

Wonderware® FactorySuite™

InTouch™ New Features User's Guide

Revision A

Last Revision: 4/21/04

Invensys Systems, Inc.

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

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

© 2004 Invensys Systems, Inc. All Rights Reserved.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200
<http://www.wonderware.com>

Trademarks

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

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

Contents

New Features in InTouch5

SmartSymbol Manager.....	5
New Script Function to Redirect Remote References.....	6
Tablet PC Support	6
Improved Support for FDA 21 CFR Part 11 Requirements.....	12
InTouch Support of Slow and Intermittent Networks	12
Time and Date Functions	13
New Script Function to Display InTouch Version Information	14
Cross-Reference Query Display.....	14
Alarm Viewer ActiveX Control Display -Flash Unack Alarms.....	14
Additional Colors for Animation Links	16
Changing Window Colors in WindowViewer at Runtime	16
String Data for Quick Functions	17
Support for Tagnames with Numeric First Characters.....	18
Increased Number of Characters Allowed for Conditional Expressions.....	18
New Script Function to Notify Users of Password Expiration	18
Vertical Mouse Release Slider.....	18
Bypassing the Creation of Placeholders.....	19

InTouch™ SmartSymbols21

Overview	21
The SmartSymbol Manager	23
Generating SmartSymbols.....	26
Instantiating SmartSymbols	31
Editing Existing SmartSymbols	36
The SmartSymbol Library.....	38
The SmartSymbol Properties Dialog Box	43
Error Handling for the SmartSymbol Manager.....	44
SmartSymbols at Runtime.....	45
Redirect Remote References Using a Script	45

CHAPTER 1

New Features in InTouch

This chapter contains important information about the new features provided in this release of InTouch. Please review this chapter carefully for information on the new functionality provided in this release.

Contents

- SmartSymbol Manager
- New Script Function to Redirect Remote References
- Tablet PC Support
- Improved Support for FDA 21 CFR Part 11 Requirements
- InTouch Support of Slow and Intermittent Networks
- Time and Date Functions
- New Script Function to Display InTouch Version Information
- Cross-Reference Query Display
- Alarm Viewer ActiveX Control Display -Flash Unack Alarms
- Additional Colors for Animation Links
- Changing Window Colors in WindowViewer at Runtime
- String Data for Quick Functions
- Support for Tagnames with Numeric First Characters
- Increased Number of Characters Allowed for Conditional Expressions
- New Script Function to Notify Users of Password Expiration
- Vertical Mouse Release Slider
- Bypassing the Creation of Placeholders

SmartSymbol Manager

The InTouch SmartSymbol Manager allows InTouch application developers to create symbols as templates to connect to ArchestrA objects, local InTouch tags or InTouch tags using remote references. You can create Smart Symbols that can be imported into any InTouch application without the need to import whole application windows. For more information on the SmartSymbol Manager, see Chapter 2, "InTouch™ SmartSymbols."

New Script Function to Redirect Remote References

A new script function is provided to enable InTouch application developers to redirect ArchestrA object references or references to remote InTouch tags. The new script function used to redirect remote references at runtime is `IOSetRemoteReferences`. For more information on `IOSetRemoteReferences`, see Chapter 2, *Redirect Remote References Using a Script*.

Tablet PC Support

InTouch enables users to work with InTouch on Tablet PCs. Tablet PCs enable users to use the tablet display like a writing tablet, while retaining the ability to use the keyboard and/or mouse. Users can also control the Tablet PC and Windows XP-compatible applications using the stylus. InTouch Terminal Services are not supported on Tablet PCs.

You can also use the Tablet PC digital pen to perform commands instead of the mouse. The table below shows examples of common actions performed with the mouse as well as the corresponding default settings for actions using the digital pen. For more information on Tablet PC functionality, refer to the Tablet PC documentation.

Mouse	Digital Pen
Point	Point
Click	Tap
Double-click	Double-tap
Right-click	Tap and hold
Click and drag	Drag

Pen Input Panel

The InTouch WindowViewer displays edit boxes when the user clicks touch links of Analog or Message types. In addition to these edit boxes, WindowViewer now displays the Pen Input Panel to facilitate data entry using the digital pen. Note that the Pen Input Panel will only be displayed if WindowViewer is running on a Tablet PC.

The Pen Input Panel is displayed only when the Keypad option of the touch link is set to "No."

The default input method for the Pen Input Panel is handwriting. However, users can switch between handwriting and keyboard input methods using buttons on the user interface.

AnnotateLayout Script Function

InTouch includes a new script function called AnnotateLayout. This script function allows users to 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.

When AnnotateLayout is called, the Annotate Layout dialog box appears. This dialog box has the following functionality:

- Captures the screen image of WindowViewer at the time this dialog box is displayed.
- Allows for annotation of the screen capture using the pen in conjunction with tool bar and menu item settings.
- Allows the image and the annotation to be saved as a GIF or JPEG file.
- Allows the image and the annotation to be printed (if a printer is configured).
- Allows the image and the annotation to be sent as an attachment via e-mail (if SMTP is configured).

The Annotate Layout dialog box contains a toolbar and menu items and shows the InTouch WindowViewer screen capture in its client area. The dialog box has a Close button on the caption bar, but does not have minimize or maximize buttons.

Annotate Layout Toolbar

The toolbar includes the Save, E-Mail, Print, Cut, Copy, Paste, Pen, Highlighter, Eraser and Lasso functions. The following table explains the functions of the toolbar commands:

Command	Functionality
Save	Saves the image and annotation as a GIF or JPEG file. When the user clicks Save, the Save an Image File dialog box appears. The default directory is the current user's "My Documents." By default, the image is saved as a JPEG file.
E-Mail	Invokes an E-Mail dialog box. The dialog box includes the "From," "To," "Cc", "Bcc," "Subject", and "Attachment" text boxes and the "Send" button. The text box for the E-Mail message text is not labeled. Users must set up an E-Mail server by clicking E-Mail Configuration on the Edit menu. The E-Mail functionality in the Annotate Layout dialog box uses SMTP (Simple Mail Transfer Protocol).

Command	Functionality
Print	Invokes the standard "Print" dialog box and allows the user to print the annotated image to the selected printer. If a printer has not been configured, this toolbar function will not be available.
Cut	Cuts the annotation selected using the lasso. Accelerator key is Ctrl + X.
Copy	Copies the annotation selected using the lasso. Accelerator key is Ctrl + C.
Paste	If any previously cut or copied annotations are available in the clipboard, this function allows the user to paste it in the current window. Accelerator key is Ctrl + V.
Pen	<p>Enables the user to annotate the screen capture in the Annotate Layout dialog box. Expands to the "Color", "Size", "Tip" and "Transparency" options.</p> <ul style="list-style-type: none"> • Click Color to choose the pen color. The default is Red. Click Size to choose the thickness of the pen line. The options are 1pt to 5 pt thickness and the default is 1pt. • Click Tip to expand to "Ball" and "Rectangle." The default is "Ball." A check mark on the left side of the menu indicates the current selection. • Click Transparency to select the transparency. The default is 0%.
Highlighter	<p>Enables the user to highlight annotations on the screen capture in the Annotate Layout dialog box. Expands to the Color and Transparency options.</p> <ul style="list-style-type: none"> • Click Color to select the highlighter color. The default color is yellow. • Click Transparency to select the transparency. The default setting is 50%.

Command	Functionality
Eraser	<p>Enables the user to erase annotations on the screen capture in the Annotate Layout dialog box. Expands to the Mode and Size options.</p> <ul style="list-style-type: none"> Click Mode to select the Stroke or Point options. The default is Point. A check mark on the left side of the menu indicates the current selection. Click Size to select either Small, Medium or Large. The default eraser size is Medium. A check mark on the left side of the menu indicates the current selection.
Lasso	The Lasso is a drawing tool that allows users to select annotations on the screen capture to Cut or Copy.

Annotate Layout Menus

The Annotate Layout dialog box includes the following menus with the following functions:

File Menu

Menu Item	Description	Shortcut Keys
Save	Saves the current screen capture.	Ctrl + S
E-Mail	Invokes the New Message dialog box.	Ctrl + M
Print	Invokes the Print dialog box and allows the user to select a printer.	Ctrl + P
Exit	Exits the Annotate Layout dialog box.	Alt + F4

Edit Menu

Menu Item	Description	Shortcut Keys
Undo	Negates the most recent annotation.	Ctrl + Z
Redo	Reapplies the most recently negated annotation.	Ctrl + Y

Menu Item	Description	Shortcut Keys
Clear	Expands to the following options: <ul style="list-style-type: none">• All - Clears all annotations on the screen capture in the Annotate Layout dialog box.• Selection - Clears the annotations selected using the Lasso.	N/A
Cut	Cuts the annotation selected using the lasso.	Ctrl + X
Copy	Copies the annotation selected using the lasso.	Ctrl + C
Paste	Pastes the cut or copied data into the current window.	Ctrl + V
E-Mail Configuration	Allows the user to set up the SMTP server name.	N/A

Tools Menu

Menu Item	Description
Pen	<p>Enables the user to configure the pen. Expands to the "Color", "Size", "Tip" and "Transparency" options.</p> <ul style="list-style-type: none"> • Click Color to choose the pen color. The default is red. • Click Size to choose the thickness of the pen line. The options are 1pt to 5 pt thickness and the default is 1pt. • Click Tip to expand to "Ball" and "Rectangle." The default is "Ball." A check mark on the left side of the menu indicates the current selection. • Click Transparency to select the transparency. The default is 0%.
Highlighter	Allows the user to select Color and Transparency options for the highlighter.
Eraser	Allows the user to select Mode and Size options for the eraser.
Lasso	Allows the user to select annotations on the screen capture to Cut or Copy.
Restore Defaults	Resets all options for the tools such as Pen, Eraser and Highlighter, to their defaults.

Help Menu

Menu Item	Description
About	Displays the About dialog box.

All the toolbar and Tools menu settings are persisted on a per user basis. When the setting dialog box is displayed, the user's current pen/highlighter/eraser options are the last settings used and not the default settings.

Portrait / Landscape Orientation

Some tablet PCs can be used as either a tablet PC or a laptop PC. When switching from one configuration to another, the orientation by default is switched. For example, if the tablet PC running in laptop configuration is switched to tablet configuration, the screen orientation will be switched from landscape (1024x768) to portrait (768x1024).

If the tablet PC is running in tablet configuration, and WindowViewer is configured to dynamically change the application resolution to the screen resolution, an application developed in landscape orientation will be automatically scaled to fit the portrait orientation when opened in WindowViewer. If WindowViewer is not configured to dynamically change the application resolution, however, the landscape application will not be scaled. In this case, some InTouch windows may be truncated.

Improved Support for FDA 21 CFR Part 11 Requirements

In conjunction with the new security and authentication functions, new system tags enable the visibility of full names of operators, and OS group and ArchestrA role membership. Using these features improves productivity and flexibility when designing and configuring an application, which must be validated to conform to the above standard.

InTouch Support of Slow and Intermittent Networks

The Network Application Development (NAD) feature has been enhanced to support slow and intermittent networks. Now, when NAD performs an update to an application, it copies only the changed files from the master. NAD does not copy the SmartSymbol directory along with the application to the NAD client. This minimizes the time to complete the update and reduces the load on the network.

NAD has been enhanced to support customers who want to apply NAD to different environments, including networks that involve dial-up connections, limited bandwidth, and intermittent disconnections.

NAD now reduces the number of "round trips" that are needed to identify which master files need to be copied and which client files need to be erased. When you save changes in WindowMaker, an additional file NADFileData.dat, which contains the list of names, date/time stamps, and sizes for all the files in the NAD application directory, is created. The NAD client reads this file once into a memory buffer at the start of the download, and has all the information it needs for both loops.

All downloaded files are copied to a temporary sub-directory called NAD_Temp. Files will only be 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 will not be corrupted by the partial introduction of new or updated files.

NAD can detect whether additional changes have been made to the master application during application download. If such a situation arises, NAD will abandon the download of the application. If the application engineer has issued another "Notify Clients" after the latest update, NAD will automatically begin downloading the latest application files at the next polling period. Otherwise, it will wait until the next "Notify Clients" is issued before an application download takes place.

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. If contact with the NAD master file 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 will not be erased simply because a lost connection makes it impossible to confirm their existence on the master application.

Time and Date Functions

InTouch now includes additional functions providing enhanced time zone support for the Alarm DBView Control, a new UTC time and date function and a script function to adjust for daylight savings time.

Alarm DBView Control - Enhanced Time Zone Support

The Alarm DBView control allows the user to view the Origin Time, GMT Time, or Local Time for alarms. The new property, which provides this functionality is called `DisplayedTimeZone`. The Origin Time and Local Time options are in addition to GMT.

New UTC Time and Date Function

A new time and date function has been added to enable users to obtain the UTC Time, the current time zone and offset from GMT, and the daylight savings time status.

At 9:24 am Pacific time, the display shows the following:

```
UTCDateTime ("UTC_SHORT") gives 17:24:05
```

```
UTCDateTime ("UTC_LONG") gives 01/06/2003 17:24:05
```

```
UTCDateTime ("UTC_LOCAL") gives Pacific Standard Time -8:0: 1
```

If any invalid string (other than `UTC_SHORT`, `UTC_LONG`, `UTC_LOCAL`) is passed as an argument, the return value will be the default (ddd mmm dd hh:mm:ss yyyy)

For the above date and time (9:24 AM on 6th Jan 2003) the return value would be - Mon Jan 06 17:24:05 2003.

Note The return value of the UTC time and date function is memory message tag

New Script Function to Adjust for Daylight Savings Time

A new script function called `WWIsDayLightSaving()` has been added to enable users to find out the Daylight Savings setup status from InTouch. This script function returns 1 if Daylight Savings is enabled and 0 if Daylight Savings is not enabled.

New Script Function to Display InTouch Version Information

A new script function called `InTouchVersion()` has been added to enable users to get InTouch version information through scripts.

The Script function `sVersion = InTouchVersion(VERSION_TYPE)` will return a string depending on the value of `VERSION_TYPE`.

User can enter the following 5 choices starting from 0 to 4(both inclusive). If the user enters a negative integer or an integer greater than 4, the default full version string will be displayed. The choices for the value of `VERSION_TYPE` are:

- 0: The full version will be displayed
- 1: Major version number ("9") will be displayed
- 2: Minor version number ("0") will be displayed
- 3: Patch level ("1"). For InTouch SP1 the Patch level is 1.
- 4: Build number will be displayed.

Cross-Reference Query Display

Cross-reference queries now enable users to view the usage in Quick Functions.

Alarm Viewer ActiveX Control Display -Flash Unack Alarms

Users can configure unacknowledged (UNACK) alarms to flash in the Alarm Viewer control. A property named "Flash Unack Alarms" is provided as a check box. UNACK alarms will flash only when this check box is selected. By default, this check box is not selected. When the user selects Flash Unack Alarms, only UNACK alarms will flash. UNACK_RTN alarms will not flash.

Five new properties have been added to the Alarm Viewer ActiveX control to enable the alarm flashing functionality. The five new properties are explained below:

FlashUnAckAlarms

Property:	FlashUnAckAlarms
Purpose:	This property enables or disables the flashing of Unack Alarms. It takes a discrete input value of 1 or 0. If this property is set to 1, Unack alarms will flash. If this property is set to 0, Unack alarms will not flash. At design time, this property corresponds to the "Flash Unack Alarms" check box on the Alarm Viewer's General properties tab.
Type:	Discrete
Default:	0 (not flashing)
Example:	If the control name is "AlarmViewerCtrl1" then the property is accessed as #AlarmViewerCtrl1.FlashUnAckAlarms.

FlashUnAckAlmColorRange1

Property:	FlashUnAckAlmColorRange1
Purpose:	This property sets the flashing color for Unack Alarms belonging to Alarm Priority Range 1.
Type:	Integer
Default:	Green
Example:	If the control name is "AlarmViewerCtrl1" then the property is accessed as #AlarmViewerCtrl1.FlashUnAckAlmColorRange1.

FlashUnAckAlmColorRange2

Property:	FlashUnAckAlmColorRange2
Purpose:	This property sets the flashing color for Unack Alarms belonging to Alarm Priority Range 2.
Type:	Integer
Default:	Green
Example:	If the control name is "AlarmViewerCtrl1" then the property is accessed as #AlarmViewerCtrl1.FlashUnAckAlmColorRange2.

FlashUnAckAlmColorRange3

Property:	FlashUnAckAlmColorRange3
Purpose:	This property sets the flashing color for Unack Alarms belonging to Alarm Priority Range 3.
Type:	Integer
Default:	Green
Example:	If the control name is "AlarmViewerCtrl1" then the property is accessed as #AlarmViewerCtrl1.FlashUnAckAlmColorRange3.

FlashUnAckAlmColorRange4

Property:	FlashUnAckAlmColorRange4
Purpose:	This property sets the flashing color for Unack Alarms belonging to Alarm Priority Range 4.
Type:	Integer
Default:	Green
Example:	If the control name is "AlarmViewerCtrl1" then the property is accessed as #AlarmViewerCtrl1.FlashUnAckAlmColorRange4.

Additional Colors for Animation Links

Users can assign 9 breakpoints and 10 colors for analog fill color, line color, and text color animation links. Previously, users were able to set 4 breakpoints and 5 colors for animation links.

Changing Window Colors in WindowViewer at Runtime

Users can dynamically change window colors in WindowViewer. This can be done by using the new script function ChangeWindowColor(). Changes in window color are not maintained between sessions.

ChangeWindowColor()

Purpose	Returns the Status - Success or Fail for the window's background color change.	
Syntax	<code>IntegerResult = ChangeWindowColor("Window Name", RValue,GValue,BValue);</code>	
	Parameter	Description
	WindowName	The Name of the Window for which the background color must be changed. Actual string or message Tagname.
	RValue	Specifies the intensity of the red color. Actual integer value or memory integer tagname. (valid range 0 - 255).
	GValue	Specifies the intensity of the green color. Actual integer value or memory integer tagname. (valid range 0 - 255).
	BValue	Specifies the intensity of the blue color. Actual integer value or memory integer tagname. (valid range 0 - 255).
Remarks	Return type is 0, 1 or 2 and is assigned to the IntegerResult Tag. 0 - Failure - Window not defined, or RGB value out of range. 1 - Success. 2 - Failure - Window exists and is currently not open.	
Example(s)	<p><code>ChangeWindowColor("Win1", 255,255,255)</code> will return 1, where "Win1" is the name of the defined window and is currently open.</p> <p><code>ChangeWindowColor("Win2", 255,255,255)</code> will return 0, where "Win2" is the name of the window and the window by this name does not exists or RGB value is out of range (Valid range is 0-255).</p> <p><code>ChangeWindowColor("Win3", 255,255,255)</code> will return 2, where "Win3" is the name of the defined window and is not currently open.</p>	

String Data for Quick Functions

Quick functions can now return message tag and string data up to 131 characters.

Support for Tagnames with Numeric First Characters

InTouch tagnames can consist of both numeric and alphabetic characters and the first character can be numeric. If the first character of an InTouch tagname is numeric, then the tagname must contain at least one alphabetic character other than "e" or "E." The creation of tags prohibits the use of any valid numeric expressions as tagnames. For example, 123E4 or strings that could be hexadecimal numbers such as 0xABCD are not permitted. For tagnames beginning with a numeric character, dashes (-) are not allowed to be contained within the tagname.

Increased Number of Characters Allowed for Conditional Expressions

The number of characters allowed in the conditional field for conditional expressions has been increased from 256 characters to 1023 characters.

New Script Function to Notify Users of Password Expiration

Users can now use the script function `GetAccountStatus()` to determine when a user's password will expire. A script can be written using the return value of this script to pop up a window displaying the number of days until the user's password expires. The positive return value of function shows the number of days before the user's password expires.

`GetAccountStatus()` also displays the following account status values:

Account password expired = -1

Account password will never expire = -2

Account locked out = -3

Account disabled = -4

Account info failed = -5

Users can also write a script to display the proper message/warning which corresponds to the return value of the function.

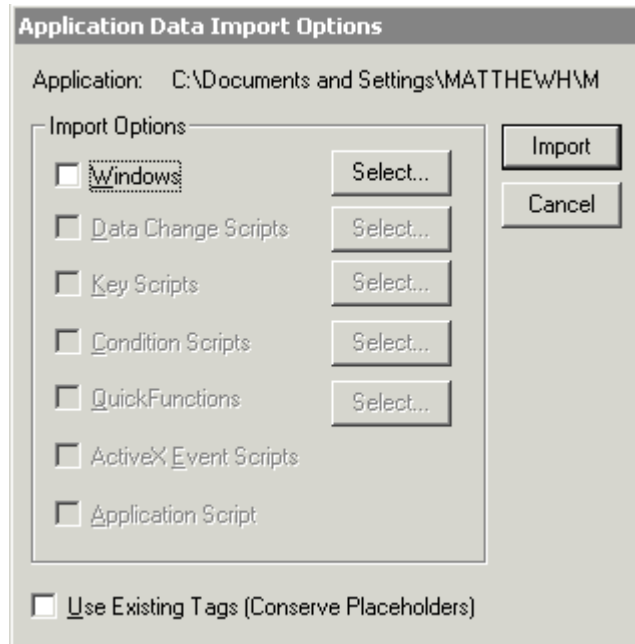
Vertical Mouse Release Slider

InTouch provides a slider control for use with ArchestrA. A value is written to the tag only when the slider is released.

Bypassing the Creation of Placeholders

When windows and scripts are imported from one application to another, placeholders are used in front of each tag. This poses a problem when many tags are used in the imported windows and scripts, because there is a limit in the number of placeholders allowed (approximately 4095). Prior to the enhancement, big applications could not be imported all at once. Instead, chunks of windows had to be imported at a time, to make sure that the number of placeholders was not exceeded.

This new enhancement introduces an additional check box "Use Existing Tags (Conserve Placeholders)" at the lower part of the Import dialog box. When the



Conserve Placeholders option is enabled, you can import scripts and windows with animation links without creating placeholders, if the imported tag names exist in the current Tagname Dictionary and the basic tag types match. Also, in the Import dialog box, some of the import-script selection dialog box controls are dimmed if those script types do not exist in the importing application's folder.

This new feature will not conserve placeholder tags or expressions contained in Wizards, Trend objects, SPC objects, Alarm objects, or ActiveX objects.

CHAPTER 2

InTouch™ SmartSymbols

The InTouch SmartSymbol Manager allows InTouch application developers to create and utilize ArchestrA integrated symbols easily. The SmartSymbol Manager can also be used to create SmartSymbols for use with InTouch Tags. The SmartSymbol Manager is a tightly integrated and user friendly interface in which application developers can create graphic symbols and integrate them with ArchestrA Objects and their data attributes.

Contents

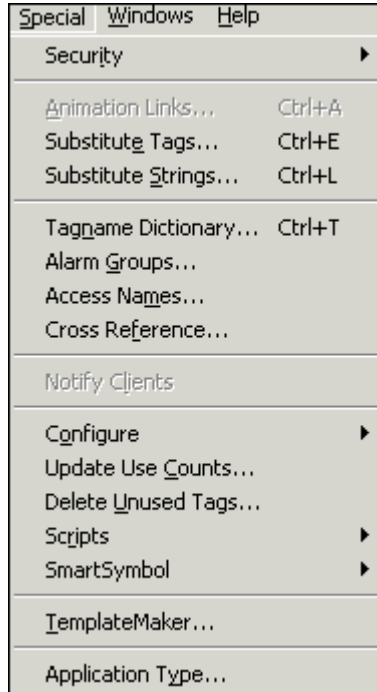
- Overview
- The SmartSymbol Manager
- Generating SmartSymbols
- Instantiating SmartSymbols
- Editing Existing SmartSymbols
- The SmartSymbol Library
- The SmartSymbol Properties Dialog Box
- Error Handling for the SmartSymbol Manager
- SmartSymbols at Runtime
- Redirect Remote References Using a Script

Overview

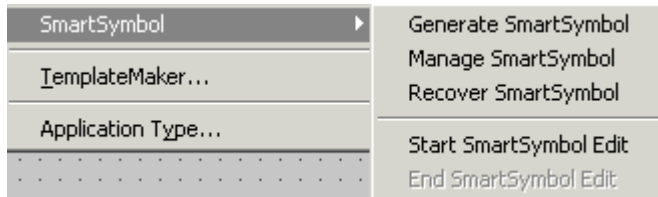
Using the SmartSymbol Manager, application developers can create symbols as templates to connect either to ArchestrA objects, to local InTouch tags or to InTouch tags through remote references. Once you have created symbol templates and saved them into the Symbol Library, you can instantiate the symbols by selecting them and dropping them into an InTouch window.

The SmartSymbol Manager provides application developers with browsers that are easy to navigate and dialog boxes to select and configure the AutomationObjects. It also provides a Tag Browser interface. A scripting function has been added to InTouch to allow users to modify the remote ArchestrA Object or InTouch Tag references at runtime. The scripting function also allows users to modify the Access Name (Galaxy or InTouch Tag Server node). The main functional components of the SmartSymbol utility are the Symbol Library and the Symbol Manager.

The SmartSymbol Manager is accessible in InTouch WindowMaker via the **Special** menu.



On the **Special** menu, point to **SmartSymbol** to access the SmartSymbol menu.



The options on the SmartSymbol menu are as follows:

Menu Item	Description
Generate SmartSymbol	Generates a SmartSymbol from the selected cell.
Manage SmartSymbol	Displays a list of folders containing existing SmartSymbols. You can move existing SmartSymbols or folders around within the Symbol Library by dragging and dropping them in the desired location. For more information, see The SmartSymbol Library.
Recover SmartSymbol	Allows users to recover SmartSymbols from instances in windows that were deleted from the SmartSymbol Library. For more information, see Recovering Orphaned Symbol Instances.

Menu Item	Description
Start SmartSymbol Edit	Allows users to make changes to a SmartSymbol in the library. For more information, see Editing Existing SmartSymbols.
End SmartSymbol Edit	Allows users to apply changes made when editing an existing SmartSymbol. For more information, see Editing Existing SmartSymbols.

The SmartSymbol Manager

The SmartSymbol Manager allows application developers to create, instantiate, and manage the InTouch Application's SmartSymbols. With its import and export features, the SmartSymbol Manager also allows developers to transport SmartSymbols among various InTouch applications and across different physical systems.

Note SmartSymbols are supported in ArcestrA platforms and architectures with Industrial Application Server 2.0 or higher. The Integrated Development Environment must be installed on the InTouch node for using the ArcestrA related features like Template Browser, Object Browser and new instance creation when working with SmartSymbols.

The SmartSymbol Manager Menu Items

The SmartSymbol Manager has a menu bar with the following menus and menu items:

File Menu (Alt + F)

Menu Item	Description	Shortcut Keys
New Folder	Creates a new folder in the selected location in the Symbol Manager. This function is only available when the focus is on a folder in the tree.	Alt + F, then press F
New Template Folder	Creates a template folder in the selected location in the Symbol Manager. This function is only available when the focus is on a folder under the "ArchestrA Symbols" folder or on a folder in the tree.	Alt + F, then press T
Browse New Template Folder	Allows the user to browse the Galaxy for the name of an existing Object Template to use in creating a new Template Folder. This function is only available when the focus is on a folder under "ArchestrA Symbols" folder or on a folder in the tree.	Alt + F, then press B
Delete	Generates a prompt to delete the selected folder, template folder or symbol.	Alt + F, then press D
Import	Allows the user to import a symbol package file from another application's Symbol Library.	Alt + F, then press I
Export	Allows the user to export one or more symbols to a package file.	Alt + F, then press X
Close	Closes the SmartSymbol Manager dialog box.	Alt + F, then press C

Edit Menu (Alt + E)

Menu Item	Description	Shortcut Keys
Cut	Removes a folder, template folder or symbol to be pasted in another location. The cut is not executed until the item is pasted in another location.	Ctrl + X
Copy	Copies a folder, template folder or symbol to be pasted in another location.	Ctrl + C
Paste	Pastes a folder, template folder or symbol copied from another location.	Ctrl + V
Rename	Allows you to rename the selected folder, template folder or symbol.	Alt + E, then press R

Tools Menu (Alt + T)

Menu Item	Description	Shortcut Keys
Galaxy Configuration	Enables the user to select or change the GR Node and Galaxy name.	Alt + T, the press G

Help Menu (Alt + H)

Menu Item	Description	Shortcut Keys
SmartSymbol Help	Displays the online help for SmartSymbols.	F1

Storage Hierarchy for SmartSymbols in the Symbol Library

SmartSymbols are stored in the Symbol Library in a standard hierarchical folder structure. Two standard folders are included to simplify the organization of SmartSymbols in the Symbol Library. The standard folders include a top level folder for ArchestrA Object Symbol Templates and for InTouch Tag Symbol Templates. Users can create folders for object templates using the Symbol Manager. Users should store ArchestrA Object symbol templates in the object template folder with which they should be associated upon instantiation. For example, if the user instantiates a SmartSymbol to use for \$Valve objects, the user should store the symbol template in the "\$Valve" template folder.

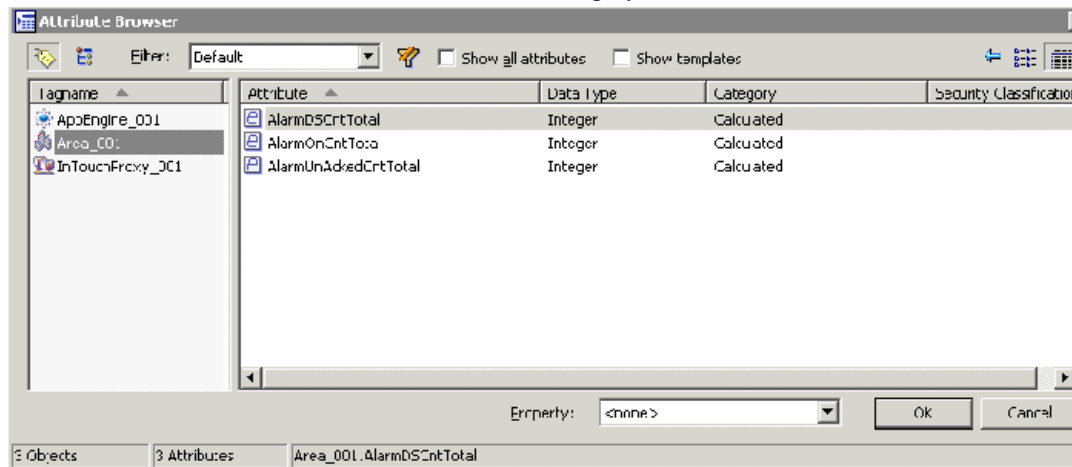
Generating SmartSymbols

Using the SmartSymbol Manager, you can generate SmartSymbols that are associated with ArcestrA object templates, InTouch remote tags or generic SmartSymbols. This section includes steps on how to generate these three types of SmartSymbols.

Generating a new SmartSymbol for use with an ArcestrA object template

To generate a new SmartSymbol for use with an ArcestrA Object template

1. Create a new InTouch window in WindowMaker.
2. Using the InTouch graphics drawing tools and/or Wizards, create a graphic that you want to make into a SmartSymbol.
3. From the **Special** menu, click **Animation Links**, click the type of animation link to configure, then double-click in the Expression text box to pick from a list of Object Templates to define the animation reference. The Attribute Browser is displayed.



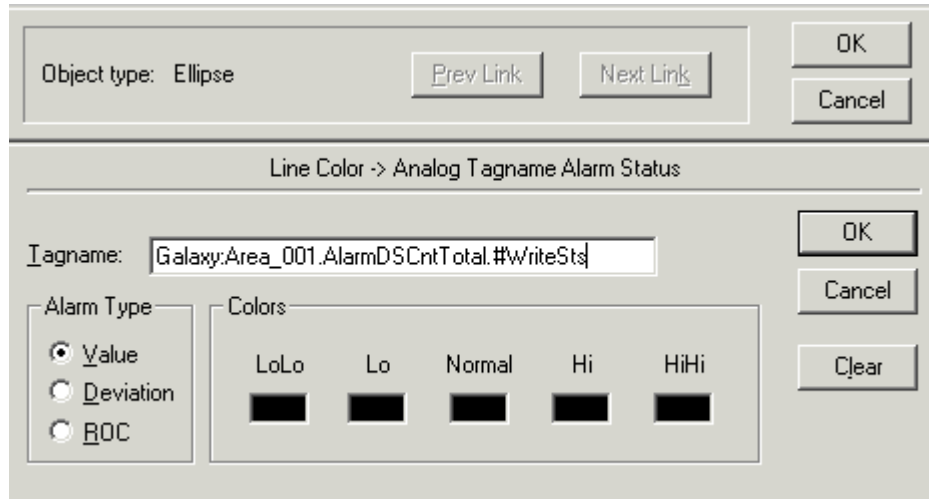
Note The first time you double-click in the Expression text box, the Tag Selection dialog box is displayed. The user must define a new Galaxy tag source in order to access the Attribute Browser.

4. Configure the animation for the graphical elements using references to Galaxy Object or template data. If you enter the references manually, you must make sure that the Galaxy Object data references follow the "Galaxy:\$Template.attribute" syntax.

The animation references for all the connections to an ArcestrA template are defined as:

Galaxy:\$<TemplateName>.<AttributeName>

where the <TemplateName> may be followed by more contained object names and the <AttributeName> may be a hierarchical set of attribute names.



A SmartSymbol consists of all instances of the object template associated with the folder in which the symbol template is stored. For example, a symbol placed in the "Pump" folder in the Symbol Manager will have the "\$Pump" object template associated with it.

Note Where animation points reference Galaxy:\$TemplateName.PV, the TemplateName will be changed to the instance name only if TemplateName matches an actual template name. Where animation points reference Galaxy:me.PV, the me will always be replaced by the instance name. Within InTouch, "Galaxy" is a key word which identifies that the reference is to an ArchestrA Galaxy. The actual galaxy is identified when you instantiate SmartSymbols and specify the GR Node and Galaxy Name.

5. Select the specific attribute to reference and click **OK**. The Object Template name and the selected attribute name using the proper ArchestrA syntax (i.e. "galaxy:\$valve.pv") will be displayed in the reference text box.

Tip You can also use the Attribute Browser to select a specific Galaxy Object and attribute.

6. Click the graphic, then go to the **Arrange** menu and click **Make Cell**. As an alternative, right click the graphic, point to **Cell/Symbol** and click **Make Cell**.

Tip To select multiple graphics to make into a cell, click the arrow icon, then click in WindowMaker and drag the arrow until the select box contains the desired graphics, then right-click in the selected area, point to **Cell/Symbol** and click **Make Cell**.

7. From the **Special** menu, point to **SmartSymbol** and click **Generate SmartSymbol**. The SmartSymbol Manager creates the new SmartSymbol for Galaxy data and creates an ArchestrA template folder if one does not already exist. The Symbol Manager dialog box displays thumbnails of the new symbol you created as well as any symbols in the Symbol Library. A default name is automatically assigned to the symbol (for example, New Symbol).

The new SmartSymbol is stored in either the top level ArchestrA Symbols folder or in the template folder created or associated with it. To rename the symbol, right click the symbol and click **Rename**. If you enter an invalid name or a duplicate name, the system will prompt you to enter a valid or unique name.

8. Click **OK** in the Symbol Manager dialog box to save the new symbol in the SmartSymbol Manager library. The symbol will be available for future use in this or other windows of this InTouch application.

Note The graphics used to create the SmartSymbol in the InTouch application are cells, not SmartSymbols. In the application window, you can delete a displayed cell, then use the SmartSymbol Manager to instantiate a SmartSymbol.

Generating a generic SmartSymbol without a specific ArchestrA Object Template binding in the references

To generate a generic SmartSymbol

1. Create a new InTouch window in WindowMaker.
2. Using the InTouch graphics drawing tools and/or Wizards, create a graphic that you want to make into a SmartSymbol.
3. From the **Special** menu, click **Animation Links** and configure the applicable animation for the various graphic elements using references to the data.
4. Manually enter a reference to Galaxy data using the "me." syntax and specifying the Galaxy and the specific attribute name. For example, "Galaxy:me.pv."
5. Click the graphic, then go to the **Arrange** menu and click **Make Cell**. As an alternative, right click the graphic, point to **Cell/Symbol** and click **Make Cell**.

Tip To select multiple graphics to make into a cell, click the arrow icon, then click in WindowMaker and drag the arrow until the select box contains the desired graphics, then right-click in the selected area, point to **Cell/Symbol** and click **Make Cell**.

6. From the **Special** menu, click **Generate SmartSymbol**. The Symbol Manager dialog box is displayed. By default, the new SmartSymbol is placed in the ArchestrA Symbols top level folder. You can move it to any existing subfolder.
7. To rename the symbol, click the **Edit** menu, then click **Rename** or right-click on the symbol, then click **Rename** and enter the desired name.

Note The name of the SmartSymbol must be unique within the folder where it is stored. If you enter an invalid or duplicate name, the system will indicate the invalid entry and replace the entry with the previous valid name.

Note When you rename a Smart Symbol in the Library, the instances of that Symbol will not be affected by that rename. Instead, the new name for the symbol template will be visible the next time you access the Properties dialog box.

8. Click **OK** in the Symbol Manager dialog box to save the new symbol in the SmartSymbol Manager library. The symbol will be available for future use in this or other windows of this InTouch application.

Note The graphics used to create the SmartSymbol in the InTouch application are cells, not SmartSymbols. In the application window, you can delete a displayed cell, then use the SmartSymbol Manager to instantiate a SmartSymbol.

Generating a new SmartSymbol that uses InTouch Data

To generate a new SmartSymbol that uses InTouch data

1. Create a new InTouch window in WindowMaker.
2. Using the InTouch graphics drawing tools and/or Wizards, create a graphic that you want to make into a SmartSymbol.
3. From the **Special** menu, click **Animations Links** and configure the applicable animation for the various graphic elements using references to the tag data, either directly to local tags or as remote tag references.

Note You can either select references using the Tag Browser or manually type in references using the correct syntax. To access the Tag Browser, double-click in the Expression text box. The Tag Browser is displayed.

4. Click the graphic, then go to the **Arrange** menu and click **Make Cell**. As an alternative, right click the graphic, point to **Cell/Symbol** and click **Make Cell**.

Tip To select multiple graphics to make into a cell, click the arrow icon, then click in WindowMaker and drag the arrow until the select box contains the desired graphics, then right-click in the selected area, point to **Cell/Symbol** and click **Make Cell**.

5. From the **Special** menu, click **Generate SmartSymbol**. The Symbol Manager dialog box is displayed. By default, the new SmartSymbol is placed in the InTouch Symbols top level folder. You can move it to any existing subfolder.
6. To rename the symbol, click the **Edit** menu, then click **Rename** or right-click on the symbol, then click **Rename** and enter the desired name.

Note The name of the SmartSymbol must be unique within the folder where it is stored. If you enter an invalid or duplicate name, the system will indicate the invalid entry and replace the entry with the previous valid name.

7. Click **OK** in the Symbol Manager dialog box to save the new symbol in the SmartSymbol Manager library. The symbol will be available for future use in this or other windows of this InTouch application.

Note The graphics used to create the SmartSymbol in the InTouch application are cells, not SmartSymbols. In the application window, you can delete a displayed cell, then use the SmartSymbol Manager to instantiate a SmartSymbol.

Limitations of SmartSymbols

The following are known limitations of SmartSymbols:

- A SmartSymbol cannot contain a trend object. If you attempt to instantiate a SmartSymbol that contains trend object (historical or real-time), an error message will be displayed.
- You cannot use the standard InTouch Substitute Tags or Substitute Strings options on the right-click menu in WindowMaker. However, you can substitute tags or strings from the SmartSymbol Properties dialog box.
- If you have InTouch 9.0 and Industrial Application Server 1.5 installed and you click the ellipses in the Symbol Properties dialog box to browse for an ArchedrA instance, an error message will be displayed because new features in the Attribute Browser are not available in Industrial Application Server 1.5. In order to browse for ArchedrA instances, you must have Industrial Application Server 2.0 installed.
- Generating a SmartSymbol with an SPC Charts wizard is not supported.
- The Attribute Browser does not show derived object instances. To address this issue, create the derived template in the Symbol Manager or create a custom filter in the browser.

Instantiating SmartSymbols

Once you have generated a SmartSymbol, the SmartSymbol can be instantiated onto an InTouch window. You can instantiate a SmartSymbol with references to an ArchestrA object, create a new ArchestrA object instance to reference in the new SmartSymbol reference or change references in a SmartSymbol instance.

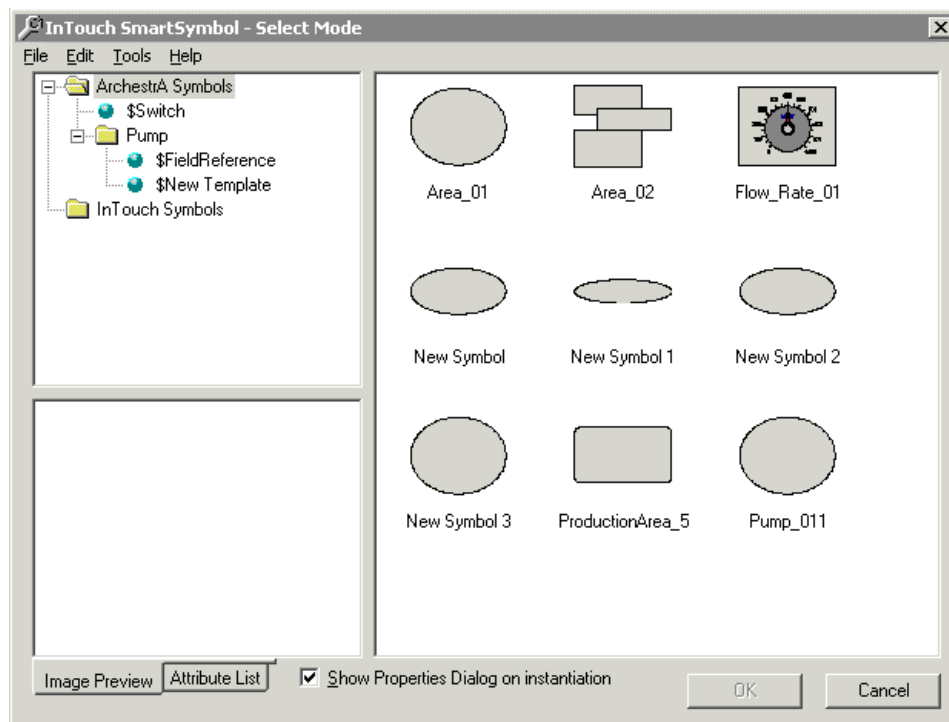
Instantiating a SmartSymbol with a reference to an ArchestrA object

To instantiate a SmartSymbol with a reference to an ArchestrA object

1. Open WindowMaker and create a new window.
2. Click the Wizard hat, click **SmartSymbol**, then click **OK** or click the SmartSymbol icon on the toolbar.

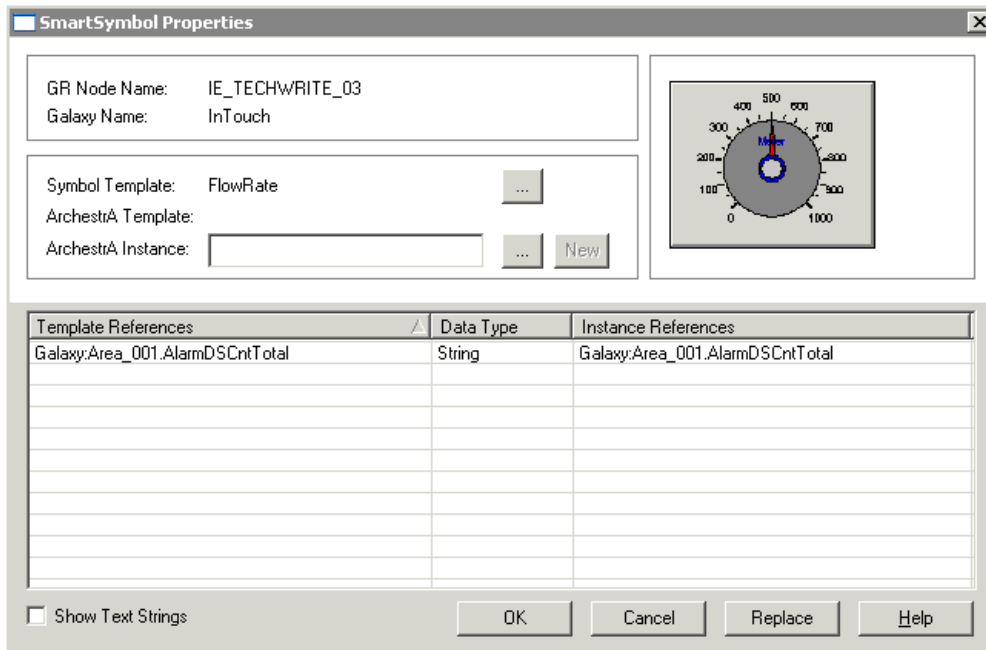
Note The SmartSymbol icon can be added to the toolbar. To add the SmartSymbol icon, click the Wizard icon, select the SmartSymbol Manager, then click **Add to Toolbar** and click **OK**.

3. Click in the area of the WindowMaker window where you want to place the symbol. The Select Mode dialog box of the Symbol Manager is displayed.



- Double-click the symbol or click the symbol, then click **OK**. If the **Show Properties Dialog on instantiation** checkbox is checked (default), the Properties dialog box is displayed. Otherwise, you will return to InTouch WindowMaker, where the new symbol is displayed.

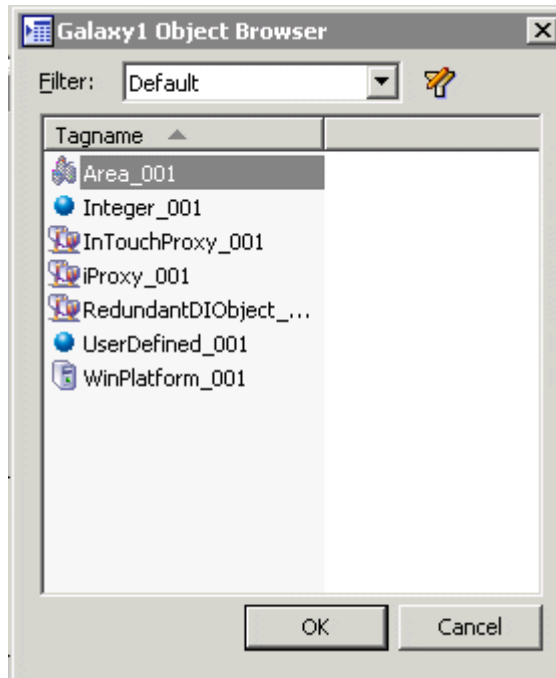
Note In WindowMaker, double-click the SmartSymbol to display the Properties dialog box.



The SmartSymbol dialog box displays the following information:

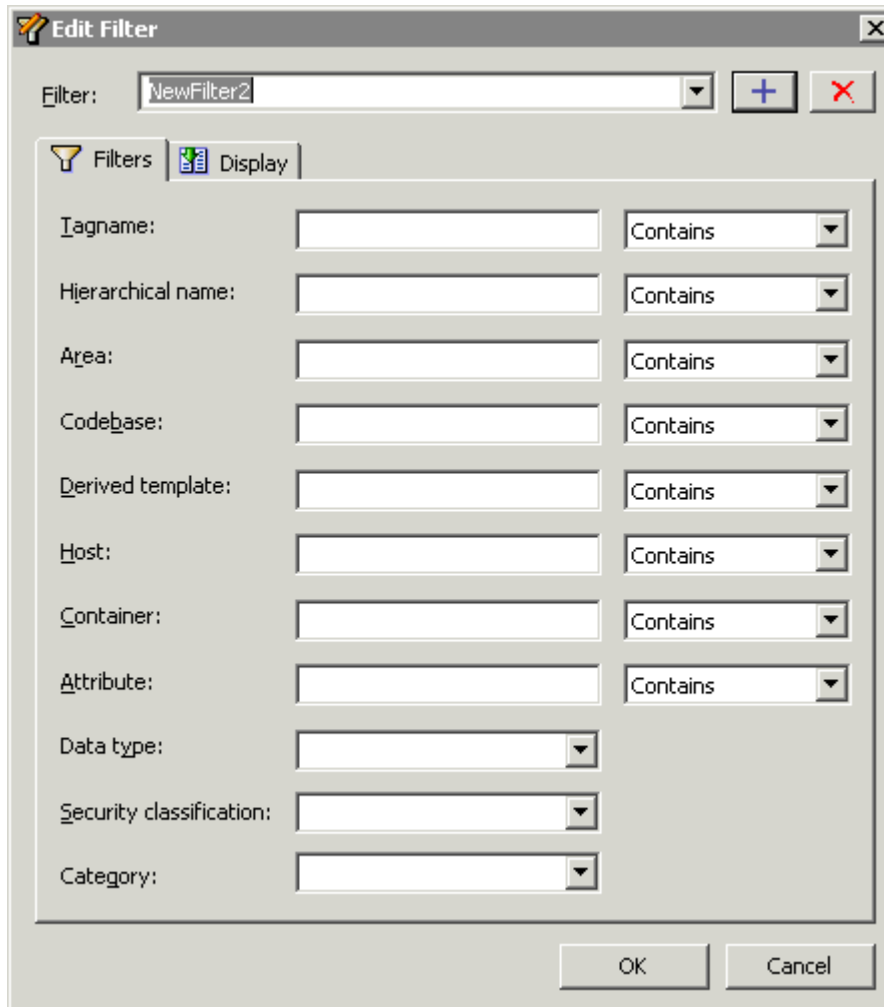
- Symbol Template name
 - GR Node name
 - Galaxy name
 - ArchestrA Template name, if one has been associated with the symbol
 - A text box to select an ArchestrA Object instance (used to resolve the unresolved references)
 - A list of the reference strings defined within the symbol (as an option, you can view static text strings as well by checking the Show Text Strings check box)
- In the ArchestrA Instance text box, click the ellipses to browse for an Object Instance. These will be Object Instances of the Object Template associated with the Symbol Folder that the Symbol Template is stored in. If the Symbol Template does not belong to a Symbol Template folder (belongs to only ArchestrA Symbols folder), then all object instances will be listed.

Tip Depending on the number of objects deployed in your Galaxy application, the amount of data displayed in the Attribute Browser can be large. To make this information more manageable, reduce the data shown by creating a filter. Creating a filter is explained below.



Note If there is no template configured for the symbol, all object instances are listed.

6. To create a filter, click the **Edit Filter** icon  , then click the plus sign  . The Edit Filter dialog box is displayed.



Note The Attribute, Data type, Security Classification and Category fields are available, but they will not filter for those attributes. Also, the Attribute Browser does not show derived object instances. To address this issue, create the derived template in the Symbol Manager or create a custom filter in the browser.

7. Select the instance to map to the SmartSymbol. The system displays the instance attribute references in the list that contain either "Galaxy:me.<attr>" or "Galaxy:\$<templatename>.<attr>" references updated with the newly selected Object instance name in place of the "me" or the "\$<templatename>", where the \$<templatename> matches the template or a parent template of the object instance selected.

Note Other reference strings in the symbol that contain either direct object instance references or other \$Template names not related to the upward template hierarchy (exact Template or parent and above) of the selected object instance will not be affected by the selection and will remain unchanged.

8. Click **OK** to close the Symbol Properties dialog box.
9. From the **File** menu in WindowMaker, click **Save Window**.

Creating an ArcestrA Object Instance to Reference in the New SmartSymbol Instance

To create a new ArcestrA object instance

1. Click the Wizard hat, click **SmartSymbol**, then click **OK** or click the SmartSymbol icon on the toolbar.
2. Click in the area of the WindowMaker window where you want to place the symbol. The Select Mode dialog box of the Symbol Manager is displayed.
3. Double-click the symbol or click the symbol, then click **OK**. If the **Show Properties Dialog on instantiation** checkbox is checked (default), the Properties dialog box is displayed. Otherwise, you will return to InTouch WindowMaker, where the new symbol is displayed.

Note In WindowMaker, double-click the SmartSymbol to display the Properties dialog box.

Template References	Data Type	Instance References
Galaxy:Area_001.AlarmDSCntTotal	String	Galaxy:Area_001.AlarmDSCntTotal

4. Enter any valid object name in the ArchestrA Instance field and click **New**. If this is the first time you have attempted this action during the current session, a login dialog box with the Domain Name, User name and Password fields will be displayed.
5. Enter a valid User Name, Password and Domain Name (If the IDE is set to a mode other than None, a Domain Name is required only for OS User or OS Group Based security) and click **OK**.

Note If you enter the name of an existing object instance, you will be prompted to enter a new name.

Changing References in a SmartSymbol Instance

Once one or more SmartSymbols have been placed onto a window as instances, their remote references can be changed to point to something different, whether it be another object or a different InTouch tag.

To edit references in a SmartSymbol instance

1. In WindowMaker, open the window in which the SmartSymbol linked to an AutomationObject is located.
2. Double-click the symbol. The Symbol Properties dialog box is displayed.
3. Click the ellipses button next to the Symbol Template text box to select a new symbol template or click the ellipses button next to the ArchestrA Instance text box to browse for an object instance.
4. Selects a different object instance to map to the SmartSymbol and click **OK**.
5. From the **File** menu in WindowMaker, click **Save Window**.

Editing Existing SmartSymbols

Application developers can edit existing SmartSymbols. Editing involves breaking the cell and using the drawing tools and the animation links dialog box to make changes.

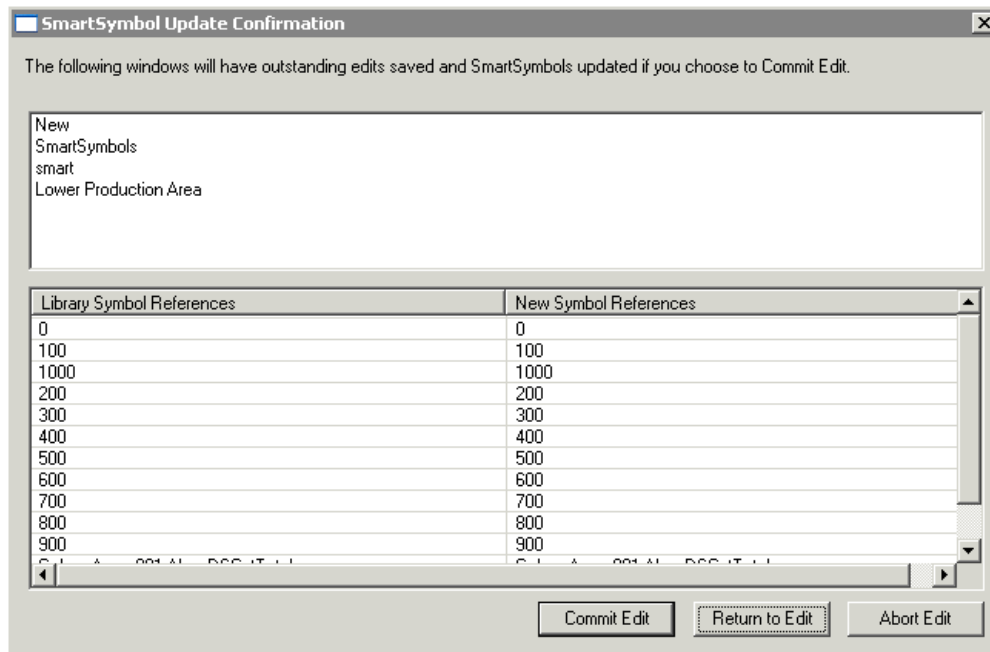
To edit existing SmartSymbols

1. Open WindowMaker and open an existing window or create a new window.
2. From the **Special** menu, point to **SmartSymbol** and point to **Start SmartSymbol Edit**. The Symbol Manager dialog box is displayed.
3. Select a SmartSymbol that you want to edit from those listed in the Library and place it in a window. It is recommended that you edit SmartSymbols in a temporary window rather than in an application window.
4. Right-click the symbol in the window and click **Break Cell**. The symbol is broken up and each of the elements is selected in the window.

5. Selects one of the symbol's elements to edit and change the size or shape, etc. as desired using the InTouch graphic editing functions. Edits must be performed on one element at a time.

Note If you add elements to a cell that is part of a SmartSymbol, this can result in a bigger cell. When you propagate, the size of existing SmartSymbol instances is protected.

6. Hold the Shift key and click all elements of the symbol, the right-click in the selected area and click **Make Cell**.
7. From the **Special** menu, click **End Symbol Edit**. A confirmation dialog box is displayed.



The confirmation dialog box has the following options:

Function	Description
Commit Edit	Updates the Symbol with the changes made during the SmartSymbol Edit.
Return to Edit	Cancels the End Symbol Edit command and allows the user to continue editing the Symbol.
Abort Edit	Terminates the Edit mode and leaves the Symbol unchanged.

8. Click **Commit Edit** to apply the changes. The Symbol Manager updates the Symbol to reflect the changes made. The changes made to the symbol template are propagated automatically to the symbol instances.

The SmartSymbol Library

The SmartSymbol Library contains the SmartSymbols for an InTouch application. The SmartSymbol Manager is used to manage the contents of the Symbol Library. Reference information about the symbols in the library is stored in an associated XML file. The XML file includes a version number at the root of the data structure indicating the XML version. There is also a version number in the Symbol Library file, which is located in the Symbol folder under the application directory. The logic and additional data for the symbols is stored in a binary format within the SmartSymbol Manager.

Symbols and their related files are stored at the application level under the Symbols folder in the application directory. The symbols can be exported from and imported to other InTouch applications.

Under the Symbols folder, there is a folder for each symbol. Each folder contains any bitmaps included in the symbol.

Moving SmartSymbols around in the Symbol Library

To move existing SmartSymbols or folders around within the Symbol Library, click the symbol, then drag and drop the symbol(s) and/or folders to a new folder.

You can drop symbols on folders or on symbols under folders. When you drop a symbol on another symbol, the symbol you are moving will be stored in the same folder as the other symbol.

When you drop a SmartSymbol into an ArcestrA Object Template folder, the ArcestrA-related symbols will be assigned to the new Object Template in the target folder. The references to InTouch tags are not changed.

Note SmartSymbols and Folders can only be moved within in their respective top-level folder, "ArcestrA Symbols" or "InTouch Symbols," but not from one top-level folder to the other.

Exporting SmartSymbols from the Symbol Library

When you have created or imported SmartSymbols and have them in your application's Symbol Library, you can export symbols to other libraries. Exporting symbols to other libraries allows you to use symbols that you have already created in other InTouch applications.

To export a SmartSymbol

1. From the **Special** menu, click **SmartSymbol**, then click **Manage SmartSymbol**.
2. From the Symbol Manager's list of symbols and folders, click the SmartSymbol(s) and/or folder(s) to export. To select multiple symbols and/or folders, hold the **Ctrl** key and click the desired items.

3. From the **File** menu, click **Export**. The Export Symbol dialog box is displayed.
4. Browse for the file directory to which you want to export the symbol, enter a file name for the package file and click **OK**.

Note If a SmartSymbol package file of the same name already exists in the directory, you will be prompted to overwrite the existing package file or choose a different directory and/or file name.

The system then exports the selected SmartSymbols defined in the Library to the package file stored in the specified directory.

Importing SmartSymbols into the Symbol Library

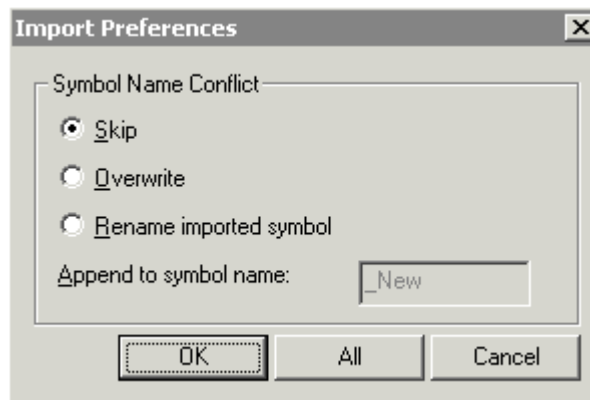
You can import SmartSymbols from other InTouch applications into your application's symbol library. Importing symbols from other applications allows you to reuse symbols that you have already created instead of creating those symbols once again.

To import SmartSymbols into the Symbol Library

1. From the **Special** menu, click **SmartSymbol**, then click **Manage SmartSymbol**.

Note You cannot import SmartSymbols unless you have closed all windows in WindowMaker first.

2. From the **File** menu, click **Import**. The Import Symbol dialog box is displayed.
3. Browse for the file directory that contains the SmartSymbol package file to import, click the package file, then click **OK**. The Import Preferences dialog box is displayed.



Note You can only import folders that contain one or more SmartSymbols. Importing an empty folder into the SmartSymbol Library is not supported.

If the name of the symbol package file you are importing conflicts with that of a SmartSymbol already in the Symbol Library, you can choose from the following options:

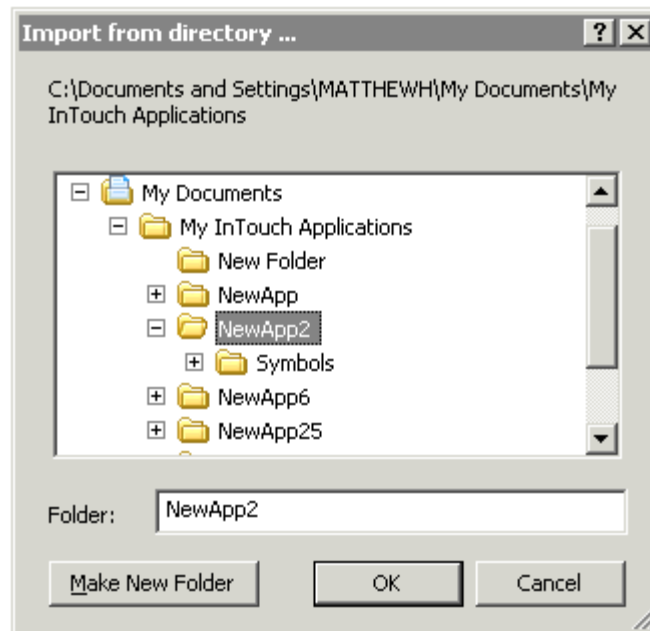
- Click the **Skip** radio button to skip the import of this symbol. If you are importing multiple symbols, the system will continue importing the rest of the symbols.
- Click the **Overwrite** radio button to overwrite the existing symbol with the new one.
- Click **Rename** radio button to name the new symbol with an unused name.
- Click **OK** to apply the selected radio button command to the SmartSymbol.
- Click **All** to apply the selected radio button command to all SmartSymbols in the package file.
- Click **Cancel** to cancel the entire import operation and do not import any of the subsequent specified symbols. Any symbols imported prior to choosing Cancel will remain in the Symbol Library.

Importing Windows that Contain SmartSymbols

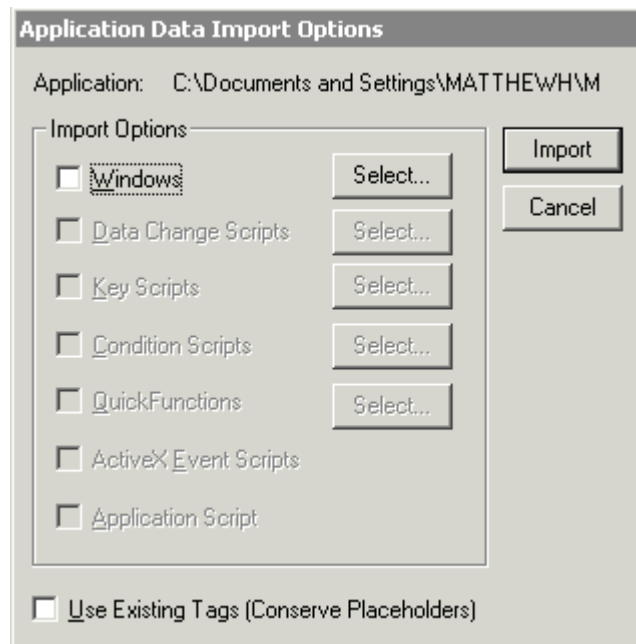
You can import windows containing SmartSymbols from other InTouch applications using the **Import** option on the **File** menu in WindowMaker.

To import a window containing SmartSymbols

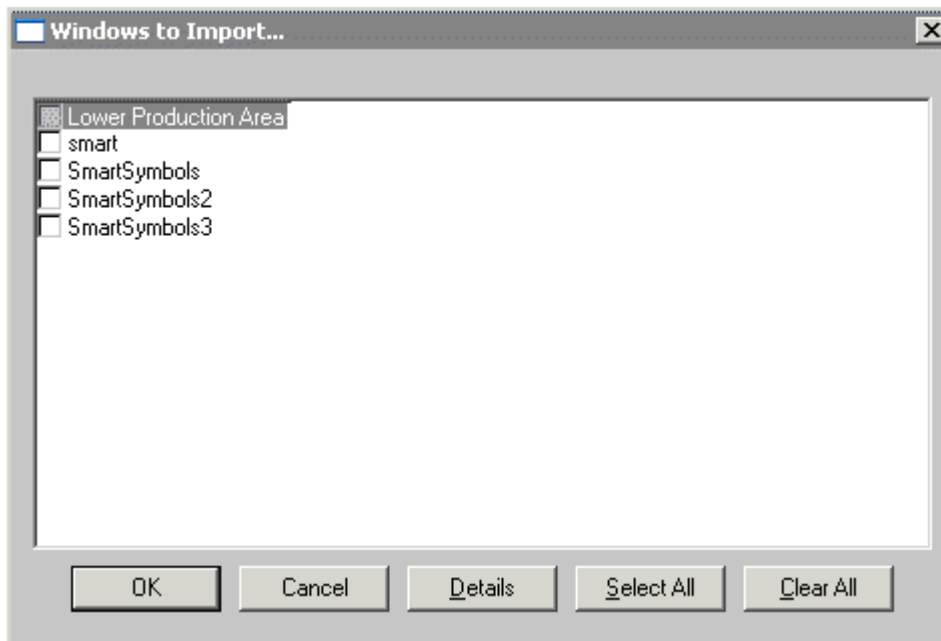
1. In WindowMaker, click **File**, then click **Import**. The Import from directory dialog box is displayed.



2. Browse to the directory where the InTouch window is stored, click the folder, then click **OK**. The Application Import Data Options dialog box is displayed.



- Click **Select**. The Windows to Import dialog box is displayed.



- Click the check box(es) for the window(s) to import, then click **OK**.
- Click **Import**. The selected windows are imported to your InTouch applications.

Renaming SmartSymbols within the Library

Use the Symbol Manager to rename SmartSymbols as needed. To rename a SmartSymbol, click the **Edit** menu, click **Rename** and type the new name for the corresponding symbol. Renaming a SmartSymbol has no impact on any Symbol instances.

Deleting SmartSymbols from the Library

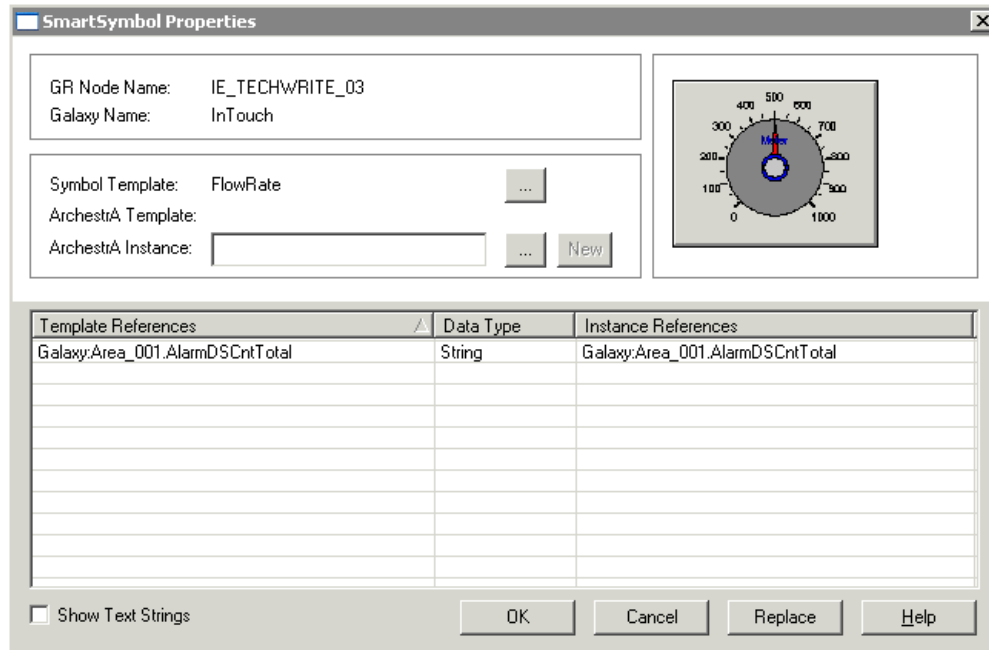
When you delete one or more SmartSymbols from the Symbol Library, a warning dialog box is displayed to inform you of the consequences of such deletions. Deleting the SmartSymbols will affect any previously placed instances of that Symbol in existing InTouch windows in that you will no longer be able to access or edit or view the properties of that Symbol instance. The runtime state of those symbol instances will not be affected by the deletion.

Recovering Orphaned Symbol Instances

To recover SmartSymbol instances that have been orphaned due to the deletion of the symbol template, click the SmartSymbol, then click the **Special** menu and click **Recover SmartSymbol**.

Note If you attempt to access the properties of an instance for which the Library Symbol has been deleted, a warning message will be displayed telling you that the Symbol no longer exists in the library. You can choose to recover the Library Symbol from the instance, placing it back into the library.

The SmartSymbol Properties Dialog Box



The SmartSymbol Properties dialog box allows users to perform the following functions:

- Browse for ArcestrA instances

Users can type the name of the ArcestrA instance or browse for it using the integrated Object Browser. The user must identify a Galaxy connection by specifying a Galaxy Repository (GR Node) Name and Galaxy Name for the Object Instance for the browsing function to be available. The Galaxy Name combo box will be unavailable until the user enters a valid, accessible GR Node name. The ArcestrA Instance combo box will be unavailable until the user enters a valid, accessible Galaxy Name. The listed References and Instance Attribute information will be updated when the user has completed the browsing operation.
- Search for and replace instance attributes

The SmartSymbol Properties dialog box supports partial string search and replacement. Even the user clicks the Search button, a search and replace dialog box is displayed. The search and replace function does not modify text strings if they are not shown (**via unchecked checkbox**).
- Display and edit text strings

This function shows text strings in the list view and allows users to edit the string manually or by using the search/replace function.

- Change the symbol template after instantiation
Use the Symbol Manager's Browse function. The template data is merged into the instance data and the instance references are refreshed by the template references.
- Edit the Instance Attributes information (double click access to the Tagname Dictionary and manual edit methods) - this is accomplished via an update button and panel.
- Type check instance attribute references
This function is supported only for InTouch tag references. Tags that do not type match will show the Required Type text in red. Non-ArchestrA Templates other than ArchestrA templates display "N/A" in the ArchestrA Template text box and the ArchestrA Instance fields and the associated Browse button is unavailable.

Note The SmartSymbol Properties dialog box does not support searching and replacing static string information without modifying the tag references. The Properties dialog box also does not support the verification of ArchestrA instances.

Error Handling for the SmartSymbol Manager

Error	Result
Error in creating a new object or reference link	The object or reference link creation fails and the error is logged in the Logger.
Missing file or failure opening a file	A placeholder graphic is placed in the window where the SmartSymbol was to be located and the error is logged in the Logger.
Corrupt XML file structure	An error message stating, "Unable to load xml file..." is displayed during the attempt to instantiate the SmartSymbol and the warning is logged in the Logger.
Missing or incorrect property fields	The object or reference link creation fails and the error is logged in the Logger.
Missing Nodes	The object or reference link creation fails and the error is logged in the Logger.
Nodes of the wrong type	The object or reference link creation fails and the error is logged in the Logger.
Missing Variables	The object or reference link creation fails and the error is logged in the Logger.
Missing Buffer	The object or reference link creation fails and the error is logged in the Logger.
Missing Point	The object or reference link creation fails and the error is logged in the Logger.

Error	Result
Missing Rectangle	The object or reference link creation fails and the error is logged in the Logger.
Null Bitmap	A bitmap without an image is created for the symbol and the error is logged in the Logger.
Unknown Object	The error condition is ignored and the event is logged as a warning in the Logger.
Unknown Link	The error condition is ignored and the event is logged as a warning in the Logger.
Unknown Node	The error condition is ignored and the event is logged as a warning in the Logger.

SmartSymbols at Runtime

At runtime, SmartSymbol instances in InTouch windows function much like other InTouch graphics. However, SmartSymbols are also able to display animation or data values from Industrial Application Server AutomationObject attributes when configured to do so.

Redirect Remote References Using a Script

InTouch application developers can redirect ArchestrA object references or remote references to InTouch tags at runtime by writing a script. This enables developers to switch object instances for a particular graphic symbol based on particular conditions being met or directly via user interaction such as clicking a button. This new script function is applicable to all objects, not just SmartSymbols.

The new script function used to redirect remote references at runtime is `IOSetRemoteReferences`.

IOSetRemoteReferences()

	Changes object references at runtime. This can be used to change ArchestrA references, remote InTouch references and references to IO tag servers.	
Category	Data Change	
Syntax	IOSetRemoteReferences(BaseAccess, NewAccess, MatchString, SubstituteString, Mode);	
	Parameter	Description
	BaseAccess	This string parameter specifies the original configured Access Name to match to in the reference(s)
	NewAccess	The new Access Name. This is applied to all references for which the original Access Name matches the string provided in BaseAccess and for which the original Item Name matches the MatchString (if one is specified).
	MatchString	The string to match in the original configured Item Name in the reference(s). If this is an empty string, it is treated as a match for any Item Name.
	SubstituteString	The string to substitute into the original configured Item Name, replacing the MatchString, to produce the active Item Name for the reference(s). If this is an empty string, no substitution is made.
	Mode	Determines the way in which the MatchString must match the original configured Item Name. Matching is always from the beginning of the Item Name, never in the middle. 0 denotes that the match must be for the entire Item Name or up to a "." 1 denotes that a partial match is allowed, even if the next character is not a "."
Example(s)	<pre>IOSetRemoteReferences("Galaxy", "", "pumpX", "pump001", 0); IOSetRemoteReferences("Galaxy", "TagServer1", "pumpX", "p2", 0); IOSetRemoteReferences("TagServer1", "TagServer2", "pumpX", "backpump3", 0) IOSetRemoteReferences("TagServer1", "", "Tank", "Plant", 1)</pre>	

Redirecting References

- IOSetRemoteReferences only changes remote references. It only affects those for which the original, configured Access Name matches the indicated BaseAccess, and for which the original, configured Item Name string begins with the indicated MatchString.
- A single call to IOSetRemoteReferences affects all remote references in open windows, for which their original, configured name strings match BaseAccess and MatchString.
- If you leave BaseAccess empty (i.e. ""), IOSetRemoteReferences does not perform any functions.

- If MatchString is empty (i.e. ""), IOSetRemoteReferences affects all remote references for which the original AccessName matches BaseAccess.
- The new active name for a remote reference is obtained by using NewAccess as the "active" Access Name and by copying the original Item Name and replacing SubstituteString with NewString.
- When the Match Mode flag is set to '0', replacement in the Item Name is only done for full object names (or tagnames), or full attribute names (or dot fields). MatchString must match the entire original Item Name or up to a character followed by a period (".").
- When the Match Mode flag is set to '1', partial replacement of the item string will be allowed when the item string starts with the match item string. That is, MatchString must match some portion of the original item string, but that sub-portion must start at the beginning of the item string and the last character does not need to be followed by a ".".
- The original, configured names for a remote reference remain unchanged. Subsequent calls to IOSetRemoteReferences do not need to recognize the current active name. Calls to IOSetRemoteReferences() can take place in any order.

Restoring References

- If SubstituteString is empty (i.e. ""), IOSetRemoteReferences restores the active Access Name to the original base Access Name.
- If MatchString is empty (i.e. ""), IOSetRemoteReferences restores the active Item Name to the original Item Name.

Note Even if SubstituteString is not empty, if MatchString is empty, the Item Name will be restored to the original Item name. Inserting text at the beginning of the name is not allowed. For example, running the script `IOSetRemoteReferences("Access1", "", "", "Valve", 0);` will not append the string Valve at the beginning or end of the all original item names.

- If SubstituteString is empty (i.e. ""), the IOSetRemoteReferences restores the active Item Name to the original Item Name. Using a non-empty MatchString with an empty SubstituteString enables you to select a subset of remote references on the indicated original base access and restore them to their original Item names.

IOSetRemoteReferences and InTouch Windows

IOSetRemoteReferences only affects the links in windows that are already open. The following scenarios explain the behavior of IOSetRemoteReferences when it applies to links in newly opened windows.

Scenario 1

1. Create two InTouch windows, Window1 and Window2.
2. On Window2, insert the link `xxx:TankLevel.Value`. This window will open only via an action button.

3. On Window1, add a text link to xxx:TankLevel.Value and add a second button to call `IOSetRemoteReferences("xxx","TagServer1","", "",0);`
4. In WindowMaker, click **Runtime**.
5. Run the script function.
The link on Window1 is updated with the value of `TagServer1:TankLevel.Value`.
6. Click the button to open the InTouch Window2.
The link on Window2 does not show the value of `TagServer1:TankLevel.Value`. because the script only affects the links in windows that are already open.

Scenario 2

1. In InTouch Window1, configure an action script to run the following:

```
Show "Window2" ;
IOSetRemoteReferences("xxx","TagServer1","", "",0);
```

2. In WindowMaker, click **Runtime**.

It may appear that links on Window2 would be updated correctly because the script function is run after opening the window. However, the script function runs before the window is fully initialized. Therefore, the references on the window are not redirected.

To address this, you could use the action button to set values for tags that would represent the script function parameters and open a popup window. You can configure a window script on the target window for OnShow that would that would use these tags to call the `IOSetRemoteReferences`. For example, on the action button, you could configure the following:

```
NewAccessNameTag="Tagserver1 "
Show "Window2" ;
```

On the OnShow window script, configure the following:

```
IOSetRemoteReferences("xxx",NewAccessNameTag,"","",0);
```

Script changes on Animation Link references within an Application

The scope of a change caused by the script function call is to change all related remote references in an application. This is to handle complex scenarios of symbols and remote references, where a number of different attributes of the same object may be accessed as a group. For example, a complex Pump symbol might access speed, flow, and data quality for whichever actual pump is selected. With a single script function call, all the related references throughout the application are updated to the proper target.

Therefore a single script function call could affect any of the following across multiple Windows, within a single InTouch application:

- Individual Symbol reference changes

"Galaxy:pump.pv"

- Group of Related Symbols with multiple related references where the call could switch between the ".pv", ".state", and ".cmd" attributes for the entire group:

Galaxy:mainpump.pv", "Galaxy:mainpump.state", "Galaxy:mainpump.cmd"
--

Galaxy:fwdpump.pv", "Galaxy:fwdpump.state", "Galaxy:fwdpump.cmd"

Galaxy:rsvpump.pv", "Galaxy:rsvpump.state", "Galaxy:rsvpump.cmd"

Galaxy:bckpump.pv", "Galaxy:bckpump.state", "Galaxy:bckpump.cmd"

Galaxy:altpump.pv", "Galaxy:altpump.state", "Galaxy:altpump.cmd"
--

- The references within Contained Symbols of Contained Objects, where one call to change the upper level object name "tank" to another instance, like "tank1" affects all the contained object reference:

Galaxy:tank.pump1.pv", "Galaxy:tank1.pump1.pv"
--

Galaxy:tank.invalve1.pv", "Galaxy:tank1.invalve1.pv"
--

Galaxy:tank.invalve2.pv", "Galaxy:tank1.invalve2.pv"
--

Galaxy:tank.outvalve.pv", "Galaxy:tank1.outvalve.pv"
--

Galaxy:tank.level.pv", "Galaxy:tank1.level.pv"
--

Galaxy:tank.mixer.pv", "Galaxy:tank1.mixer.pv"
--

Index

C

changing references in an instance 36

D

deleting SmartSymbols 42

E

Edit Menu 9, 11, 25

Editing SmartSymbols 36

error handling 44

F

File Menu 24

G

Generating SmartSymbols 26

H

Help Menu 25

I

Instantiation

SmartSymbols 31, 35

InTouch

New Features 5

L

Library 38

limitations 30

M

Manager

SmartSymbol 23

menu items

SmartSymbol Manager 23

Menus

Edit 9, 11, 25

File 24

Help 25

Tools 25

N

New Features 5

Alarm Viewer Control Display 14

Cross-Reference Query Display 14

Display Version Information 14

Enhanced NAD 12

SmartSymbol Manager 5

Tablet PC Support 6

Time and Date Functions 13

O

Overview

SmartSymbol 21

P

Properties

SmartSymbol 43

R

recovery 42

redirect remote references 45

renaming SmartSymbols 42

runtime

SmartSymbols 45

S

scripting

redirect remote references 45

SmartSymbol

Overview 21

Properties 43

SmartSymbol instances

change references 36

SmartSymbol Library 38

SmartSymbol library

exporting 38

importing 39

SmartSymbol Manager 23

error handling 44

SmartSymbolManager

menu items 23

SmartSymbols

deleting 42

Editing 36

generating 26

instantiating 31, 35

limitations 30

recovering 42

renaming 42

runtime 45

symbol recovery 42

T

Tools menu 25

