for continued availability of the ICS, and should cover the following:

- When the DRP should be activated depending upon an event, its duration and its severity.
- Detailed course of action for operating the ICS manually until external connections are secured.
- Personnel responsible for each procedure.
- Processes for securely backing up data and restoring it. This should cover:
  - Requirements for building redundancy.
  - File backup procedures.
  - Frequency of backups.
  - Storage mechanism for full and incremental backups.
  - Safe storage of installation media, license keys and configuration information.
  - List of individuals responsible for performing, testing, maintaining and restoring backups.
- List of personnel with physical and virtual access to the ICS.
- Detailed configuration information about the components of the ICS.
- Schedule for testing the DRP.

Conclusion

Security lapses present a serious threat to ICS software and infrastructure. Therefore, it is important for every organization to:

- Be proactive about preventing security lapses
- Identify potential lapses
- Detect them in a timely manner when they occur
- Address lapses to ensure minimum disruption and maximum availability

To this end:

- Computers and networks must be secured
- Users and groups must be authenticated and authorized
- Contingency plans must be in place to recover from untoward or intentional events

Refer to the document "Guide to Industrial Control Systems (ICS) Security" [NIST Special Publication 800-82 Revision 2](*https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf*) for additional details and recommendations.

Security Configuration for InTouch HMI

The table below lists the security areas that you need to configure for InTouch HMI and the details of the section(s) in this guide that provide the corresponding instructions.

Security Area(s)	**Topic(s) in this guide**	**Summary**
Protecting the Applications and Content on the Host	*Configuring an Inactivity Time-Out* on page 175	Configure WindowViewer to automatically log off an inactive operator from an InTouch application.
Securing Systems through Authentication and Authorization	*Authentication and Authorization Based Security*	Users needs to authenticate themselves before using an InTouch application. InTouch verifies whether the authenticated user is authorized to use specific functionality.
	Using Virtual Accounts	Using virtual accounts provides an additional layer of security when accessing alarm functions. See Using Virtual Accounts in the InTouch HMI Alarms and Events Guide.
Protecting the Applications and Content on the Host	*Locking System Keys*	Restrict operator access to standard Windows functions by disabling system keys on the computer running an InTouch application.
Securing Systems through Authentication and Authorization		
Managing Users and Groups through ICS Software	*Using InTouch-Based Security*	Restrict which functions an operator is allowed to perform by linking those functions to internal tags.
Managing Users and Groups through Windows	*Using Operating System-Based Security*	Inherit some user/group account policies from the Windows operating system.

Index

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

W