

Guide to Creating InTouch Windows Programmatically

Invensys Systems, Inc.

Revision A

Last Revision: 4/28/09



Copyright

© 2009 Invensys Systems, Inc. All Rights Reserved.

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

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

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

<http://www.wonderware.com>

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

Trademarks

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

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

Contents

Welcome.....	7
Documentation Conventions.....	7
Technical Support	8
 Chapter 1 Preparing an XML File	 9
 Chapter 2 Using an XML File to Import a Window.....	 11
Preparing a Command File.....	12
Creating a Minimum Command File	13
Sending Print Information to a File	13
Sending a Cross Reference to a File	13
Creating a Log File.....	14
Command Syntax	14
Creating an Application	16
Adding Tags to Newly Generated Application.....	17
Deleting a Window	17
Renaming a Window	18
Importing a Window.....	18
Handling Errors	19
Missing SmartSymbols	20
Missing Archestra Symbols.....	20
Expressions, Tag Names, and Scripts	20
Running WindowMaker from the Command Prompt	21

Archestra IDE Extension	22
Running WindowMaker from Managed InTouch Applications	22
Running DBDump from the Command Prompt	24
Running DBDump for Managed InTouch Applications	25
Running DBLoad from the Command Prompt	26
Running DBLoad for Managed InTouch Applications	27

Chapter 3 XML Formats.....29

General XML File Format	29
Using a Schema	29
XML File Header	30
Extraneous XML Elements	30
User-Supplied Text	30
Preserving Text White Space	30
Common Element Definitions	31
Color Elements	31
RGB Elements	31
Color Name Elements	31
Color Reference Elements	32
Color Value Elements	32
TextInfo Elements	33
Point Elements	34
Pen Elements	34
Dimension Elements	35
Expressions	35
Virtual Key Codes and Virtual Key Flags	36
Window Element	38
Window Definition	39
Activating Schema Validation	41
Window Scripts	41
OnShow Window Script Element	41
WhileShowing Window Script Element	42
OnHide Window Script Element	42
Window Objects	43
Default Element Values	44
Rectangle Object	45
Rounded Rectangle Object	47
Ellipse Object	49
Line Object	50

Horizontal Line Object	51
Vertical Line Object.....	52
Polyline Object.....	53
Polygon Object	54
Text Object.....	55
Bitmap Object.....	57
Button Object.....	60
Smart Symbols.....	61
Archestra Symbols	65
Unsupported Window Objects	69
Window Animation Links	69
Script/Expression/Tag Name	
Requirements Matrix.....	70
Discrete User Input.....	72
Analog User Input	73
String User Input	75
Discrete Line Color.....	77
Analog Line Color.....	78
Discrete Alarm Line Color	81
Analog Alarm Line Color	82
Discrete Fill Color	84
Analog Fill Color.....	85
Discrete Alarm Fill Color.....	87
Analog Alarm Fill Color	88
Discrete Text Color.....	90
Analog Text Color.....	91
Discrete Alarm Text Color	93
Analog Alarm Text Color	94
Vertical Slider.....	96
Horizontal Slider	97
Object Height.....	98
Object Width.....	99
Vertical Location	100
Horizontal Location.....	101
Vertical Percent Fill	102
Horizontal Percent Fill.....	103
Discrete Pushbutton.....	104
Show Window Pushbutton.....	105
Hide Window Pushbutton	105
Visibility.....	106
Blink.....	107
Orientation.....	108
Disable	109

Static Tooltip	109
Dynamic Tooltip	110
Discrete Value Display	110
Analog Value Display	111
String Value Display	112
Pushbutton Action Scripts	112
On Left Key Down/On Key Down	113
While Left Key Down/While Key Down	113
On Left Key Up/On Key Up	113
On Left Key Double Click	114
On Right Key Down/On Right Down	114
While Right Key Down/While Right Down	115
On Right Key Up/On Right Up	115
On Right Key Double Click	116
On Middle Key Down/On Middle Down	116
While Middle Key Down/While Middle Down	117
On Middle Key Up/On Middle Up	117
On Middle Key Double Click	118
On Mouse Over	118
Left Key Equivalent	119
Right Key Equivalent - Not Supported	120
Middle Key Equivalent - Not supported	120
Unsupported InTouch Features	121
Index	123

Welcome

You can import window definitions into InTouch applications using XML formatted files. How to create an XML file is outside the scope of this guide.

This guide assumes you know how to create InTouch applications using WindowMaker.

This guide assumes you can create, format, and test XML files and know the general requirements for using XML files.

Note You need to install Wonderware Application Server 3.1 before installing InTouch 10.1.050. If you have InTouch 10.1 installed, uninstall it before installing InTouch 10.1.050.

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

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

Documentation Conventions

This documentation uses the following conventions:

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

Technical Support

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

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

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

Chapter 1

Preparing an XML File

The following procedure describes how to create an XML file and import a window definition into an InTouch application.

- 1 Create an XML file. This guide does not define the tools to use, or basic XML syntax.
- 2 Check the XML file for adherence to XML formatting standards. You can verify the syntax of your XML file with the help of XML validation tools. To view your XML file, open it in Internet Explorer.
- 3 Prepare the tags in the application. You can use the InTouch DBDump and DBLoad utilities to extract, modify, and reload tags. For more information on DBDump, DBLoad, and the CSV file, see the *InTouch Data Management Guide*.
- 4 Prepare a WindowMaker command file to perform the functions you need to accomplish.
- 5 Exit from your InTouch application, WindowMaker, and WindowViewer.
- 6 Run the WindowMaker command file from a command line prompt, or run the command file from a program or from an IDE extension.
- 7 Check for errors. If you defined an error log file, check it and check the ArchestrA Logger. If you ran the command file from a program, check the return code.

The following sample files help you create and import window definitions using the XML import functionality:

- Sample XML file that loads a window definition into an InTouch application.
- Sample command file.
- Two schema files. The import parser does not use these schema files. The schema files are more restrictive than the XML import parser.

Note Schema files are available in the InTouch installation folder after the installation is complete.

Chapter 2

Using an XML File to Import a Window

You create a command file, and then run WindowMaker with the command file to import a window from an XML file. In addition to importing a window, the command file can be used to:

- Create a new application.
- Delete a window.
- Rename a window.
- Send print information to a file or printer.
- Send a cross reference to a file.

When the XML import is in progress, the following operations are not supported on a referenced symbol:

- Deleting a symbol
- Renaming a symbol
- Swapping symbol names

However, you can perform these operations after the XML import is complete.

The following system level operations are not supported:

- Launching console sessions
- Switching focus to other applications

You can run the InTouch DBDump and DBLoad utilities from the command line. However, for managed InTouch applications, run the DBDump and DBLoad utilities from the ArchestrA IDE extension. DBDump creates an export file containing tag information from an InTouch application. DBLoad imports tag information into an InTouch application. For more information about the DBDump and DBLoad utilities, see the InTouch *Data Management Guide*.

Preparing a Command File

The WindowMaker command file is a text file. Commands are read from the file and then actions are taken against the InTouch application data set.

The command file uses the single-byte ANSI character set. You cannot use MultiByte Character Set (MBCS) or Unicode characters.

Specifics of the text file syntax are:

- The 8-bit ANSI character set is used, but only characters under 127 are allowed. Most control characters are stripped out.
- Lines that begin with the # character are treated as comments.
- Blank lines are ignored.
- Lines are terminated with a carriage return line feed (CRLF) sequence.
- Spaces can occur at the beginning of a line, at the end of a line, between the commands, the equal character, and its argument.
- For any arguments requiring quotation marks, spaces are not allowed between the quotation marks and other text.
- All leading and trailing white space is removed before each line is processed.
- Each command starts with a period.
- Each command file must start with the WindowMaker command-file command (**.WINDOWMAKERCOMMANDFILE**).

A command file can contain multiple commands. If a command fails, the command processor continues with the next command in the file.

Creating a Minimum Command File

An example of the minimum command file is:

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
```

Sending Print Information to a File

You can send a listing of the entire InTouch application to a printer or to a text file. If the InTouch window contains ArchestrA graphics then the output will be sent to an HTML file.

The following example command file prints application information to the printer named Printer01:

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.PRINTAPPLICATIONINFORMATION
.OUTPUTTARGET=Printer
.OUTPUTTARGETNAME=Printer01
.GO
```

The following is an example command file that prints to a file. If the file exists, it is overwritten.

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.PRINTAPPLICATIONINFORMATION
.OUTPUTTARGET=TextFile
.OUTPUTTARGETNAME=C:\MyApps\AppInfo.txt
.GO
```

Sending a Cross Reference to a File

You can create a cross reference and send it to a file. If the file exists, it is overwritten. No user interaction is required.

An example command file:

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.CROSSREFERENCE
.SEARCHFOR=TagName
.REFERENCETYPE=ByWindow
.OUTPUTFILE=C:\MyApps\AppCrossRef.Csv
.GO
```

Creating a Log File

You can log informational messages, warnings, and errors to a text file. If you do not specify this command, the processing status is sent to the ArchestrA Logger. If the file already exists, it is overwritten.

An example command file that logs errors to a file.

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.COMMANDLOGFILE=C:\MyApps\LogFile.Txt
```

Ensure that the folder MyApps exists. If the folder MyApps does not exist, the log file is not created.

Command Syntax

The following table lists the commands you can use in a WindowMaker command file.

Command	Description
.WINDOWMAKERCOMMANDFILE	Identifies the file as being a WindowMaker command file. If this command is not found, the file is not processed and WindowMaker exits with an error code.
.VERSION=1	This command indicates to WindowMaker that the file is not too new to be read.
.COMMANDLOGFILE=<Full File Path>	Information, warnings, and errors are logged to this file. If the file already exists, it is overwritten. If this command is omitted, check the ArchestrA Logger for information about the processing status.
.GO	Runs the command. Required after a command with one or more parameters.
.WINDOWCREATE	Creates a window from an XML file.
.XMLFILEPATH=<Full File Path>	The full file path to an XML file containing a window specification. Required when the WINDOWCREATE command is used.

Command	Description
.WINDOWDELETE	Deletes the specified InTouch application window by name. An error is not generated if the window name is not found.
.WINDOWNAME=<Window Name>	Name of the window to delete. Required when the WINDOWDELETE command is used.
.WINDOWRENAME	Renames a window. If the 'old' name cannot be found, the command issues a warning. If the new name exists, an error message is generated. If the window cannot be renamed, an error occurs. If the old name matches the new name, then nothing is done.
.OLDWINDOWNAME=<Existing Window Name>	Current window name. Required when the WINDOWRENAME command is used.
.NEWWINDOWNAME=<New Window Name>	New window name. A window with this name must not exist. Required when the WINDOWRENAME command is used.
.PRINTAPPLICATIONINFORMATION	Prints or dumps the InTouch application data set to a text file. This is the same as clicking the Print option on the File menu commands from WindowMaker.
.OUTPUTTARGET=Printer TextFile	Sends output to a printer or text file.
.OUTPUTTARGETNAME=<Printer Name> <Full Text File Path>	Name of the printer or output file. If the file exists, it is overwritten.
.CROSSREFERENCE	Generates cross reference information in Comma Separated Variable (CSV) format.
.SEARCHFOR=TagName QuickFunctions	Searches for tags or QuickFunctions by name.
.REFERENCETYPE=ByTagName ByWindow	Cross-references by tag or window name.
.OUTPUTFILE=<Full File Path>	Full path to the output file. If the file exists, it is overwritten.

Creating an Application

You can use a WindowMaker command file to create a new default InTouch application. You create a blank application using the minimum command file. However, you cannot create a blank managed InTouch application using the minimum command file. For more information, see [Creating a Minimum Command File](#) on page 13.

The command file must be placed in the folder where you want the application created. If the folder contains an existing InTouch.ini file, the application is not generated.

The path to the new application folder cannot contain an embedded “-I” or “-L” string sequence. For example, the folder C:\MyApps\App-Large cannot be created.

The created InTouch.ini file has contents similar to the following example. Window positions vary with the screen resolution in which the application is shown. The application has the title and description “Generated InTouch Application.” The default language is English.

Example InTouch.ini file contents:

```
[InTouch]
AppMode=2
AppName0=Generated InTouch Application
AppName1=
AppName2=
AppName3=
AppDesc0=Generated InTouch Application
AppDesc1=
AppDesc2=
AppDesc3=
LanguageBase=English (United States)
LanguageBaseID=1033
InTouchView=0
SAOConverted=1
WinFullScreen=1
WinLeft=-4
WinTop=-4
WinWidth=1288
WinHeight=1004
SnapOn=1
```

Adding Tags to Newly Generated Application

A new InTouch application contains only system tags. If you need to add tags to an application before importing a window, then:

- 1 Create a new application.
- 2 Run DBLoad to import tags to the new application.
- 3 Import the window.

For example:

```
WM.Exe C:\MyApps\App001 COMMANDFILE="C:\BlankFile.Txt"
DBLoad C:\MyApps\App001,C:\TagDumps\App001.Csv,0
WM.Exe C:\MyApps\App01 COMMANDFILE="C:\Commands.Txt"
```

Deleting a Window

If you add a window to an existing application that has a window with the same name, you must delete the existing window. You can delete the existing window using the WindowMaker command file.

Important An error message does not appear if the window selected to delete does not exist.

A message appears in the ArchestrA Logger if an existing window cannot be deleted. No user interaction is required.

You enter a sequence of commands in your command file to delete a window from an InTouch application. A command file can include several delete window command sequences.

The following example shows the command sequence to delete a window from an InTouch application:

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.WINDOWDELETE
.WINDOWNAME=Window002
.GO
```

Renaming a Window

You can change the name of a window in an InTouch application.

You enter a sequence of commands in your command file to rename a window from an InTouch application. A command file can include several rename window command sequences.

An error message does not appear if the window selected to be renamed does not exist.

If the new window name already exists, nothing is done, and a warning message is logged.

If the old name exists and the new name does not exist, but the rename fails for some other reason, an error message is logged.

No user interaction is required. Warnings and error messages appear in the ArchedrA Logger.

If you rename a window with the name of an existing window, a warning message is logged in the log file or the ArchedrA Logger. In the following example, renaming the Window005 to an existing window Window006, logs a warning message in the log file or the ArchedrA Logger.

```
.WINDOWRENAME  
.OLDWINDOWNAME=Window005  
.NEWWINDOWNAME=Window006  
.GO
```

Importing a Window

You can import a window into an InTouch application. The full file path to the XML file containing a single InTouch window's specification is required. The new window name is contained within the XML file.

The window is not imported if:

- A window with the same name exists in the application.
- An unresolved error occurs in one of the window scripts.
- An unresolved error occurs trying to add any part of an object to the window.

User interaction may be required if:

- Elements specified in scripts or expressions contain errors or are missing.
- Scripts contain syntax errors.

It is possible that nothing is shown to allow a user to correct the problem. Examine the log file and the ArchestrA Logger for details.

You enter a sequence of commands in your command file to import a window.

For example:

```
.WINDOWCREATE  
.XMLFILEPATH=C:\WMCommandTest\WMCreateFile.Xml  
.GO
```

A command file may contain more than one import window command and several create window command sequences.

Handling Errors

Your XML file must follow general XML formatting rules. If you cannot open your XML file with Internet Explorer, your XML file contains XML formatting errors. Fix all errors before using the file with WindowMaker.

If you use a file containing general format errors, WindowMaker logs the error message “XML file could not be loaded” in the ArchestrA Logger.

WindowMaker stops when any element is missing from a script, on any script error, or any other element specification error occurs. For example:

- Missing tags
- Missing external WindowMaker script extension DLL
- Missing ActiveX controls
- Missing Wizard DLL

In some cases, such as for particular animation links or custom property overrides, no message box appears if the tag is missing. A note is written to the Archestra Logger, the animation link and object are not created, and the window is not created. In other cases, the script and expression parsing stops. You are given an opportunity to create the tag, and if it is successfully created, processing continues.

Missing SmartSymbols

If a SmartSymbol template does not exist in the target application, the window is not imported. An error message appears in the logger and the output text file.

If a SmartSymbol instance fails to be created for any reason, the window is not created. A SmartSymbol import can fail even though the SmartSymbol template exists. There may be additional information about the failure in the ArcestrA Logger.

Missing ArcestrA Symbols

If an ArcestrA Symbol reference does not exist in the Galaxy repository, the window is not created. An error message appears in the ArcestrA Logger and the output text file.

The XML import process can fail even though the ArcestrA Symbol reference exists. Additional information about the failure is logged in the ArcestrA Logger.

Expressions, Tag Names, and Scripts

When expressions, tag names, and scripts are required, they cannot be empty text, or text that is only blanks. If these text items are not formulated correctly; the expression, tag name, or script do not parse correctly and result in an error.

In the case of tag names, the tag type must match the type expected for an object. In most cases, it is possible to use dotfields to access tag properties. For example, the second bit of an integer tag iTag005 can be accessed as:

```
iTag005.02
```

In this case, the tag and dotfield evaluates to a discrete value and can be used where a discrete tag name is required, whereas the iTag005 alone would result in an error.

Running WindowMaker from the Command Prompt

You can run WindowMaker from the command prompt.

The command line is:

```
WM.EXE AppPath,Commandfile="Command.txt"
```

Parameters

AppPath

Defines the full path to the InTouch application. This parameter is optional if you are running WindowMaker in the application's folder.

CommandFile

Defines the full path to the command file. Quotation marks must enclose the path name. Extra spaces are not allowed around the equal sign.

Suppose the InTouch application is in the C:\MyApps folder, and the command file is C:\WMCommandFile.txt. The following example shows the WindowMaker command you enter from the command prompt:

```
WM.EXE C:\MyApps,COMMANDFILE="C:\WMCommandFile.Txt"
```

Note You cannot create managed InTouch applications from the command prompt.

ArchestrA IDE Extension

Use the ArchestrA IDE extension to run the InTouch XML import functionality for managed applications. You can use the IDE extension to select a command file for the XML import process. After you select a command file, WindowMaker starts up and the associated XML file is parsed.

Important WindowMaker stops responding if you start a new console session or maximize an existing console session when the XML import is in progress.

The ArchestrA IDE has a context menu item used to select a command file for the XML import. The menu item **Process InTouch Command File** is shown in the context menu when you right click an InTouchViewApp derived template. However, the menu item will not be shown, when you select:

- Multiple InTouchViewApp templates
- InTouchViewApp base template
- InTouchViewApp instance

If you migrate a managed InTouch application from older version of InTouch, then you must open the InTouch application in WindowMaker at least once before processing the command file.

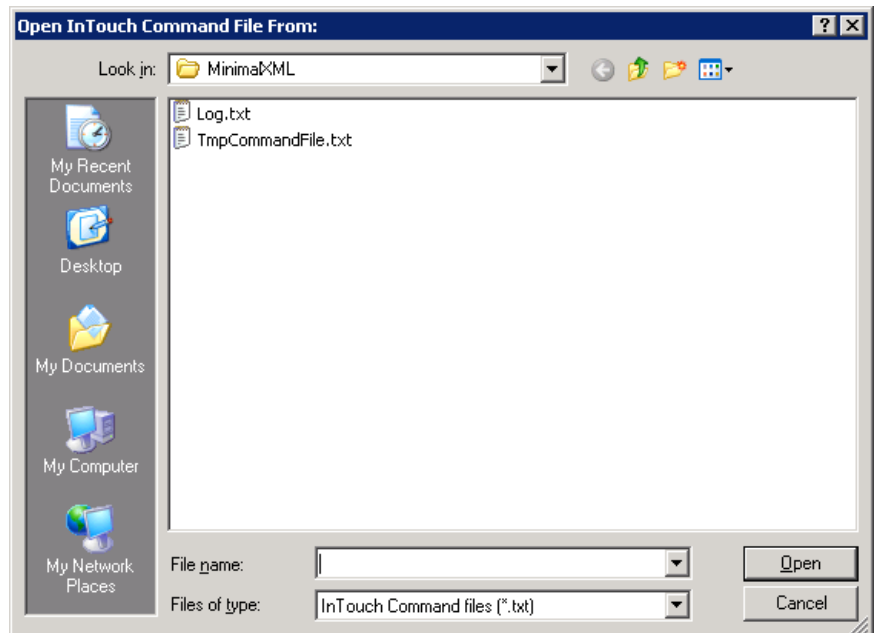
Running WindowMaker from Managed InTouch Applications

You can run WindowMaker from a managed InTouch application.

To run WindowMaker from a managed InTouch application

- 1 Open the ArchestrA IDE.
- 2 In the **Template Toolbox**, create a derived template from the InTouchViewApp template.
- 3 Associate an InTouch application with the derived template by creating a new InTouch application or by importing an existing standalone InTouch application.

- 4 Right-click the derived template and then click **Process InTouch Command File**. The **Open InTouch Command File From:** dialog box appears.



Note The InTouchViewApp derived template must be configured in WindowMaker at least once, before executing the **Process InTouch Command File** command.

- 5 Browse to the location of the command file, and click **Open**. The WindowMaker starts up.

Running DBDump from the Command Prompt

You use DBDump to extract tag information from an InTouch application. You can run DBDump from the command prompt. You must stop WindowMaker before running DBDump.

Commas are required between DBDump command parameters. Parameters are position dependent. If you want to omit a parameter between included parameters, you must include a comma for the missing parameter. The following example shows the syntax of the DBDump command when you run it from the command prompt:

```
DBDump AppPath,CsvPath,GroupTypes,OverwriteCsvFile,  
MessageBoxes
```

Parameters

AppPath

Defines the path to the InTouch application.

CsvPath

Defines the path to the export file containing tag definitions from the Tagname Dictionary of the InTouch application.

GroupTypes

Specifies whether tags are grouped in the DBDump export file by InTouch tag types. 1 indicates the tag database names should be sorted by tag group type. 0 indicates the tag names are not to be sorted by type.

OverwriteCsvFile

Specifies if the export file should be overwritten. 1 indicates the export file should be overwritten. 0 indicates the export file should not be overwritten.

MessageBoxes

Specifies whether messages appear when the DBDump utility exports the contents of the application's Tagname Dictionary. 1 indicates the message boxes are to be displayed. 0 indicates message boxes are not to be displayed.

Example

Suppose the InTouch application is located in the C:\MyInTouchApps\App001 folder and you want to write the contents of the tag database to the C:\TagDumps\App001.csv file. You do not want any message boxes to appear, and you want to overwrite the target export file if it exists. The command is:

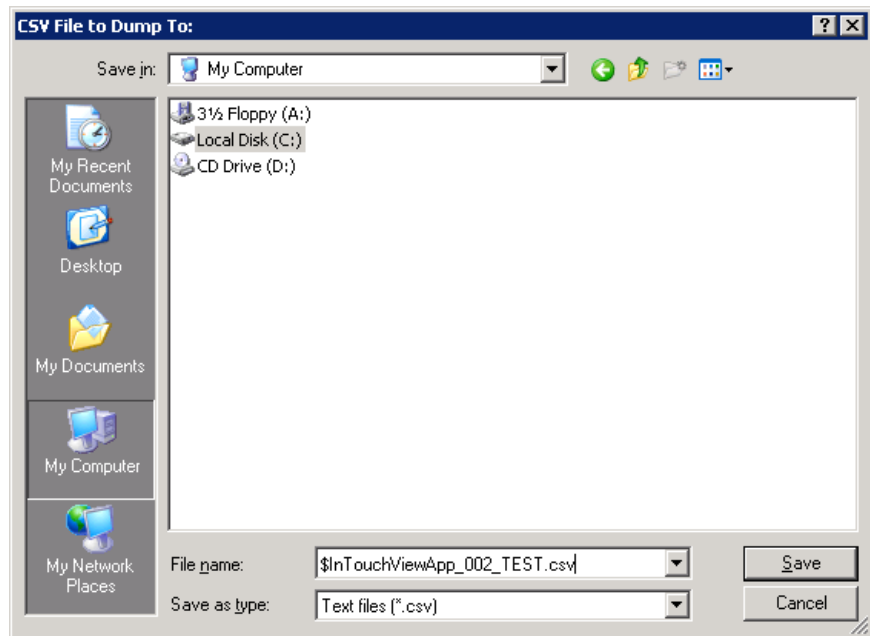
```
DBDump C:\MyInTouchApps\App001,C:\TagDumps\App001.Csv,1,1,0
```

Running DBDump for Managed InTouch Applications

You can run DBDump for managed InTouch applications.

To run DBDump for a managed InTouch application

- 1 Open the Archestra IDE.
- 2 In the **Template Toolbox**, right-click an InTouchViewApp derived template, point to **Export**, and then click **DB Dump**. The **CSV File to Dump To:** dialog box appears.



- 3 Type the name of the CSV file and click **Save**. The application tag data is successfully dumped into the CSV file.

Running DBLoad from the Command Prompt

You use DBLoad to import tag information into an InTouch application. You can run DBLoad from the command prompt. You must stop WindowMaker before running DBLoad.

DBLoad command parameters are position dependent, if you want to omit a parameter between included parameters, you must include a comma for the missing parameter. The following example shows the syntax of the DBLoad command when you run it from the command prompt:

```
DBLoad AppPath,CsvPath,MessageBoxes
```

Parameters

AppPath

Specifies the path to the InTouch application.

CsvPath

Specifies the path to the file containing tag definitions to be imported into an application's Tagname Dictionary.

MessageBoxes

Specifies whether messages appear when the DBLoad utility imports the contents of the import file into the application's Tagname Dictionary. 1 indicates the message boxes are to be displayed. 0 indicates message boxes are not to be displayed.

Example

Suppose the InTouch application is located in the C:\MyInTouchApps\App001 folder and you want to read the tag database information from the C:\TagDumps\App001.Csv file. You do not want the message boxes to appear. The following example shows the DBLoad command you enter at the command prompt:

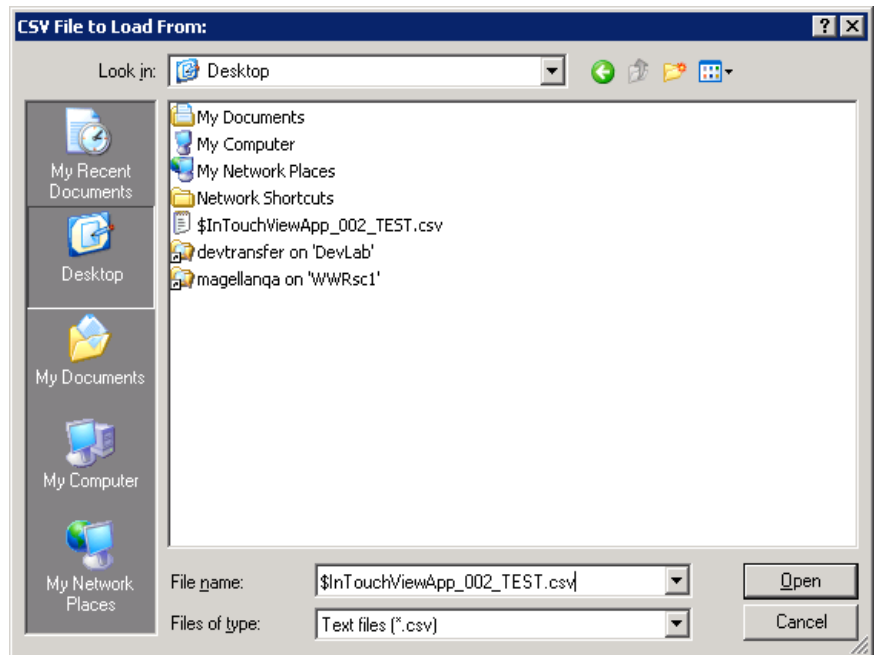
```
DBLoad C:\MyInTouchApps\App001,C:\TagDumps\App001.Csv,0
```

Running DBLoad for Managed InTouch Applications

You can run DBLoad for managed InTouch applications.

To run DBLoad for a managed InTouch application

- 1 Open the Archestra IDE.
- 2 In the **Template Toolbox**, right-click an InTouchViewApp derived template, point to **Import**, and then click **DB Load**. The **CSV File to Load From:** dialog box appears.



- 3 Browse to locate the CSV file and click **Open**. The tag data is successfully loaded into the InTouch application.

Chapter 3

XML Formats

You can import a window and most window elements defined in an XML file. Tags cannot be imported from an XML file. You must use the DBLoad utility instead.

General XML File Format

Your XML file must comply with all general XML formatting rules. You should be able to open your XML file with Internet Explorer. If not, your XML file contains formatting errors.

Using a Schema

You can validate the XML file against the schema file. Validating your XML file with the schema file detects most XML formatting errors.

The schema is more restrictive than the XML input parser.

- The schema requires the order of elements to match the order listed in the tables in this guide.
- Element names and values are case sensitive.
- In some cases, the schema requires an element to be explicitly defined.

You invoke schema validation in the window element. For information about how to specify a schema, see Window Definition on page 39.

XML File Header

The XML declaration must be included at the top of the file. The minimal declaration is:

```
<?xml version="1.0"?>
```

Extraneous XML Elements

The InTouch XML import functionality does not generate errors or warnings for undefined elements, attributes, or nodes.

You must type everything correctly. If you mean to have an `<ONRIGHTDOWN>` action script, but accidentally type `<ONRITEDOWN>`, no warning is generated, and your action script is not added to the window object.

Warnings and errors are created for XML definitions that are missing required elements.

User-Supplied Text

If your script uses XML field delimiter characters, the script text must be encapsulated within a CDATA element. Text in a CDATA element is not parsed by the XML file import functionality.

You can enter text without enclosing it within a CDATA element provided the text does not contain any XML field delimiter characters.

Preserving Text White Space

When text not in a CDATA element is processed, the leading and trailing spaces are removed. As a result, an element such as `<Title> MyWindowName</Title>` results in a window name of 'MyWindowName' without leading spaces.

To preserve leading and trailing white space in any text, enclose the text within a CDATA element. For example, an element `<Title><![CDATA[MyWindowName]]></Title>` results in a window name of "MyWindowName".

Common Element Definitions

Some elements or definitions are shared by many elements.

Color Elements

Color elements can be specified by RGB value, name, reference value, or integer value. The color elements are R, G, B, Name, Ref, and Value. These elements are used in other elements. For example, FillColor, TextColor, and BGCOLOR.

RGB Elements

You use RGB values to specify a color. The values assigned to RGB elements range from 0 to 255. When an element is missing, a default is used. The default depends on the window object.

Examples:

```
<FillColor>
    <R>192</R>
    <G>192</G>
    <B>192</B>
</FillColor>

<TextColor> <R>0</R><G>0</G><B>0</B> </TextColor>
```

Color Name Elements

You can specify a color using a name supported by Internet Explorer version 3.0 or later. The color name is case sensitive and must match a known HTML color.

The following web sites have a list of the color names and values:

http://www.w3schools.com/html/html_colornames.asp

<http://www.learningwebdesign.com/colornames.html>

http://www.oreilly.com/catalog/wdnut/excerpt/color_names.html

The following web sites have a list of the color names:

<http://www.geocities.com/SiliconValley/Pines/6986/colortbl.html>

Examples:

```
<FillColor>
    <Name>White</Name>
</FillColor>

<TextColor> <Name>White</Name> </TextColor>
```

Color Reference Elements

You can specify a color using a hexadecimal color value. The color value must always contain six hexadecimal digits.

The following web sites list colors by name and their corresponding hexadecimal values:

http://www.w3schools.com/html/html_colornames.asp

<http://www.learningwebdesign.com/colornames.html>

This web site has a list of non-dithering color values:

<http://www.htmlgoodies.com/tutorials/colors/article.php/3479001>

This site has 4096 colors and their hex codes:

<http://yorktown.cbe.wvu.edu/sandvig/MIS314/Assignments/A03/ColorHexCodes.asp>

Examples:

```
<FillColor>
    <Ref>#FF00FF</Ref>
</FillColor>

<TextColor> <Ref>#FF00FF</Ref> </TextColor>
```

Color Value Elements

You can specify a color using a positive integer color value. The value must be in the range 0 to 16777215.

Examples:

```
<FillColor>
    <Value>16711935</Value>
</FillColor>

<TextColor> <Value>16711935</Value> </TextColor>
```

TextInfo Elements

You can specify how text appears on the screen with the TextInfo element.

The following elements are available for specifying text:

Elements	Description
Font	Name of font families, such as 'System' or 'Arial'.
FontStyle	Values can be {Regular, Italic, Bold, BoldItalic}.
FontSize	Need to indicate font size unit of measure. Usually, points. Values can be 0 or larger.
Underline	Underlined text: {true or false}.
Strikeout	Struck out text: {true or false}.
TextColor	Color element for text color.
TextJustify	Text justification: {Left, Center, or Right}

Example:

```
<TextInfo>
  <Font>Arial</Font>
  <FontStyle>Regular</FontStyle>
  <FontSize>12</FontSize>
  <Underline>>false</Underline>
  <Strikeout>>false</Strikeout>
  <TextColor>
    <R>0</R>
    <G>0</G>
    <B>0</B>
  </TextColor>
  <TextJustify>Left</TextJustify>
</TextInfo>
```

Point Elements

You use a point element to define the position of other elements. Point elements contain two elements, X and Y. X and Y elements must contain values in the range -32000 and 32000.

Elements	Description
X	Left coordinate in pixels. Required.
Y	Top coordinate in pixels. Required

Example:

```
<Point>
  <X>10</X>
  <Y>25</Y>
</Point>
```

Pen Elements

You use pen elements to specify the line characteristics of an object's border.

The following elements are available for specifying the pen element:

Field	Description
PenColor	Uses color elements: RGB element, name element, reference element, or value element.
PenWidth	Pen width in pixels. Values can be 1, 2, 4, 6, 9, or 11.
PenStyle	Values can be None, Solid, Dash, Dot, DashDot, DashDotDot.

Example:

```
<Pen>
  <PenColor>
    <R>0</R>
    <G>0</G>
    <B>0</B>
  </PenColor>
  <PenWidth>4</PenWidth>
  <PenStyle>Solid</PenStyle>
</Pen>
```

Dimension Elements

The dimension element contains elements to specify the coordinates of the upper left corner of an object on the screen. The dimension element also includes elements to specify the width and height of a rectangular object.

In WindowMaker, the coordinate limits are -32000 to +32000 in both the vertical and horizontal directions. If you specify a combination of X and Y location with width or height values that place the calculated coordinates outside of a (-32000, -32000, 32000, 32000) boundary, a warning appears and the values are clamped to the maximum values. Height and width values should be positive.

You use the following elements within a dimension element to specify an area on the screen.

Elements	Description
Left	Left edge coordinate. Required.
Top	Top edge coordinate. Required.
Width	Width in pixels.
Height	Height in pixels.

Example:

```
<Dimension>
  <Left>4</Left>
  <Top>4</Top>
  <Width>632</Width>
  <Height>278</Height>
</Dimension>
```

Expressions

You enclose expression text in CDATA sections. This prevents XML delimiters, which are valid in expressions, from causing file parsing to fail.

Example:

```
<EXPRESSION>
<![CDATA[
  First line of expression text
  Second line of expression text
  N-th line of expression text
]]>
</EXPRESSION>
```

Virtual Key Codes and Virtual Key Flags

Some InTouch animation links support keyboard input.

The XML parser translates the name of a virtual key into its virtual key code. Also, the flags for the modifier keys are specified using text instead of numeric bit combinations. Key names are not case sensitive.

InTouch applications can use the virtual key names listed in the following table.

Key Represented	Virtual Key Names
ADD	Add
Alpha Keys	A through Z
BACKSPACE	Backspace
CANCEL	CtrlBreak
CLEAR	Clear
Copy	Copy
Decimal	Decimal
DELETE	Delete
Divide	Divide
DOWN ARROW	Down
Empty string means no assignment.	<Blank>
END	End
ENTER	Return
ESC	Escape
Execute	Execute
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9

Key Represented	Virtual Key Names
F10	F10
F11	F11
F12	F12
F13	F13
F14	F14
F15	F15
F16	F16
HELP	Help
HOME	Home
INSERT	Insert
LEFT ARROW	Left
MULTIPLY	Multiply
Numeric keys	1 through 9
Numeric Keypad 0	NUMPAD0
Numeric Keypad 1	NUMPAD1
Numeric Keypad 2	NUMPAD2
Numeric Keypad 3	NUMPAD3
Numeric Keypad 4	NUMPAD4
Numeric Keypad 5	NUMPAD5
Numeric Keypad 6	NUMPAD6
Numeric Keypad 7	NUMPAD7
Numeric Keypad 8	NUMPAD8
Numeric Keypad 9	NUMPAD9
NUM LOCK	NumLock
PAGE UP	PageUp
PAGE DOWN	PageDown
PRINT SCREEN	Print
RIGHT ARROW	Right
SELECT	Select
Separator	Separator

Key Represented	Virtual Key Names
SPACEBAR	Space
SUBTRACT	Subtract
TAB	Tab
UP ARROW	Up

Two modifier keys are supported. Modifier keys can be specified separately or in combination for the CKEYFLAGS element. The modified name is the element value. A modifier key is applied as long as the modifier name is in the attribute value string.

Name	Key Represented
CTRL	CONTROL key must be pressed with regular key.
SHIFT	SHIFT key must be pressed with regular key.
CTRL + SHIFT	Both the CONTROL and SHIFT keys must be pressed.

To generate a CTRL+A sequence, the XML is:

```
<VirtualKeyType>
  <KeyCode>A</KeyCode>
  <KeyFlags>CTRL</KeyFlags>
</VirtualKeyType>
```

Window Element

You can include only a single window element in your XML file. The window element must be the first element specified in the XML file.

The window name must not match an existing window name in the InTouch application.

Window Definition

You can use the following elements to specify a window.

Elements	Description
Title	Window name as non-empty string. Maximum of 32 characters and trailing spaces are not counted. Required.
Comment	Comment text. Maximum of 59 characters.
Dimension	Required when using the optional schema. When the schema is not used, a default dimension is applied if a dimension element is not included. The default values are 4 4 632 278.
WindowStyle	Window Type = {Replace Overlay Popup}.
BackgroundColor	Window background color specified by a color element: RGB element, name element, reference element, or value element.
FrameStyle	{Single Double None}.
TitleBar	Title bar enabled = {true false}.
SizeControls	Size controls enabled = {true false}.
ScriptOnShow	Script element.
ScriptWhileShowing	Script element.
ScriptOnHide	Script element.
ObjectList	Objects contained by the window, including SmartSymbols.

This example creates an empty window:

```
<?xml version="1.0" encoding="UTF-8"?>
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/
  Window
  wwInTouchWindow.xsd" Version="1">
  <Title>ApplicationSecondWindow</Title>
  <Comment>Main application window</Comment>
  <Dimension>
    <Left>4</Left>
    <Top>4</Top>
    <Width>632</Width>
    <Height>278</Height>
  </Dimension>
  <BackgroundColor>
    <R>192</R>
    <G>192</G>
    <B>192</B>
  </BackgroundColor>
  <WindowStyle>Overlay</WindowStyle>
  <FrameStyle>Single</FrameStyle>
  <TitleBar>true</TitleBar>
  <SizeControls>true</SizeControls>
</iw:InTouchWindow>
```

Minimal example:

```
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/
  Window
  wwInTouchWindow.xsd" Version="1">
  <Title>ApplicationSecondWindow</Title>
  <Dimension><Left>4</Left><Top>4</Top>
    <Width>632</Width>
    <Height>278</Height></Dimension>
</iw:InTouchWindow>
```

Activating Schema Validation

The schema validation and processing are activated by placing specific data within the `InTouchWindow` element. If you activate the schema, you must place the `InTouchCommon.Xsd` and `InTouchWindow.Xsd` files in the same folder as the XML file.

Example of using a schema:

```
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/
  Window wwInTouchWindow.xsd" Version="1">
</iw:InTouchWindow>
```

Example of not using a schema:

```
<InTouchWindow Version="1">
</InTouchWindow>
```

Window Scripts

You can create three types of window scripts in your XML import file: `OnShow`, `WhileShowing`, and `OnHide`. You place the script elements within the window element.

Script text can be enclosed in a CDATA section to prevent XML delimiter characters in the script from interfering with XML file parsing.

OnShow Window Script Element

The `OnShow` script element contains the script text.

Example:

```
<ScriptOnShow><![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
]]></ScriptOnShow>
```

Minimal example:

```
<ScriptOnShow>Single line of script text</ScriptOnShow>
```

WhileShowing Window Script Element

The WhileShowing script has two elements. The script text can be placed in a CDATA section.

Elements	Description
Text	Text of the window script. Required.
FREQUENCY	Script execution frequency in milliseconds. Required when using the optional schema. Default is 1000.

Example:

```
<ScriptWhileShowing>
  <Text><![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]></Text>
  <Frequency>1000</Frequency>
</ScriptWhileShowing>
```

Minimal example:

```
<ScriptWhileShowing>
  <Text>Single line of script text</Text>
  <Frequency>1000</Frequency>
</ScriptWhileShowing>
```

OnHide Window Script Element

The OnHide window script has one element. The script text can be placed in a CDATA section.

Example:

```
<ScriptOnHide>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</ScriptOnHide>
```

Minimal example:

```
<ScriptOnHide>Line of script text</ScriptOnHide>
```

Window Objects

Window objects are placed in an object list defined by the `ObjectList` element.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<iw:InTouchWindow
  xmlns:iw="http://www.wonderware.com/InTouch/Window"
  xmlns:itc="http://www.wonderware.com/InTouch/Common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.wonderware.com/InTouch/Window.xsd" Version="1">
  <Title>Window0001</Title>
  <Comment>Sample simple window</Comment>
  <Dimension>
    <Left>10</Left>
    <Top>10</Top>
    <Width>400</Width>
    <Height>400</Height>
  </Dimension>
  <BackgroundColor>
    <R>255</R>
    <G>0</G>
    <B>255</B>
  </BackgroundColor>
  <WindowStyle>Replace</WindowStyle>
  <FrameStyle>Double</FrameStyle>
  <TitleBar>true</TitleBar>
  <SizeControls>true</SizeControls>
  <ObjectList>
    <Rectangle>
      <Title>Rectangle1</Title>
      <Pen>
        <PenColor>
          <Name>Black</Name>
        </PenColor>
        <PenWidth>4</PenWidth>
        <PenStyle>Solid</PenStyle>
      </Pen>
      <Dimension>
        <Left>100</Left> <Top>50</Top>
        <Width>270</Width> <Height>80</Height>
      </Dimension>
      <FillColor>
        <R>128</R> <G>128</G> <B>128</B>
      </FillColor>
    </Rectangle>
  </ObjectList>
</iw:InTouchWindow>
```

Each window object can contain animation links. A window object containing animation links includes an animation links element.

```
<Animationlinks> </Animationlinks>
```

Default Element Values

When window object elements are not specified, the element values in the following table are used as defaults.

Elements	Default Value
FILLCOLOR	<R>212</R><G>208</G> 200
PENWIDTH	1
PENCOLOR	Black
PENSTYLE	Solid
CORNERDIMENSION	<Width>20</Width> <Height>20</Height>
TEXTCOLOR	Black
TEXTJUSTIFY	Left
ROTATION	0
FONT	System
FONTSTYLE	regular
FONTWEIGHT	Ignored
FONTSIZE	10
UNDERLINE	False
STRIKEOUT	False
FLIP	None
TRANSPARENT	<R>0</R><G>255</G> 0

Pen Style Limitations

If you specify a pen style other than solid, the pen width is forced to 1. To specify pen widths larger than 1, the pen style option must be removed or set to SOLID.

Pen Dimensions

The dimension specification is for the centerline of an object. If an object's pen width is larger than 1, then an object may not fit within the specified dimension. Instead, the pen width straddles the boundary of the object. Some pixels are inside the object's boundary; others are outside of the boundary.

Rectangle Object

The following elements can be specified for a rectangle element.

Elements	Description
FillColor	Fill color of a rectangle. Specified with a color element. Default is rgb(212, 208, 200).
Pen	Pen element.
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. An object is not created if the Dimension element is assigned invalid values or is missing values.
Title	Object name. Optional.
Animationlinks	Optional animation links list.

Example:

```
<Rectangle>
  <Title>Rectangle1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>4</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Dimension>
    <Left>100</Left> <Top>50</Top>
    <Width>270</Width> <Height>80</Height>
  </Dimension>
  <FillColor>
    <R>128</R> <G>128</G> <B>128</B>
  </FillColor>
  <AnimationLinks> </AnimationLinks>
</Rectangle>
```

Minimal example:

```
<Rectangle>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</Rectangle>
```

Rounded Rectangle Object

The following elements can be specified for a rounded rectangle object.

Elements	Description
FillColor	Fill color. Specified by a color element. Default is rgb(212, 208, 200).
Pen	Pen element.
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. An object is not created if the Dimension element is assigned invalid values or is missing values.
CornerDimension	Elements width and height. Corner width cannot exceed the width of the rectangle. Corner height cannot exceed the height of the rectangle. Corner width and corner height must be at least 1. Default is 20, 20.
Title	Object name. Optional.
Animationlinks	Optional animation links list.

Example:

```
<RoundedRectangle>
  <Title>RoundedRectangle1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>4</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Dimension>
    <Left>100</Left>
    <Top>50</Top>
    <Width>270</Width>
    <Height>80</Height>
  </Dimension>
  <FillColor>
    <R>128</R> <G>128</G> <B>128</B>
  </FillColor>
  <AnimationLinks>
  </AnimationLinks>
  <CornerDimension>
    <Width>8</Width>
    <Height>8</Height>
  </CornerDimension>
</RoundedRectangle>
```

Minimal example:

```
<RoundedRectangle>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</RoundedRectangle>
```

Ellipse Object

The following elements can be specified for an ellipse object.

Elements	Description
FillColor	Fill color. Specified with a color element. Default is rgb(212, 208, 200).
Pen	Pen element.
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. An object is not created if the Dimension element is assigned invalid values or is missing values.
Title	Object name. Optional.
Animationlinks	Optional animation links list.

Example:

```
<Ellipse>
  <Title>Ellipse1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>4</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Dimension>
    <Left>100</Left>
    <Top>50</Top>
    <Width>270</Width>
    <Height>80</Height>
  </Dimension>
  <FillColor>
    <R>128</R> <G>128</G> <B>128</B>
  </FillColor>
  <AnimationLinks>
  </AnimationLinks>
</Ellipse>
```

Minimal example:

```
<Ellipse>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</Ellipse>
```

Line Object

The following elements can be specified for a line object. From and to points cannot be the same point.

Elements	Description
Pen	Pen element
Title	Object name. Optional.
Animationlinks	Optional animation links list.
Points	Point elements. Must contain two. Extra point elements are ignored.

Example:

```
<Line>
  <Title>Line1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>1</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>150</X><Y>160</Y></Point>
  </Points>
  <AnimationLinks>
  </AnimationLinks>
</Line>
```

Minimal example:

```
<Line>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>150</X><Y>160</Y></Point>
  </Points> </Line>
```

Horizontal Line Object

The following elements can be specified for a horizontal line object. This results in a WindowMaker HV Line object.

Elements	Description
Pen	Pen element.
Title	Object name. Optional.
Animationlinks	Optional animation links list.
Points	Must contain two points. The Y coordinate of the second point is ignored and set to the Y coordinate of the first point. Extra point elements are ignored. Required. An object is not created if the Points element is assigned invalid values or is missing values.

Example:

```
<HorizontalLine>
  <Title>Line1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>1</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>150</X><Y>150</Y></Point>
  </Points>
  <AnimationLinks>
  </AnimationLinks>
</HorizontalLine>
```

Minimal example:

```
<HorizontalLine>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>150</X><Y>150</Y></Point>
  </Points>
</HorizontalLine>
```

Vertical Line Object

The following elements can be specified for a vertical line object. This will result in a WindowMaker HV Line object.

Elements	Description
Pen	Pen element.
Title	Object name. Optional.
Animationlinks	Optional animation links list.
Points	Must contain two points. The X coordinate of the second point is ignored and set to the X coordinate of the first point. Extra point elements are ignored. Required. An object is not created if the Points element is assigned invalid values or is missing values.

Example:

```
<VerticalLine>
  <Title>Line1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>1</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>50</X><Y>250</Y></Point>
  </Points>
  <AnimationLinks>
  </AnimationLinks>
</VerticalLine>
```

Minimal example:

```
<VerticalLine>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>50</X><Y>250</Y></Point>
  </Points>
</VerticalLine>
```

Polyline Object

You must define at least two points in order to load the polyline object. Using the same coordinates for the two points is not recommended.

The following elements can be specified for a polyline object.

Elements	Description
Points	Point elements, at least two required. An object is not created if the Points element is assigned invalid values or is missing values.
Pen	Pen element.
Title	Object name. Optional.
Animationlinks	Optional animation links list. Fill Color, Text Color, Percent Fill, Value Display are not permitted.

Example:

```
<Polyline>
  <Title>polyline1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>1</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>60</X><Y>250</Y></Point>
    <Point><X>70</X><Y>350</Y></Point>
    <Point><X>80</X><Y>450</Y></Point>
  </Points>
  <AnimationLinks>
  </AnimationLinks>
</Polyline>
```

Minimal example:

```
<Polyline>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>80</X><Y>450</Y></Point>
  </Points>
</Polyline>
```

Polygon Object

You must define at least two points to load a polygon object. Using the same coordinates for the two points is not recommended.

The following elements can be specified for a polygon object.

Elements	Description
Points	Contains Point elements. At least two points are required. X and Y values are in pixels.
FillColor	Fill color of polygon object. Contains a color element. Default is rgb(212, 208, 200).
Pen	Pen element.
Title	Object name. Optional.
Animationlinks	Optional animation links list. Text Color, Value Display animation links are not permitted.

Example:

```
<Polygon>
  <Title>polygon1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>1</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>60</X><Y>250</Y></Point>
    <Point><X>70</X><Y>350</Y></Point>
    <Point><X>80</X><Y>450</Y></Point>
  </Points>
  <FillColor>
    <R>128</R> <G>128</G> <B>128</B>
  </FillColor>
  <AnimationLinks>
  </AnimationLinks>
</Polygon>
```

Minimal example:

```
<Polygon>
  <Points>
    <Point><X>50</X><Y>150</Y></Point>
    <Point><X>80</X><Y>450</Y></Point>
  </Points>
</Polygon>
```

Text Object

The following elements can be specified for a text object.

Elements	Description
Title	Object name. Optional.
TextString	Displayed text string. Required. Text object is not created if the Textstring element is assigned an invalid value or is missing a value.
TextInfo	Defines how the text will be shown.
Rotation	Defines orientation of the text in degrees. Possible values:{0, 90, 180, 270}. Default is 0.
Animationlinks	Optional animation links list.
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. Object is not created if the Dimension element is assigned invalid values or is missing values.

Example:

```
<Text>
  <Title>text1</Title>
  <TextString>This is some text to
display</TextString>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
  <TextInfo>
    <Font>Arial</Font>
    <FontStyle>Regular</FontStyle>
    <FontSize>12</FontSize>
    <Underline>false</Underline>
    <Strikeout>false</Strikeout>
    <TextColor>
      <R>0</R>
      <G>0</G>
      <B>0</B>
    </TextColor>
    <TextJustify>Left</TextJustify>
  </TextInfo>
  <Rotation>0</Rotation>
  <AnimationLinks>
  </AnimationLinks>
</Text>
```

Minimal example:

```
<Text>
  <TextString>This is some text to
display</TextString>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</Text>
```

Bitmap Object

If a source image is specified, it must exist. Supported graphic file formats are: JPG, JPEG, BMP, TIF, PCX, BIF, and TGA.

A bitmap without a specified source image appears as a default rectangle element on a window.

Transparent Colors

The InTouch internal application object data structure for bitmaps contains a Boolean field indicating that transparent color should be applied. Another field stores the actual transparent color.

In WindowMaker, if you create a new bitmap object, it initially has a default transparent color that is black and is unavailable. After you select the transparent color tool, you have applied a permanent transparent color to the bitmap object. From the transparent color tool, there is no visual cue that the transparent color has been assigned to a bitmap.

For bitmap objects imported from an XML file, the transparent color node for a bitmap is an optional entry, but if it is present, the transparent color is enabled and the transparent color is assigned to the bitmap object. There is no mechanism for the end user to open the window in WindowMaker, to select the bitmap object, and then to disable the transparent color.

Bitmap Object Elements

The following elements can be specified for a bitmap object.

Elements	Description
Title	Bitmap object name. Optional.
Fillcolor	Interior RGB color of the bitmap object. Contains a color element. Default is rgb(0, 0, 0).
SourceImage	Path to the bitmap file. Default is Null.
Transparent	RGB color for transparent color. Default is rgb(0, 255, 0).
Pen	Defines the line around the bitmap object.
Animationlinks	Optional animation links list.

Elements	Description
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. An object is not created if the Dimension element is assigned invalid values or is missing values.

Example:

```

<Bitmap>
  <Title>bitmap1</Title>
  <Pen>
    <PenColor><Name>Black</Name></PenColor>
    <PenWidth>1</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
  <Dimension>
    <Left>100</Left>
    <Top>50</Top>
    <Width>270</Width>
    <Height>80</Height>
  </Dimension>
  <FillColor>
    <R>128</R>
    <G>128</G> <
    B>128</B>
  </FillColor>
  <Transparent>
    <R>0</R> <G>128</G> <B>255</B>
  </Transparent>
  <SourceImage>C:\MyPictures\hello.jpg</SourceImage>
  <AnimationLinks>
  </AnimationLinks>
</Bitmap>

```

Example:

```
<Bitmap>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
  <SourceImage>C:\MyPictures\hello.jpg</SourceImage>
</Bitmap>
```

Example with empty SourceImage node:

```
<Bitmap>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
  <FillColor>
    <R>128</R> <G>128</G> <B>128</B>
  </FillColor>
  <SourceImage></SourceImage>
</Bitmap>
```

Minimal example:

```
<Bitmap>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</Bitmap>
```

Button Object

The following elements can be specified for a button object.

Elements	Description
Title	Button object name. Optional.
TextString	Displayed caption. Default string is “Text”.
TextInfo	Defines how the caption is shown.
Animationlinks	Optional animation links list. Line Color, Fill Color, Text Color, Percent Fill, Orientation animation links are not permitted.
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. An object is not created if the Dimension element is assigned invalid values or is missing values.

Example:

```

Button>
  <Title>button1</Title>
  <TextInfo>
    <Font>Arial</Font>
    <FontStyle>Regular</FontStyle>
    <FontSize>12</FontSize>
    <Underline>false</Underline>
    <Strikeout>false</Strikeout>
    <TextColor>
      <R>0</R>
      <G>0</G>
      <B>0</B>
    </TextColor>
    <TextJustify>Left</TextJustify>
  </TextInfo>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
  <TextString>Stop All Robots</TextString>
  <AnimationLinks>
  </AnimationLinks>
</Button>

```

Minimal example:

```

<Button>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</Button>

```

Smart Symbols

A SmartSymbol template must be defined in the application before importing a window that uses the SmartSymbol template. If the SmartSymbol template does not exist, the import fails, and the window is not created.

Multiple SmartSymbols can be specified for a window. Each SmartSymbol is declared in a separate <SmartSymbol> </SmartSymbol> element.

The following elements can be specified for a Smart Symbol.

Elements	Description
SymbolName	SmartSymbol name. Required. An object is not created if the name does not match a SmartSymbol template name in the application, or the name is missing.
Dimension	Location and size of an object. Elements are top, left, width, height. Top, left, width, and height values are pixels. The resulting coordinates must be within -32000 and +32000. Both width and height cannot be zero. Required. An object is not created if the Dimension element is assigned invalid values or is missing values.
TagReplace	Instance tag replacement.
StringReplace	Instance string replacement.

Example:

```
<SmartSymbol>
  <SymbolName>MyCoolSymbol</SymbolName>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
  <TagReplace>
    <Find>Tag001</Find>
    <Replace>Tag007</Replace>
  </TagReplace>
  <StringReplace>
    <Find>Find This Text</Find>
    <Replace>Replace it with this text</Replace>
  </StringReplace>
</SmartSymbol>
```

Minimal example:

```

<SmartSymbol>
  <SymbolName>MyCoolSymbol</SymbolName>
  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>
</SmartSymbol>

```

Tag Replacement

The tags within a SmartSymbol instance can be replaced. You should only replace tags within a SmartSymbol instance that currently exists in the application.

Tag replacement is case insensitive and the entire tag name string must match. You can add one or more <TagReplace> elements to the SmartSymbol node to replace multiple tags.

For example:

```

<TagReplace>
  <Find>Tag1</Find>
  <Replace>DTagA</Replace>
</TagReplace>
<TagReplace>
  <Find><![CDATA[Tag1]]></Find>
  <Replace><![CDATA[DTagA]]></Replace>
</TagReplace>

```

If the tag names specified for replacement do not exist in an application, the SmartSymbol is not created. Also, the window containing the SmartSymbol is not created.

The replaced tag's type must be the same as the original tag's type.

String Replacement

Strings can be replaced within a SmartSymbol instance. String replacements are case sensitive and the entire source string must match. Multiple string replacement elements can be used within a single SmartSymbol element.

```
<StringReplace>
  <FIND>
    <![CDATA[Open]]>
  </FIND>
  <REPLACE>
    <![CDATA[Off]]>
  </REPLACE>
</StringReplace>

<StringReplace>
  <FIND>Open</FIND>
  <REPLACE>Off</REPLACE>
</StringReplace>
```

SmartSymbol Example

This is an example of a SmartSymbol element containing tag and string replacement elements.

```
<SmartSymbol>

  <SymbolName>MyCoolSymbol</SymbolName>

  <Dimension>
    <Left>100</Left><Top>50</Top>
    <Width>270</Width><Height>80</Height>
  </Dimension>

  <TagReplace>
    <Find>DTag1</Find>
    <Replace>DTagA</Replace>
  </TagReplace>

  <TagReplace>
    <Find>ATag001</Find>
    <Replace>ATag002</Replace>
  </TagReplace>
```

```

<StringReplace>
  <Find><![CDATA[Open]]> </Find>
  <Replace><![CDATA[On]]> </Replace>
</StringReplace>

```

```

<StringReplace>
  <FIND>
    <![CDATA[Closed]]>
  </FIND>
  <REPLACE>
    <![CDATA[Off]]>
  </REPLACE>
</StringReplace>

```

```

</SmartSymbol>

```

ArchestrA Symbols

An ArchestrA Symbol must be defined in the Galaxy before you import a window that uses the ArchestrA Symbol reference. If the ArchestrA Symbol reference does not exist, the import fails, and the window is not created.

You can specify symbols from an Instance or from the Graphic Toolbox in the symbol reference. However, you cannot specify Template symbols in the symbol reference.

Multiple ArchestrA Symbols can be specified for a window. Each ArchestrA Symbol is declared in a separate <ArchestrASymbol> </ArchestrASymbol> element.

The following elements can be specified for an ArchestrA Symbol.

Elements	Description
SymbolReference	Reference to look up the symbol in the Galaxy. Required. An object will not be created if the ArchestrA symbol specified by the symbol reference does not exist or if the symbol reference field is missing or empty.

Elements	Description
Dimension	The location and size of an object. The Dimension sub-elements are top, left, width, and height. Required. Specifying width and height is optional. If width and height are not specified, the original width and height of the ArchestrA Symbol is used. If only width is specified, then the height is calculated using the aspect ratio and vice versa. For example, if the original width was 200 and height was 100, and if you specify width as 100, the height is changed to 50. An object is not created if the Dimension element is assigned invalid values or does not contain a value. A window containing this symbol is also not created.
Title	The text name for an element. Optional. This name should be unique within the XML file; otherwise, it is ignored.
Flip	The flip type of an element. Optional.
Rotation	The rotation angle of an element. Optional.
StringReplace	One or more string replacement nodes. Optional.
CustomPropertyOverride	One or more custom property overrides. Optional.

Elements	Description
AnimationLinks	The list of animation links. Optional. Specifying Line Color, Fill Color, Text Color, Percent Fill, Orientation, Slider, Tooltip, ValueDisplay, Blink or UserInput is not permitted.

Example:

```

<ArchestrASymbol>

  <Title>EmbedSym1</Title>
  <Dimension>
    <Left>200</Left>
    <Top>200</Top>
    <Width>150</Width>
    <Height>150</Height>
  </Dimension>

  <Flip>None</Flip>
  <Rotation>0</Rotation>

  <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>

  <AnimationLinks>
  </AnimationLinks>

  <StringReplace>
    <Find>LABEL</Find>
    <Replace>OFF</Replace>
  </StringReplace>

</ArchestrASymbol>

```

Minimal example:

```

<ArchestrASymbol>

  <Dimension>
    <Left>200</Left>
    <Top>200</Top>

```

```
</Dimension>

<SymbolReference>ButtonChromeMomentaryRed</SymbolReference>

</ArchestrASymbol>
```

CustomPropertyOverride

You can override only those custom properties which are already defined for the ArchestrA Symbol.

To override multiple custom properties, add one or more `<CustomPropertyOverride>` elements to the ArchestrA Symbol node.

Example:

```
<ArchestrASymbol>
  <Title>EmbedSym1</Title>
  <Dimension>
    <Left>200</Left>
    <Top>200</Top>
    <Width>150</Width>
    <Height>150</Height>
  </Dimension>

  <Flip>None</Flip>
  <Rotation>0</Rotation>

  <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>

  <AnimationLinks>
  </AnimationLinks>

  <CustomPropertyOverride>
    <CustomPropertyName>cp1</CustomPropertyName>
    <OverrideValue>DTagA</OverrideValue>
    <IsConstant>false</IsConstant>
  </CustomPropertyOverride>
</ArchestrASymbol>
```

In this example, `cp1` is the name of the existing custom property. The override is applied to the custom property with the new value set to `DTagA`. `IsConstant` is an optional field used to indicate whether the value should be interpreted as a constant. The `IsConstant` flag is only applicable if the type of the custom property is String, Time, or Elapsed Time. The `IsConstant` flag is set to `false` by default.

Note If the tag name specified for `OverrideValue` does not exist in the tag database, the ArchestrA Symbol reference is not created on the window and import for that particular window fails. Error messages are logged in the log file or in the ArchestrA Logger.

ArchestrA Symbol String Replacement Type

The existing strings can be replaced within the ArchestrA Symbol instance. The string replacements are case sensitive. You can use multiple string replacement nodes within a single ArchestrA Symbol node.

```
<StringReplace>
  <FIND>
    <![CDATA[Open]]>
  </FIND>
  <REPLACE>
    <![CDATA[Off]]>
  </REPLACE>
</StringReplace>

<StringReplace>
  <FIND>Open</FIND>
  <REPLACE>Off</REPLACE>
</StringReplace>
```

Unsupported Window Objects

Cell, Symbol, Real-Time Trend, and Historical Trend objects cannot be imported. If elements for unsupported objects are included in the XML file, they are ignored.

Window Animation Links

Window object animation links are declared within the `<Animationlinks>` `</Animationlinks>` element.

There can be zero or more animation links within a window element.

Not all elements support all animation link types. The ArchestrA Symbols support the following animation links:

- `ObjSize_Height`
- `ObjSize_Width`
- `Location_Vert`
- `Location_Hori`
- `TPushB_Disc`

- TPushB_Action
- TPushB_ShowWin
- TPushB_HideWin
- Misc_Visib
- Misc_Disable

For more information about Window Animation Links, see the InTouch documentation.

Some animation link types prevent other animation link types from being created. Animation link processing occurs in the order specified in the XML file.

Script/Expression/Tag Name Requirements Matrix

Each animation link has a control field. The control field can be a script, an expression, or a tag name. Some control fields are limited by the type of tags or expressions allowed. The following table lists the control field required for each animation link and any limitation of the control field.

Animation Link Type	Control Field	Control Field Limit
Discrete User Input	Tag	Discrete tag
Analog User Input	Tag	Analog tag
String User Input	Tag	String tag
Discrete Line Color	Expression	Discrete expression
Analog Line Color	Expression	Analog expression
Discrete Alarm Line Color	Tag	Discrete tag alarm status
Analog Alarm Line Color	Tag	Analog tag alarm status
Discrete Fill Color	Expression	Discrete expression
Analog Fill Color	Expression	Analog expression
Discrete Alarm Fill Color	Tag	Discrete tag alarm status

Animation Link Type	Control Field	Control Field Limit
Analog Alarm Fill Color	Tag	Analog tag alarm status
Discrete Text Color	Expression	Discrete expression
Analog Text Color	Expression	Analog expression
Discrete Alarm Text Color	Tag	Discrete tag alarm status
Analog Alarm Text Color	Tag	Analog tag alarm status
Vertical Slider	Tag	Valid analog tag
Horizontal Slider	Tag	Valid analog tag
Object Size Height	Expression	Analog value
Object Size Width	Expression	Analog value
Vertical Location	Expression	Valid expression
Horizontal Location	Expression	Valid expression
Vertical Percent Fill	Expression	Analog value
Horizontal Percent Fill	Expression	Analog value
Discrete Touch Pushbutton	Tag	Discrete value
Action Touch Pushbutton	Script	Valid script
Show Window Touch Pushbutton	Window Name	Window must exist.
Hide Window Touch Pushbutton	Window Name	Window must exist.
Visibility	Expression	Discrete value

Animation Link Type	Control Field	Control Field Limit
Blink	Expression	Discrete value
Orientation	Expression	Analog value
Disable	Expression	Discrete value
Tooltip	Expression	String tag
Discrete Value Display	Expression	Discrete expression
Analog Value Display	Expression	Analog expression
String Value Display	Expression	String expression

Discrete User Input

The following elements can be specified for a discrete user input animation link.

Elements	Description
Title	Object name. Optional.
Message	Message text to user. Default is no message text. Required if the optional schema is used.
InputOnly	If input only: {true, false}. Default is false.
OnMessage	On message text for user. Default is On. Cannot be empty text.
OffMessage	Off message text for user. Default is Off. Cannot be empty text.
SetPrompt	Set prompt message text for user. Default is On. Cannot be empty text.
ResetPrompt	Reset prompt message text for user. Default is Off. Cannot be empty text.
KeyAssignment	Virtual key element. Default is no assignment. An empty string means no assignment occurs.
Expression	Discrete tag. Required. Object is not created if invalid or missing.

Example:

```

<UserInputDiscrete>
  <Title>UserInputDiscrete1</Title>
  <InputOnly>false</InputOnly>
  <KeyAssignment>
    <KeyCode>F1</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </KeyAssignment>
  <Message>Pump Valve State</Message>
  <Expression><![CDATA[dTag001]]></Expression>
  <OnMessage>On Message Text</OnMessage>
  <OffMessage>Off Message Text</OffMessage>
  <ResetPrompt>
    <![CDATA[Reset Prompt Text]]>
  </ResetPrompt>
  <SetPrompt>Set Prompt Text</SetPrompt>
</UserInputDiscrete>

```

Minimal example:

```

<UserInputDiscrete>
  <Message>Pump Valve State</Message>
  <Expression>dTag001</Expression>
</UserInputDiscrete>

```

Analog User Input

The following elements can be specified for an analog user input animation link.

Elements	Description
Title	Object name. Optional.
Message	Message text to user. Required if optional schema is used. Default is no message text.
InputOnly	If input only: true, false. Default is false.
MinAnalogValue	Minimum floating point value allowed. Required if optional schema is used. Default is 0.0

Elements	Description
MaxAnalogValue	Maximum floating point value allowed. Required if optional schema is used. Default is 100.0. Must be larger than the value assigned to the MINANALOGVALUE element.
KeyPadEnabled	Specifies whether a keypad is visible. Possible values are Yes or No. The default is No.
KeyAssignment	Virtual key element, empty string, or absent. Default is no assignment. Empty string means no assignment occurs.
Expression	Analog type tag name. Required. Object is not created if invalid or missing.

Example:

```
<<UserInputAnalog>
  <Title>UserInputAnalog1</Title>
  <InputOnly>>false</InputOnly>
  <KeyAssignment>
    <KeyCode>F1</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </KeyAssignment>
  <Message>Flush Pump Speed</Message>
  <Expression>aTag001</Expression>
  <KeyPadEnabled>>false</KeyPadEnabled>
  <MinAnalogValue>0.0</MinAnalogValue>
  <MaxAnalogValue> 100.0</MaxAnalogValue>
</UserInputAnalog>
```

Minimal example:

```
<UserInputAnalog>
  <Message>Pump Valve State</Message>
  <Expression><![CDATA[aTag001]]></Expression>
  <MinAnalogValue>0.0</MinAnalogValue>
  <MaxAnalogValue>100.0</MaxAnalogValue>
</UserInputAnalog>
```

String User Input

You should use either `EchoEnabled` or `EchoMode`, but not both. `EchoMode` allows you to specify the password mode, whereas the `EchoEnabled` allows only the states enabled or disabled.

You use the `Echo Character` element to control how user input is shown during run time. Possible values are `Yes`, `No`, and `Password`.

- If the element is set to `Yes`, then during run time the string characters are shown in the input edit box. Input can only be enabled. Password character and encryption are disabled.
- If the element is set to `No`, then during run time the input characters are not shown in the input edit box. Input can only be enabled. Password character and encryption are disabled.
- If the element is set to `Password`, then during run time the password character is shown instead of the password typed by the user. A password character is optional, but if specified, cannot be empty. The default password character is an asterisk. Encryption is optional and is off by default. Input only is mandatory and is forced on.

If the elements do not match these criteria, then default options are used.

You can specify the following elements for a string user input animation link.

Elements	Description
Title	Object name. Optional.
Message	Message text to user. Required if optional schema is used. Default is no message text.
InputOnly	If input only: {true, false}. Default is false.
EchoCharacters	Echo characters: no, yes, password. Required if optional schema is used. Default is yes.
KeyPadEnabled	If keypad is visible: true, false. Default is false.
PasswordCharacter	Password character. Default is “*”

Elements	Description
EncryptEnabled	Encryption enabled: yes, no. Default is no.
KeyAssignment	Virtual key element, empty string, or absent. Default is no assignment. Empty string means no assignment occurs.
Expression	Message-type tag. Required. Object is not created if invalid or missing.

Example:

```

<UserInputString>
  <Title>UserInputString1</Title>
  <InputOnly>false</InputOnly>
  <KeyAssignment>
    <KeyCode>F1</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </KeyAssignment>
  <Message>Select Pump</Message>
  <Expression>mTag001</Expression>
  <EchoCharacters>true</EchoCharacters>
  <EncryptEnabled>false</EncryptEnabled>
  <PasswordCharacter>*</PasswordCharacter>
</UserInputString>

```

Minimal example:

```

<UserInputString>
  <Message>Select Pump</Message>
  <Expression>mTag001</Expression>
  <EchoCharacters>true</EchoCharacters>
</UserInputString>

```

Discrete Line Color

The following elements can be specified for a discrete line color animation link.

Elements	Description
Title	Object name. Optional.
Oncolor	Contains a color element. Default is rgb(0,0,0).
Offcolor	Contains a color element. Default is rgb(0,0,0).
Expression	Discrete tag or expression. Required. Object is not created if the expression element values are missing or invalid.

Example:

```
<LineColorDiscrete>
  <Title>LineColorDiscrete1</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <OnColor><Name>Green</Name></OnColor>
  <OffColor><Name>Red</Name></OffColor>
</LineColorDiscrete>
```

Minimal example:

```
<LineColorDiscrete>
  <Expression>dTag001</Expression>
</LineColorDiscrete>
```

Analog Line Color

The required breakpoint values must have increasing values. If this is not the case, a warning is logged, and the offending value is one plus the previous value.

The object containing the animation link is not created if a required value is missing or invalid.

You can specify the following elements for an analog line color animation link.

Elements	Description
Title	Object name. Optional.
Color1	Contains a color element. Default is rgb(0,0,0).
Color2	Contains a color element. Default is rgb(0,0,0).
Color3	Contains a color element. Default is rgb(0,0,0).
Color4	Contains a color element. Default is rgb(0,0,0).
Color5	Contains a color element. Default is rgb(0,0,0).
Color6	Contains a color element. Default is rgb(0,0,0).
Color7	Contains a color element. Default is rgb(0,0,0).
Color8	Contains a color element. Default is rgb(0,0,0).
Color9	Contains a color element. Default is rgb(0,0,0).
Color10	Contains a color element. Default is rgb(0,0,0).
Values	Constant analog values. Must contain nine Value elements. Required.
Expression	Analog tag or expression. Required.

Example:

```
<LineColorAnalog>
  <Title>LineColorAnalog1</Title>
  <Expression>aTag001</Expression>
  <Colors>
    <Color1><Name>White</Name></Color1>
    <Color2><Name>Red</Name></Color2>
    <Color3><Name>Orange</Name></Color3>
    <Color4><Name>Yellow</Name></Color4>
    <Color5><Name>Green</Name></Color5>
    <Color6><Name>Blue</Name></Color6>
    <Color7><Name>Cyan</Name></Color7>
    <Color8><Name>Magenta</Name></Color8>
    <Color9><Name>Violet</Name></Color9>
    <Color10><Name>Black</Name></Color10>
  </Colors>
  <Values>
    <Value>10.0</Value>
    <Value>20.0</Value>
    <Value>30.0</Value>
    <Value>40.0</Value>
    <Value>50.0</Value>
    <Value>60.0</Value>
    <Value>70.0</Value>
    <Value>80.0</Value>
    <Value>90.0</Value>
  </Values>
</LineColorAnalog>
```

Minimal example:

```
<LineColorAnalog>
  <Expression>aTag001</Expression>
  <Values>
    <Value>10.0</Value>
    <Value>20.0</Value>
    <Value>30.0</Value>
    <Value>40.0</Value>
    <Value>50.0</Value>
    <Value>60.0</Value>
    <Value>70.0</Value>
    <Value>80.0</Value>
    <Value>90.0</Value>
  </Values>
</LineColorAnalog>
```

Discrete Alarm Line Color

The following elements can be specified for a discrete alarm line color animation link.

Elements	Description
Title	Object name. Optional.
NormalColor	Contains a color element. Default is rgb(0,0,0.)
AlarmColor	Contains a color element. Default is rgb(0,0,0.)
Expression	Discrete tag. Required. Object is not created if expression is invalid or missing.

Example:

```
<LineColorDiscreteAlarm>
  <Title>LineColorDiscreteAlarm1</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <NormalColor><Name>Black</Name></NormalColor>
  <AlarmColor><Name>Red</Name></AlarmColor>
</LineColorDiscreteAlarm>
```

Minimal example:

```
<LineColorDiscreteAlarm>
  <Expression>dTag001</Expression>
</LineColorDiscreteAlarm>
```

Analog Alarm Line Color

The following elements can be specified for an analog alarm line color animation link.

Elements	Description
Title	Object name. Optional.
Value: LOLOCOLOR	Contains a color element. Default is rgb(0,0,0).
Value:LOCOLOR	Contains a color element. Default is rgb(0,0,0).
Value: NORMALCOLOR	Contains a color element. Default is rgb(0,0,0).
Value:HICOLOR	Contains a color element. Default is rgb(0,0,0).
Value: HIHICOLOR	Contains a color element. Default is rgb(0,0,0).
Deviation: NORMALCOLOR	Contains a color element. Default is rgb(0,0,0).
Deviation: MINORCOLOR	Contains a color element. Default is rgb(0,0,0).
Deviation: MAJORCOLOR	Contains a color element. Default is rgb(0,0,0).
ROC: NORMALCOLOR	Contains a color element. Default is rgb(0,0,0).
ROC: ROCCOLOR	Contains a color element. Default is rgb(0,0,0).
Expression	Analog tag. Required. Object is not created if expression is invalid or missing.

Example of a value alarm:

```

<LineColorAnalogAlarm>
  <Title>LineColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <Value>
    <LoLoColor><Name>Red</Name></LoLoColor>
    <LoColor><Name>DarkRed</Name></LoColor>
    <NormalColor><Name>Black</Name></NormalColor>
    <HiColor><Name>DarkGreen</Name></HiColor>
    <HiHiColor><Name>Green</Name></HiHiColor>
  </Value>
</LineColorAnalogAlarm>

```

Example of a deviation alarm:

```

<LineColorAnalogAlarm>
  <Title>LineColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <Deviation>
    <NormalColor><Name>Black</Name></NormalColor>
    <MinorColor><Name>Green</Name></MinorColor>
    <MajorColor><Name>Red</Name></MajorColor>
  </Deviation>
</LineColorAnalogAlarm>

```

Example of a ROC alarm:

```

<LineColorAnalogAlarm>
  <Title>LineColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <ROC>
    <NormalColor><Name>Black</Name></NormalColor>
    <ROCColor><Name>Red</Name></ROCColor>
  </ROC>
</LineColorAnalogAlarm>

```

Discrete Fill Color

The following elements can be specified for a discrete fill color animation link.

Elements	Description
Title	Object name. Optional.
OnColor	Contains a color element. Default is rgb(0,0,0).
OffColor	Contains a color element. Default is rgb(0,0,0).
Expression	Discrete tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<FillColorDiscrete>
  <Title>FillColorDiscrete1</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <OnColor><Name>Green</Name></OnColor>
  <OffColor><Name>Red</Name></OffColor>
</FillColorDiscrete>
```

Minimal example:

```
<FillColorDiscrete>
  <Expression>dTag001</Expression>
</FillColorDiscrete>
```

Analog Fill Color

The object containing the animation link is not created if a required value is missing or invalid.

You can specify the following elements for an analog fill color animation link.

Elements	Description
Title	Object name. Optional.
Color1	Contains a color element. Default is rgb(0,0,0).
Color2	Contains a color element. Default is rgb(0,0,0).
Color3	Contains a color element. Default is rgb(0,0,0).
Color4	Contains a color element. Default is rgb(0,0,0).
Color5	Contains a color element. Default is rgb(0,0,0).
Color6	Contains a color element. Default is rgb(0,0,0).
Color7	Contains a color element. Default is rgb(0,0,0).
Color8	Contains a color element. Default is rgb(0,0,0).
Color9	Contains a color element. Default is rgb(0,0,0).
Color10	Contains a color element. Default is rgb(0,0,0).
Values	Constant analog values. Must contain nine value elements. Required.
Expression	Analog tag or expression. Required.

Example:

```
<FillColorAnalog>
  <Title>FillColorAnalog1</Title>
  <Expression>aTag001</Expression>
  <Colors>
    <Color1><Name>White</Name></Color1>
    <Color2><Name>Red</Name></Color2>
    <Color3><Name>Orange</Name></Color3>
```

```
<Color4><Name>Yellow</Name></Color4>
<Color5><Name>Green</Name></Color5>
<Color6><Name>Blue</Name></Color6>
<Color7><Name>Cyan</Name></Color7>
<Color8><Name>Magenta</Name></Color8>
<Color9><Name>Violet</Name></Color9>
<Color10><Name>Black</Name></Color10>
</Colors>
<Values>
  <Value>10.0</Value>
  <Value>20.0</Value>
  <Value>30.0</Value>
  <Value>40.0</Value>
  <Value>50.0</Value>
  <Value>60.0</Value>
  <Value>70.0</Value>
  <Value>80.0</Value>
  <Value>90.0</Value>
</Values>
</FillColorAnalog>
```

Minimal example:

```
<FillColorAnalog>
  <Expression>aTag001</Expression>
  <Values>
    <Value>10.0</Value>
    <Value>20.0</Value>
    <Value>30.0</Value>
    <Value>40.0</Value>
    <Value>50.0</Value>
    <Value>60.0</Value>
    <Value>70.0</Value>
    <Value>80.0</Value>
    <Value>90.0</Value>
  </Values>
</FillColorAnalog>
```

Discrete Alarm Fill Color

The following elements can be specified for a discrete alarm fill color animation link.

Elements	Description
Title	Object name. Optional.
NormalColor	Contains a color element. Default is rgb(0,0,0).
AlarmColor	Contains a color element. Default is rgb(0,0,0).
Expression	Discrete tag. Required. Object is not created if the expression is invalid or missing.

Example:

```
<FillColorDiscreteAlarm>
  <Title>FillColorDiscreteAlarm1</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <NormalColor><Name>Black</Name></NormalColor>
  <AlarmColor><Name>Red</Name></AlarmColor>
</FillColorDiscreteAlarm>
```

Minimal example:

```
<FillColorDiscreteAlarm>
  <Expression>dTag001</Expression>
</FillColorDiscreteAlarm>
```

Analog Alarm Fill Color

The following elements can be specified for an analog alarm fill color animation link. The object containing the animation link is not created if a required value is missing or invalid.

Elements	Description
Title	Object name. Optional.
Value: LoLoColor	Contains a color element. Default is rgb(0,0,0).
Value: LoColor	Contains a color element. Default is rgb(0,0,0).
Value: NormalColor	Contains a color element. Default is rgb(0,0,0).
Value: HiColor	Contains a color element. Default is rgb(0,0,0).
Value: HiHiColor	Contains a color element. Default is rgb(0,0,0).
Deviation: NormalColor	Contains a color element. Default is rgb(0,0,0).
Deviation: MinorColor	Contains a color element. Default is rgb(0,0,0).
Deviation: MajorColor	Contains a color element. Default is rgb(0,0,0).
ROC: NormalColor	Contains a color element. Default is rgb(0,0,0).
ROC: ROCColor	Contains a color element. Default is rgb(0,0,0).
Expression	Analog tag. Required. Object is not created if the expression is invalid or missing.

Example of a value alarm:

```

<FillColorAnalogAlarm>
  <Title>FillColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <Value>
    <LoLoColor><Name>Red</Name></LoLoColor>
    <LoColor><Name>DarkRed</Name></LoColor>
    <NormalColor><Name>Black</Name></NormalColor>
    <HiColor><Name>DarkGreen</Name></HiColor>
    <HiHiColor><Name>Green</Name></HiHiColor>
  </Value>
</FillColorAnalogAlarm>

```

Example of a Deviation Alarm:

```

<FillColorAnalogAlarm>
  <Title>FillColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <Deviation>
    <NormalColor><Name>Black</Name></NormalColor>
    <MinorColor><Name>Green</Name></MinorColor>
    <MajorColor><Name>Red</Name></MajorColor>
  </Deviation>
</FillColorAnalogAlarm>

```

Example of a ROC Alarm:

```

<FillColorAnalogAlarm>
  <Title>FillColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <ROC>
    <NormalColor><Name>Black</Name></NormalColor>
    <ROCColor><Name>Red</Name></ROCColor>
  </ROC>
</FillColorAnalogAlarm>

```

Discrete Text Color

The following elements can be specified for a discrete text color animation link.

Elements	Description
Title	Object name. Optional.
OnColor	Contains a color element. Default is rgb(0,0,0).
OffColor	Contains a color element. Default is rgb(0,0,0).
Expression	String tag expression. Required.

Example:

```
<TextColorDiscrete>
  <Title>TextColorDiscretel</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <OnColor><Name>Green</Name></OnColor>
  <OffColor><Name>Red</Name></OffColor>
</TextColorDiscrete>
```

Minimal example:

```
<TextColorDiscrete>
  <Expression>dTag001</Expression>
</TextColorDiscrete>
```

Analog Text Color

The following elements can be specified for an analog text color animation link. The object containing the animation link is not created if a required value is missing or invalid.

Elements	Description
Title	Object name. Optional.
Color1	Contains a color element. Default is rgb(0,0,0).
Color2	Contains a color element. Default is rgb(0,0,0).
Color3	Contains a color element. Default is rgb(0,0,0).
Color4	Contains a color element. Default is rgb(0,0,0).
Color5	Contains a color element. Default is rgb(0,0,0).
Color6	Contains a color element. Default is rgb(0,0,0).
Color7	Contains a color element. Default is rgb(0,0,0).
Color8	Contains a color element. Default is rgb(0,0,0).
Color9	Contains a color element. Default is rgb(0,0,0).
Color10	Contains a color element. Default is rgb(0,0,0).
Values	Constant analog values. Must contain nine value elements. Required.
Expression	Analog tag expression. Required.

Example:

```
<TextColorAnalog>
  <Title>TextColorAnalog1</Title>
  <Expression>aTag001</Expression>
  <Colors>
    <Color1><Name>White</Name></Color1>
    <Color2><Name>Red</Name></Color2>
    <Color3><Name>Orange</Name></Color3>
    <Color4><Name>Yellow</Name></Color4>
    <Color5><Name>Green</Name></Color5>
    <Color6><Name>Blue</Name></Color6>
    <Color7><Name>Cyan</Name></Color7>
    <Color8><Name>Magenta</Name></Color8>
    <Color9><Name>Violet</Name></Color9>
    <Color10><Name>Black</Name></Color10>
  </Colors>
  <Values>
    <Value>10.0</Value> <Value>20.0</Value>
    <Value>30.0</Value> <Value>40.0</Value>
    <Value>50.0</Value> <Value>60.0</Value>
    <Value>70.0</Value> <Value>80.0</Value>
    <Value>90.0</Value>
  </Values>
</TextColorAnalog>
```

Minimal example:

```
<TextColorAnalog>
  <Expression>aTag001</Expression>
  <Values>
    <Value>10.0</Value>
    <Value>20.0</Value>
    <Value>30.0</Value>
    <Value>40.0</Value>
    <Value>50.0</Value>
    <Value>60.0</Value>
    <Value>70.0</Value>
    <Value>80.0</Value>
    <Value>90.0</Value>
  </Values>
</TextColorAnalog>
```

Discrete Alarm Text Color

The following elements can be specified for a discrete alarm text color animation link.

Elements	Description
Title	Object name. Optional.
NormalColor	Contains a color element. Default is rgb(0,0,0).
AlarmColor	Contains a color element. Default is rgb(0,0,0).
Expression	Discrete tag. Required. Object is not created if the expression is invalid or missing.

Example:

```
<TextColorDiscreteAlarm>
  <Title>TextColorDiscreteAlarm1</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <NormalColor><Name>Black</Name></NormalColor>
  <AlarmColor><Name>Red</Name></AlarmColor>
</TextColorDiscreteAlarm>
```

Minimal example:

```
<TextColorDiscreteAlarm>
  <Expression>dTag001</Expression>
</TextColorDiscreteAlarm>
```

Analog Alarm Text Color

The following elements can be specified for an analog alarm text color animation link.

Elements	Description
Title	Object name. Optional.
Value: LoLoColor	Contains a color element. Default is rgb(0,0,0).
Value: LoColor	Contains a color element. Default is rgb(0,0,0).
Value: NormalColor	Contains a color element. Default is rgb(0,0,0).
Value: HiColor	Contains a color element. Default is rgb(0,0,0).
Value: HiHiColor	Contains a color element. Default is rgb(0,0,0).
Deviation: NormalColor	Contains a color element. Default is rgb(0,0,0).
Deviation: MinorColor	Contains a color element. Default is rgb(0,0,0).
Deviation: MajorColor	Contains a color element. Default is rgb(0,0,0).
ROC: NormalColor	Contains a color element. Default is rgb(0,0,0).
ROC: ROCColor	Contains a color element. Default is rgb(0,0,0).
Expression	Analog tag. Required. Object is not created if the expression is invalid or missing.

Example:

```

<TextColorAnalogAlarm>
  <Title>TextColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <Value>
    <LoLoColor><Name>Red</Name></LoLoColor>
    <LoColor><Name>DarkRed</Name></LoColor>
    <NormalColor><Name>Black</Name></NormalColor>
    <HiColor><Name>DarkGreen</Name></HiColor>
    <HiHiColor><Name>Green</Name></HiHiColor>
  </Value>
</TextColorAnalogAlarm>

```

Example of a Deviation Type Alarm:

```

<TextColorAnalogAlarm>
  <Title>TextColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <Deviation>
    <NormalColor><Name>Black</Name></NormalColor>
    <MinorColor><Name>Green</Name></MinorColor>
    <MajorColor><Name>Red</Name></MajorColor>
  </Deviation>
</TextColorAnalogAlarm>

```

Example of a ROC Type Alarm:

```

<TextColorAnalogAlarm>
  <Title>TextColorAnalogAlarm1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <ROC>
    <NormalColor><Name>Black</Name></NormalColor>
    <ROCColor><Name>Red</Name></ROCColor>
  </ROC>
</TextColorAnalogAlarm>

```

Vertical Slider

You can specify the following elements for a vertical slider animation link.

Elements	Description
Title	Object name. Optional.
ReferenceLocation	Vertical reference: top, middle, bottom. Default is bottom.
TopValue	Top value. Cannot be the same as BottomValue. Default is 0.
BottomValue	Bottom value. Default is 100.
UpwardMovement	Upward movement. Must range from 0 to 32767. Out of range values are clamped and an error message is logged. Default is 50.
DownwardMovement	Downward movement. Must range from 0 to 32767. Out of range values are clamped and an error message is logged. Default is 50.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<SliderVertical>
  <Title>SliderVertical1</Title>
  <ReferenceLocation>Bottom</ReferenceLocation>
  <TopValue>10.0</TopValue>
  <BottomValue> 110.0</BottomValue>
  <UpwardMovement> 20.0</UpwardMovement>
  <DownwardMovement> 120.0</DownwardMovement>
  <Expression> <![CDATA[aTag001]]> </Expression>
</SliderVertical>
```

Minimal example:

```
<SliderVertical>
  <Expression>aTag001</Expression>
</SliderVertical>
```

Horizontal Slider

The following elements can be specified for a horizontal slider animation link.

Elements	Description
Title	Object name. Optional.
ReferenceLocation	Reference location: left, center, right. Default is left.
LeftValue	Left value of a slider. Default is 0.
RightValue	Right value of a slider. Default is 100.
LeftMovement	Left horizontal movement. Must range from 0 to 32767. Out of range values are clamped and an error message is logged. Default is 50.
RightMovement	Right horizontal movement. Must range from 0 to 32767. Out of range values are clamped and an error message is logged. Default is 50.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<SliderHorizontal>
  <Title>SliderHorizontal1</Title>
  <ReferenceLocation>Left</ReferenceLocation>
  <LeftValue>10.0</LeftValue>
  <RightValue>120.0</RightValue>
  <LeftMovement>20.0</LeftMovement>
  <RightMovement>150.0</RightMovement>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
</SliderHorizontal>
```

Minimal example:

```
<SliderHorizontal>
  <Expression><![CDATA[aTag001]]></Expression>
</SliderHorizontal>
```

Object Height

The following elements can be specified for an object height animation link. An object height animation link cannot be used with an orientation animation link.

Elements	Description
Title	Object name. Optional.
SizeAnchor	Defines where on the object the anchor is located. Values can be bottom, middle, top. Default is bottom.
SizeMin	Value at minimum height. Default is 0.
SizeMax	Value at maximum height. Default is 100. Must be greater than the value assigned to the SizeMin element.
MinPercent	Minimum percentage height. Default is 0. Range is from 0 to 100.
MaxPercent	Maximum percentage height. Default is 100. Must be more than minimum percent. Range is from 0 to 100.
Expression	Analog tag name or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<ObjectSizeHeight>
  <Title>ObjectSizeHeight1</Title>
  <Expression> <![CDATA[aTag001]]> </Expression>
  <SizeMin>0.0</SizeMin>
  <SizeMax>100.0</SizeMax>
  <MinPercent>0.0</MinPercent>
  <MaxPercent>100.0</MaxPercent>
  <SizeAnchor>Top</SizeAnchor>
</ObjectSizeHeight>
```

Minimum example:

```
<ObjectSizeHeight>
  <Expression>aTag001</Expression>
</ObjectSizeHeight>
```

Object Width

The following elements can be specified for an object size width animation link. An object width animation link cannot be used with an orientation animation link.

Elements	Description
Title	Object name. Optional.
SizeAnchor	Defines where on the object the anchor is located. Values can be: {Left, center, right}. Default is left.
SizeMin	Value at minimum width. Default is 0.
SizeMax	Value at maximum width. Default is 100. Must be larger than SizeMin.
MinPercent	Minimum percentage width. Default is 0. Range is from 0 to 100.
MaxPercent	Maximum percentage width. Default is 100. Must be more than MinPercent. Range is from 0 to 100.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
< ObjectSizeWidth>
  <Title>ObjectSizeWidth1</Title>
  <Expression> <![CDATA[aTag001]]> </Expression>
  <SizeMin>0.0</SizeMin>
  <SizeMax>100.0</SizeMax>
  <MinPercent>0.0</MinPercent>
  <MaxPercent>100.0</MaxPercent>
  <SizeAnchor>Center</SizeAnchor>
</ObjectSizeWidth>
```

Minimum example:

```
<ObjectSizeWidth>
  <Expression>aTag001</Expression>
</ObjectSizeWidth>
```

Vertical Location

The following elements can be specified for a vertical location animation link. An object vertical location animation link cannot be used with an orientation animation link.

Elements	Description
Title	Object name. Optional.
MinValue	Value at top. Default is 0. Cannot be the same value as at bottom.
MaxValue	Value at bottom. Default is 100. Cannot be same value as at top.
DecreaseMovement	Vertical movement upward. Default is 0. Cannot be less than 0.
IncreaseMovement	Vertical movement downward. Default is 100. Cannot be greater than 100.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<LocationVertical>
  <Title>LocationVertical1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <MinValue>0.0</MinValue>
  <MaxValue>100.0</MaxValue>
  <DecreaseMovement>0</DecreaseMovement>
  <IncreaseMovement>100</IncreaseMovement>
</LocationVertical>
```

Minimal example:

```
<LocationVertical>
  <Expression>aTag001</Expression>
</LocationVertical>
```

Horizontal Location

The following elements can be specified for a horizontal location animation link. An object horizontal location animation link cannot be used with an orientation animation link.

Elements	Description
Title	Object name. Optional.
MinValue	Value at left. Default is 0. Cannot be same value as at right end.
MaxValue	Value at right. Default is 100. Cannot be same value as at left end.
DecreaseMovement	Horizontal movement to left. Default is 0. Must not be less than 0.
IncreaseMovement	Horizontal movement to right. Default is 100. Must not be greater than 100.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<LocationHorizontal>
  <Title>LocationHorizontal1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
  <MinValue>0.0</MinValue>
  <MaxValue>100.0</MaxValue>
  <DecreaseMovement>0</DecreaseMovement>
  <IncreaseMovement>100</IncreaseMovement>
</LocationHorizontal>
```

Minimal example:

```
<LocationHorizontal>
  <Expression>aTag001</Expression>
</LocationHorizontal>
```

Vertical Percent Fill

The following elements can be specified for a vertical percent fill animation link. A vertical percent fill animation link cannot be used with an orientation animation link.

Elements	Description
Title	Object name. Optional.
FillDirection	Defines the direction of motion. Possible values are: {Up, down}. Default is up. Required.
FillColor	Solid fill color element. Default is rgb(0, 0, 0).
FillMin	Defines minimum value. Default is 0.
FillMax	Defines maximum value. Default is 100. Must be more than the minimum fill.
FillMinPercent	Defines minimum value as a percentage. Default is 0. Range from 0 to 100.
FillMaxPercent	Defines maximum value as a percentage. Default is 100. Range from 0 to 100. Must be more than the minimum percentage.
Expression	Analog tag or expression. Required.

Example:

```
<PercentFillVertical>
  <Title>PercentFillVertical1</Title>
  <Expression><![CDATA[aTag001]]></Expression>
  <FillMin>0.0</FillMin>
  <FillMax>100.0</FillMax>
  <FillMinPercent>0</FillMinPercent>
  <FillMaxPercent>100</FillMaxPercent>
  <FillColor><Name>Purple</Name></FillColor>
  <FillDirection>Up</FillDirection>
</PercentFillVertical>
```

Minimal example:

```
<PercentFillVertical>
  <Expression>aTag001</Expression>
  <FillDirection>Up</FillDirection>
```

```
</PercentFillVertical>
```

Horizontal Percent Fill

The following elements can be specified for a horizontal percent fill animation link. A horizontal percent fill animation link cannot be used with an orientation animation link.

Elements	Description
Title	Object name. Optional.
FillDirection	Left, right. Default is right.
FillColor	Contains a color element. Default is rgb(0, 0, 0).
FillMin	Value at minimum fill. Default is 0.
FillMax	Value at maximum fill. Default is 100. Must be more than the minimum fill.
FillMinPercent	Minimum percent to fill object. Default is 0. Range from 0 to 100.
FillMaxPercent	Maximum percent to fill object. Default is 100. Range from 0 to 100. Must be more than the minimum percent fill.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<PercentFillHorizontal>
  <Title>PercentFillHorizontal1</Title>
  <Expression><![CDATA[aTag001]]></Expression>
  <FillMin>0.0</FillMin>
  <FillMax>100.0</FillMax>
  <FillMinPercent>0</FillMinPercent>
  <FillMaxPercent>100</FillMaxPercent>
  <FillColor><Name>Purple</Name></FillColor>
  <FillDirection>Right</FillDirection>
</PercentFillHorizontal>
```

Minimal example:

```
<PercentFillHorizontal>
  <Expression>aTag001</Expression>
  <FillDirection>Left</FillDirection>
```

```
</PercentFillHorizontal>
```

Discrete Pushbutton

The following elements can be specified for a discrete pushbutton animation link.

Elements	Description
Title	Object name. Optional.
ButtonType	Type of discrete button: {direct, reverse, toggle, reset, or set}. Required. Object is not created if invalid or missing.
Expression	Discrete type tag or expression. Required. Object is not created if the expression element is invalid or missing.
KeyAssignment	Virtual key element, empty string, or absent. Default is no assignment. An empty string means no assignment occurs.

Example:

```
<ButtonDiscreteValue>
  <Title>ButtonDiscreteValue1</Title>
  <ButtonType>Reverse</ButtonType>
  <KeyAssignment>
    <KeyCode>F2</KeyCode>
    <KeyFlags>Shift</KeyFlags>
  </KeyAssignment>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
</ButtonDiscreteValue>
```

Minimal example:

```
<ButtonDiscreteValue>
  <ButtonType>Reverse</ButtonType>
  <Expression><![CDATA[dTag001]]></Expression>
</ButtonDiscreteValue>
```

Show Window Pushbutton

If a named window does not exist at the time the link is generated, a warning is logged. The link is generated without any action for the window. At least one of the named windows must exist, or the ShowWindow animation link is not imported.

You can specify the following elements for a show window pushbutton animation link.

Element	Description
Title	Object name. Optional.
WindowName	Name of window to show. There must be at least one window name specified or this animation link is not added to the object.

Example:

```
<ButtonShowWindow>
  <Title>ButtonShowWindow1</Title>
  <WindowName>
    <![CDATA[Window006]]>
  </WindowName>
  <WindowName>Window007</WindowName>
  <WindowName><![CDATA[SliderWindow]]></WindowName>
</ButtonShowWindow>
```

Hide Window Pushbutton

If a named window does not exist at the time the link is generated, a warning is logged. The link is generated without any action for the window. At least one of the named windows must exist, or the HideWindow animation link is not imported.

The following elements can be specified for a hide window pushbutton animation link.

Element	Description
Title	Object name. Optional.
WindowName	Name of window to hide. There must be at least one window name specified or this animation link is not added to the object.

Example:

```

<ButtonHideWindow>
  <Title>ButtonHideWindow1</Title>
  <WindowName>
    <![CDATA[Window001]]>
  </WindowName>
  <WindowName>Window002</WindowName>
  <WindowName> <![CDATA[SliderWindow]]> </WindowName>
</ButtonHideWindow>

```

Visibility

The following elements can be specified for a visibility animation link.

Elements	Description
Title	Object name. Optional.
State	Defines if the object can be seen. Possible values are: {On, off}. Default is on. Required if optional schema is used.
Expression	Discrete tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```

<Visibility>
  <Title>Visibility1</Title>
  <Expression>dTag001</Expression>
  <State>On</State>
</Visibility>

```

Minimal example:

```

<Visibility>
  <Expression>dTag001</Expression>
  <State>On</State>
</Visibility>

```

Blink

The following elements can be specified for a blink animation link. The animation link blinks when the expression is true.

Elements	Description
Title	Object name. Optional.
BlinkAttribute	{Invisible, visible}. Default is visible.
BlinkSpeed	Defines the rate the object blinks. Possible values are: {Slow, medium, fast}. Default is medium.
TextColor	Defines the color of the text that blinks. Contains a color element. Default is rgb(0,0,0)
LineColor	Contains a color element. Default is rgb(0,0,0)
FillColor	Contains a color element. Default is rgb(0,0,0)
Expression	Discrete tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<Blink>
  <Title>Blink1</Title>
  <Expression>dTag001</Expression>
  <TextColor><R>0</R><G>0</G><B>0</B></TextColor>
  <LineColor><R>0</R><G>0</G><B>0</B></LineColor>
  <FillColor><R>0</R><G>255</G><B>0</B></FillColor>
  <BlinkAttribute>Invisible</BlinkAttribute>
  <BlinkSpeed>Slow</BlinkSpeed>
</Blink>
```

Minimal example:

```
<Blink>
  <Expression>dTag001</Expression>
  <BlinkAttribute>Visible</BlinkAttribute>
  <BlinkSpeed>Fast</BlinkSpeed>
</Blink>
```

Orientation

The following elements can be specified for an orientation animation link. An orientation animation link cannot be used with slider, size, location, or percent fill animation links.

Elements	Description
Title	Object name. Optional.
X	Horizontal offset from object center point. Optional. Default is 0.
Y	Vertical offset from object center point. Optional. Default is 0.
CWMax	Value at maximum clockwise rotation. Default is 100.
CWRotation	Clockwise rotation. Default is 360. Must range from 0 to 360. CWROTATION+CCWROTATION cannot exceed 360.
CCWMax	Value at maximum counter clockwise rotation. Default is 0.
CCWRotation	Counter clockwise rotation. Default is 0. Must range from 0 to 360. CWRotation+CCWRotation cannot exceed 360.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<Orientation>
  <Title>Orientation1</Title>
  <Expression> <![CDATA[aTag001]]> </Expression>
  <X>0</X> <Y>0</Y>
  <CWMax>100.0</CWMax>
  <CWRotation>360.0</CWRotation>
  <CCWMax>0.0</CCWMax>
  <CCWRotation>0.0</CCWRotation>
</Orientation>
```

Minimal example:

```
<Orientation>
  <Expression>aTag001</Expression>
</Orientation>
```

Disable

The following elements can be specified for a disable animation link.

Elements	Description
Title	Object name. Optional.
State	{On, off}. Default is On. Required if the optional schema is used.
Expression	Discrete tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<Disable>
  <Title>Disable1</Title>
  <Expression>dTag001</Expression>
  <State>On</State>
</Disable>
```

Minimal example:

```
<Disable>
  <Expression>dTag001</Expression>
  <State>On</State>
</Disable>
```

Static Tooltip

The following elements can be specified for a static tooltip animation link.

Elements	Description
Title	Object name. Optional.
Message	Static text message. Required. Object is not created if invalid or missing.

Example:

```
<TooltipStatic>
  <Title>TooltipStatic1</Title>
  <Message>
    <![CDATA[ Click here to win a million dollars!
  ]]>
  </Message>
</TooltipStatic>
```

Dynamic Tooltip

The following elements can be specified for a dynamic tooltip animation link.

Elements	Description
Title	Object name. Optional.
Expression	Expression element. Required. Object is not created if the expression is invalid or missing.

Example:

```
<TooltipTag>
  <Title>TooltipTag1</Title>
  <Expression>
    <![CDATA[sTag001]]>
  </Expression>
</TooltipTag>
```

Discrete Value Display

The following elements can be specified for a discrete value animation link.

Elements	Description
Title	Object name. Optional.
OnMessage	The string to display when the value of the animation link is True. Default text is On
OffMessage	The string to display when the value of the animation link is False. Default text is Off
Expression	Discrete tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```

<ValueDisplayDiscrete>
  <Title>ValueDisplayDiscrete1</Title>
  <Expression>
    <![CDATA[dTag001]]>
  </Expression>
  <OnMessage>
    <![CDATA[Pump is On]]>
  </OnMessage>
  <OffMessage>
    <![CDATA[Pump is Off]]>
  </OffMessage>
</ValueDisplayDiscrete>

```

Minimal example:

```

< ValueDisplayDiscrete>
  <Expression>dTag001</Expression>
</ValueDisplayDiscrete>

```

Analog Value Display

The following elements can be specified for an analog value animation link.

Elements	Description
Title	Object name. Optional.
Expression	Analog tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```

<<ValueDisplayAnalog>
  <Title>ValueDisplayAnalog1</Title>
  <Expression>
    <![CDATA[aTag001]]>
  </Expression>
</ValueDisplayAnalog>

```

String Value Display

The following elements can be specified for a string value animation link.

Elements	Description
Title	Object name. Optional.
Expression	Message tag or expression. Required. Object is not created if the expression is invalid or missing.

Example:

```
<ValueDisplayString>
  <Title>ValueDisplayString1</Title>
  <Expression>
    <![CDATA[sTag001]]>
  </Expression>
</ValueDisplayString>
```

Pushbutton Action Scripts

Pushbutton action script elements are declared within a button action script element, which is declared within an animation links element.

Example:

```
<AnimationLinks>
  <ButtonActionScripts>
    <OnLeftDown>
      <![CDATA[
        First line of script text
        Second line of script text
        N-th line of script text
      ]]>
    </OnLeftDown>
  </ButtonActionScripts>
</AnimationLinks>
```

On Left Key Down/On Key Down

Do not use both OnLeftDown and OnKeyDown animation links in the same object.

Example:

```
<OnLeftDown>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</ONLEFTDOWN>
```

While Left Key Down/While Key Down

The following elements can be specified for an WhileLeftDown or WhileKeyDown animation link.

Elements	Description
Text	Script text. Required.
Frequency	Script execution frequency in milliseconds. Range from 1 to 360000. Required if optional schema is used. Default is 1000.

Do not use WhileLeftDown and WhileKeyDown animation links in the same object.

Example:

```
<WhileLeftDown>
  <Text>
    <![CDATA[
      First line of script text
      Second line of script text
      N-th line of script text
    ]]>
  </Text>
  <Frequency>1000</Frequency>
</WhileLeftDown>
<
```

On Left Key Up/On Key Up

You place the text of a script in the OnLeftUp element. Do not use OnLeftUp and OnKeyUp animation links in the same object.

Example:

```
<OnLeftUp>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnLeftUp>
```

On Left Key Double Click

You place the text of a script in the OnLeftDoubleClick animation link.

Example:

```
<OnLeftDoubleClick>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnLeftDoubleClick>
```

On Right Key Down/On Right Down

You place the text of a script in the OnRightDown element. Do not use OnRightDown and OnRightKeyDown animation links in the same object.

Example:

```
<OnRightDown>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnRightDown>
```

While Right Key Down/While Right Down

The following elements can be specified for an WhileRightKeyDown or WhileRightDown animation link.

Elements	Description
Text	Script text. Required
Frequency	Script execution frequency in milliseconds. A number between 1 and 360000. Default is 1000.

Do not use both WhileRightDown and WhileRightKeyDown animation links in the same object.

Example:

```
<WhileRightDown>
  <Text>
    <![CDATA[
      First line of script text
      Second line of script text
      N-th line of script text
    ]]>
  </Text>
  <Frequency>1000</Frequency>
</WhileRightDown>
```

On Right Key Up/On Right Up

You place the text of a script in the OnRightUp and OnRightKeyUp elements. Do not use both OnRightUp and OnRightKeyUp animation links in the same object.

Example:

```
<OnRightUp>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnRightUp>
```

On Right Key Double Click

You place the text of a script in the `OnRightDoubleClick` element.

Example:

```
<OnRightDoubleClick>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnRightDoubleClick>
```

On Middle Key Down/On Middle Down

You put the text of a script in the `OnMiddleKeyDown` element. Do not use `OnMiddleDown` and `OnMiddleKeyDown` animation links in the same object.

Example:

```
<OnMiddleDown>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnMiddleDown>
```

While Middle Key Down/While Middle Down

The following elements can be specified for an WhileMiddleKeyDown or WhileMiddleDown animation link.

Elements	Description
Text	Script text. Required
Frequency	Script execution frequency in milliseconds. A number between 1 and 360000. Required if optional schema is used. Default is 1000.

Do not use WhileMiddleDown and WhileMiddleKeyDown animation links in the same object.

Example:

```
<WhileMiddleDown>
  <Text>
    <![CDATA[
      First line of script text
      Second line of script text
      N-th line of script text
    ]]>
  </Text>
  <Frequency>1000</Frequency>
</WhileMiddleDown>
```

On Middle Key Up/On Middle Up

You place the text of a script in the OnMiddleKeyUp and OnMiddleUp animation links. Do not use OnMiddleUp and OnMiddleKeyUp animation links in the same object.

Example:

```
<OnMiddleUp>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnMiddleUp>
```

On Middle Key Double Click

You place the text of a script in the OnMiddleDoubleClick element.

Example:

```
<OnMiddleDoubleClick>
  <![CDATA[
    First line of script text
    Second line of script text
    N-th line of script text
  ]]>
</OnMiddleDoubleClick>
```

On Mouse Over

The following elements can be specified for an OnMouseOver animation link.

Elements	Description
Text	Script text. Required.
Timeout	Timeout in milliseconds. Required if optional schema is used. Default is 250.

Example:

```
<OnMouseOver>
  <Text>
    <![CDATA[
      First line of script text
      Second line of script text
      N-th line of script text
    ]]>
  </Text>
  <Timeout>250</Timeout>
</OnMouseOver>
```

Left Key Equivalent

The following elements can be specified for a Left Key equivalent animation link.

Elements	Description
KeyCode	Virtual key code name or empty string. Required. An empty string means no assignment occurs.
KeyFlags	Virtual key flag combination or empty string. Required. An empty string means no assignment occurs. Disabled if no KeyCode is specified.

Example for "CTRL+B":

```
<LeftKey>
  <KeyCode>B</KeyCode>
  <KeyFlags>Ctrl</KeyFlags>
</LeftKey>
```

Example of a No Key Flag Modifier for "L":

```
<<LeftKey>
  <KeyCode>L</KeyCode>
</LeftKey>
```

Example of a Both Key Flag Modifiers for CTRL+SHIFT+F7:

```
<LeftKey>
  <KeyCode>F7</KeyCode>
  <KeyFlags>CtrlShift</KeyFlags>
</LeftKey>
```

Right Key Equivalent - Not Supported

The right key equivalent is not supported by the InTouch XML import functionality.

The following elements can be specified for a right key equivalent animation link.

Elements	Description
KeyCode	Virtual key code name or empty string. Required. An empty string means no assignment occurs.
KeyFlags	Virtual key flag combination or empty string. Required. An empty string means no assignment occurs. Disabled if no KeyCode is specified.

Example:

```
<RightKey>
  <KeyCode>B</KeyCode>
  <KeyFlags>Ctrl</KeyFlags>
</RightKey>
```

Middle Key Equivalent - Not supported

The middle key equivalent is not supported by the InTouch XML import functionality.

The following elements can be specified for a middle key equivalent animation link.

Elements	Description
KeyCode	Virtual key code name or empty string. Required. An empty string means no assignment occurs.
KeyFlags	Virtual key flag combination or empty string. Required. An empty string means no assignment occurs. Disabled if no KeyCode is specified.

```
<MiddleKey>
  <KeyCode>B</KeyCode>
  <KeyFlags>Ctrl</KeyFlags>
</MiddleKey>
```

Unsupported InTouch Features

Wizards and ActiveX controls cannot be imported using the XML import functionality.

You cannot import tags or the tag database using the XML import functionality. You can create a file containing tag definitions and use the DBLoad utility to import them to an application's Tagname Dictionary. For more information about using DBLoad, see the InTouch Data Management Guide. For information about running DBLoad from the command prompt, see Running DBLoad from the Command Prompt on page 26.

Index

A

- action scripts 112
- alarmcolor element
 - discrete alarm fill color 87
 - discrete alarm line color 81
 - discrete alarm text color 93
- analog alarm fill color animation links 88
- analog alarm line color animation links 82
- analog alarm text color animation links 94
- analog fill color animation links 85
- analog line color animation links 78
- analog text color animation links 91
- analog user input animation links 73
- analog value display animation links 111
- animation links
 - analog alarm fill color 88
 - analog alarm line color 82
 - analog alarm text color 94
 - analog fill color 85
 - analog line color 78
 - analog text color 91
 - analog user input 73
 - analog value display 111
- bitmap objects 57
- blink element 107
- button objects 60
- disable 109
- discrete alarm fill color 87
- discrete alarm line color 81
- discrete alarm text color 93
- discrete fill color 84
- discrete line color 77
- discrete pushbutton 104
- discrete text color 90
- discrete user input 72
- discrete value display 110
- ellipse objects 49
- hide window pushbutton 105
- horizontal location 101
- horizontal percent fill 103
- horizontal slider 97
- hv line objects 51
- left key equivalent 119
- line objects 50
- object height 98
- object width 99
- on left key double click 114
- on left key down 113
- on left key up 113
- on middle key double click 118
- on middle key down 116
- on middle key up 117
- on mouse over 118
- on right key double click 116

- on right key down 114
- on right key up 115
- orientation 108
- polygon objects 54
- polyline objects 53
- rectangle objects 45
- requirements matrix 70
- rounded rectangle objects 47
- show window pushbutton 105
- static tooltip 109
- string user input 75
- string value display 112
- text objects 55
- vertical line objects 52
- vertical location 100
- vertical percent fill 102
- vertical slider 96
- visibility 106
- while left key down 113
- while middle key down 117
- while right key down 115
- windows 69
- application, creating 16
- ArchestrA IDE Extension 22
- ArchestrA Symbols 20

B

- bitmap objects 57
- blink element, animation links 107
- blinkattribute element 107
- blinkspeed element 107
- bottomvalue element, vertical slider 96
- button objects 60
- buttondiscretevalue element 104
- buttonhidewindow element 106
- buttonshowwindow element 105
- buttontype element 104

C

- ccwmax element 108
- ccwrotation element 108
- CDATA
 - user supplied text 30
 - window scripts 41
- cell objects, unsupported 69
- ckeyflags element 119
- color elements 31
- colors

- name element 31
- reference element 32
- RGB element 31
- value element 32
- colors element 85
- command files
 - minimum 13
 - syntax 12
- command line
 - running DBDump 24
 - running DBLoad 26
 - running WindowMaker 21
- COMMANDLOGFILE command 14
- commands
 - COMMANDLOGFILE 14
 - GO 14
 - NEWWINDOWNAME 15
 - OLDWINDOWNAME 15
 - OUTPUTFILE 15
 - OUTPUTTARGET 15
 - OUTPUTTARGETNAME 15
 - PRINTAPPLICATIONINFORMATION 15
 - REFERENCETYPE 15
 - SEARCHFOR 15
 - syntax 14
 - VERSION 14
 - WINDOWCREATE 14
 - WINDOWDELETE 15
 - WINDOWMAKERCOMMANDFILE 14
 - WINDOWNAME 15
 - WINDOWRENAME 15
 - XMLFILEPATH 14
- cornerdimension element 44, 47
- creating an application 16
- cross reference 13
- CROSSREFERENCE command 15
- cwmax element 108
- cwrotation element 108

D

- DBDump, running 24
- DBLoad
 - adding tags 17
 - command line 26
- declaration, XML 30
- decreasemovement element
 - horizontal location 101

- vertical location 100
- deviation element 83
- dimension elements
 - bitmap objects 58
 - button objects 60
 - definition 35
 - ellipse objects 49
 - rectangle objects 45
 - rounded rectangle objects 47
 - smartsymbols 62, 66
 - text objects 55
- disable animation links 109
- discrete alarm line color animation links 81
- discrete alarm text color animation links 93
- discrete fill color animation links 84
- discrete line color animation links 77
- discrete pushbutton animation links 104
- discrete text color animation links 90
- discrete user input animation links 72
- discrete value display, animation links 110
- documentation conventions 7
- downwardmovement element 96
- dynamic tooltip element 110

E

- echo character element 75
- echocharacters element 75
- ellipse objects 49
- encryptenabled element 76
- error handling 19
- examples
 - analog alarm fill color 89
 - analog alarm line color 83
 - analog alarm text color 95
 - analog fill color 85
 - analog line color 79
 - analog text color 92
 - analog user input 74
 - analog value display 111
 - bitmap objects 58
 - blink animation 107
 - button objects 61
 - deviation alarm 83
 - dimension element 35
 - disable animation 109

- discrete alarm fill color 87
- discrete alarm line color 81
- discrete alarm text color 93
- discrete fill color 84
- discrete line color 77
- discrete pushbutton 104
- discrete text color 90
- discrete user input 73
- discrete value display 111
- dynamic tooltip animation 110
- ellipse objects 49
- expression 35
- hide window pushbutton 106
- horizontal line objects 51
- horizontal location 101
- horizontal percent fill 103
- horizontal sliders 97
- ini file 16
- key code strings 38
- left key equivalent 119
- line objects 50
- object height 98
- object width 99
- on left key double click 114
- on left key down 113
- on left key up 114
- on middle key double click 118
- on middle key down 116
- on middle key up 117
- on mouse over 118
- on right key double click 116
- on right key down 114
- on right key up 115
- onhide 42
- onshow window script 41
- orientation animation 108
- pen elements 34
- polygon objects 54
- polyline objects 53
- rectangle objects 46
- RGB elements 31
- ROC alarm 83
- rounded rectangle objects 48
- show window 105
- smartsymbols 62, 64
- static tooltip animation 109
- string value display 112
- text objects 56

- user input string 76
- vertical location 100
- vertical percent fill 102
- vertical slider 96
- visibility animation links 106
- while left key down 113
- while middle key down 117
- while right key down 115
- whilshowing 42
- window elements 40
- XML declaration 30
- expression elements
 - analog user input 74
 - analog value display 111
 - blink 107
 - disable animation 109
 - discrete alarm fill color 87
 - discrete alarm line color 81
 - discrete alarm text color 93
 - discrete fill color 84
 - discrete line color 77
 - discrete pushbutton 104
 - discrete text color 90
 - discrete user input 72
 - discrete value display 110
 - dynamic tooltip 110
 - horizontal location 101
 - horizontal percent fill 103
 - horizontal sliders 97
 - object height 98
 - object width 99
 - orientation animation 108
 - string value displays 112
 - vertical location 100
 - vertical percent fill 102
 - vertical slider 96
 - visibility animation links 106
- expressions 35

F

- file header, XML 30
- fillcolor element
 - bitmap objects 57
 - blink element 107
 - default value 44
 - ellipse objects 49
 - horizontal percent fill 103
 - polygon objects 54

- rectangle objects 45
- rounded rectangle objects 47
- vertical percent fill 102
- fillcoloranalog element 85
- fillcoloranalogalarm element 89
- fillcolordiscrete element 84
- fillcolordiscretealarm element 87
- filldirection element
 - horizontal percent fill 103
 - vertical percent fill 102
- fillmax element
 - horizontal percent fill 103
 - vertical percent fill 102
- fillmaxpercent element
 - horizontal percent fill 103
 - vertical percent fill 102
- fillmin element
 - horizontal percent fill 103
 - vertical percent fill 102
- fillminpercent element
 - horizontal percent fill 103
 - vertical percent fill 102
- flip element 44
- font element 44
- fontsize element 44
- fontstyle element 44
- fontweight element 44
- framestyle elements 39
- frequency element
 - while left key down 113
 - while middle key down 117
 - while right key down 115
 - whilshowing window 42

G

- GO command 14

H

- hide window pushbutton animation
 - links 105
- highcolor element 82
- hihicolor element 82
- historical trend objects, unsupported 69
- horizonatal line objects 51
- horizontal location animation links 101
- horizontal percent fill animation
 - links 103
- horizontal slider animation links 97

I

- increasemovement element
 - horizontal location 101
 - vertical location 100
- inputonly element
 - analog user input 73
 - discrete user input 72

K

- keyassignment element
 - analog user input 74
 - discrete pushbutton 104
 - discrete user input 72
- keycode element 119
- keypadenabled element 74, 75

L

- left key equivalent animation links 119
- leftmovement element 97
- leftvalue element 97
- line objects 50
- linecolor element 107
- linecoloranalog element 78
- linecoloranalogalarm element 82
- linecolordiscrete element 77
- linecolordiscretealarm element 81
- locationhorizontal element 101
- locationvertical element 100
- locolor element 82
- log file, creating 14
- lolocolor element 82

M

- majorcolor element 82
- Managed InTouch Application 22, 25, 27
- maxanalogvalue element 74
- maxvalue element
 - horizontal location 101
 - vertical location 100
- message elements
 - analog user input 73
 - discrete user input 72
 - static tooltip animation links 109
- middle key equivalent, not supported 120
- minanalogvalue element 73
- minimum command file 13
- minorcolor element 82

- minvalue element
 - horizontal location 101
 - vertical location 100

N

- name element 31
- NEWWINDOWNAME command 15
- normalcolor element 82
 - discrete alarm fill color 87
 - discrete alarm line color 81
 - discrete alarm text color 93

O

- object height, animation links 98
- object width animation links 99
- objectlist elements 39
- objects, windows 43
- objectsizeheight element 98
- objectsizewidth element 99
- offcolor element
 - discrete fill color 84
 - discrete line color 77
 - discrete text color 90
- offmessage elements
 - discrete user inputs 72
 - discrete value display 110
- OLDWINDOWNAME command 15
- on left key double click animation 114
- on left key down, animation links 113
- on left key up animation links 113
- on middle key double click
 - animations 118
- on middle key down animation links 116
- on middle key up animation links 117
- on mouse over animation links 118
- on right key double click animation
 - links 116
- on right key down animation 114
- on right key up animation links 115
- oncolor element
 - discrete fill color 84
 - discrete line color 77
 - discrete text color 90
- onhide window scripts 42
- onleftdown element 113
- onleftup element 114
- onmessage elements
 - discrete user input 72

- discrete value display 110
- onshow window script 41
- orientation animation links 108
- OUTPUTFILE command 15
- OUTPUTTARGET command 15
- OUTPUTTARGETNAME command 15

P

- passwordcharacter element 75
- pen color elements 34
- pen dimensions 45
- pen element
 - bitmap objects 57
 - definition 34
 - ellipse object 49
 - horizontal line objects 51
 - line objects 50
 - polygon object 54
 - polyline objects 53
 - rectangle object 45
 - rounded rectangle object 47
 - vertical line objects 52
- pen style elements 34
- pen width elements 34
- pencolor element 44
- penstyle element
 - default value 44
 - limitations 44
- penwidth element
 - default value 44
 - limitations 45
- percentfillhorizontal element 103
- percentfillvertical element 102
- point element 34
- points element
 - horizontal line objects 51
 - line objects 50
 - polygon objects 54
 - polyline objects 53
 - vertical line objects 52
- polygon objects 54
- polyline objects 53
- PRINTAPPLICATIONINFORMATION command 15
- printing to a file 13
- Process InTouch Command File 22
- pushbutton action scripts 112

R

- real-time trend objects, unsupported 69
- rectangle objects 45
- reference element 32
- referencelocation element
 - horizontal slider 97
 - vertical slider 96
- REFERENCETYPE command 15
- resetprompt element
 - discrete user input 72
- RGB elements 31
- right key equivalent, not supported 120
- rightmovement element 97
- rightvalue element, horizontal slider 97
- ROC element 83
- roccolor element 82
- rotation element 44, 55
- rounded rectangle objects 47

S

- schemas
 - limitations 29
 - using 41
- script elements
 - on mouse over 118
 - while left key down 113
 - while middle key down 117
 - while right key down 115
 - whileshowing window element 42
- scripts 41
- SEARCHFOR command 15
- setprompt element 72
- show window pushbutton animation
 - links 105
- sizeanchor element
 - object height 98
 - object width element 99
- sizecontrols element 39
- sizemax element
 - object height 98
 - object width 99
- sizemaxpercent element
 - object height 98
 - object width 99
- sizemin element
 - object height 98
 - object width 99
- sizeminpercent element

- object height 98
- object width 99
- smartsymbols 61
 - example 64
 - missing 20
 - string replacement 64
 - stringreplace element 62
 - tag replacement 63
- soruceimage element 57
- state element
 - disable animation links 109
 - visibility animation links 106
- static tooltip animation links 109
- strikeout element 44
- string replacement 64
- string user input animation links 75
- string value display animation links 112
- stringreplace elment 62
- symbol objects, unsupported 69
- symbolname element 62, 65
- syntax
 - command file 12
 - commands 14

T

- tag replacement, smartsymbols 63
- tagreplace element 62, 66
- tags
 - adding 17
- technical support, contacting 8
- text objects 55
- text, user supplied 30
- textcolor element
 - blink element 107
 - default value 44
- textcoloranalog element 92
- textcoloranalogalarm element 95
- textcolordiscretealarm element 93
- textinfo element
 - button objects 60
 - text objects 55
- textjustify element 44
- textstring element
 - button objects 60
 - text objects 55
- timeout element 118
- title element
 - bitmap objects 57

- button objects 60
- ellipse objects 49
- horizontal line objects 51
- line objects 50
- polygon objects 54
- polyline objects 53
- rectangle objects 45
- rounded rectangle objects 47
- text objects 55
- vertical line objects 52
- tooltipstatic element 109
- tooltiptag element 110
- topvalue element 96
- transparent colors, bitmap objects 57
- transparent element
 - bitmap objects 57
 - default value 44

U

- underline element 44
- unsupported InTouch features 121
- upwardmovement element 96
- user supplied text 30
- userstringinput element 75

V

- value element
 - color element 32
 - linecoloranalogalarm element 83
- valuedisplayanalog element 111
- valuedisplaydiscrete element 111
- valuedisplaystring element 112
- values element 85
- VERSION command 14
- vertical line objects 52
- vertical location animation links 100
- vertical percent fill animation links 102
- vertical slider animation links 96
- virtual key codes 36
- virtual key flags 36
- visibility animation link 106
- visibility element 106

W

- while left key down animation links 113
- while middle key down animation
 - links 117
- whileleftdown element 113

- whileshowing window scripts 42
- white space 30
- window animation links 69
- window elements
 - backgroundcolor element 39
 - comment element 39
 - definition 39
 - dimension element 39
 - framestyle element 39
 - objectlist 39
 - sizecontrols 39
 - title element 39
 - titlebar element 39
 - windowstyle element 39
- window objects
 - cornerdimension 44
 - default attribute values 44
 - explanation 43
 - fillcolor element 44
 - flip element 44
 - font element 44
 - fontsize element 44
 - fontstyle element 44
 - fontweight element 44
 - pencolor element 44
 - penstyle element 44
 - penwidth element 44
 - rectangle 45
 - rotation element 44
 - strikeout element 44
 - textcolor element 44
 - textjustify element 44

- transparent element 44
 - underline element 44
 - unsupported 69
- window scripts
 - definition 41
 - onhide 42
 - onshow element 41
 - whileshowing 42
- WINDOWCREATE command 14
- WINDOWDELETE command 15
- WindowMaker, command line 21
- WINDOWMAKERCOMMANDFILE command 14
- WINDOWNAME command 15
- windowname element 105
- WINDOWRENAME command 15
- windows
 - deleting 17
 - importing 18
 - renaming 18
- windowstyle elements 39

X

- X element 34, 108

XML

- declaration 30
- extraneous elements 30
- file header 30
- XMLFILEPATH command 14

Y

- Y element 34, 108