# Contents

Contents ......................................................................................................................... 1  
Introduction .................................................................................................................... 8  
Scripting Enhancements ................................................................................................. 9  
  Introduction Autocomplete .......................................................................................... 9  
  Intellisense (Autosuggest) ....................................................................................... 9  
    Appearing .............................................................................................................. 9  
    Places where Auto correct can be used in Graphics ................................................... 9  
    Places where Auto Complete can be used in Objects: ............................................... 12  
  Position ..................................................................................................................... 13  
  Content ..................................................................................................................... 13  
  Icons that are displayed in AutoComplete. ................................................................. 13  
  Order ....................................................................................................................... 16  
  Navigating ................................................................................................................ 16  
  Typing ...................................................................................................................... 16  
  Accepting ................................................................................................................ 16  
  Cancelling .............................................................................................................. 17  
  Method Parameter Help ............................................................................................ 17  
  Line Numbers ......................................................................................................... 17  
  Red Error Squiggles ................................................................................................. 17  
  Tool Tip Help ......................................................................................................... 17  
  Color Indicators for Script Elements .................................................................... 17  
  Multi-level Undo and Redo ...................................................................................... 18  
  Line Numbers ......................................................................................................... 18  
  TRY ... CATCH ...................................................................................................... 19  
  New Error log script functions ................................................................................ 20  
  Show Graphic Function .......................................................................................... 20  
  Known Limitations ................................................................................................. 20  
  Alarm Severity Mapping ......................................................................................... 21
Severity Levels ......................................................................................................................... 21
Event Types ............................................................................................................................. 21
Global Configuration of Alarm and Event Mapping .................................................................... 22
Alarm Aggregation at Runtime ................................................................................................. 22
Alarm Aggregation Enable/Disable .......................................................................................... 25
Aggregating Alarm State Information ....................................................................................... 25
Configuring Alarm State Aggregation ....................................................................................... 25
Aggregation upon migration ....................................................................................................... 26
Monitoring Alarm State Information at Run Time ..................................................................... 27
Support for Severity in Runtime .............................................................................................. 28
Multi Galaxy Improvements ..................................................................................................... 29
Application Server Improvements .......................................................................................... 29
Silenced Alarms ....................................................................................................................... 29
UDO Standard Object Description ......................................................................................... 29
Remove Platform Breach Message ........................................................................................... 30
Protecting Graphic Symbols ..................................................................................................... 30
Protecting Objects On Export .................................................................................................. 30
Element or Function Description ............................................................................................. 30
Exporting Objects As Protected ............................................................................................... 31
OPC Object Improvement ........................................................................................................ 33
Adjusted Execution Times ......................................................................................................... 34
Engine Execution Times ............................................................................................................ 34
DI Object Execution Times ........................................................................................................ 34
Checkpoint Files ....................................................................................................................... 34
Changes To Hostname References On Platforms .................................................................... 34
Platform’s Localhost Is Valid: ................................................................................................. 34
Historian Localhost In AppEngine Is Valid: ............................................................................. 34
For Historical Connections to A2ALMDB Localhost is Valid: ............................................... 35
Cancel buttons removed that were not functional ...................................................................... 36
Migration and import operations: .............................................................................................. 36
Deployment ............................................................................................................................... 37
Changed text for redundancy settings on platform ................................................................. 38
Script and Client Controls Improvement ................................................................................ 39
Alarm Manager Enhancement ................................................................................................... 39
Information Technology Compliance ........................................................................................................... 41
Machine Hostname Rename After Installation .......................................................................................... 41
User Account Control (UAC) Fully Supported ............................................................................................ 41
Application Server SQL Express .................................................................................................................. 41
ArchestrA User ........................................................................................................................................... 41
Support for Network-based Installation ...................................................................................................... 41
Silent Installation ......................................................................................................................................... 41
Response files for silent install .................................................................................................................... 42
New Icon Set Alarm Management Tools .................................................................................................... 42
Historian Architecture for Alarm and Event Support .................................................................................. 43
Sending Alarms/Events to Historian ............................................................................................................ 43
Event Historization Service (EHZ) ............................................................................................................... 44
Global Filtering Capability of Alarms & Events .......................................................................................... 44
Configuration alarm control to show severity .............................................................................................. 45
New Animations for ArchestrA Graphics ...................................................................................................... 46
  Configuring Point Animation ...................................................................................................................... 46
  Using Alarm Border Animations ................................................................................................................ 47
  States of the alarm border animation ......................................................................................................... 48
Alarm Styles .................................................................................................................................................. 49
Icons For Alarm Borders ............................................................................................................................. 50
Requirements of Alarm Border Animations ................................................................................................ 51
Behavior of Alarm Border Animations ......................................................................................................... 51
Configuring Optional Alarm Border Animation Characteristics ............................................................... 51
Configuring Alarm Border Animation ......................................................................................................... 53
  Configure An Alarm Border animation for a symbol using field attributes or objects ............................. 54
Using Polar Star Symbols ............................................................................................................................ 55
Sweep Angle animation ................................................................................................................................. 57
  How to configure a sweep angle animation ............................................................................................... 57
Graphic Performance Index (GPI) ................................................................................................................ 59
Trend Pen Animation ................................................................................................................................. 60
  Trend overview and introduction .............................................................................................................. 60
  Trend Pen Example ................................................................................................................................. 61
Style Libraries ............................................................................................................................................. 61
  Managing Style Libraries .......................................................................................................................... 61
## Contents

Creating Composite ArchestrA Symbols with Symbol Wizard ................................................................. 83
Embedding Composite Symbols into an Application ................................................................................... 83
Creating Multiple Configurations of a Symbol ............................................................................................ 84
Understanding Visual and Functional Symbol Configurations .................................................................. 85
Visual Symbol Configurations .................................................................................................................... 85
Functional Symbol Configurations ............................................................................................................ 85
Different Symbol Wizard Work Flows ......................................................................................................... 86
Using the Symbol Wizard ............................................................................................................................ 86
Understanding Choice Groups and Choices ............................................................................................... 88
Defining Symbol Configuration Rules .......................................................................................................... 90
Designing Symbols with the Symbol Wizard ............................................................................................... 92
Creating Symbol Choice Groups, Choices, and Options .............................................................................. 93
Assigning Symbol Configuration Rules ....................................................................................................... 93
Updating Symbol Layers ............................................................................................................................. 94
To update a layer rule..................................................................................................................................... 95
Associating Configuration Elements to Symbol Layers ............................................................................... 95
Associating Graphic Elements to Symbol Layers ......................................................................................... 96
Associating Custom Properties to Symbol Layers ....................................................................................... 96
Associating Named Scripts to Symbol Layers ............................................................................................ 97
Verify Symbol Configurations .................................................................................................................... 97
Using Multi-Configuration Symbols in an Application .............................................................................. 99

### Situational Awareness Library

Introduction and Philosophy ......................................................................................................................... 102
General ......................................................................................................................................................... 103
Meters ......................................................................................................................................................... 103
Meter Digital PV ........................................................................................................................................ 103
Optimal Range .......................................................................................................................................... 103
Normal Operating Range vs. Full Range ...................................................................................................... 104
Auto-Scaling .............................................................................................................................................. 104
Auto Ranging ............................................................................................................................................. 105
Graphic toolbox overview symbols ........................................................................................................... 105
Using SA Symbols ...................................................................................................................................... 106
In its most simple form the wizard is configured for minimal implementation: ...................................... 106
Symbol Mode Advanced ............................................................................................................................. 108
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphic Symbols Situational Awareness Library</td>
</tr>
<tr>
<td>Control Meter</td>
</tr>
<tr>
<td>Analyzer Object</td>
</tr>
<tr>
<td>Deviation Symbol</td>
</tr>
<tr>
<td>Flow Meter Symbol</td>
</tr>
<tr>
<td>Level Object Symbol</td>
</tr>
<tr>
<td>Level comparison Symbol</td>
</tr>
<tr>
<td>Multi-Level Meter Symbol</td>
</tr>
<tr>
<td>Miscellaneous Meter Symbol</td>
</tr>
<tr>
<td>Pressure Meter Symbol</td>
</tr>
<tr>
<td>RPM Indication Symbol</td>
</tr>
<tr>
<td>Target Meter Symbol</td>
</tr>
<tr>
<td>Temperature Meter Symbol</td>
</tr>
<tr>
<td>KPI Bar Symbol</td>
</tr>
<tr>
<td>Invisible Profile Meter Symbol</td>
</tr>
<tr>
<td>Flow Tracker Symbol</td>
</tr>
<tr>
<td>L3 Data object Symbol</td>
</tr>
<tr>
<td>OP Bar Output Symbol</td>
</tr>
<tr>
<td>Loopblock Symbol</td>
</tr>
<tr>
<td>Meter Voting Symbol</td>
</tr>
<tr>
<td>3 Spoke Polar Star Symbol</td>
</tr>
<tr>
<td>4 Spoke Polar Star Symbol</td>
</tr>
<tr>
<td>6 Spoke Polar Start Symbol</td>
</tr>
<tr>
<td>8 Spoke Polar Star Symbol</td>
</tr>
<tr>
<td>Agitator Symbol</td>
</tr>
<tr>
<td>Feeder Symbol</td>
</tr>
<tr>
<td>HV/LV Switch</td>
</tr>
<tr>
<td>Heat Exchanger Symbol</td>
</tr>
<tr>
<td>Pump Blower Rotary Valve Symbol</td>
</tr>
<tr>
<td>Rotating Equipment Symbol</td>
</tr>
<tr>
<td>Gap Control Valve Symbol</td>
</tr>
<tr>
<td>Valve and Damper Symbol</td>
</tr>
<tr>
<td>Parallel Control Valve Symbol</td>
</tr>
<tr>
<td>Vessel Symbol</td>
</tr>
</tbody>
</table>
Direction Arrow Symbol ...................................................................................................................... 189
Multiple Equipment Object .................................................................................................................. 191
Meter Connector Symbol ..................................................................................................................... 194
INTRODUCTION

Wonderware System Platform 2014 has some significant new content. Obtaining the full value from this new content may require that you reconsider how you have approached and delivered HMI projects in the past.

For the Wonderware software Product Managers, it seems all very natural because we work for years on these first visions and then on the implementation of these new features.

We realize that for many of you this is the first time you have seen many of these new features, particularly the new Situational Awareness Library. So therefore we endeavored to create this document to assist in pointing out the new features and differences from the previous release.

This document is intended as comprehensive introductory overview of what has changed from a Product Management perspective. This is absolutely not intended to replace or supplement official documentation or detailed instructions of how to implement certain features. For that please refer to the official product documentation included with the released software and/or posted on the Wonderware WDN support site.

This document is intended to provide a quick and easy reference document regarding what is new with the System Platform 2014 release, and where you can find these features in the IDE.
SCRIPTING ENHANCEMENTS

INTRODUCTION AUTOCOMPLETE

QuickScript autocomplete incorporates several features for use while authoring object and client scripts:

- Resolves an Attribute reference when you type a generic object name, such as "me." Run-time attributes appear in an autocomplete list box.
- Provides method parameter help in an autocomplete list box including context-specific suggestions covering definitions, keywords, script elements, and programmatic constructs such as try ... catch or while ... endwhile.
- Automatic word completion of Attribute references, methods, programmatic constructs, and other script elements.

These features serve as convenient documentation of method parameters and scripting syntax as well as an enhanced input method.

Autocomplete displays a context-sensitive list of options for script elements, keywords, object and attribute names, and programmatic constructs. You can

INTELLISENSE (AUTOSUGGEST)

While typing, the editor suggests options based on the language keywords, the names of Script Functions, and the attributes of the object that the script is attached to.

APPEARING

The autocomplete dropdown will appear when the user types any alphanumeric character, period, or space following a non-space. It will also appear if the user types Ctrl-space.

PLACES WHERE AUTO CORRECT CAN BE USED IN GRAPHICS

CUSTOM PROPERTIES, ME. REFERENCES.
SYMBOl SCriPTS:

INTOUCH REFERENCES

Choose IT app for the right namespace
CLIENT CONTROLS:

SCRIPTS:
EXPRESSIONS IN ANIMATIONS:

PLACES WHERE AUTO COMPLETE CAN BE USED IN OBJECTS:

SCRIPTS

DECLARATIONS:
**POSITION**

The autocomplete dropdown will appear below the current caret location (unless there is no room below, in which case above) and to the right of the caret (unless there is no room in which case it will shift left to avoid the right border of the editor window).

**CONTENT**

In general, the dropdown content of the autocomplete box should include as items any keyword, namespace, .Net type name, method name, property, field, symbol name, graphic element name, custom property, properties of graphic elements and symbols, graphic element methods, client control properties and methods, client script functions, third party control name, tagname, attribute name, attribute extension, attribute property, Me, MyEngine, MyPlatform, MyHost, MyContainer, MyArea, script alias etc. that is syntactically valid at the current caret location. It should not include anything that is not valid at the current caret location. Validity is a function of the scripting language and will not be described here, except that there will additional checking for attribute categories so that config-time only attributes do not appear in the autocomplete list. An item should not contain any dots (periods). If an attribute or namespace eligible for autocomplete contains periods, it will be shown in autocomplete dotted segment at a time.

Each item in the autocomplete box list should have an appropriately evocative icon accompanying it.

With the exception of keywords, the casing of the dropdown content of the autocomplete box is preserved. Keywords are all lowercase except for relative references (Me, MyEngine, MyPlatform, MyHost, MyContainer, and MyArea) which are mixed case.

**ICONS THAT ARE DISPLAYED IN AUTOCOMPLETE.**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon1" alt="MxBoolean attribute icon" /></td>
<td>MxBoolean attribute</td>
</tr>
<tr>
<td><img src="icon2" alt="MxInteger attribute icon" /></td>
<td>MxInteger attribute</td>
</tr>
<tr>
<td><img src="icon3" alt="MxFloat attribute icon" /></td>
<td>MxFloat attribute</td>
</tr>
<tr>
<td><img src="icon4" alt="MxDouble attribute icon" /></td>
<td>MxDouble attribute</td>
</tr>
<tr>
<td><img src="icon5" alt="MxString attribute icon" /></td>
<td>MxString attribute</td>
</tr>
<tr>
<td><img src="icon6" alt="MxTime attribute icon" /></td>
<td>MxTime attribute</td>
</tr>
<tr>
<td><img src="icon7" alt="MxElapsedTime attribute icon" /></td>
<td>MxElapsedTime attribute</td>
</tr>
<tr>
<td>MxReference attribute</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>MxStatus attribute</td>
<td></td>
</tr>
<tr>
<td>MxDataTypeEnum attribute</td>
<td></td>
</tr>
<tr>
<td>MxSecurityClassification attribute</td>
<td></td>
</tr>
<tr>
<td>MxDataQuality attribute</td>
<td></td>
</tr>
<tr>
<td>MxQualifiedEnum attribute, CustomEnum</td>
<td></td>
</tr>
<tr>
<td>MxQualifiedStruct attribute</td>
<td></td>
</tr>
<tr>
<td>MxInternationalizedString attribute</td>
<td></td>
</tr>
<tr>
<td>.Net Method</td>
<td></td>
</tr>
<tr>
<td>.Net Property</td>
<td></td>
</tr>
<tr>
<td>.Net Field or Variable</td>
<td></td>
</tr>
<tr>
<td>.Net Namespace</td>
<td></td>
</tr>
<tr>
<td>.Net Struct</td>
<td></td>
</tr>
<tr>
<td>.Net Class</td>
<td></td>
</tr>
<tr>
<td>.Net Interface</td>
<td></td>
</tr>
<tr>
<td>.Net Enumeration</td>
<td></td>
</tr>
<tr>
<td>.Net Enum Value</td>
<td></td>
</tr>
</tbody>
</table>
### QuickScript Keyword

Contains object name, or any partial attribute name like a UDA, field attributes, primitive (example `me.alarm` will show blue ball for alarm) that has a dot in the middle of the name, or any attribute of mx type MxNone (example input/output extension attribute `WriteValue`) or if there are several type choices among objects and attributes. In other words, if the attribute type cannot be exactly or unambiguously determined, this icon will be used.

**Note:** Please note that, in case, if name collision appear in Bitfields properties then the icon displayed will be the icon of MxType of the attribute rather than the blue icon appear otherwise.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Rectangle" /></td>
<td>Rectangle</td>
</tr>
<tr>
<td><img src="image" alt="Rounded Rectangle" /></td>
<td>Rounded Rectangle</td>
</tr>
<tr>
<td><img src="image" alt="Line" /></td>
<td>Line</td>
</tr>
<tr>
<td><img src="image" alt="Horizontal or Vertical Line" /></td>
<td>Horizontal or Vertical Line</td>
</tr>
<tr>
<td><img src="image" alt="Text" /></td>
<td>Text</td>
</tr>
<tr>
<td><img src="image" alt="Ellipse" /></td>
<td>Ellipse</td>
</tr>
<tr>
<td><img src="image" alt="Curve" /></td>
<td>Curve</td>
</tr>
<tr>
<td><img src="image" alt="Closed Curve" /></td>
<td>Closed Curve</td>
</tr>
<tr>
<td><img src="image" alt="Button" /></td>
<td>Button</td>
</tr>
<tr>
<td><img src="image" alt="Polygon" /></td>
<td>Polygon</td>
</tr>
<tr>
<td><img src="image" alt="Polyline" /></td>
<td>Polyline</td>
</tr>
<tr>
<td><img src="image" alt="Connect" /></td>
<td>Connect</td>
</tr>
<tr>
<td><img src="image" alt="Image" /></td>
<td>Image</td>
</tr>
<tr>
<td><img src="image" alt="Group or Embedded Symbol" /></td>
<td>Group or Embedded Symbol</td>
</tr>
<tr>
<td><img src="image" alt="Alarm Control" /></td>
<td>Alarm Control</td>
</tr>
<tr>
<td><img src="image" alt="Edit box" /></td>
<td>Edit box</td>
</tr>
<tr>
<td><img src="image" alt="Arc" /></td>
<td>Arc</td>
</tr>
<tr>
<td><img src="image" alt="Pie" /></td>
<td>Pie</td>
</tr>
<tr>
<td><img src="image" alt="Chord" /></td>
<td>Chord</td>
</tr>
<tr>
<td><img src="image" alt="Circle" /></td>
<td>Circle</td>
</tr>
<tr>
<td><img src="image" alt="Status" /></td>
<td>Status</td>
</tr>
<tr>
<td>Radio buttons</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>Checkbox</td>
<td></td>
</tr>
<tr>
<td>Editbox</td>
<td></td>
</tr>
<tr>
<td>Combobox</td>
<td></td>
</tr>
<tr>
<td>Calendar</td>
<td></td>
</tr>
<tr>
<td>DatePicker</td>
<td></td>
</tr>
<tr>
<td>Listbox</td>
<td></td>
</tr>
</tbody>
</table>

**ORDER**

In general, the autocomplete box is ordered alphabetically, although keywords and dimmed variables will float to the top of the list given their importance to the script, each in alphabetical order.

**NAVIGATING**

While the autocomplete box is showing, the keyboard up and down arrow keys will serve to move the autocomplete selection up and down respectively. Similarly, the keyboard page-up and page-down keys will serve to move up and down an autocomplete page worth of content at a time respectively. In addition, the user may use the mouse to scroll and select items in the autocomplete box. A single-click on an item selects that item but does not insert the item into the editing session nor does it close the autocomplete box.

**TYPING**

As the user continues typing alphanumeric characters while the autocomplete box is visible, items in the list will be winnowed away where it the typed characters no longer prefix them. Typing the backspace key will erase the last character of the prefix and so the cause the winnowed away items to reappear.

**ACCEPTING**

If the user double-clicks an item in the autocomplete box, or types the Enter or Tab key, the highlighted item will be inserted at the caret (without an end-line or tab appended). If the user types a space, period, comma, open or closed parenthesis, or other punctuation, the highlighted item in the autocomplete box will be inserted at the caret WITH the aforementioned character appended.
CANCELING

While the autocomplete box is showing the user may cancel it by clicking away from it with the mouse or typing the Esc key. The user can also cancel the autocomplete box by typing left or right arrow keys which will also move the editor caret left or right.

METHOD PARAMETER HELP

If the caret is located between the open and close parentheses of a function call, a parameter help box will appear below the caret with the method name and names and types of the arguments. The argument at the current caret location will show in bold. If a method is overloaded, up and down arrows on the parameter help box will allow the user to peruse the various overloads.

LINE NUMBERS

The script editor will display line numbers in the left margin. Up to 4 digits will fit in the left margin when the script editor is not zoomed. The line number may appear clipped for scripts longer than 9999 lines or when the script editor is zoomed. To go to a specific line in the script you can use the right-click menu Go To function (this is an existing function, not new to Apollo).

RED ERROR SQUIGGLES

The script editor will display a red error squiggly underline under verification errors in the script text. The squiggly will appear after about 2 and a half seconds of keyboard inactivity. In some cases, but not all, more than one error will be underlined. This is not always possible because some errors prevent the compiler from continuing forward after the error. When hovering over the error with the mouse, the error text message will appear as a tool tip. The error text message should be identical to that shown when pressing the red check verification button.

TOOLTIP HELP

In addition to the error tool tips described in the previous section, the script editor will also display a tool tip when hovering over a variable name in the script. The variable’s name and type will show in the tool tip.

COLOR INDICATORS FOR SCRIPT ELEMENTS

The QuickScript .NET editor uses different text colors to identify different script elements. The following table shows the text colors associated with script elements.
MULTI-LEVEL UNDO AND REDO

You can selectively undo a history of changes to your script. The number of changes that can be undone is limited only by the amount of available memory.

An undone change can be redone. Redo mirrors undo changes.

A single undo typically is comprised of sequences of typing or deleting, which can be interrupted by interaction with an autocomplete list or by moving the cursor with the mouse, or by clicking elsewhere in the script.

All pending undo and redo actions will be lost if you close the object editor, switch to another script within the object editor, or switch among Startup, OnScan, Execute, OffScan, and Shutdown scripts.

LINE NUMBERS

The script editor displays line numbers in the left margin.

- Line numbers of up to four digits will display when the script editor is not zoomed.
- The line number may appear clipped for scripts longer than 9999 lines or when the script editor is zoomed.
- Use the right-click context menu Go To function to go to a specific line in the script.

<table>
<thead>
<tr>
<th>Element</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>Blue Syntax highlighted while typing.</td>
</tr>
<tr>
<td>Comments (both single line and multi-line)</td>
<td>Green Syntax highlighted while typing.</td>
</tr>
<tr>
<td>Strings</td>
<td>Purple Syntax highlighted while typing.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function names, numeric constants, operators, semicolons, dim variables, alias variables, and so on</td>
<td>Black See descriptions for Attribute names and Reserved words.</td>
</tr>
<tr>
<td>Attributes names</td>
<td>Maroon, bold face</td>
</tr>
<tr>
<td>Reserved words</td>
<td>Red, non-bold face</td>
</tr>
</tbody>
</table>
TRY ... CATCH

TRY ... CATCH provides a way to handle some or all possible errors that may occur in a given block of code, while still running rather than terminating the program. The TRY part of the code is known as the *try block*. Deal with any exceptions in the CATCH part of the code, known as the *catch block*.

The general format for TRY ... CATCH is as follows:

TRY
[try statements] ‘guarded section
CATCH
[catch statements]
ENDTRY

Where:

*tryStatements*
Statement(s) where an error can occur. It can be a compound statement. The *tryStatement* is a guarded section.

*catchStatements*
Statement(s) to handle errors occurring in the associated Try block. It can be a compound statement.

**Note:** Statements inside the Catch block may reference the reserved ERROR variable, which is a .Net System.Exception thrown from the Try block. The statements in the Catch block run only if an exception is thrown from the Try block.

TRY ... CATCH is executed as follows:

1) Run-time error handling starts with TRY. Put code that might result in an error in the try block.
2) If no run-time error occurs, the script will run as usual. Catch block statements will be ignored.
3) If a run-time error occurs, the rest of the try block does not execute.
4) When a run-time error occurs, the program immediately jumps to the CATCH statement and executes the catch block. The simplest kind of exception handling is to stop the program, write out the exception message, and continue the program.

**Example:**
```plaintext
dim command = new System.Data.SqlClient.SqlCommand;
dim reader as System.Data.SqlClient.SqlDataReader;
try
    command.Connection.ConnectionString = "Integrated Security=SSPI";
    command.CommandText="select * from sys.databases";
    command.Connection.Open();
    reader = command.ExecuteReader();
    while reader.Read()
        me.name = reader.GetString(0);
        LogMessage(me.name);
    endWhile;
catch
    LogMessage(error);
```
endtry;
if reader <> null and not reader.IsClosed then
 reader.Close();
endif;
 then
 command.Connection.Close();
endif;

NEW ERROR LOG SCRIPT FUNCTIONS

LogMessage was existing but there was always a wish to be able to log different messages to the logger so we added:

LogTrace

LogWarning

LogError

LogCustom

SHOW GRAPHIC FUNCTION

Show Graphic library is part of Auto Complete.

KNOWN LIMITATIONS

There are several areas where Auto Complete can suggest something that is not very useful. For example the .Net client parser can bring up functions just as .Net does. These functions are not are not valid and are marked as such. In a later release we will have multiple versions of the parser, but in this release that issue will not be addressed.
SEVERITY LEVELS

Application Server has four Severity levels for Alarms. These can be configured to correspond to specific ranges of Alarm Priority. You can also configure each Severity level to enable or disable historization in the Alarm Database.

The default mapping for Alarm Severity is as follows:

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
<th>Historize</th>
<th>From Priority Range</th>
<th>To Priority Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Critical</td>
<td>Y</td>
<td>1</td>
<td>250</td>
</tr>
<tr>
<td>2</td>
<td>High</td>
<td>Y</td>
<td>251</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>Medium</td>
<td>Y</td>
<td>501</td>
<td>750</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
<td>Y</td>
<td>751</td>
<td>999</td>
</tr>
</tbody>
</table>

EVENT TYPES

There are also 3 event types, which can also be configured to enable or disable historization:

The default mapping for Event Types is as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Historize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Application</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>User</td>
<td>Y</td>
</tr>
</tbody>
</table>
GLOBAL CONFIGURATION OF ALARM AND EVENT MAPPING

You can configure Alarm and Event mapping for the entire Galaxy in the IDE. On the IDE main menu, click **Galaxy**, then click **Configure**, then click **Alarm Priority Mapping**. The **Alarm and Event Priority Mapping and Historization** dialog appears.

Alarm Severties can be mapped to priorities in ascending or descending order. For example, severity 1 can map to priority range 1-250 or it can map to priority range 999-751.

Severity 4 ends at 999 by default, but this is not required to be the end-of-range number. For example, severity 4 can be configured to end at 900, leaving all alarms of priority above 900 unmapped to any severity level. If there are any gaps in the priority range mapping, alarms with priorities in those gaps correspond to severity 0, not mapped.

ALARM AGGREGATION AT RUNTIME

Every alarm has two status attributes: **InAlarm** and **Acked**. Together, these constitute four possible states for any given alarm, as shown in the table below:

<table>
<thead>
<tr>
<th>InAlarm</th>
<th>Acked</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FALSE</strong></td>
<td><strong>FALSE</strong></td>
</tr>
<tr>
<td><strong>FALSE</strong></td>
<td><strong>TRUE</strong></td>
</tr>
<tr>
<td><strong>TRUE</strong></td>
<td><strong>FALSE</strong></td>
</tr>
<tr>
<td><strong>TRUE</strong></td>
<td><strong>TRUE</strong></td>
</tr>
</tbody>
</table>

These are the states that you see in the Alarm Client grid controls (the ArchestrA Alarm Client and the ActiveX Alarm Viewer).

The relation between these states can be seen in the following State Diagram:
ACK_RTN is sort of a “ground state” for alarms. When an alarm becomes active, the state changes to UNACK_ALM. Then one of two things can happen: If the alarm is then acknowledged, then the state goes to ACK_ALM because the alarm is still in the alarmed state. If on the other hand the alarm returns to normal before it is acknowledged, it goes to the UNACK_RTN state. When the alarm is both returned to normal and acknowledged, it again returns to the ACK_RTN state.

When you display the **current alarms** (also known as Summary List in the ActiveX Alarm Viewer), you only see the three states UNACK_ALM, ACK_ALM, and UNACK_RTN. That is because these are the states that need attention – something needs to be done to return the alarm to normal condition and to acknowledge that the alarm has been seen by an operator. ACK_RTN does not need any attention, so it does not appear on the **current alarms** display.

Also, each Alarm can be **ENABLED**, **SILENCED**, or **DISABLED**. The meanings are as follows:

- **Enabled**: Alarm is fully functional. When the condition becomes TRUE the alarm state becomes active and everybody can see it on the InTouch display screen.
- **Silenced**: Alarm is fully functional. When the condition becomes TRUE the alarm state becomes active. However, it does NOT get displayed on the screen. We ONLY record it in the Alarm Database. (In other words, this is a Historization-only setting.)
- **Disabled**: Alarm is forced OFF. Even when the condition becomes TRUE, the alarm state remains FORCED to the ACK_RTN state. It is never waiting for an acknowledgement.

The Enable/Silence/Disable setting can be done at the Area level, Object level, and individual Alarm level.

Alarm Aggregation is a way of summarizing multiple alarms on a complex object to make it easy to identify which of several objects has alarms that need attention, and to compare objects to see which set of alarms is more important than another – that is, which object has the “most urgent” alarms.
We only aggregate alarms in the UNACK_ALM, ACK_ALM, and UNACK_RTN states. (As mentioned above, ACK_RTN alarms do not need attention.)

Alarm Aggregation ranks alarms by four criteria, in the following order:

- Alarm Mode: Enabled is more urgent than Silenced
- InAlarm: TRUE precedes FALSE (that is, Alarmed precedes Returned to Normal)
- Acked: FALSE precedes TRUE (that is, Unacknowledged precedes Acknowledged)
- Severity Level: 1 is most important, 4 is least important.

The “most urgent” alarm on an AppObject is the one at the top of this sorting order. If two or more alarms on an Object have the same “most urgent” combination of Alarm Mode, InAlarm status, Acked status, and Severity, they have the same ranking for Alarm Aggregation.

On each AppObject, we summarize the statuses of all alarms on the Object and its descendants using five Attributes.

- AlarmMostUrgentSeverity: Severity 1-4 of the “most urgent” alarm(s)
- AlarmMostUrgentMode: Alarm Mode of the “most urgent” alarm(s)
- AlarmMostUrgentInAlarm: TRUE/FALSE flag indicating whether the alarm is presently in the alarm state, for the “most urgent” alarm(s)
- AlarmMostUrgentAcked: TRUE/FALSE flag indicating whether the alarm has been acknowledged for the “most urgent” alarm(s)
- AlarmCntsBySeverity: Array of counts of all alarms that are InAlarm or waiting to be Acked, separated by Severity. Element 1 contains the count of alarms of Severity 1, element 2 contains the count of alarms of Severity 2, etc.

If no alarms are configured on an Object or its descendants, or if Alarm Aggregation is disabled, these attributes have the following default values:

- AlarmMostUrgentSeverity = 0
- AlarmMostUrgentMode = Alarm Mode of the Object itself
- AlarmMostUrgentInAlarm = FALSE
- AlarmMostUrgentAcked = TRUE
- AlarmCntsBySeverity = {0, 0, 0, 0}

These Attributes summarize the status of alarms on the Object itself AND on all descendant Objects. That is, if an Object has a child Object in the Model View hierarchy, the Aggregation values represent all alarms on the parent Object PLUS all the alarms on the child Object. For an Area, the Aggregation values represent all alarms on the Area object itself, all AppObjects assigned to that Area, and all sub-Areas of that Area.
Alarm Aggregation also applies to individual Attributes. This applies only to Analog Field Attributes. An Analog Field Attribute can have several different types of alarms, and multiple alarms configured on it: Level Alarms (HiHi, Hi, Lo, LoLo), Deviation Alarms (MinorDev and MajorDev), Rate of Change Alarms (ROC Rising, ROC falling), and Bad Quality Alarms. Alarm Aggregation summarizes all the alarms on the Analog Field Attribute, whether it has a single alarm or several alarms configured.

**Alarm Aggregation Enable/Disable**

Alarm Aggregation can be enabled or disabled at the Area level. By default, Alarm Aggregation is enabled. It can be configured at design time in the IDE by checking or unchecking the Alarm Aggregation Cmd checkbox in the General tab of the Area editor dialog. This setting can be Locked, so it propagates to all derived templates and instances.

This Attribute cannot be changed at runtime.

If an Area has Alarm Aggregation disabled, this does not affect sub-Areas. Sub-Areas may aggregate alarms if they have Alarm Aggregation enabled and the parent Area has Alarm Aggregation disabled.

**Aggregating Alarm State Information**

Alarm aggregation provides an efficient way to identify whether any alarms on an object are currently in the InAlarm state, the overall status of the most important of those alarms, and how many alarms are active at each level of alarm severity and at each level of containment. This makes it possible to follow a trail from one level to the next to find the underlying cause of a complex object’s alarms.

You can view alarm aggregation statuses in run-time clients such as Object Viewer. You can model alarm aggregation in Managed InTouch applications by using animations such as the alarm border animation with Situational Awareness Library symbols or with ArchestrA symbols.

**Configuring Alarm State Aggregation**

Configuring alarm state aggregation consists of normal alarm configuration procedures plus an added step of enabling the aggregation feature on each relevant Area object.

**To configure alarm state aggregation:**

1. Accept the default settings or configure alarm severities for each of the four severity levels you want to aggregate. This is a global configuration.
2. Configure an object with one or more alarms. All alarms configured on all objects will be aggregated. Aggregation on an attribute-by-attribute basis applies only to Analog Field Attribute alarms.
3. Set the `AlarmModeCmd` of the object to enabled or silenced, but not to disabled.
4. Set the `AlarmModeCmd` of at least one alarm on the object to enabled or silenced, but not to disabled.
5. Set the `AlarmModeCmd` of the Area object to enabled or silenced, but not to disabled.
6. Check the `Enable Alarm Aggregation` checkbox on the Area object editor to enable alarm aggregation. This sets the value of the `AlarmAggregationStateCmd` attribute to True.
AGGREGATION UPON MIGRATION

Migrated and new Galaxies will have alarm aggregation enabled by default, since the performance impact is minimal, only in extreme cases this is an option or for other reasons than performance.
MONITORING ALARM STATE INFORMATION AT RUN TIME

You can obtain run-time alarm state information using clients such as Object Viewer or InTouch Tag Viewer. You can also create InTouch applications with Situational Awareness Library symbols configured with alarm border animations that display alarm state aggregation status information.

For example, you have configured a galaxy with ProcessArea1 and ProcessArea2 contained in Plant1. You have two tanks in each process area, with alarms configured on each tank. You can watch the aggregated alarm severity counts at each containment level in Object Viewer.

You can visualize the same objects in an InTouch application, using the Alarm Client along with the alarm border animation to represent the aggregated alarms at each level of containment.
SUPPORT FOR SEVERITY IN RUNTIME

Adding support for Severity in runtime, and being able to change the priority of Platform alarms and or disable them.
Multi Galaxy Improvements

- Added custom serialization to MxDataConsumer/MxDataProvider. This improves CPU performance across the board for better overall system performance, and has the side effect of allowing more data to be transmitted per second (dependent on the size of your system, how many processors, available memory, etc.).
- Added asynchronous capability to the MxDataConsumer’s registration/unregistration API. (Prevents delays in data receipt in multi-system scenarios).
- Fixed an issue that was causing the Matrikon OPC client to never complete registration in multi-galaxy scenarios.
- Added compressions and optimization for low bandwidth networks.
- Added support for secured writes.
- Added Support for enumeration, and arrays.
- Reduced the chatter between nodes, low bandwidth networks.
- Increased the throughput from 20,000 per sec to 70,000 per sec
- Being able to unpair a Galaxy if the connection is lost or node is not there anymore

Application Server Improvements

Silenced Alarms

According to international alarm standards when an alarm is silenced and it makes state transitions this should still be logged in the historical database, so silence is the occurrence where the detection logic does not get switch off but the alarm is taken of the alarm summary display. The new Alarm border animation will clearly indicate this:

Note: this will only work with the new Application Server alarming and storage, the ALMDB logger still has the old behavior and will not log the state changes on silence.

UDO Standard Object Description

The standard text in UDO objects was removed so these do not fill up the ALMDB.
REMOVE PLATFORM BREACH MESSAGE

Platform license count breach messages have been removed. It was confusing for end-users since it was not enforced and so did not do anything. In the future we will change our licensing around this.

PROTECTING GRAPHIC SYMBOLS

PROTECTING OBJECTS ON EXPORT

Protect symbols and derived templates by flagging the objects as protected in the galaxy database. Protected symbol and template behavior is similar to that of a base template. This option is available through a specialized form of the Export Object(s) operation. System Integrators and other system designers can use this functionality to protect objects designed in a master galaxy that are intended for use in production galaxies or galaxies on run-time nodes.

Protecting an object on export does not change the object in the galaxy from which it was exported. Protection is effective only on import of protected objects. Specific behavior is described as follows:

ELEMENT OR FUNCTION DESCRIPTION

*Base* templates are protected by default and cannot be checked out, edited or renamed. *Protected* templates and symbols cannot be checked out, edited, or renamed, but can be deleted. A template derived from a protected template is unprotected.

Protected templates and symbols are marked in the Template Toolbox, Graphic Toolbox, or Application views (Model, Deployment, or Derivation View) with a gold-colored padlock icon. Ancestor objects are protected in the exported .aaPKG file when a child object is protected. Protection is effective on import of protected objects. A protected object retains its protected status even when exported using the standard *Export Object(s)* workflow.

Instances cannot be protected. Only templates and symbols can be protected. The new export workflow is similar as that used for unprotected objects except that you now can select the option to export *As Protected Object(s).* Graphics Symbols and client controls directly or indirectly embedded in a protected graphic are also protected.

Protected symbols can be opened in the Graphic Editor as read-only.
EXPORTING OBJECTS AS PROTECTED

The export objects workflow is the same as for all objects with the exception of specifying the export of selected objects as protected.

To protect objects on export

1 In the Template Toolbox, Graphic Toolbox, or in the Application Views (Model, Deployment, or Derivation), select one or more templates or symbols to export.

2 On the Galaxy menu or context menu, click Export and then click As Protected Object(s). The Export Automation Object(s) dialog box appears.

3 In the Export Automation Object(s) dialog box, browse to a path and type a name for the exported file.

4 Click Save. The file is saved with the specified name and an .aaPKG extension.

5 When the export is complete, click Close. Now you can import the .aaPKG file into another existing Galaxy.

The UI to import packages was updated and by default it has a new option “Never overwrite existing object with a protected object”, this was done to avoid accidental import of a package into a standards Galaxy and mark the objects as protected, if this behavior is needed for example the intent is to overwrite unprotected objects the user has to make this choice by Un-checking this check box.
Template Protection Change Management
- Never overwrite an unprotected object with a protected object
Prior to this release: If you had to modify an existing alias to link to a different OPC tag in an OPC object there was no method of using the browser again the only way to do this is to delete the tag and add it again and then browse.

With this release: The browse button is always there:

The browser itself was updated to the newest layout and is resizable:
ADJUSTED EXECUTION TIMES

ENGINE EXECUTION TIMES
The engine execution times for new Galaxies were adjusted to 500ms. This change only applies to new galaxies. It does not apply to migrated existing galaxies.

DI OBJECT EXECUTION TIMES
DI object default scan group settings were adjusted to 250ms. This change only applies to new galaxies. It does not apply to migrated existing galaxies.

CHECKPOINT FILES
The frequency of writing a checkpoint file was changed from “scan cycle” default to 10 seconds. This change only applies to new galaxies. It does not apply to migrated existing galaxies.

CHANGES TO HOSTNAME REFERENCES ON PLATFORMS
In previous releases: When users created demo or other single node systems they had to enter the machine’s IP address or resolvable hostname in the platform, engine, and Historian configuration fields.

Beginning with this release: “localhost” is a valid local machine reference.

PLATFORM’S LOCALHOST IS VALID:

HISTORIAN LOCALHOST IN APPENGINE IS VALID:
FOR HISTORICAL CONNECTIONS TO A2ALMDB LOCALHOST IS VALID:

Note InTouch alarm database does not support this and only can leverage SA password and node name.
This only applies for the A2ALMDB new storage and Application Server generated alarms.
CANCEL BUTTONS REMOVED THAT WERE NOT FUNCTIONAL

MIGRATION AND IMPORT OPERATIONS:

Non-Cancellable operations. Changed the text from Cancel to Close for disabled state.

Non-Cancellable operations. At the end of operation, close button is enabled.
Cancellable operation. During operation, button is enabled and text says Cancel.

![Galaxy Dump](image1)

Cancellable operation. At the end of operation, button text is changed to Close from Cancel.

![Galaxy Dump](image2)

DEPLOYMENT

![Deploy](image3)
### CHANGED TEXT FOR REDUNDANCY SETTINGS ON PLATFORM

<table>
<thead>
<tr>
<th>Redundancy Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundancy message channel IP address:</td>
</tr>
<tr>
<td>Redundancy message channel port: 30001</td>
</tr>
<tr>
<td>Redundancy primary channel port: 30000</td>
</tr>
</tbody>
</table>

**To:**

<table>
<thead>
<tr>
<th>Redundancy Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local redundancy message IP Address:</td>
</tr>
<tr>
<td>Local redundancy message port: 30001</td>
</tr>
<tr>
<td>Local redundancy primary port: 30000</td>
</tr>
<tr>
<td>Store forward redundancy port: 32568</td>
</tr>
</tbody>
</table>
SCRIPT AND CLIENT CONTROLS IMPROVEMENT

Within the import mechanism a check was added to make sure the user gets flagged if they import a script function library or a client control.

Additionally, the affected objects are indicated to the user.

For Example:

![Import Client Control(s):](image)

ALARM MANAGER ENHANCEMENT

Starting with Microsoft Windows Vista, the operating system imposes "Session 0 Isolation" as a security enhancement. All Windows Services and associated programs are required to run in Session 0, and no GUI applications are allowed to run in Session 0. This is to protect important system programs from being hacked by malware running in a user's log-in session.

The consequence is that programs that previously ran in the same Windows Session -- such as Wonderware Application Server and InTouch WindowViewer -- now run in separate Windows Sessions. Alarms that are reported by the Galaxy are handled by the Session 0 instance of AlarmMgr, which is now different from the Console Session instance of AlarmMgr that handles InTouch alarms. A simple alarm query in an InTouch alarm display such as

```
\InTouch!$System \Galaxy!Area_001
```

is now serviced by two separate instances of AlarmMgr -- one running in the Console Session for InTouch, another running in Session 0 for the Galaxy.

When Windows Vista was released in 2006, Invensys made modifications to AlarmMgr so that all alarm queries for Galaxy alarms are directed to the instance of AlarmMgr in Session 0. However, to preserve backward compatibility with previous versions of Windows, the instances of AlarmMgr in both the Console Session and Session 0 retained the same name "AlarmMgr" when registering with the SuiteLink Name Service (SLSSVC.exe). This is a "discovery service" that one instance of AlarmMgr can use to make contact with another instance of AlarmMgr, either on a remote computer or on the same computer but in a different Session. SLSSVC.exe allows only ONE registration for each unique program name -- so if AlarmMgr.exe is launched in both the Console Session and Session 0 and both
of those register with the name "AlarmMgr", only the FIRST instance to start up is able to register successfully with SLSSVC.exe and the other instance is "invisible" to attempts to connect with it.

You may have noticed that if you start up InTouch Window Viewer on a computer BEFORE you start up the WinPlatform, the alarm displays are unable to show the Galaxy alarms. And if you start up the WinPlatform first and InTouch second, alarm displays on remote computers are able to show the Galaxy alarms, but not the InTouch alarms.

A similar problem occurs on an InTouch-only system where you configure Alarm DB Logger to run as a Windows Service. If Alarm DB Logger is running as an ordinary application, it can log the InTouch alarms -- but if it's running as a Service, it cannot access the InTouch alarms.

This inability to access alarms is because the second instance of AlarmMgr.exe that starts up (which contains the InTouch alarms) is unable to register successfully with SLSSVC.exe and as a result is invisible to attempts to connect from outside the Session.

If you inspect the SMC Logger output for when the WinPlatform Starts up and when InTouch View.exe starts up, you will see a message from the component AlarmMgr of the form

    Registering AlarmMgr with SLSSVC as "AlarmMgr"

In Wonderware Application Server 4.0 and InTouch 11.0 and later you will also see a message indicating whether this attempt was successful or if it failed. Failure occurs because SLSSVC.exe does not accept registration of two programs with the same name. But even with earlier versions of Application Server and InTouch, if you are running on Windows Vista or later, this problem occurs.

Starting with Wonderware Application Server 4.0 and InTouch 11.0, you can configure the AlarmMgr so it uses separate names for Session 0 and the Console Session -- the Session 0 instance of AlarmMgr registers with the name "AlarmMgr0" and the Console Session continues to register as "AlarmMgr". As a result BOTH instances of AlarmMgr are able to register successfully and are visible to external alarm clients. Also, AlarmMgr automatically handles your existing alarm queries to direct them to the appropriate sessions for Galaxy alarms and InTouch alarms.

Further details will be posted in a future TechNote.
INFORMATION TECHNOLOGY COMPLIANCE

MACHINE HOSTNAME RENAME AFTER INSTALLATION

ArchestrA Service Bus (ASB) and multi-galaxy communications were introduced in the previous System Platform 2012 R2 release. A side effect of this new technology was that the machine could not be renamed after installing the ArchestrA components.

New in the 2014 release: The machine can be renamed after installing ArchestrA components, but before running configurations or the IDE. Upon machine rename, and after reboot, the system will update the associated ASB services with the new Machine name. No additional user interaction is required for this.

USER ACCOUNT CONTROL (UAC) FULLY SUPPORTED

We used to require UAC to be turned off in order to work with our system, as of this release we have accommodated UAC and are able to function properly with UAC turned on this will lead to better IT compliance.

APPLICATION SERVER SQL EXPRESS

Adding support for SQL Server express in Application Server, that way a system that is smaller does not need a custom install of SQL, and everything can be installed from our Media. No limitations are stated, SQL server express on our media is SQL Server express 2012 SP1, 64 bit.

ARCHESTRA USER

The ArchestrA user service account can now be non-interactive, meaning this can be an account not permitted to logon to the Machine. It can also be a domain account, permitting the ArchestrA user service to be a domain controlled service account.

SUPPORT FOR NETWORK-BASED INSTALLATION

Support product to completely install from a network location.

SILENT INSTALLATION

Rather than embed the response files in the Install Guide, we update the documentation with links to folder location on the DVD where the response files reside.

The Response Files are automatically generated by the Build system based, which means they are automatically updated when any team changes the component / feature set for their product.

This will help create virtual appliances and deploy configurations automatically.
RESPONSE FILES FOR SILENT INSTALL

To support virtual appliances on the media there is a directory that has the response files to install the product selections, in the R2 release we documented them static, and if things changed they were outdated, now they get generated every time we build the product.

NEW ICON SET ALARM MANAGEMENT TOOLS
Events (including alarms as described here) are intercepted from Notification Distributor at each area level. They are sent to Event Historization Service and then HCAL to Historian, following the similar communication path as process data:

1. Events are stored in the new A2ALMDB database with the same schema as existing WWALMDB.
2. Events and process data follow the same route through HCAL to Historian.
3. Existing events message flow is not altered. In other words, events can still be sent to Alarm Manager and historized to WWALMDB.
4. Events and process data share the same historian configuration such as historian node, store/forward path etc. “Enable storage to historian” must be enabled to historize either process data or events.
5. Similar to process data, events historization also support store/forward, application engine fail-over and dual Historians.
EVENT HISTORIZATION SERVICE (EHZ)

The Event Historization Service is an ASB Service that provides the ability for clients to historize Events to a configured Historian. EHz can be hosted in-process or as a regular ASB service. EHz also provides connection pooling making sure that multiple connections to the same Historian are avoided.

GLOBAL FILTERING CAPABILITY OF ALARMS & EVENTS

Events intercepted at Notification Distributor can be filtered at Galaxy level for historization. The filter is configured through “Alarm Priority Mapping” under Galaxy configuration menu as shown below. Only checked items are historized.

Storage of the severity will be in the Historian logged within the column USR1 since this is not use by Application Server.
CONFIGURATION ALARM CONTROL TO SHOW SEVERITY
NEW ANIMATIONS FOR ARCHESTRA GRAPHICS

CONFIGURING POINT ANIMATION

After selecting point animation, a list of configurable points is retrieved from the graphic element based on the following conditions.

• If the graphic element is a multi-point graphic type (Line, HV/Line, Polyline, Curve, Polygon, Closed curve), animation control points appear on the graphic element in preview mode.

If the graphic element is not a supported multi-point graphic, then the top left X and Y coordinate of the graphic element is selected as the animation point.
In the case of an element group consisting of several symbol elements, the animation point is the top left corner of the rectangle around all grouped elements.
To configure point animation

1) Open the symbol in Symbol Editor.
2) Select a graphic element.
3) On the Special menu, click Edit Animations. The Edit Animations dialog box appears.
4) Click the Add Animation button to show a list of Visualization and Interaction animations.
5) Select Point from the Visualization animation list. The Point dialog box appears with a list of points and a preview of the points on the symbol. The list shows each point as a pair of X and Y fields to enter an expression or a reference that evaluates to a floating point value.
6) Select a point from the list of points. The selected point changes to orange in the preview of the symbol.
7) Enter an expression, constant, or reference in the Point field.
8) Repeat steps 6-7 to animate other points in the symbol.
9) Save your changes.

USING ALARM BORDER ANIMATIONS

Alarm border animation shows a highly visible colored border around a graphic element when alarm conditions occur. The color and fill pattern of the border indicates the severity and current state of the alarm.

Alarm border animation also shows a triangular indicator icon at the top left corner of the border. Alarm severity appears within the indicator icon as a number from 1 to 4. The indicator icon can be shown or hidden as a configurable option of Alarm border animation.

Together, the alarm border and indicator icon enable operators to quickly recognize alarm conditions.

Any non-embedded graphic can represent an alarm border animation.
Only the reference needs to be supplied to configure the alarm border. The Alarm border will auto configure and bind properly.

In Runtime a border will be shown for an active alarm.

The colors are defined with element styles as global definitions

**STATES OF THE ALARM BORDER ANIMATION**

Unacked, flashes, acked will be continues color, and unacked return as following:

- Unacked return
- Acked Active
- Silenced
ALARM STYLES

- Alarm Critical Unack
- Alarm High UNACK
- Alarm Medium UNACK
- Alarm Low UNACK
- Alarm Critical ACK
- Alarm High ACK
- Alarm Medium ACK
- Alarm Low ACK
- Alarm Critical RTN
- Alarm High RTN
- Alarm Medium RTN
- Alarm Low RTN
- Alarm Inhibited
- Alarm Supressed
- Alarm Shelved
ICONS FOR ALARM BORDERS

The images are also global defined and can be changed a 24 by 24 pixel image (PNG) is required.

In the alarm and event configuration this can be changed.

Colors of alarm borders are globally defined in the style library.
**REQUIREMENTS OF ALARM BORDER ANIMATIONS**

Alarm border animation can be applied to all types of symbols except embedded symbols and nested groups. Alarm border animation can also be applied to symbol elements and element groups.

**BEHAVIOR OF ALARM BORDER ANIMATIONS**

Alarm Border animation shows on a symbol based on the current state of the symbol’s aggregated alarm attributes. The appearance of the alarm border itself reflects the current alarm state and the user’s interaction with the alarm. If the user does not override the *Outline* property of the assigned Element Style, the default Alarm Border outline is rendered.

- **When an alarm is generated for a symbol attribute:** The alarm border and indicator icon animation appear around the Situation Awareness Library symbol’s graphic element with blinking.

- **When the user acknowledges an alarm condition:** The alarm border and indicator icon animation appear around the Situation Awareness Library symbol’s graphic element without blinking.

- **When the alarm value returns to normal after the user acknowledges the alarm condition:** The alarm border and indicator icon animation no longer appear on the Situation Awareness Library symbol’s graphic element.

- **When the alarm condition returns to normal without the user acknowledging the alarm:** The alarm border and indicator icon animation remain around the Situation Awareness Library symbol’s graphic element in a defined Return to Normal visual style.

- **When the user inhibits, suppresses, or disables an alarm:** The alarm border and indicator icon animation remain around the Situation Awareness Library symbol’s graphic element in a defined inhibited visual style.

- **When an alarm condition changes when Alarm Border animation is visible on a graphic element, animation updates to show the new alarm state. In the case of aggregation alarms, Alarm Border animation show the highest current alarm state within a set of aggregated alarms.**

**CONFIGURING OPTIONAL ALARM BORDER ANIMATION CHARACTERISTICS**

Users can complete the following optional tasks to configure the appearance of the border around an alarmed graphic and map alarm priority levels that appear in the indicator icon:

- Map alarm priority numbers to alarm severity ranges for Alarm Border indicator icons. By default, an Alarm Border indicator icon is assigned to the four alarm severity levels and an alarm’s Inhibited or Suppressed states.

- The outline of an alarm border is set by the Outline property of an Element Style. Other properties of an Element Style are ignored. Only an alarm border shows the visual characteristics of an associated alarm Element Style because the alarm indicator icon is an image. For more information about Element Styles,

- The color and fill pattern of an alarm border is set by a defined Element Style. Only an alarm border shows the visual characteristics of an associated alarm Element Style because the alarm indicator icon is an image.

The following table shows the assigned mapping of Element Styles to alarm severity and state in Alarm Border animations.

Users cannot change the mapping of Element Styles to alarm states. Users can only change the values assigned to the Element Style’s Outline properties.
The following default colors and fill patterns are assigned to alarm borders. The outline of an alarm border can be configured as part of an assigned Element Style. Otherwise, the default Outline property values are set for the alarm border.

<table>
<thead>
<tr>
<th>Alarm Severity</th>
<th>Alarm State</th>
<th>Element Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UnAck</td>
<td>Alarm_Critical_UnAck</td>
</tr>
<tr>
<td>1</td>
<td>Ack</td>
<td>Alarm_Critical_Ack</td>
</tr>
<tr>
<td>1</td>
<td>RTN</td>
<td>Alarm_Critical_RTN</td>
</tr>
<tr>
<td>2</td>
<td>UnAck</td>
<td>Alarm_High_UnAck</td>
</tr>
<tr>
<td>2</td>
<td>Ack</td>
<td>Alarm_High_Ack</td>
</tr>
<tr>
<td>2</td>
<td>RTN</td>
<td>Alarm_High_RTN</td>
</tr>
<tr>
<td>3</td>
<td>UnAck</td>
<td>Alarm_Medium_UnAck</td>
</tr>
<tr>
<td>3</td>
<td>Ack</td>
<td>Alarm_Medium_Ack</td>
</tr>
<tr>
<td>3</td>
<td>RTN</td>
<td>Alarm_Medium_RTN</td>
</tr>
<tr>
<td>4</td>
<td>UnAck</td>
<td>Alarm_Low_UnAck</td>
</tr>
<tr>
<td>4</td>
<td>Ack</td>
<td>Alarm_Low_Ack</td>
</tr>
<tr>
<td>4</td>
<td>RTN</td>
<td>Alarm_Low_RTN</td>
</tr>
<tr>
<td>All</td>
<td>Inhibited</td>
<td>Alarm_Inhibited</td>
</tr>
<tr>
<td>All</td>
<td>Suppressed</td>
<td>Alarm_Suppressed</td>
</tr>
<tr>
<td>All</td>
<td>Shelved</td>
<td>Alarm_Shelved</td>
</tr>
</tbody>
</table>
The Alarm Border animation dialog box contains mutually exclusive fields to set the referenced attributes for aggregate or individual alarms.

For aggregation alarms, users specify Alarm Border animation by entering an attribute or object name in the Use Standard Alarm-Urgency References field of the Alarm Border dialog box.

The selected object attributes map to the following aggregation alarm attributes:

- AlarmMostUrgentAcked
- AlarmMostUrgentInAlarm
- AlarmMostUrgentMode
- AlarmMostUrgentSeverity

To set Alarm Border animation for individual alarms, users specify references to the following alarm attributes or tags:

- InAlarm Attribute

### Alarm Border Fill Color and Pattern

<table>
<thead>
<tr>
<th>Alarm Severity</th>
<th>Alarm State</th>
<th>Alarm Border Fill Color and Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UnAck</td>
<td>Purple, solid</td>
</tr>
<tr>
<td>1</td>
<td>Ack</td>
<td>Purple, solid</td>
</tr>
<tr>
<td>1</td>
<td>RTN</td>
<td>Purple, diagonal cross fill</td>
</tr>
<tr>
<td>2</td>
<td>UnAck</td>
<td>Red, solid</td>
</tr>
<tr>
<td>2</td>
<td>Ack</td>
<td>Red, solid</td>
</tr>
<tr>
<td>2</td>
<td>RTN</td>
<td>Red, diagonal cross fill</td>
</tr>
<tr>
<td>3</td>
<td>UnAck</td>
<td>Yellow solid</td>
</tr>
<tr>
<td>3</td>
<td>Ack</td>
<td>Yellow, solid</td>
</tr>
<tr>
<td>3</td>
<td>RTN</td>
<td>Yellow, diagonal cross fill</td>
</tr>
<tr>
<td>4</td>
<td>UnAck</td>
<td>Cyan solid</td>
</tr>
<tr>
<td>4</td>
<td>Ack</td>
<td>Cyan, solid</td>
</tr>
<tr>
<td>4</td>
<td>RTN</td>
<td>Cyan, diagonal cross fill</td>
</tr>
<tr>
<td>All</td>
<td>Inhibited</td>
<td>White, solid</td>
</tr>
<tr>
<td>All</td>
<td>Suppressed</td>
<td>White, solid</td>
</tr>
<tr>
<td>All</td>
<td>Shelved</td>
<td>White, solid</td>
</tr>
</tbody>
</table>

---

**CONFIGURING ALARM BORDER ANIMATION**
Alarm Border animation subscribes to these attributes. Based on the alarm state of these attributes, Alarm Border animation is applied to the graphic element in run time.

**Configure an Alarm Border Animation for a Symbol Using Field Attributes or Objects**

1. Open a Situation Awareness Library symbol in the Symbol Editor.
2. Select the symbol.
3. Click Add Animation to show the list of animation types.
4. Select Alarm Border from the list of animations. The Alarm Border dialog box appears with a set of configuration options.
5) Select either Use Standard Alarm-Urgency References or Use Customized Alarm-Urgency References.
   a. If selecting Use Standard Alarm-Urgency References:
      i. From the drop-down list, select an attribute or object name. Both direct and relative
         references to an object are supported. Field Attributes are also permitted. An
         expression cannot be used to reference the object.
      ii. Click “OK”.
   b. If selecting Use Customized Alarm-Urgency References:
      i. From the drop-down list, select an attribute, a symbol element, or an InTouch tag name
         for all fields. All fields must contain values and cannot be left blank. Expressions,
         external references, and custom properties can be entered in all fields.
      ii. Click OK.

6) Enter a custom property, a constant, an external reference, or an expression in the Show Alarm Indicator
   field to set the conditions when an alarm indicator icon is shown or hidden.

For UDA’s, Digital field attributes and or InTouch alarms choose customized and fill out the following.

Note Priority can only be 1, 2, 3 or 4 indicating severity level, since those are the only severities supported. If this is
not the case instead of a reference in Severity source an integer constant of 1, 2, 3 or 4 is allowed.

**USING POLAR STAR SYMBOLS**

The Graphic Toolbox includes 3, 4, 6, and 8 spoke Situation Awareness Library polar star symbols. A polar star
symbol shows a set of related process values on the spokes that are part of the symbol. As values change along the
length of the spokes, the changing shape of the polar star’s polygon is easily recognizable by operators who can
react quickly to abnormal process conditions.
The following examples show the appearance of Polar Star symbols when measured process values are within normal limits and process values outside of the normal range.

Each spoke contains a set of custom properties to set value set points, alarm limits, and coordinate set point locations for the normalized process value. When a process value changes from its set point location on a spoke, the animation changes the shape of the polar star polygon.

Polar Star spoke attributes can be referenced in scripts or directly from the **Custom Properties** dialog box. The values assigned to a Polar Star spoke attributed can be a constant, an expression, or a remote reference.

For minimal animation, each spoke’s **PV** and **SP** attributes should be assigned values. The following procedure explains how to configure animation by setting values to a Polar Star’s Custom Properties spoke attributes.

To configure polar star point animation

1. Open the polar star symbol in Symbol Editor.
2. Click the polar star symbol to select it.
3. Right-click on the symbol to show the action menu and select **Custom Properties**.
4) Enter expressions or references for each spoke’s PV and other attributes.
5) Enter expressions or references for the remaining spokes of a Polar Star symbol.
6) Save your changes.

**Sweep Angle Animation**

**How to Configure a Sweep Angle Animation.**

Draw a two pie and draw the following:

Set sweep angle to 360 degrees.
Link Slider PV to sweep angle of the Pie graphic.

Test in runtime the sweep angle animation.
GRAPHIC PERFORMANCE INDEX (GPI)

Depending on what is on a particular screen and the density of the object the call up time is affected.

By selecting the meter in the icon set

A performance index can be retrieved

A number of 5 means a sub second call up time and a number closer to 0 will indicate a poor performance of the graphic.

Graphic Performance Index addition – Indication in status bar updates every 15 seconds or when saved.
Trend pen animation was created with the intent to be able to create small real time and historical trends that either use real-time data or append Historical data if present.

Configuration is very simple:

If there is a Historian and the tag is present then data will be back filled from the Historian, the maximum time is 7 days, the trend pen was designed to have a large number of instances on the screen, very different from .net controls this is animation meaning the user can add animations, and draw over the trend pen. They can be attached to Meters.
A Galaxy Style Library includes a set of Element Styles. A Galaxy Style Library can be exported from the IDE and imported into other Galaxies.

Style definitions:
NOTE: ONLY ONE GALAXY STYLE LIBRARY CAN BE LOADED AND USED IN A GALAXY AT A TIME.

Two types of Element Styles are included in a Galaxy Style Library:

- Pre-Defined Element Styles. These Element Styles have been configured with default values, but you can modify them to your requirements.
- User-Defined Element Styles. These Element Styles do not contain preset values and can be configured to meet the needs of your applications.

You can create your own Galaxy Style Library by overriding the configuration of Element Styles and exporting the library.

If you want to use the original default visual properties for your graphic elements, you can reset Pre-Defined and User-Defined Element Styles to their defaults.

**IMPORTING A GALAXY STYLE LIBRARY**

To import a Galaxy Style Library

1) On the Galaxy menu, click Import and click Theme. The Select a Theme File to Import dialog box appears.
2) Browse for the library file. Galaxy Style Library files are XML files.
3) Select the file and click Open. The selected Galaxy Style Library is loaded in the IDE.

The imported library is now the active Galaxy Style Library for the entire Galaxy.
NOTE: IF A GALAXY STYLE LIBRARY IS IMPORTED TO THE IDE WHILE WINDOWVIEWER RUNNING, APPLICATION GRAPHICS ARE REFRESHED WITH NEW ELEMENT STYLES AND APPEAR AUTOMATICALLY IN WINDOWVIEWER WITHOUT REQUIRING A RESTART.

ADDITIONAL STYLE LIBRARIES

We are shipping and installing alternate style libraries and this can be helpful for demonstrating the feature. Can import from here:

C:\Program Files (x86)\Arches\Framework\Bin\AdditionalElementStyles
EXPORTING A GALAXY STYLE LIBRARY

You can create your own Galaxy Style Library by overriding the configuration of Element Styles and exporting the library. For information about setting Element Style overrides, see "Chapter 7,

To export a Galaxy Style library:

1) On the Galaxy menu, click Export and click Galaxy Style Library. The Export Galaxy Style Library dialog box appears.
2) Browse to a path and type a name for the exported library file.
3) Click Save. The file is saved with the specified name and a .xml file extension.

RESETTING A PRE-DEFINED OR USER-DEFINED GALAXY STYLE LIBRARY

If you changed the values of Pre-Defined or User-Defined Galaxy Element Styles, you can restore the default Element Styles by clicking the Reset to Default button.

MODIFYING GRAPHICS WITH ELEMENT STYLES

Element Styles define one or more visual properties like fill, line, text, blink, outline, and status properties of graphic elements. You can apply an Element Style to a graphic element to set preconfigured properties defined in that Element Style. Element Styles establish consistent visual standards for symbols.

WORKING WITH ELEMENT STYLES

An Element Style defines a set of visual properties that determine the appearance of text, lines, graphic outlines, and interior fill shown in ArchestrA Symbols or graphics. An Element Style that is applied to a symbol sets pre-configured visual property values that take precedence over a symbol’s native visual properties.
UNDERSTANDING ELEMENT STYLES

Element Styles provide the means for developers to establish consistent visual standards in their ArchestrA applications. An Element Style can define the same visual properties of text, lines, fill, and outlines for all symbols or graphics that belong to an application. Likewise, Element Styles can show the current status of an object represented by a symbol. For example, a different Element Style can be applied to a symbol when an object transitions to an alarm state.

GALAXY STYLE LIBRARY

A set of Element Styles is provided in the predefined Galaxy Style Library. The pre-defined values of the Element Styles in this library can be changed. However, existing Element Styles cannot be renamed or deleted, and new Element Styles cannot be added to the library.
VISSUAL PROPERTIES DEFINED BY ELEMENT STYLES

The following table lists the visual properties of graphic elements defined in an Element Style.

<table>
<thead>
<tr>
<th>Graphic Element</th>
<th>Element Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>• Font family</td>
</tr>
<tr>
<td></td>
<td>• Font size</td>
</tr>
<tr>
<td></td>
<td>• Font style</td>
</tr>
<tr>
<td></td>
<td>• Font color</td>
</tr>
<tr>
<td></td>
<td>• Blink On/Off</td>
</tr>
<tr>
<td>Fill</td>
<td>• Fill color</td>
</tr>
<tr>
<td></td>
<td>• Fill gradient</td>
</tr>
<tr>
<td></td>
<td>• Fill pattern</td>
</tr>
<tr>
<td></td>
<td>• Fill texture</td>
</tr>
<tr>
<td></td>
<td>• Blink On/Off</td>
</tr>
<tr>
<td>Line</td>
<td>• Line pattern</td>
</tr>
<tr>
<td></td>
<td>• Line weight</td>
</tr>
<tr>
<td></td>
<td>• Line color</td>
</tr>
<tr>
<td></td>
<td>• Blink On/Off</td>
</tr>
<tr>
<td>Outline</td>
<td>• Outline Show/Hide</td>
</tr>
<tr>
<td></td>
<td>• Outline Pattern</td>
</tr>
<tr>
<td></td>
<td>• Outline Weight</td>
</tr>
<tr>
<td></td>
<td>• Outline Color</td>
</tr>
<tr>
<td></td>
<td>• Blink On/Off</td>
</tr>
</tbody>
</table>

An Element Style may not define every visual property. If a property value is not defined in an applied Element Style, the element’s native style is used and can be changed. However, if an element’s property value is defined in an applied Element Style, the element’s native properties are disabled and cannot be changed.

ELEMENT STYLES IN ANIMATIONS

You can configure an element or a group of elements with Boolean or truth table animations that determine whether Element Styles are applied based on evaluated conditions or expressions.

PROPERTY STYLE ORDER OF PRECEDENCE

To understand the behavior of an element’s properties when an Element Style is applied, you should understand the order of precedence for the levels at which property styles are applied.
**UPDATING ELEMENT STYLES AT APPLICATION RUN TIME**

You can update the Elements Styles applied to symbols or graphics included in a running application.

- Updating Element Styles from the IDE: When an application is deployed and updates were made to the applied Element Styles from the ArchestrA IDE, those updates will be propagated to the graphic elements in a running application without requiring WindowViewer to be closed and re-opened.
- Importing an updated Graphic Style Library: Importing an updated Graphic Style Library that includes different applied Element Styles will propagate those changes to graphic elements in a running application without requiring WindowViewer to be closed and re-opened.

**CHANGING VISUAL PROPERTIES OF AN ELEMENT STYLE**

You can modify the visual properties of any Element Style in the currently loaded Galaxy Styles Library. You modify properties by setting overrides on the **Element Styles** tab in the **Configure Galaxy Style Library** dialog box. In the **Configure Galaxy Style Library** dialog box you can:

- Modify the appearance of text by setting overrides for the text font, text size, text style, text color, and blinking.
- Modify the appearance of graphic fill by setting overrides for fill color and blinking.
- Override the appearance of the line pattern, weight, color, and blinking.
- Override the appearance of the outline line pattern, weight, color, and blinking.
- Preview the appearance of an Element Style.
- Reset Element Style visual properties to their default values.

To show the current Element Styles of a Galaxy:

1) On the **Galaxy** menu, click **Configure** and click **Galaxy Style Library**. The **Configure Galaxy Style Library** dialog box appears.
2) Click the **Element Styles** tab.
The Element Styles tab includes the following fields:

- The **Preview** field shows the appearance of an element when the current Element Style is applied.
- The **Reset to Default** button returns all modified Element Styles to their default values.
- The property tabs include related fields to set values for each property defined in the selected Element Style.

You can modify an Element Style’s text visual properties by setting alternative values for text font, text color, text style, and blink rate.
To change the appearance of text in an Element Style:

1) On the Galaxy menu, point to Configure, and then click Galaxy Style Library. The Configure Galaxy Style Library dialog box appears.
2) Select an Element Style from the Element Style Overrides list.
3) Click the Text (Ts) tab.

4) To change the font, select Font Override, click the browse button, and select a font from the Font dialog box.

5) To override the font color:
   a. Select Font Color Override.
   b. Click the color box.
   c. Select a color from the Select Font Color dialog box.

6) To override the text blink behavior:
   a. Select Blink.
   b. Select a blinking speed from the Speed list.
   c. Click the color box.
   d. Select a style from the Select Blink Color dialog box.
OVERRIDING THE ELEMENT STYLE FILL PROPERTIES

You can modify an Element Style’s fill visual properties by setting alternative values for fill color and blink rate.

To override the fill appearance of an Element Style

1) On the Galaxy menu, point to Configure, and then click Galaxy Style Library. The Configure Galaxy Style Library dialog box appears.
2) Select an Element Style from the Element Style Overrides list.
3) Click the Fill tab.

4) To override the fill style:
   a. Select Fill Style Override.
   b. Click the color box.
   c. Select a style from the Style Selection dialog box.

5) To override the fill blink behavior:
   a. Select Blink.
   b. Select a blinking speed from the Speed list.
   c. Click the color box.
   d. Select a style from the Style Selection dialog box.
OVERRIDING THE ELEMENT STYLE LINE PROPERTIES

You can modify an Element Style’s line visual properties by setting alternative values for line color, line pattern, and line weight.

1) On the Galaxy menu, point to Configure, and then click Galaxy Style Library. The Configure Galaxy Style Library dialog box appears.

2) Select an Element Style from the Element Style Overrides list.

3) Click the Line tab.

4) To override the line pattern, select Line Pattern Override and select a line pattern from the adjacent list.

5) To override the line weight, select Line Weight Override and type a new line weight in the adjacent box.

6) To override the line style:
   a. Select Line Color Override.
   b. Click the color box.
   c. Select a style from the Style Selection dialog box.

7) To override the line blink behavior:
   a. Select Blink.
   b. Select a blinking speed from the Speed list.
   c. Click the color box.
   d. Select a style from the Style Selection dialog box.

8) Click OK
OVERRIDING THE ELEMENT STYLE OUTLINE PROPERTIES

You can modify an Element Style’s outline visual properties by setting alternative values for text font, text color, text style, and blink rate.

To override the outline appearance of an Element Style

1) On the Galaxy menu, point to Configure, and then click Galaxy Style Library. The Configure Galaxy Style Library dialog box appears.
2) Select an Element Style from the Element Style Overrides list.
3) Click the Outline tab.

4) Select Show Outline.
5) To set the line pattern, select Line Pattern and select a line pattern from the adjacent list.
6) To set the line weight, select Line Weight and type a line weight in the adjacent box.
7) To set the line style:
   a. Click the color box next to Line Color.
   b. Select a style from the Style Selection dialog box.
8) To set the line blink behavior:
   a. Select Blink.
   b. Select a blinking speed from the Speed list.
   c. Click the color box.
   d. Select a style from the Style Selection dialog box.
PREVIEWING AN ELEMENT STYLE

The Preview area shows the appearance of an Element Style’s current assigned property values.

RESETTING AN ELEMENT STYLE TO DEFAULT VALUES

You can reset an Element Style to its original default property values.

NOTE: resetting an Element Style resets visual properties to original default values, no to any previous override settings.

To reset an Element Style to default values:

1) On the Galaxy menu, point to Configure, and then click Galaxy Style Library. The Configure Galaxy Style Library dialog box appears.
2) Select one or more Element Styles from the Element Style Overrides list.
3) Click Reset to Default. All Element Style properties

CHANGING THE VISUAL PROPERTIES OF USER-DEFINED ELEMENT STYLES

The Galaxy Style Library includes a set of 25 user-defined Element Styles. User-defined Element Styles appear near the bottom of the list of the Element Style Overrides field and are named User_DEFINED_01 to User_DEFINED_25.

All visual properties of user-defined Element Styles are initially set to default values. Visual properties can be individually configured for each user-defined Element Style by setting overrides for text, fill, line, and outline as other pre-defined Element Styles.
APPLYING ELEMENT STYLES TO ELEMENTS

You can apply Element Styles to one or more elements. Unlike setting Element Style overrides that change the appearance of an Element Style’s properties, applying an Element Style to an element overrides the element’s native properties. Applying Element Styles to elements can help standardize the appearance of those elements in the Galaxy and show the current state of an object represented by a symbol or graphic.

USING THE ELEMENT STYLE LIST

The Symbol Editor menu bar contains an Element Style list to select an Element Style and apply it to a selected element of a symbol or graphic.

To apply an Element Style to a graphic element:

1) Open the symbol or graphic in the Symbol Editor.
2) Select one or more elements from the graphic or symbol.
3) Select an Element Style from the Element Styles list to apply the selected elements.
USING THE PROPERTIES GRID

The Symbol Editor Properties view contains an Element Style Appearance item to select an Element Style and apply it to a selected element of a symbol or graphic.

To apply an Element Style from the Properties Editor:

1) Open the symbol or graphic in the Symbol Editor.
2) Select one or more elements from the graphic or symbol.
3) In the Appearance category of the Properties Editor, select an Element Style from the Element Style list.

USING FORMAT PAINTER

You can use the Symbol Editor’s Format Painter to copy the Element Style from one graphic element to another.

To apply an Element Style using Format Painter:

1) Open a symbol or graphic in the Symbol Editor.
2) Select the element with the Element Style you want to copy.
3) On the Edit menu, click Format Painter. The pointer appears as the format painter cursor.
4) Select the elements you want to apply the Element Style to. The Element Style is applied to the clicked element.
CLEARING AN ELEMENT STYLE

When an Element Style is applied to an element, you cannot edit the element’s styles that are controlled by the applied Element Style. However, you can clear the application of the Element Style so that all of the styles can be edited.

To clear an Element Style

1) Select the element.
2) Elect None in the Element Style list.

SELECTING AN ELEMENT STYLE AS A DEFAULT FOR A CANVAS

You can select an Element Style at the canvas level. The selected Element Style is applied to any graphic element or groups that you create on the canvas.

APPLYING ELEMENT STYLES TO GROUPS OF ELEMENTS

You can apply an Element Style on a group of elements in the same way that you apply an Element Style to an element. However, the group’s run-time behavior must be set to TreatAsIcon.

SETTING A GROUP’S RUN-TIME BEHAVIOR TO TREATASICON

To apply an Element Style to a graphic element group, the group’s TreatAsIcon property must be set to True. Otherwise, the Element Style lists are disabled when an element group is selected.

To set a group’s TreatAsIcon property to true:

1) Select the element group to which the Element Style will be applied.
2) On the Properties menu, click Run-time Behavior and click TreatAsIcon.
3) Select True from the drop-down list.
UNDERSTANDING ELEMENT STYLE BEHAVIOR WITH A GROUP OF ELEMENTS

- The Element Style applied to a group has higher precedence than the property styles applied to individual graphic elements in the group.
- If the Element Style applied to a group of elements has undefined property styles, then the element continues to use its Element Style or element-level settings for undefined property styles.
- If the Element Style that is applied to a group of elements has defined property styles, then those property styles override the property styles defined at the element level for elements in the group.
- An Element Style cannot be applied to a nested element group.
- If you add an element to a group that has a group-level Element

CONFIGURING AN ANIMATION USING ELEMENT STYLES

You can configure an element or a group of elements with a:
- Boolean animation that applies Element Styles based on a binary Yes/No condition.
- Truth table animation that applies Element Styles based on a range of possible values. The truth table animation that applies Element Styles:
  - Associates expressions of any data type supported by Application Server or InTouch to an Element Style.
  - Defines as many conditions as required and applies a separate Element Style for each condition
  - Defines the conditions to apply an Element Style by specifying a comparison operator (=, >, >=, <, <=) and a breakpoint, which itself can be a value, an attribute reference, or an expression.
  - Arranges conditions in the order that Element Styles are processed.

CONFIGURING A BOOLEAN ANIMATION USING ELEMENT STYLES

You can configure an element or a group of elements with a Boolean animation that uses only two Element Styles.

To configure an element or a group of elements with an Element Style that uses Boolean animation

1) Open the symbol or graphic in the IDE Symbol Editor.
2) Select the element or element group.
3) On the Special menu, click Edit Animations. The Edit Animations dialog box appears.
4 Click the Add icon and select Element Style. The Element Style animation is added to the Animation list and the Element Style state selection panel appears.

5 Click the Boolean button. The Boolean Element Style configuration panel appears.

6) In the Boolean text box, enter a Boolean numeric value, attribute reference, or expression.

7) Clear Element Style in the True, 1, On area or False, 0, Off area if you do not want a different Element Style for the true or false condition than the default Element Style that is shown in the Element Style list.

8) In the True, 1, On area, select the Element Style in the list to use when the expression is true.
9) In the **False**, 0, Off area, select the Element Style in the list to use when the expression is false.

10) Click **OK**.

### CONFIGURING A TRUTH TABLE ANIMATION WITH ELEMENT STYLES

You can configure an element or a group of elements with a Truth Table animation that selects multiple Element Styles based on a set of evaluated values or expressions.

To configure an element or a group of elements with an Element Style that uses Truth Table animation

1) Open the symbol or graphic in the IDE Symbol Editor.

2) Select the element or group.

3) On the Special menu, click Edit Animations. The Edit Animations dialog box appears.

4) Click the Add icon and select Element Style. The Element Style animation is added to the Animation list and the Element Style state selection panel appears.

5) Click the Truth Table button. The Truth Table Element Style configuration panel appears. The Element Style that is applied to the element

6) In the Expression Or Reference area:
   a. Select the data type of the expression from the list.
   b. Type a value, attribute reference or expression in the text box.

7) If the data type of the expression is string or internationalized string, you can specify to ignore the case by selecting Ignore Case.

8) In the Truth Table, select the Element Style check box and select the Element Style for one of the conditions to be defined in the truth table.

9) In the Operator column, select a comparison operator.

10) In the Value or Expression column, type a value, attribute reference, or expression.
11) To add other conditions:
   a. Click the Add icon. An additional condition is added to the truth table.
   b. Select the Element Style check box, select the Element Style for the condition, select an operator, and enter the condition value or expression.

12) After adding all truth table conditions, click OK. Truth Table animation is typically used to set different Element Styles to represent an object’s current alarm state. You can set Truth Table conditions to show different Element Styles that represent the following alarm conditions:
   a. When the attribute TankLevel_001.PV is 0 then no Element Style is applied.
   b. When the attribute TankLevel_001.PV is less than 20, then the Element Style is Alarm_Minor_Dev.
   c. When the attribute TankLevel_001.PV is greater than the attribute Standards.TankMax then the Element Style is Alarm_Major_Dev.

DELETING A CONDITION FROM AN ANIMATION TRUTH TABLE

You can delete a condition from an animation Truth Table to remove the associated Element Style from the animation.

To delete a condition from a Truth Table animation that uses Element Styles
   1) Open the Edit Animations dialog box, Truth Table Element Style panel.
   2) Select the condition you want to delete.
   3) Click the Remove icon. The condition is removed.
CHANGING THE PROCESSING ORDER OF ELEMENT STYLES IN A TRUTH TABLE ANIMATION

You can change the processing order of Element Styles by moving the conditions up or down in the Truth Table list. The Element Style at the top of the Truth Table list is processed first. The remaining Element Styles are processed in order based on their position from the top of the list.

To change the processing order of Element Style conditions:

1) Open the Edit Animations dialog box, Truth Table Element Style panel.
2) Select the condition you want to move up or down the condition list in order for it to be processed sooner or later.
3) Click the:
   a. Arrow up icon to move the condition up in the truth table.
   b. Arrow down icon
Using the Symbol Wizard

The ArchestrA Symbol Editor includes Symbol Wizard, which can be used to create composite ArchestrA Symbols containing different visual and functional configurations. Composite symbols containing multiple configurations can be embedded into managed InTouch applications. A symbol’s default configuration can be changed to meet the requirements of an application. Composite symbols with multiple configurations reduce the number of required ArchestrA Symbols that need to be created for an application.

Symbol Wizard can create multi-configuration composite symbols from traditional ArchestrA Symbols and Situational Awareness Library symbols.

Both types of symbols are located in the Graphic Toolbox in separate folders. Situational Awareness Library symbols provide an additional benefit of including defined properties and their associated attributes to more easily create configurations.

Some Situational Awareness Library symbols include a Type property to assign a specific function for a symbol configuration. For example, a meter symbol can be configured to represent a thermometer, a pressure meter, or flow meter by changing the attribute assigned to the Type property.

Composite ArchestrA Symbols created with Symbol Wizard are saved in the IDE’s Graphic Toolbox and are not associated with any specific ArchestrA object template or object instance. Except for the ability to select a specific symbol configuration, composite ArchestrA Symbols behave like standard ArchestrA Symbols.
Typically, the process of creating and embedding a composite ArchestrA Symbol in an application requires the involvement of a Designer and a Consumer.

A Designer creates composite ArchestrA Symbols using Symbol Wizard. A Consumer selects configurations of composite symbols and embeds them into managed InTouch applications.

**CREATING COMPOSITE ARCHESTRA SYMBOLS WITH SYMBOL WIZARD**

A Designer uses Symbol Wizard to define the various required symbol configurations based on a set of rules and symbol layers. A Designer defines a set of layers, which are used to group a set of graphic elements, custom properties, and named scripts. Graphic elements and other symbol properties can be assigned to no layers or multiple layers. Graphic elements that are not assigned to any layer always appear in all symbol configurations.

A Designer can create a rule for each layer that defines the conditions when the layer is included in a symbol configuration. Rules are assigned with Choice Groups, Choices, and Options. A Designer selects a configuration to be the symbol default that appears when the symbol is embedded in a managed InTouch application.

After creating all symbol configurations, the Designer verifies how each configuration of a symbol using the Symbol Wizard Preview. Designers set values in the Wizard Options view to verify that each configuration appears as designed based on the layer rules set for the symbol.

When a composite ArchestrA Symbol is complete, the Designer saves it to the Galaxy library so that it is available for use in managed InTouch applications. For more information about creating composite

**EMBEDDING COMPOSITE SYMBOLS INTO AN APPLICATION**

Composite ArchestrA Symbols created with Symbol Wizard are stored in a Galaxy library just like standard ArchestrA Symbols.

Composite symbols are identified in the Galaxy Browser by a Symbol Wizard icon to differentiate them from standard ArchestrA Symbols.

When a Consumer selects a composite ArchestrA Symbol and embeds it into a managed InTouch application, the composite ArchestrA Symbol’s default configuration is selected. The Consumer can change a composite symbol’s configuration by changing the rules assigned to the symbol’s properties from the Symbol Wizard’s Wizard Options section of the Properties view. Depending on the selected configuration, there can be additional configuration-related properties that can be selected by the consumer.
After selecting a symbol configuration and changing any properties, the Consumer saves the composite ArchestrA Symbol so that it can be imported into WindowMaker. While the InTouch application is running, the composite ArchestrA Symbol appears as the configuration selected by the Consumer. A composite symbol configuration cannot be changed during run-time.

**CREATING MULTIPLE CONFIGURATIONS OF A SYMBOL**

The Symbol Wizard is a feature of the Symbol Editor to create multiple configurations of an ArchestrA or Situation Awareness Library symbol. A symbol configuration represents different visual or functional variations of a symbol.

Symbol configurations are created using layers containing associated graphic elements, custom properties, and named scripts. Based on symbol properties and possible values of these properties, rules are applied that specify when a layer is part of a symbol configuration.
UNDERSTANDING VISUAL AND FUNCTIONAL SYMBOL CONFIGURATIONS

Standard ArchestrA symbols available from the IDE Graphic Toolbox show reasonably realistic views of process objects. These symbols can be modified with the Symbol Wizard to incorporate multiple visual configurations in a symbol.

Situation Awareness Library symbols include properties to create functional symbol configurations in addition to visual properties.

VISUAL SYMBOL CONFIGURATIONS

Using an example of a centrifugal pump with separate inlet and outlet pipes, there are four practical visual configurations. The pump’s blade housing is common and appears in all possible configurations. But, the pump’s inlet and outlet pipes can be placed at the left or right in a horizontal direction or at the top or bottom when the pump is oriented vertically.

Horizontal direction and vertical orientation are the two visual properties that identify the different configurations of a pump symbol.

The attributes associated with these two pump properties are left and right for the direction property and top and bottom for vertical orientation property.

FUNCTIONAL SYMBOL CONFIGURATIONS

Situation Awareness Library symbols include functional properties in addition to visual properties. For example, a multi-stage pump symbol includes a NumberOfStages property to select either a five-stage, three-stage, or single stage pump in addition to a visual Orientation property to select left or right pump configurations.
DIFFERENT SYMBOL WIZARD WORK FLOWS

There are two types of ArchestrA users who work with the Symbol Wizard:

- **Designers** are ArchestrA users responsible for creating different symbol configurations. After verifying that all configurations built for a symbol are correct, the Designer saves the symbol to the Galaxy library just like standard ArchestrA symbols.

- **Consumers** embed symbols with multiple configurations as part of creating managed InTouch applications. The symbol’s default configuration is selected when a symbol is embedded. The Consumer can change a symbol’s configuration based on the needs of the managed InTouch application. A Consumer selects the needed configuration by changing values or rules in the **Wizard**.

USING THE SYMBOL WIZARD

Designers and Consumers work with the Symbol Wizard to create symbols with multiple configurations that can be embedded into managed InTouch applications. Designers begin editing an ArchestrA or Situation Awareness Library symbol with the Symbol Editor. Designers show the Symbol Wizard by clicking the Symbol Wizard icon from the Symbol Editor’s menu bar, selecting it as an option of the **View** menu, or pressing Alt+W.
The Symbol Editor window updates to show two tabbed Symbol Wizard views at the left. **Elements, Options, and Layers** views at the left and a **Properties** view at the right of the Symbol Editor’s window.

After creating all configurations of a symbol with the Symbol Wizard, Designers use the Symbol Wizard Preview to verify that all configurations are correct. The Symbol Wizard Preview can be launched by clicking it from the menu bar, selecting it as an option of the **View** menu, or pressing Alt+P.

Beneath the **Tools** view, a tabbed view shows the graphic elements, custom properties, and named scripts that belong to the symbol. The symbol’s graphic elements, custom properties, and named scripts from this view are assigned to symbol layers by dragging them to corresponding folders in the **Layers** view.

- The tabbed **Options** view shows a hierarchical list of Choice Groups, Choices, and Options that define symbol configurations.
The tabbed Layers view includes a list of defined symbol layers. Beneath each layer, separate folders contain the symbol’s graphic elements, custom properties, and named scripts associated with the layer.

**UNDERSTANDING CHOICE GROUPS AND CHOICES**

The Symbol Wizard’s Options view includes buttons to create Choice Groups, Choices, and Options.

- A **Choice Group** represents a unique property of a symbol and appears as the top level property node in the Options view.

- A **Choice** represents a possible value or attribute of a Choice Group property. Choices are indented beneath the associated Choice Group node in the Options view. Choices are mutually exclusive and only one choice can be selected from a Choice Group for a single configuration of a symbol.

- An item shown in the Options view list can be moved by selecting it, and then clicking the Up or Down arrow. If no Choice is specified as the default value for a Choice Group, the default value is always the first Choice added to the Choice Group.

In the example of an ArchestrA centrifugal pump symbol, two possible Choice Groups are Direction and Vertical for the two horizontal and vertical properties of different pump configurations. The Left, Right, Bottom, and Top choices appear as the associated Choice attributes of the two visual orientation properties.
In the example of a Situation Awareness Library pump symbol, the choice groups are the number of pump stages, whether the pump includes a VSD display, and the pump's left/right orientation.

**IMPORTANT:** CHOICE GROUPS AND THEIR ASSIGNED CHOICES ARE AUTOMATICALLY CREATED FOR SITUATION AWARENESS LIBRARY SYMBOLS.

ADDITIONAL CHOICE GROUPS, CHOICES, AND OPTIONS CAN BE ADDED FROM THE OPTIONS VIEW.
DEFINING SYMBOL CONFIGURATION RULES

An Option rule defines an expression that determines if the associated option or layer is visible or invisible based on the evaluation of the rule to true or false.

Option rules can consist of a single expression or compound expressions using Boolean keywords or operator characters:

<table>
<thead>
<tr>
<th>Boolean Keywords</th>
<th>AND, OR, NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator Characters</td>
<td>• Period (.)</td>
</tr>
<tr>
<td></td>
<td>A period concatenates a Choice Group to a Choice in a hierarchical expression.</td>
</tr>
<tr>
<td></td>
<td>• Pipe (</td>
</tr>
<tr>
<td></td>
<td>A pipe evaluates to a Boolean OR.</td>
</tr>
<tr>
<td></td>
<td>• Ampersand (&amp;)</td>
</tr>
<tr>
<td></td>
<td>An ampersand evaluates to a Boolean AND.</td>
</tr>
<tr>
<td></td>
<td>• Exclamation point (!)</td>
</tr>
<tr>
<td></td>
<td>An exclamation point evaluates to a Boolean NOT.</td>
</tr>
<tr>
<td></td>
<td>• Parentheses ( )</td>
</tr>
<tr>
<td></td>
<td>A compound expression enclosed within parentheses is evaluated before other expressions in a rule</td>
</tr>
</tbody>
</table>

Any other unlisted keywords or operator characters are treated as part of the references in a rule.

Any other unlisted keywords or operator characters are treated as part of the references in a rule.
• Compound expressions that include a Boolean keyword must include blank spaces around the keyword.
  
  \textit{ConditionA OR ConditionB}

• Compound expressions that include an operator character that evaluates to a Boolean condition do not require blank spaces.
  
  \textit{ConditionA | ConditionB}

• A property attribute must be referenced by its hierarchal Choice Group name.
  
  \textit{ChoiceGroup.Choice}

• When an Option is renamed, the name change is updated in all referenced rule expressions.

• An Option or a Choice can be deleted only if no graphics are associated with their default layers.

\section*{Associating Elements to Symbol Wizard Layers}

Symbol layers associate graphic elements, custom properties, and named scripts to a unique symbol configuration defined by a rule.

When the rule is True, the layer’s associated elements, custom properties, and named scripts are part of the symbol’s configuration.

In the example of an ArchestrA centrifugal pump symbol, Option rules determine whether the pump should be oriented horizontally or vertically and what directional attribute should be selected for the orientation property. When the rule for the Right configuration is True, the Right layer containing the inlet and outlet pipes is part of the symbol’s configuration.
The blade housing does not belong to a layer because it is common to all pump symbol configurations. Graphic elements of a symbol that do not belong to a layer appear in all symbol configurations. As a result, the pump’s blade housing appears in the Left, Right, Top, and Bottom configurations of the pump by default.

Associating elements to symbol layers involves working with the Symbol Wizard views shown to the left of the graphic canvas.

**DESIGNING SYMBOLS WITH THE SYMBOL WIZARD**

The process of creating and implementing a composite ArchestrA symbol created with Symbol Wizard has two work flows, a Designer workflow and a Consumer workflow:

- A Designer uses Symbol Wizard to create composite ArchestrA symbols containing multiple configurations.
- A Consumer embeds a composite symbol created by Symbol Wizard into a managed InTouch application and selects the configuration needed by the application.

The following list summarizes the tasks completed by a Designer to create multiple configurations of a symbol with Symbol Wizard.

- Define a symbol’s Choice Groups, their Choices, and Options
- Assign rules to Choice Groups, Choices, and Options
- Update symbol layers if necessary
- Associate graphic elements, custom properties, and named scripts to symbol layers
- Verify each symbol configuration with Symbol Wizard Preview
CREATING SYMBOL CHOICE GROUPS, CHOICES, AND OPTIONS

After planning the possible configurations for a symbol, Designers should know the properties and the possible attributes associated with each configuration. Designers create Choice Groups, Choices, and Options to define a symbol’s properties and attributes.

Important: Situation Awareness Library symbols have predefined Choice Groups, Choices, and Options.

To create symbol choice groups, choices, and options:

1) In the ArchestrA IDE, create a copy of a symbol in the Graphic Toolbox that you want to create multiple configurations. You can also build an entirely unique symbol from scratch and create multiple configurations of it with Symbol Wizard.

2) Check out and open the copied symbol in the Symbol Editor’s canvas drawing area.

3) Click the Symbol Wizard icon shown on the Symbol Editor menu bar. You can also show Symbol Wizard by pressing [Alt+W] or selecting it as an option from the View menu. The Symbol Editor updates to show the Symbol Wizard’s Options and Layers views at the left of the window.

4) Click the Options tab.

5) Click Add Choice Group to create a Choice Group. A Choice Group folder appears in the Options window.

6) Rename the Choice Group to assign an easily identifiable name of a property used in a symbol configuration. Creating a Choice Group automatically sets it to rename mode. You can also manually rename a Choice Group by right-clicking on the Choice Group and select Rename from the menu.

7) Repeat steps 5-6 to create as many Choice Groups as needed to define all properties of a symbol that determine its configurations.

8) Select a Choice Group folder and click Add Choice to add a choice beneath the selected Choice Group.

9) Rename the Choice to assign an easily identifiable name of a property attribute used in a symbol configuration.

10) Repeat steps 8-9 to assign all possible Choice attributes to the Choice Groups.

11) Click Add Option to add an Option, which appears in the window at the same hierarchical level as Choice Groups.

12) Right-click the Option and select Rename to assign a name.

13) Repeat steps 11-12 to create as many Options needed to define a symbol’s configurations.

ASSIGNING SYMBOL CONFIGURATION RULES

Designers can specify rules for a symbol’s defined Choices and Options. Choice Groups should not be included in symbol configuration rules.

These rules determine the graphic elements, custom properties, and scripts that belong to a symbol configuration.
To define symbol configuration rules:

1) Show the selected symbol in the Symbol Editor with the Symbol Wizard enabled.
2) Select a Choice from the Options view. The Properties view updates to show Option Properties fields. The Name field shows the name of the Choice you selected from the Options view. The Rule field is blank.
3) If necessary, enter a rule for the Choice. Important: Not all Choices require rules. Specify only those rules necessary to create symbol configurations. Choices without rules are always visible.
4) Repeat steps 2-3 to specify rules for the remaining Choices of the symbol.
5) Select an Option from the Options view. The Name field of the Option Properties view updates to show the name of the Option you selected from the Options view.

6) Enter a rule for the Option that defines the conditions to show or hide the Choice Groups and Choices in a configuration.
7) Enter True or False in the Default Value field to set the Option as part of the symbol’s default configuration or not.
8) In the Description field, enter a description of the Option. The description appears when the Consumer embeds the symbol and clicks on the option to configure it.
9) Repeat steps 5-8 to specify rules and optional default values for the remaining Options of the symbol.

**UPDATING SYMBOL LAYERS**

Symbol Wizard automatically creates a set of default layers that match the hierarchical set of Choices and Options defined for a symbol. Each Choice layer has an assigned default rule containing the expression `ChoiceGroup.Choice` that defines an attribute of a symbol’s property.

The default rule for an Option layer is simply the name of the Option itself. Renaming an Option automatically renames any layer rules that reference the Option.

**IMPORTANT:** UPDATING SYMBOL LAYERS MAY NOT BE NECESSARY IF THE DEFAULT SET OF LAYERS CREATED FOR CHOICES AND OPTIONS CAN CREATE ALL SYMBOL CONFIGURATIONS.

Designers can update layers by adding layers to or deleting layers from the set of default layers. Also, layers can be renamed and the default rule assigned to a layer can be changed.

If a symbol layer is renamed, it loses the link to the Option. When the Option name is updated, the layer name will not get updated with changed Option name.

**To add or delete a symbol layer**

1) Show the selected symbol in the Symbol Editor with the Symbol Wizard selected.
2) Click the **Layers** tab to show the list of layers.
3) To add a layer, do the following:
Using the Symbol Wizard

a. Click the Add Layer icon above the Layers list. You can also add a layer by right-clicking within the layers list to show the action menu and selecting Add. The new layer appears at the bottom of the list with an assigned default name.

b. Click on the new layer to select it.

c. Rename the new layer. Creating a layer automatically sets it to rename mode. You can also manually rename a layer by right-clicking on the layer and select Rename from the menu.

4) To delete a layer, do the following:
   a. Click on the layer within the list to be deleted.
   b. Delete the layer by clicking the Delete Layer icon above the Layers list or right clicking

TO UPDATE A LAYER RULE

1) Show the selected symbol in the Symbol Editor with the Symbol Wizard selected.
2) Click the Layers tab to show the list of layers.
3) Select a layer from the list whose rule needs to be updated. The Layer Properties view appears and shows the current rule assigned to the selected layer Choice or Option.
4) Click within the Rule field to select it.
5) Update the rule.
6) Click Save to save the changes to the layer rule.

ASSOCIATING CONFIGURATION ELEMENTS TO SYMBOL LAYERS

The basic workflow to associate graphic elements, custom properties, or named scripts to a symbol layer consists of these general steps:
1) Select a symbol layer from the Layers view.
2) Select items from the tabbed Elements, Named Scripts, and Custom Properties views to associate with the selected layer.

NOTE: MULTIPLE GRAPHIC ELEMENTS, CUSTOM PROPERTIES, OR NAMED SCRIPTS CAN BE SELECTED USING THE [SHIFT] KEY TO SELECT A RANGE OF LISTED ITEMS OR THE [CTRL] KEY TO SELECT INDIVIDUAL ITEMS FROM A LIST. SCRIPTS INTO THE LAYERS VIEW.

Configuration elements can be associated with a symbol layer by two methods:
• Active layer method: Select the check box to the left of the layer name. Then, drag and drop the configuration element anywhere within the Layers view. The configuration element is automatically associated to the correct folder of the active layer.
• Direct folder method: Select a layer and expand it to show the folders for the different types of configuration elements. Then, drag and drop the configuration element directly on the folder that matches the type of configuration element.
ASSOCIATING GRAPHIC ELEMENTS TO SYMBOL LAYERS

Graphic elements show the visual properties of a symbol. Designers must associate graphic elements to the defined layers of a symbol.

**To associate graphic elements to symbol layers:**

1) Show the selected symbol in the Symbol Editor with the Symbol Wizard selected.
2) Click the **Elements** tab to show the graphic elements that belong to the symbol.
3) Click the **Layers** tab.
4) Activate a layer from the **Layers** view by selecting the check box next to the layer. If you prefer to add graphic elements directly to a layer’s **Graphic Elements** folder with the direct folder method, simply click the layer name from the list to select it.
5) Click the box to the left of the check box to expand the layer view and show the **Graphic Elements** folder.
6) Click on the graphic element in the **Elements** view to be associated with the active symbol layer. You can also select the symbol element group by clicking it on the displayed symbol.
   a. Using standard Windows drag and drop technique, drag the graphic element from the **Elements** view and drop it anywhere within the **Layers** view.
   b. If you are using the direct folder method, you must drop the graphic element directly on the selected layer’s **Graphic Elements** folder.
7) The selected element appears beneath the active layer’s **Graphic Elements** folder.
8) Repeat steps 6-7 to select all element groups that belong to the symbol layer. You can also select multiple graphic elements from the **Elements** view and drop them as a set. Repeat steps 4-8 to select all elements for the different layers of a symbol.
9) The **Show/Hide** icon appears to the left of the **Graphic Elements** folder in the **Layers** view. Clicking the icon shows or hides the graphic elements in a layer’s **Graphic Elements** folder on the symbol itself.
10) Click the **Show/Hide** icon to verify the graphic elements associated to a layer are correct for the symbol configuration.
11) Save your changes to the symbol.

ASSOCIATING CUSTOM PROPERTIES TO SYMBOL LAYERS

Associating custom properties to a symbol layer uses a procedure similar to associating graphic elements. Selected custom properties are dragged and dropped on the **Custom Properties** folder to associate them to a symbol layer. You can associate custom properties to layers with the active layer or director folder methods.

**To associate custom properties to symbol layers:**

1) Open the selected symbol in the Symbol Editor with the Symbol Wizard selected.
2) Click the **Custom Properties** tab to show the locally defined custom properties of the symbol. Custom properties of embedded symbols are not listed.
3) Click the **Layers** tab.
4) Select a layer from the **Layers** view to add custom properties by selecting the check box next to the layer.
5) Click the box to the left of the check box to expand the layer view and show the **Custom Properties** folder.
6) Click on a custom property in the **Custom Properties** view that belongs to the selected symbol layer.
7) Using standard Windows drag and drop technique, drag the custom property from the `Custom Properties` view and drop it on the `Custom Properties` folder. The selected custom property appears beneath the `Custom Properties` folder.

8) Repeat steps 6-7 to select all custom properties that belong to the symbol layer.

9) Repeat steps 4-7 to select the remaining custom properties for the different layers of a symbol.

10) Save your changes to the symbol.

ASSOCIATING NAMED SCRIPTS TO SYMBOL LAYERS

Associating named scripts to a symbol layer uses a similar procedure to associate graphic elements or custom properties. You can associate named scripts to layers with the active layer or director folder methods.

To associate named scripts to symbol layers:

1) Show the selected symbol in the Symbol Editor with the Symbol Wizard selected.

2) Click the `Named Scripts` tab to show the scripts associated with the symbol.

3) Click the `Layers` tab.

4) Select a layer from the `Layers` view by selecting the check box next to the layer.

5) Click the box to the left of the check box to expand the layer view and show the `Named Scripts` folder.

6) Click on a script in the `Named Scripts` view that belongs to the selected symbol layer.

7) Using standard Windows drag and drop technique, drag the script from the `Named Scripts` view and drop it on the `Named Scripts` folder. The selected script appears beneath the `Named Scripts` folder.

8) Repeat steps 6-7 to select all scripts that belong to the symbol layer.

9) Repeat steps 4-7 to select the remaining scripts for the different layers of a symbol.

10) Save your changes to the symbol.

VERIFY SYMBOL CONFIGURATIONS

After creating the different configurations of a symbol, Designers use the Symbol Wizard Preview to verify each configuration works as designed. Also, Designers can validate the symbol to identify any invalid references to other objects or values.
To verify symbol configurations

1. Open the composite symbol created with Symbol Wizard in the Symbol Editor.
   Click Symbol Wizard Preview shown on the menu bar of the Symbol Editor.
   You can also open the Symbol Wizard Preview as a View menu option or by pressing Alt+P.

The Symbol Editor updates to show the Wizard Options view with a set of drop-down lists to select different symbol property attributes and options. The default symbol configuration should be selected.

Select the different combinations of property values and view options from Wizard Options fields.

4. Verify the symbol that appears is correct for the specified configuration Choices and Option rule.
5. Click the Validation icon to see if the symbol contains any invalid references.

The Validation view lists any invalid references within the symbol that need to be corrected.

**IMPORTANT:** INVALID REFERENCES ALSO INCLUDE REFERENCES TO PROPERTIES OR ELEMENTS IN HIDDEN SYMBOL LAYERS.
USING MULTI-CONFIGURATION SYMBOLS IN AN APPLICATION

Composite ArchestrA Symbols created with Symbol Wizard are stored in a Galaxy library just like standard ArchestrA Symbols.

When a Consumer selects a composite ArchestrA Symbol and embeds it into a managed InTouch application, the composite ArchestrA Symbol’s default configuration is selected. The Consumer can change a composite symbol’s configuration by changing the values assigned to the symbol’s properties from the Symbol Wizard’s section of the Properties view.

After selecting a symbol configuration and changing any properties, the Consumer saves the composite ArchestrA Symbol. While the InTouch application is running, the composite ArchestrA Symbol appears as the configuration selected by the Consumer. A composite symbol configuration cannot be changed during application run time.

Embedding Multi-Configuration Symbols

Consumers embed composite ArchestrA symbols from the Graphic Toolbox. Embedding a composite symbol is similar to embedding a standard ArchestrA symbol.

A composite symbol appears with its default configuration when it is embedded. The Consumer can select another configuration by changing the configuration values shown in the Wizard Options section of the Properties view.

To embed a composite ArchestrA Symbol

1) Create a new symbol from the Graphic Toolbox or add a symbol to a derived AutomationObject from the Template Toolbox.
2) Open the symbol to show the Symbol Editor.
3) On the Edit menu, click Embed ArchestrA Graphic. You can also click the Embed ArchestrA Graphic icon from the menu bar. The Galaxy Browser appears.
4) Locate the folder containing the composite symbol.
5) Click the symbol to select it and click OK.
6) Position the pointer at the location where the composite symbol should be placed.
7) Click once to embed the composite symbol. The composite symbol appears with handles on the Symbol Editor canvas.
8) Select the symbol’s configuration by selecting values for the various options shown in the Wizard Options view.
9) Rename the symbol.

10) Right-click on the symbol and select **Custom Properties** from the menu. The **Edit Custom Properties** dialog box appears with the set of custom properties defined for the symbol.

11) Configure the custom properties with the required references for the application.

12) Press [F10] to show the **Edit Scripts** dialog box.

13) Verify if any changes need to be made to the symbol’s named scripts to run within the application.

14) Save the changes made to the symbol.
Beneath the Properties view, the **Wizard Options** area includes drop-down menus to select options to show different symbol configurations. After verifying that all symbol configurations are correct, Designers save the symbol into the Graphic Toolbox.
**SITUATIONAL AWARENESS LIBRARY**

**INTRODUCTION AND PHILOSOPHY**

The Graphic Toolbox includes Situation Awareness Library symbols in a set of folders separate from other symbols in the ArchestrA Symbol Library. As their name suggests, Situation Awareness Library symbols are designed to enhance an operator’s situational awareness of current process conditions using a variety of visual techniques.

Situation Awareness Library symbols use a largely monochrome color palette to increase the contrast between normal operating conditions and a vivid colored alarm state. Situation Awareness Library symbols are not photo realistic and provide just enough visual detail to recognize their functional purpose without showing extraneous information to operators.

Situation Awareness Library symbols include an extensive set of default custom properties. These custom properties can be set to show or hide parts of the symbol itself, set the full array of alarm conditions, and show reported values based on the symbol’s configuration.

Situation Awareness Library symbols also support multiple visual and functional configurations. Using the Symbol Wizard, a single Situation Awareness Library symbol can support multiple configurations by showing or hiding functional elements as options that are part of the default symbol set. For more information about configuring Situation Awareness Library symbols with the Symbol Wizard.

Situation Awareness Library symbols incorporate a variety of animations that enable operators to quickly assess current process conditions. Animations particularly suited for Situation Awareness Library symbols include:

- Point animation
- Alarm Border animation
- Polar star animation
- Sweep angle animation
GENERAL

METERS

Meters have been created for both graphical and numeric presentation of information. For the graphical meters, the intent was to continually represent PV with respect to SP, alarm limits and a reference value (tracker). Additional information, such as a numeric presentation of the PV and tagname, is toggleable thereby allowing the operator to somewhat customize the graphic page.

Meters are connected to flow lines using “meter markers” as are shown below. The site may or may not use the page navigation as shown.

![Example of flow meters connected to process line](image)

METER DIGITAL PV

Level 2 meters have the capability to toggle the visibility of the numeric PV and engineering units. The intent is to provide this functionality globally on a workstation via a button on the toolbar. Additionally, the operator can also toggle the numeric PV and engineering units from a right click menu.

On Level 1 meters, the visibility of the numeric PV will be defaulted to off. When the PV goes outside the “target range”, the visibility will be toggled to “on”. If no target range is set, then the numeric PV will toggle on when the PV hits an alarm limit.

The font chosen for the digital PV is Lucida Console 12 pt, or similar. If possible, we would prefer that the font is adjustable to accommodate the as-built system requirements.

OPTIMAL RANGE

The functionality and description for the optimal range on a meter has been described for both the target meter and the thermometer. This functionality is expected for all meters. If the range is not set, then the grey background will not appear. Additionally, an operator should have the option to show/hide and to change the range displayed. This can be done in a point detail or right click.
NORMAL OPERATING RANGE VS. FULL RANGE

Each meter has been given the functionality of setting both a normal operating range (a.k.a. practical range) as well as being able to display the meter’s full range. The reason for this is because during most operating situations, the meter’s full range is often too broad as compared to the region of interest. To differentiate a meter that is in its full range, additional flags will appear at the top of the meter and the meter background line will turn to black, as in the example below:

Example of flow meters a) normal range and b) full range

AUTO-SCALING

Auto-Scaling means fixing the setpoint at the center of the meter. This affords comparisons between meters when straight line profiles are desired. The majority of the meters will have the option to auto scale. For certain objects such as the polar star, the spokes will need to be normalized in order to obtain the desired performance.
AUTO RANGING

Every instrument has a full scale range, but it might be beyond a range that is useful for operations. For example, a temperature probe may have a full scale range of 0-1000°C, but the operators may only need to see 100-200°C when operating. Therefore, during normal operations, the meter should be “zoomed in” to its “normal” range of 100-200°C. In order to accommodate the operating situation when the PV extends beyond the normal range, the meter will automatically switch between “normal” and “full scale”. This automatic switching is referred to as “auto ranging”.

GRAPHIC TOOLBOX OVERVIEW SYMBOLS
USING SA SYMBOLS

Symbols can be used for InTouch or Application Server, All custom properties are pre populated.

IN ITS MOST SIMPLE FORM THE WIZARD IS CONFIGURED FOR MINIMAL IMPLEMENTATION:

Flow Meter:

Simple Wizard Options (default)

Resulting in these custom properties: “CTRL M”
Linking the Custom Properties: “CTRL E”

Example of linked symbol:
When Advanced is chosen on a SA symbol all the more complex and elaborate features become available:

![Symbol Options Table]

This results in more custom properties to be linked.

![Property Options Table]
### General Control Meter Object Template

#### Dynamic Objects

D1: Commanded OP

#### Static Objects

#### Other Objects used in Graphic

- O1: OP bar object object
- O2: Status Indication for Abnormal Mode
- O3: Standard Meter object (could be any meter)

#### Note:

- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
SYMBOL EXAMPLES WITH ALARMS

Flow Control Meter

Analyzer Control Meter

Temperature Control Meter

Pressure Control Meter

Target Control Meter
CONTROL METER

General Control Meter Object Template

![Diagram](image)

**DYNAMIC OBJECTS**

D1: Commanded OP

**STATIC OBJECTS**

**Other OBJECTS used in Graphic**

O1: OP bar object object
O2: Status Indication for Abnormal Mode
O3: Standard Meter object (could be any meter)

**NOTE:**

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags

The general control meter object is an extension of the standard meter indication. In addition to the parameters of a regular meter (O3), an OP bar (D1) and the status indication for abnormal mode (O2) are also shown.

Basically, a control meter allows an operator to make a control change to the system, whereas a regular meter is simply an indication.

The commanded output is represented by the triangle (D1)
**ANALYZER OBJECT**

**DYNAMIC OBJECTS**

- D1: PV
- D2: SP
- D3: High High Alarm Point
- D4: High Alarm Point
- D5: Low Alarm Point
- D6: Low Low Alarm Point
- D7: Numerical presentation of PV (works with D1), toggles on L2
- D8: Tracker
- D9: Full Range Indication
- D10: Direction of Change Indication

**STATIC OBJECTS**

- S1: Background shape (Works with D1)
- S2: Engineering Units (toggles on L2)
- S3: Analyzer Tag Number (toggles on L2)
- S4: Analyzer descriptor (toggles on L2)

**Other OBJECTS used in Graphic**

- O1: Status indication for quality of data
- O2: Alarm border
- O3: Clock timer object

**NOTE:**

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Similar to flow indications, analyzers shall be indicated on a vertical linear scale. PV (D1) will be presented as a bold rectangle. SP (D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D6) and will be set in the operator control zone.

The analyzer shall be presented adjacent to the equipment at the level where the meter is located. Because there are many types of analyzers, a short textual descriptor (S4) shall be used to indicate what is being measured, in conjunction with the tag name (S3). D7 and S2 can be toggled on or off together by the operator, as can S3 and S4.

To differentiate analyzers, the background will consist of a rectangle with rounded off corners (S1). Analyzer deviation will be presented in the same manner as flow deviations, discussed above.

Deviations from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indicators, a provision within the module should be provided that will allow engineering staff to place a tolerance (as a % change of presented scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance. The logic for the direction of change (D10) will be based on the rate of change flag.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter's full range. The full range flag (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (S1) will turn black and the digital PV value will automatically appear on the L2L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indication object description for more information.

Analyzer indications can be vertical or horizontal.

Alarms (O2) should be indicated as shown in the sub-picture examples, using the standard alarm template.

Because some analyzers are not continuously reading (not real-time), a clock object (O3) has been added. If the meter is a real-time analyzer, the clock object will not be required. For example, if the meter takes 5 minutes between readings, the clock object will take 5 minutes to fill completely before it starts over again. It will likely be too difficult to provide a "smooth sweep" for the clock fill. It would be sufficient to divide the clock fill into ten, or more, discrete "chunks" to provide the timing animation.
**DEVIATION SYMBOL**

Deviation Meter Object Template

- **D1**: PV
- **D2**: SP
- **D3**: High High Alarm Point
- **D4**: High Alarm Point
- **D5**: Low Alarm Point
- **D6**: Low Low Alarm Point
- **D7**: Numerical presentation of PV (works with **D1**), toggles on L2
- **D8**: Tracker
- **D9**: Full Range Indication
- **D10**: Direction of Change Indication

**STATIC OBJECTS**

- **S1**: Background shape (Works with **D1**)
- **S2**: Engineering Units (toggles on L2)
- **S3**: Tag Name (toggles on L2)
- **S4**: Deviation Meter descriptor

**Other OBJECTS used in Graphic**

- **O1**: Status indication for quality of data
- **O2**: Alarm border

**NOTE:**

- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
In some cases, operations might be interested in a deviation from a setpoint, rather than an absolute value for a particular parameter. The deviation meter was created for this purpose.

The deviation meter shall be presented adjacent to the equipment at the point where the meter is located. Because there are potentially many uses for the deviation meter, a short textual descriptor (S4) shall be used to indicate what is being measured, in conjunction with the tag name (S3). PV and ZP can be toggled on or off together by the operator, as can S3 and S4.

The deviation meter indication is presented on a vertical linear scale, containing a filled rectangle as a PV indicator (D1). SP(D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D9) and L (D8) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D8), on left hand side for H (D9)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D8) and will be set in the operator control zone.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indications, a provision within the module should be provided that will allow engineering staff to place a tolerance level (as a % change of presented scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance level. The logic for the direction of change (D10) will be based on the rate of change flag.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flag indication (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (S1) will turn black and the digital PV value will automatically appear on the L2L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indication object description for more information.

Deviation meter indications can be vertical or horizontal

Alarms (O2) should be indicated as shown in the sub-picture examples, using the standard alarm template.

The configured input range for the deviation meter must be in equal proportions around a zero (for example -200 to +200 or -50 to +50). The PV will be shown as a deviation from setpoint, where zero = setpoint.
FLOW METER SYMBOL

Flow Meter Object Template

DYNAMIC OBJECTS
D1: PV
D2: SP
D3: High High Alarm Point
D4: High Alarm Point
D5: Low Alarm Point
D6: Low Low Alarm Point
D7: Numerical presentation of PV (works with D1), toggles on L2
D8: Tracker
D9: Full Range Indication
D10: Direction of Change Indication

STATIC OBJECTS
S1: Background shape (Works with D1)
S2: Engineering Units (toggles on L2)
S3: Tag Name (toggles on L2)
S4: Meter limits

Other OBJECTS used in Graphic
O1: Status indication for quality of data
O2: Alarm border

NOTE: * Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Flow indicators should be presented adjacent to equipment and to a piping segment. The flow indicators are intended to show deviations from setpoint in flow.

The flow indication is presented on a vertical linear scale, containing a “floating ball” PV indicator (D1), similar to commonly found physical flow meters. SP (D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D8) and will be set in the operator control zone. The type of the meter (S4) will appear if there is an alarm limit shown in order to better define the meter range visually.

The sub-picture examples are configured such that the “standard” object features (normal, abnormal and alarm limits reached) are shown in the first row. We have provided recommendations for the site specific features, however, all efforts should be made to accommodate the site’s standards for Bad PV, Out of Service or Off and Suppressed as follows:

When a bad value signal is obtained, the PV indicator should disappear and the appropriate status indication icon (from the Status indication template) should appear. When a meter is suppressed, the PV may still be active, but the alarming will not activate and the appropriate status indication icon should appear - note that if the meter goes into alarm, the appropriate border for a suppressed, or “shelved”, alarm will appear. When a meter is Out of Service or Off, the PV indicator should disappear and the appropriate status indication icon should appear. PV status indicators should be indicated as described in the Status indicator object template description. The status indicator will appear (D1) when active.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flags indication (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (S1) will turn black and the digital PV value will automatically appear on the L2L3 meters.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indicators, a provision within the module should be provided that will allow engineering staff to place a tolerance level (as a % change of presented scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance level.

The logic for the direction of change (D10) will be based on the rate of change flag.

Flow indications can be vertical or horizontal.
LEVEL OBJECT SYMBOL

DYNAMIC OBJECTS

D1: PV
D2: SP
D3: High High Alarm Point
D4: High Alarm Point
D5: Low Alarm Point
D6: Low Low Alarm Point
D7: Numerical presentation of PV (works with D1), toggles on L2
D8: Tracker
D9: Full Range Indication
D10: Direction of Change Indication

STATIC OBJECTS

S1: Background shape (Works with D1)
S2: Engineering Units (toggles on D1)
S3: Tag Number (toggles on D1)

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border

NOTE:
* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Level indications should be presented within the vessel, if possible. The level indicators are intended to show deviations from setpoint in level.

The level indication is presented on a vertical linear scale, containing a filled rectangle as a PV indicator (D1). SP (D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D6) and will be set in the operator control zone.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indicators, a provision within the module should be provided that will allow engineering staff to place a tolerance level (as a % change of present scale, or an absolute value for an acceptable deviation) within which no arrow will be presented.

The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance level. The logic for the direction of change (D10) will be based on the rate of change flag.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flag indication (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (D1) will turn black and the digital PV value will automatically appear on the L3L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (D1) when active. See the section on the status indication object description for more information.

Level indications can be vertical or horizontal.
LEVEL COMPARISON SYMBOL

2 Level Comparison Object Template

DYNAMIC OBJECTS

D1A: PV for vessel 1
D1B: PV for vessel 2
D2: SP
D3: High High Alarm Point
D4: High Alarm Point
D5: Low Alarm Point
D6: Low Low Alarm Point
D7: Numerical presentation of PV (works with D1), toggles on L2
D8: Tracker
D9: Full Range Indication
D10: Direction of Change Indication

STATIC OBJECTS

S1: Background shape (Works with D1)
S2: Engineering Units (toggles on L2)
S3: Tag Number (toggles on L2)

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border

NOTE: *
Denotes a parameter that is already provided to the operator
#
Denotes a parameter that likely requires a tag to be created
π
Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
This object is a permutation of the Level Meter. It is used in cases where the offline tank still needs to be monitored, or when there are two tanks that have the same alarm points are monitored. As with other level indications, these should be presented within the vessel, if possible. The meter representing the online tank (or the controlling meter) is grey and the offline (or meter that isn’t controlling) is shown in a lighter grey.

The level indication is presented on a vertical linear scale, containing a filled rectangle as a PV indicator (D1A) for vessel 1 and PV indicator (D1B) for vessel 2. SP (D3) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D8) and will be set in the operator control zone.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indicators, a provision within the module should be provided that will allow engineering staff to place a tolerance level (as a % change of presented scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance level. The logic for the direction of change (D10) will be based on the rate of change flag.

Engineering staff have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flag indication (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (SI) will turn black and the digital PV value will automatically appear on the L2L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indication object description for more information.

Level indications can be vertical or horizontal.
MULTI-LEVEL METER SYMBOL

Multi Level Meter Object Template

DYNAMIC OBJECTS
D1: 1st level bar fill
D2: 2nd level bar fill
D3: 3rd level bar fill

STATIC OBJECTS
S1: Meter Label
S2: Level border
S3: Vertical axis hash marks (not shown on L1, toggles on L2)
S4: Engineering Units (not shown on L1, toggles on L2)

Other OBJECTS used in Graphics
O1: Status indication for quality of data
O2: Alarm border
O3: Alarm flags for 1st level (works with D1)
O4: Alarm flags for 2nd level (works with D2)
O5: Alarm flags for 3rd level (works with D3)

NOTE:  
* Denotes a parameter that is already provided to the operator  
# Denotes a parameter that likely requires a tag to be created  
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The Multi Level meter object is used to show an accumulated level within a vessel of distinguishable fluid levels (such as muddy and cloudy filtrate levels in a settler tank) or simply to show a total contribution of measures in a process such as total input oxygen levels in the plant from various sources.

To help reduce clutter on the operator screen and considering operational requirements, it was decided that the multi level meter shall have a maximum of three levels ($D_1$, $D_2$, $D_3$). Each level measurement may have associated alarm Hi/Lo flags ($O_3$, $O_4$, $O_5$) which are settable by the operator via the operator control zone and will trigger the meter alarm ($O_2$) as per standard alarming conventions. Status indication for quality of data is provided by $O_1$.

The meter may be stretched vertically ($S_2$ will be shorter or taller) to accommodate higher measurement resolution situations ($1^{st}$ level measure has a very narrow operating zone compared to the $2^{nd}$ level measure). The value of each level shall be measured from an origin point defined by the top of the level measurement below it. For the bottommost measurement, its origin point is measured from the bottom of the meter. Hash marks ($S_3$) shall be spaced at an appropriate interval and settable by the operator in the operator control zone. The engineer units label ($S_4$) can be toggled on an I2 screen by the operator, and $S_7$ indicates the vessel or measure in question.
MISCELLANEOUS METER SYMBOL

### Dynamic Objects

- **D1**: PV
- **D2**: SP
- **D3**: High High Alarm Point
- **D4**: High Alarm Point
- **D5**: Low Alarm Point
- **D6**: Low Low Alarm Point
- **D7**: Numerical presentation of PV (works with **D1**), toggles on L2
- **D8**: Tracker
- **D9**: Full Range Indication
- **D10**: Direction of Change Indication

### Static Objects

- **S1**: Background shape (Works with **D1**)
- **S2**: Engineering Units (toggles on L2)
- **S3**: Tag Name (toggles on L2)
- **S4**: Miscellaneous Meter descriptor

### Other Objects used in Graphic

- **O1**: Status indication for quality of data
- **O2**: Alarm border

**NOTE:**
- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Where an item being measured does not fall under categories such as temperature, pressure, flow and level, then a simple bar graph shall be used to represent the parameter. Pump amperes is an example of such a parameter.

The miscellaneous meter shall be presented adjacent to the equipment at the point where the meter is located. Because there are potentially many uses for the miscellaneous meter, a short textual descriptor (S4) shall be used to indicate what is being measured, in conjunction with the tag name (S3). L7 and S9 can be toggled on or off together by the operator, as can S3 and S4.

The miscellaneous meter indication is presented on a vertical linear scale, containing a filled rectangle as a PV indicator (D1). SP (D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D8) and will be set in the operator control zone.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indicators, a provision within the module should be provided that will allow engineering staff to place a tolerance level (as a % change of presented scale, or an absolute value for an acceptable level) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance level.

The logic for the direction of change (D10) will be based on the rate of change flag.

Operators have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flags indication (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (S1) will turn black and the digital PV value will automatically appear on the L2/L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indication object description for more information.

Miscellaneous meter indications can be vertical or horizontal.

Alarms (O2) should be indicated as shown in the sub-picture examples, using the standard alarm template.
DYNAMIC OBJECTS

D1: PV
D2: SP
D3: High High Alarm Point
D4: High Alarm Point
D5: Low Alarm Point
D6: Low Low Alarm Point
D7: Numerical presentation of PV (works with D1), toggles on L2
D8: Tracker
D9: Full Range Indication
D10: Direction of Change Indication

STATIC OBJECTS

S1: Background shape (Works with D1)
S2: Engineering Units (toggles on L2)
S3: pressure Tag Number (toggles on L2)

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border

NOTE:
* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Similar to flow indications, pressures shall be indicated on a vertical linear scale. To differentiate pressure indication (PI) from flow indication (FI), the PI scale will be drawn with an oval background (SF), as opposed to the FI, which is presented with a simple line background. PV deviation will be presented in the same manner as flow deviations, discussed above.

The pressure meter shall be presented adjacent to the equipment at the point where the meter is located. The pressure indicators are intended to show deviations from setpoint in level.

The pressure indication is presented on a vertical linear scale, containing a filled rectangle as a PV indicator (D1). SP (D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D8) and will be set in the operator control zone.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indications, a provision within the module should be provided that will allow engineering staff to set the tolerance pressure (as a % change of presented scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance pressure. The logic for the direction of change (D10) will be based on the rate of change flag.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flags indication (D8) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (SI) will turn black and the digital PV value will automatically appear on the L2L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indication object description for more information.

Pressure indications can be vertical or horizontal

Alarms (O2) should be indicated as shown in the sub-picture examples, using the standard alarm template.
RPM INDICATION SYMBOL

RPM Indication Object

DYNAMIC OBJECTS

D1: RPM indicator. Works with D2.
D3: High High Alarm Point
D4: High Alarm Point
D5: Low Alarm Point
D6: Low Low Alarm Point
D7: Numerical presentation of PV (works with D1), toggles on L2

STATIC OBJECTS

S1: Background shape of rpm meter
S2: Engineering Units of PV (toggles on L2)
S3: RPM Meter Tag Number (toggles on L2)
S4: Full Scale Indication

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm Background
O3: Controller indication (only visible if meter is a controller)

NOTE:  
* Denotes a parameter that is already provided to the operator  
# Denotes a parameter that likely requires a tag to be created  
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
RPM meter - this meter, which resembles a tachometer, is intended to show deviation from RPM setpoint (i.e. for a compressor). The black needle with the arrow (D1) indicates RPM at setpoint and the grey fill behind the arrow (D2) indicates the tolerance range for RPM fluctuation in the reading. HH (D3), LL (D6), H(D4) and L (D5) are presented as tick marks and indicate alarm limits.

PV status indicators follow the same coding used in the Level 3, AI and Loop dynamics. The status indicator when active will show to the right of the meter (D1). See the section on status indicator description for more information.

Alarms should be indicated as shown in the Examples, using the standard alarm template.
TARGET METER SYMBOL

Target Meter Object Template

DYNAMIC OBJECTS

D1: PV
D2: SP
D3: High High Alarm Point
D4: High Alarm Point
D5: Low Alarm Point
D6: Low Low Alarm Point
D7: Numerical presentation of PV (works with D1), toggles on L2
D8: Tracker
D9: Full Range Indication
D10: Direction of Change Indication
D11: Range of good operation

STATIC OBJECTS

S1: Background shape (Works with D1)
S2: Engineering Units (toggles on L2)
S3: Tag Number (toggles on L2)
S4: Meter limits

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border

NOTE:  *
Denotes a parameter that is already provided to the operator
#
Denotes a parameter that likely requires a tag to be created
π
Denotes a parameter that requires a tag to be created
and requires a calculation based on available tags
A target indicator will normally be shown as part of a calculation.

The target meter meter shall be presented adjacent to the equipment at the point where the meter is located. The target meter indicators are intended to show deviations from set point in level.

The target indication is presented on a vertical linear scale, containing a “bullseye” result indicator \( D_1 \) as a PV indicator. The bullseye consists of a ball, identical to the indicator used on a standard flow meter. A circle (no fill) is drawn around the ball to give the appearance of a bullseye. \( SP(D_2) \) is presented as a horizontal tick mark. \( HH(D_5) \) and \( LL(D_6) \) are presented as horizontal tick marks with triangles or flags (on right hand side for \( LL(D_6) \), on left hand side for \( HH(D_5) \)) at the end of the tick marks. \( H(D_4) \) and \( L(D_3) \) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for \( L(D_3) \), on left hand side for \( H(D_4) \)) at the end of the tick mark. An arrow indicating a direction of change \( (D10) \) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well \( (D8) \) and will be set in the operator control zone.

An additional “target range” feature \( (D11) \) has been provided for temperatures to accommodate profiles within vessels. This range will be set by WAPL operations.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indicators, a provision within the module should be provided that will allow engineering staff to place a tolerance (as a % change of present scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance. The logic for the direction of change \( (D10) \) will be based on the rate of change flag.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flags indication \( (D9) \) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line \( (S1) \) will turn black and the digital PV value will automatically appear on the L2/L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear \( (O1) \) when active. See the section on the status indication object description for more information.

Target meter indications can be vertical or horizontal.

Alarms \( (O2) \) should be indicated as shown in the sub-picture examples, using the standard alarm template.
### TEMPERATURE METER SYMBOL

**Temperature Object Template**

**DYNAMIC OBJECTS**

- D1: PV
- D2: SP
- D3: High High Alarm Point
- D4: High Alarm Point
- D5: Low Alarm Point
- D6: Low Low Alarm Point
- D7: Numerical presentation of PV (works with D1), toggles on L2
- D8: Tracker
- D9: Full Range Indication
- D10: Direction of Change Indication
- D11: Range of good operation

**STATIC OBJECTS**

- S1: Background shape (Works with D1)
- S2: Engineering Units (toggles on L2)
- S3: temperature Tag Number (toggles on L2)

**Other OBJECTS used in Graphic**

- O1: Status indication for quality of data
- O2: Alarm border

**NOTE:**

- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Temperature indication should be presented in a form that mimics a thermometer. Presented adjacent to equipment or to a piping segment, the temperature indicators are intended to show deviations from set point in temperature.

The thermometer is presented on a vertical linear scale, containing a bar graph, D1 (intended to mimic the behavior of mercury) as a PV indicator, similar to commonly found physical thermometers. The temperature indication is presented on a vertical linear scale, containing a filled rectangle as a PV indicator (D1). SP(D2) is presented as a horizontal tick mark. HH (D3) and LL (D6) are presented as horizontal tick marks with triangles or flags (on right hand side for LL (D6), on left hand side for HH (D3)) at the end of the tick marks. H (D4) and L (D5) are presented as horizontal tick marks with 90 degree angle lines (on right hand side for L (D5), on left hand side for H (D4)) at the end of the tick mark. An arrow indicating a direction of change (D10) will appear based on the rate of change flag. A reference for the PV, known as a tracker, will be provided as well (D8) and will be in the operator control zone.

An additional “target range” feature (D11) has been provided for temperatures to accommodate profiles within vessels. This range will be set by WAPL operations.

Deviation from the last PV is indicated with a simple arrow showing the direction of movement. If there is no change, no arrow will be presented. To prevent the proliferation of nuisance arrow indications, a provision within the module should be provided that will allow engineering staff to place a tolerance temperature (as a % change of present scale, or an absolute value for an acceptable deviation) within which no arrow will be presented. The PV rate of change arrow shows if the absolute value of the rate of change is greater than the tolerance temperature. The logic for the direction of change (D10) will be based on the rate of change flag.

Engineering have the ability to set a narrow range for each meter. By default, the narrow range of the meter will be set to the meter’s full range. The full range flags indication (D9) will appear only if the meter reading extends beyond the narrow range. In addition, the meter line (S1) will turn black and the digital PV value will automatically appear on the L22L3 meters.

PV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indication object description for more information.

Temperature indicators can be vertical or horizontal. When they are horizontal, they should be rotated 90 degrees to the right, with the bulb of the thermometer on the left hand side of the drawing. An example is shown in the Examples.

Alarms (O2) should be indicated as shown in the sub-picture examples, using the standard alarm template.
KPI BAR SYMBOL

KPI Bar Object Template

DYNAMIC OBJECTS
D1: PV (note: normalised to a dollar value)
D2: Tracker

STATIC OBJECTS
S1: Background shape (Works with D1)

Other OBJECTS used in Graphic
O1: Status indication for quality of data – standard size object used

NOTE:
• Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The KPI Bar is a special version of the Deviation Meter. It is larger, and intended to mainly be used on the Level 1 overview displays.

The intent of the KPI bar is to allow operators and management to compare KPIs on a common scale. The KPIs are normalized to a dollar value (configurable at design time). As a default, we have chosen $5000/day, but that will have to be modified to accommodate the actual plant ranges. This allows the operator to equally compare the effects of pushing each of the KPI limits.

The KPIs have a zero (neutral) dollar value in the middle. If the bar (D1) fills upward, then the plant is making money, as compared to the KPI expected value, otherwise they are losing money for the unit on a specific KPI.

A reference for the FV, known as a tracker, will be provided as well (D2) and will be set in the operator control zone. The tracker can therefore move and is not static.

FV status indicators follow the same coding used in the flow meter object. The status indicator will appear (O1) when active. See the section on the status indicator object description for more information.

KPI bars can be vertical or horizontal. When rotated horizontally, the bar fills to the right for a positive performance (plant making money).
**INVISIBLE PROFILE METER SYMBOL**

Invisible Profile Meter Object Template

![Invisible Profile Meter Object](image)

**NOTE:**

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
\(\pi\) Denotes a parameter that requires a tag to be created and requires a calculation based on available tags

<table>
<thead>
<tr>
<th>O1</th>
<th>Status indication for quality of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2</td>
<td>Alarm border</td>
</tr>
<tr>
<td>O3</td>
<td>Basic Flow Meter Object</td>
</tr>
</tbody>
</table>

The purpose of this object is to facilitate the creation of profile graphics. Meter connector objects are used to link the FVs from a series of Invisible Profile meters in order to create a profile (see examples page). At runtime, the meter itself is invisible to users. Please see sub-picture examples for various states of visibility for components.
FLOW TRACKER SYMBOL

Small Flow Tracker Object Template

STATIC OBJECTS

S1: Object fence
S2: Upper range limit (settable by operator)
S3: Lower range limit (settable by operator)

DYNAMIC OBJECTS

D1: PV
D2: Tracker

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The small flow tracker is designed to provide information on a flow that is important to a process but does not operate on any setpoints or alarms and therefore may be represented in a more visually efficient manner as compared to the existing Flow Meter object. An example application for the meter may be to measure condensate flow to a filter as it helps indicate usage of condensate (which is important in terms of dollar cost), but does not immediately translate into the quality of the filtration process (filtration flow, which has operating setpoints, is the more important measure of the filtration process).
L3 DATA OBJECT SYMBOL

L3 Data Block Object Template

![Diagram of L3 Data Block Object Template]

**DYNAMIC OBJECTS**

D1: PV

**STATIC OBJECTS**

S1: Tag Name (can be toggled)
S2: Engineering units (can be toggled)

**Other OBJECTS used in Graphic**

O1: Status identifier
O2: Alarm Object

**NOTE:**

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags

This object is used only for level 3 and 4 graphics to provide data to the operator. Its function is to allow operations an alternate visual form other than the standard meter.

Functionality for D2 is shown in the example. When the PID/OUT high limit has been exceeded, the bar will fill black. When the PID/OUT low limit has been exceeded the bar will show a black outline.

Alarming is handled using the alarm object template.
OP BAR OUTPUT SYMBOL

OP Bar Object Template

DYNAMIC OBJECTS
D1: Bar fill representing OP value

STATIC OBJECTS
S1: Background shape (works with D1)

Other OBJECTS used in Graphics
O1: Alarm border

NOTE: * Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The OP object shall be used to illustrate the current position of the OP (output) value of a tag. The grey fill (D1) fills from the left side to the right of the background shape (S1). The scale of the fill is such that the total width of S1 represents 100% of the OP value. Thus, if D1 fills exactly halfway to the right, OP value equals to 50%. If OP value is at 0%, the object will be a white rectangle, with a black outline (S1), and D1 will have no fill width. At >=100%, the fill (D1) changes to black to let the operator know that the controller is incapable of controlling further.

This object can be associated with any existing objects that have an OP bar.
LOOPBLOCK SYMBOL

L3 Loop Block Object Template

**DYNAMIC OBJECTS**

D1: PV
D3: Tag name or Set point value - these will toggle

**STATIC OBJECTS**

S1: Background lines
S2: Engineering Units

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm information
O3: Status indication for Mode (only when mode is in abnormal state)
O4: OP bar

**NOTE:**

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags

This object is used only for level 3 and 4 graphics to provide data to the operator. Its function is to allow operations an alternate visual form other than the standard meter.

**D3** can be toggled by the operator to show Set point value instead of Tagname.

When not in the normal mode, the abnormal mode indication **(O3)** pops up and the background changes colour as well. More information on the mode indication can be found in the status indication template.

Alarming is handled using the alarm object template
METER VOTING SYMBOL

Meter Voting

![Meter Voting Diagram]

NOTE:
- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags

OTHER OBJECTS

O1: Alarm Border
O2: Status Indication for the quality of data
O3: Multiple Equipment Object (see examples)
The meter voting shape is intended to show the operators whether multiple meters involved in voting agree or not.

Each grey block represents one meter (as in the “Sub-picture examples” page).

Standard alarming is used for this shape.
# 3 SPOKE POLAR STAR SYMBOL

![3-Spoke Polar Star Object Template](image)

## DYNAMIC OBJECTS

- **D1**: Polar Star Shape
- **D2**: Alarm highlight (works with \( O_2 \))
- **D3**: Alarm highlight (spoke that trips alarm is highlighted, works with \( O_2 \))

## STATIC OBJECTS

- **S1**: Spoke Identifiers. Toggleable
- **S2**: Expected polygon shape
- **S3**: Polar Star hollow center

## Other OBJECTS used in Graphic

- **O1**: Status indication for quality of data
- **O2**: Alarm border
- **O3**: Polar Star spokes: consists of a flow meter object without full range indicator, PV circle, SP, and tracker. Flow meter object is used although flow is not measured. (Works with \( D_1 \))

### NOTE:

- \* Denotes a parameter that is already provided to the operator
- \# Denotes a parameter that likely requires a tag to be created
- \( \pi \) Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The polar star has been used for monitoring purposes for many decades (Cookin, 1969). However, its use is limited to very few applications where information is for the most part static yet needs to be monitored.

The concept of the polar star has to do with the perceptual ability of humans to quickly recognize shapes. Each spoke is dedicated to one measured parameter (O3) and the 3 parameters presented are normalized such that a triangle is created (S2). Whenever one of the parameters strays from its SP, the salient shape of the triangle will be deformed (D1 and D2) and the operator will be able to quickly adapt to the abnormal parameter. The specific order in which the parameters are placed on the spokes is also of ultimate importance, as the deformation of the triangle will depend on the specific location of each of the spokes. Spokes should be placed such that similar indications would be grouped together. To enhance the capability of the operator to notice a deviation from normal, a dotted triangle will represent normal conditions (S2).

The spoke labels S1 can be toggled on or off by the operator, and the alarm flags on each spoke (O3) can be set in the operator control zone. When an alarm is triggered by an alarm flag, the spoke that triggered the alarm will be highlighted in the same color (D3) as the alarm border (O2). Status indication for quality of data is shown in O1.

The polar star object is intended to highlight movement from a stable operating position. Because unit operations have multiple stable operating conditions, e.g., at various throughputs, it is imperative that the capability to “re-normalize” the shape of the polar star is incorporated as a standard function of the object. This is not as important for the case of implementing a polar star for process utilities, but in cases where the object is used to display unit operations, the ability to “re-normalize” the shape will become absolutely necessary.
4 SPOKE POLAR STAR SYMBOL

4-Spoke Diamond Polar Star
Object Template

DYNAMIC OBJECTS
D1: Polar Star Shape
D2: Alarm highlight (works with O2)
D3: Alarm highlight (spoke that trips alarm is highlighted, works with O2)

STATIC OBJECTS
S1: Spoke Identifiers. Toggleable
S2: Expected polygon shape
S3: Polar Star hollow center

Other OBJECTS used in Graphic
O1: Status indication for quality of data
O2: Alarm border
O3: Polar Star spokes: consists of a flow meter object without full range indicator, PV circle, SP, and tracker. Flow meter object is used although flow is not measured. (Works with D1)

NOTE: * Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The polar star has been used for monitoring purposes for many decades (Coekin, 1969). However, its use is limited to very few applications where information is for the most part static yet needs to be monitored.

The concept of the polar star has to do with the perceptual ability of humans to quickly recognize shapes. Each spoke is dedicated to one measured parameter (O3) and the 4 parameters presented are normalized such that a diamond is created (S2). Whenever one of the parameters strays from its SP, the salient shape of the diamond will be disfigured (D1 and D2) and the operator will be able to quickly adapt to the abnormal parameter. The specific order in which the parameters are placed on the spokes is also of ultimate importance, as the deformation of the triangle will depend on the specific location of each of the spokes. Spokes should be placed such that similar indications would be grouped together. To enhance the capability of the operator to notice a deviation from normal, a dotted diamond will represent normal conditions (S2).

The spoke labels (S1) can be toggled on or off by the operator, and the alarm flags on each spoke (O3) can be set in the operator control zone. When an alarm is triggered by an alarm flag, the spoke that triggered the alarm will be highlighted in the same colour (D3) as the alarm border (O2). Status indication for quality of data is shown in O1.

The polar star object is intended to highlight movement from a stable operating position. Because unit operations have multiple stable operating conditions, e.g. at various throughputs, it is imperative that the capability to “re-normalize” the shape of the polar star is incorporated as a standard function of the object. This is not as important for the case of implementing a polar star for process utilities, but in cases where the object is used to display unit operations, the ability to “re-normalize” the shape will become absolutely necessary.
**6 SPEKE POLAR START SYMBOL**

![6-Spoke Polar Star Object Template]

### DYNAMIC OBJECTS

- **D1**: Polar Star Shape
- **D2**: Alarm highlight (works with O2)
- **D3**: Alarm highlight (spoke that trips alarm is highlighted, works with O2)

### STATIC OBJECTS

- **S1**: Spoke Identifiers. Toggleable
- **S2**: Expected polygon shape
- **S3**: Polar Star hollow center

### Other OBJECTS used in Graphic

- **O1**: Status indication for quality of data
- **O2**: Alarm border
- **O3**: Polar Star spokes: consists of a flow meter object without full range indicator, PV circle, SP, and tracker. Flow meter object is used although flow is not measured. (Works with D1)

### NOTE:

- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The polar star has been used for monitoring purposes for many decades (Coekin, 1969). However, its use is limited to very few applications where information is for the most part static yet needs to be monitored.

The concept of the polar star has to do with the perceptual ability of humans to quickly recognize shapes. Each spoke is dedicated to one measured parameter (O3) and the 6 parameters presented are normalized such that a hexagon is created (S2). Whenever one of the parameters strays from its SP, the salient shape of the hexagon will be deformed (D1 and D2) and the operator will be able to quickly adapt to the abnormal parameter. The specific order in which the parameters are placed on the spokes is also of ultimate importance, as the deformation of the triangle will depend on the specific location of each of the spokes. Spokes should be placed such that similar indications would be grouped together. To enhance the capability of the operator to notice a deviation from normal, a dotted hexagon will represent normal conditions (S2).

The spoke labels S1 can be toggled on or off by the operator, and the alarm flags on each spoke (O3) can be set in the operator control zone. When an alarm is triggered by an alarm flag, the spoke that triggered the alarm will be highlighted in the same colour (D3) as the alarm border (O2). Status indication for quality of data is shown in O1.

The polar star object is intended to highlight movement from a stable operating position. Because unit operations have multiple stable operating conditions, eg at various throughputs, it is imperative that the capability to "re-normalize" the shape of the polar star is incorporated as a standard function of the object. This is not as important for the case of implementing a polar star for process utilities, but in cases where the object is used to display unit operations, the ability to "re-normalize" the shape will become absolutely necessary.
8 SPEKE POLAR STAR SYMBOL

8-Spoke Polar Star Object Template

DYNAMIC OBJECTS
D1: Polar Star Shape
D2: Alarm highlight (works with O2)
D3: Alarm highlight (spoke that trips alarm is highlighted, works with O2)

STATIC OBJECTS
S1: Spoke Identifiers. Toggleable
S2: Expected polygon shape
S3: Polar Star hollow center

Other OBJECTS used in Graphic
O1: Status indication for quality of data
O2: Alarm border
O3: Polar Star spokes: consists of a flow meter object without full range indicator, PV circle, SP, and tracker. Flow meter object is used although flow is not measured. (Works with D1)

NOTE:
* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The polar star has been used for monitoring purposes for many decades (Coekin, 1969). However, its use is limited to very few applications where information is for the most part static yet needs to be monitored.

The concept of the polar star has to do with the perceptual ability of humans to quickly recognize shapes. Each spoke is dedicated to one measured parameter (O3) and the 8 parameters presented are normalized such that a octagon is created (S2). Whenever one of the parameters strays from its SP, the salient shape of the octagon will be disfigured (D1 and D2) and the operator will be able to quickly adapt to the abnormal parameter. The specific order in which the parameters are placed on the spokes is also of ultimate importance, as the deformation of the triangle will depend on the specific location of each of the spokes. Spokes should be placed such that similar indications would be grouped together. To enhance the capability of the operator to notice a deviation from normal, a dotted octagon will represent normal conditions (S2).

The spoke labels S1 can be toggled on or off by the operator, and the alarm flags on each spoke (O3) can be set in the operator control zone. When an alarm is triggered by an alarm flag, the spoke that triggered the alarm will be highlighted in the same colour (D3) as the alarm border (O2). Status indication for quality of data is shown in O1.

The polar star object is intended to highlight movement from a stable operating position. Because unit operations have multiple stable operating conditions, e.g. at various throughputs, it is imperative that the capability to “re-normalize” the shape of the polar star is incorporated as a standard function of the object. This is not as important for the case of implementing a polar star for process utilities, but in cases where the object is used to display unit operations, the ability to “re-normalize” the shape will become absolutely necessary.
AGITATOR SYMBOL

DYNAMIC OBJECTS

D1: Agitator/Rake "blade"
D2: Agitator/Rake numerical PV for Torque (toggles visibility)
D3: Agitator/Rake Amps numerical PV (toggles visibility)

STATIC OBJECTS

S1: Engineering Units (toggles on L2)
S2: Agitator/Rake Tag (toggles on L2)

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border
O3: Miscellaneous Meter
O4: Multi-equipment object (works with D2)
O5: Status indication for abnormal mode
O6: OP Bar for Controller input

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The agitator/rake is a simple object, created by combining a miscellaneous meter bar with a blade shape. The blade fills when the agitator is running and is hollow when it is idle.

Most of the time, the units measured will be amps.

Normal alarming conventions apply.
FEEDER SYMBOL

Feeder Object Template

STATIC OBJECTS

S1: Feeder label
S2: Feeder icon (one of Conveyor, Screw, Screen, Vibrating Feeder)

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border
O3: Multiple Equipment object
O4: Status indicator abnormal Mode
O5: Standard VSD object (removed for multiple equipment option is selected at design time)

NOTE:
* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
SYMBOL EXAMPLES

Conveyor 1 is ON
Conveyor 2 in STANDBY

Conveyor 1 and 2 in STANDBY

Conveyors in alarm state

VSD Screw – in abnormal mode

VSD Screen – all in abnormal mode
This object provides equipment status of a set of feeders via multi equipment objects (O3). The operator shall use the operator control zone to set the RUN and STOP commands to the conveyor. The feeders are designated using label $1$, and $2$ is the static shape representing the type of feeder.

Status indication of quality of data is provided through $O1$ and alarming follows standard alarming conventions via the alarm border (O2).
HV/LV Switch Description

**STATIC OBJECTS**

S1: Label for the specific Bus  
S2: Fixed portion of the switch

**DYNAMIC OBJECTS**

D1: Switch position (either open or closed - see examples)

**OTHER OBJECTS**

O1: Standard Alarm Border  
O2: Status Indication for quality of data  
O3: Status Indication for mode

**NOTE:**  
* Denotes a parameter that is already provided to the operator  
# Denotes a parameter that likely requires a tag to be created  
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The HV/LV Switch (BusSwitch) is used to indicate electrical bus switch positions. All states, from controllable, indication only and mnemonic are represented (see examples).

The shape contains a switch (£) with a description of the switch ($) and also indicates the abnormal mode that the switch might be in.
HEAT EXCHANGER SYMBOL

Heat Exchanger Template

Heat Exchangers

S1A
S1B

S2A
S2B

Fans

STATIC OBJECTS

S1A-B: HX representation
S2A-B: HX representation

DYNAMIC OBJECTS

D1: Area underneath the inlet and outlet temps

STATIC OBJECTS

S1: HX Identifier

Other OBJECTS used in Graphic

O1: Standard temperature meter with PV value. Some may require a calculation
O2: Standard temperature meter with PV value. Some may require a calculation
O3: Standard multiple equipment object
O4: Line connecting inlet temp and outlet temp meters

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
SYMBOL EXAMPLES WITH ALARMS

Fans
No indication, no control

Fans
Indication elsewhere

Other heat exchangers
No indication, no control

Other heat exchangers
Indication elsewhere
Heat exchanger objects as widgets on their own are very simple. Fans are represented using a propeller symbol within a circle. All other exchangers are depicted using a circle with a zigzag line through them.

The muted object (S1A, S2A) is used whenever there is no indication or control on an exchanger. S1B and S2B are used when there are inlet and outlet temperatures but they are not shown with the exchanger directly.

Typically, the representation for heat exchangers will look more like to the “5-oil” example shown in the Sub-pictures Examples page.
BLOWER SYMBOL

Blower Object Template

DYNAMIC OBJECTS
D1: Blower representation

STATIC OBJECTS
S1: Blower label (toggleable on and off)
S2: Blower shape
S3: Blower indicator (to differentiate from pumps)
S4: Mnemonic blower shape
S5: Mnemonic blower indicator (to differentiate from pumps)

Other OBJECTS used in Graphic
O1: Multiple Equipment Object (States 1-12,16,17) used to represent Blower on/standby/LOTO indication (works with D1)
O2: Meter represented within equipment shape...option at shape definition time. Not toggled at runtime
O3: Status indication for mode (only visible when mode is abnormal)

NOTE: * Denotes a parameter that is already provided to the operator
      # Denotes a parameter that likely requires a tag to be created
      π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
SYMBOL EXAMLES WITH ALARMS

1 blower on

2 blowers
1 on, 1 standby

Vert Blower

1 blower standby

Horizontal blower with
alarm - blower still
running

Horizontal blower with
alarm - blower tripped

Actual Sizes

Horizontal blower with meter
shown – meter in alarm –
abnormal run mode

4 parallel blowers –
all in normal mode

5 parallel blowers
-blowers 3 and 5
are not in normal
mode
Blowers are examples of equipment that often have backups and therefore tend to take up too much space on conventional displays due to the tendency to display every blower or exchanger individually. For the purposes of the level 1 displays, blowers will be displayed in the following way.

Instead of displaying every blower, one representative icon will be used (DI). The operator will know which is operating by the square indication below the blower (represented by the multiple equipment object, DI). One indication represents one blower, so in this example there are 3 blowers. In general, gray is on, white is off or in standby. In cases where the equipment is LOTO, then an “X” will appear to remind the operator that the equipment is inoperable. In both examples above, blower 1 (far left) is LOTO. In total, there are 16 possible states that the blower can have as described in the multiple equipment object template (states 13-15 in the multiple equipment object template are reserved for filters).

When there is flow through the blower (e.g. the blower is on) then the blower should be filled with the process line color. When the blower is off, there should be no fill (e.g. blower is hollow).

For blowers that have no indication nor control, the blower sample to the right above shall be used. The main differences are the blower’s salience and that the operator knows how many blowers there are, but does not know the state that they are in. The operator can set the LOTO state for these blowers manually, but has no idea if they have been brought back into service.

The blower label (SF) is toggleable on and off.

Alarms should be indicated as shown in the examples, using the standard alarm template.
PUMP SYMBOL

Pump Object Template

DYNAMIC OBJECTS

D1: Multi-Stage Pump representation

STATIC OBJECTS

S1: Pump label (toggleable on and off)
S2: Pump indicator

Other OBJECTS used in Graphic

O1A and B: Multiple Equipment Object (States 1-12,16,17) used to represent Blower on/standby/LOTO indication (works with D1)
O2: Standard VSD shape for equipment
O3A and B: Status indication for mode (only visible when mode is abnormal)

NOTE:  *
Denotes a parameter that is already provided to the operator
#
Denotes a parameter that likely requires a tag to be created
π
Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Multi-Stage Pumps are examples of equipment that often have backups and therefore tend to take up too much space on conventional displays due to the tendency to display every pump or exchanger individually. For the purposes of all displays, multi-stage pumps will be displayed in the following way.

Instead of displaying every pump, one representative icon will be used (Ω). The operator will know which is operating by the square indication below the pump (represented by the multiple equipment object OIA and OIB). One indication represents one pump, so in the example above right, there are 3 pumps.

In all other respects, this shape behaves as a standard pump.

The pump label (ΩJ) is toggles on and off.

Alarms should be indicated as shown in the examples, using the standard alarm template.
VSD Compressor Object Template

DYNAMIC OBJECTS

D1: Compressor object

STATIC OBJECTS

S1: Compressor label (toggleable on and off)

Other OBJECTS used in Graphic

O1: Single MEO
O2: Miscellaneous meter object
O3: Standard output for VSD
O4: Multiple MEO (up to 5)

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
DOL compressors are an extension of the basic pump object.

For compressors that are fixed-speed, in addition to the compressor indication (DF), along with the multiple equipment object for the compressor (O1 and O3), a meter indicating the compressor power may also be represented (O2). The most common units are kW or Amps.

The compressor label (SL) is toggleable on and off.

Alarms should be indicated as shown in the examples, using the standard alarm template.

VSD compressors are an extension of the VSD pump object.

For L3 VSDs, in addition to the compressor indication (DF), along with the multiple equipment object for the compressor (O1 and O4), a meter indicating the compressor power is also represented (O2). The most common units are kW or Amps.

The compressor label (SL) is toggleable on and off.
TURBINE SYMBOL

VSD Turbine Object Template

DYNAMIC OBJECTS
D1: Turbine object

STATIC OBJECTS
S1: Turbine label (toggleable on and off)

Other OBJECTS used in Graphic
O1: Single MEO
O2: Miscellaneous meter object
O3: Standard output for VSD
O4: Multiple MEO (up to 5)

NOTE:
* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
VSD turbines are an extension of the VSD pump object.

For L3 VSDs, in addition to the turbine indication (D1), along with the multiple equipment object for the turbine (O1 and O4), a meter indicating the turbine power is also represented (O2). The most common units are kW or Amps.

The turbine label (S1) is toggleable on and off.

Alarms should be indicated as shown in the examples, using the standard alarm template.
**GAP CONTROL VALVE SYMBOL**

Gap Control Valves Object Template

- **D1**: Feedback output from valve
- **D2**: Commanded output for each valve - AO Set point
- **D3**: Numerical presentation of output (same as **D1**)
- **D4**: Local Handswitch indication

**NOTE:**

- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags

Other OBJECTS used in Graphic

- **O1**: Alarm Object
- **O2**: Control valve object
### SYMBOL EXAMPLES WITH ALARMS

**Valve A operating, B closed**

- CV XXXX A/B
  - 49%
  - 0%

**Valve A closed, B open**

- CV XXXX A/B
  - 0%
  - 25%

**Valve A open and selected, B is OOS**

- CV XXXX A/B
  - 49%
  - X

**Valve A open, and in alarm, B closed, non-normal control mode M – Valve has local Handswitches**

- CV XXXX A/B
  - 33%
  - 0%
  - M

**Valve A open, B open**

- CV XXXX A/B
  - 49%
  - 12%

**Valve A fully open, B open, non-normal control mode M**

- CV XXXX A/B
  - 100%
  - 2%

- M

**Valve A open, and in alarm, B closed, non-normal control mode M**

- CV XXXX A/B
  - 49%
  - 0%
This object is used for gap control valves. They are often combined in situations where there are two valves and where both often need to be open simultaneously.

The object is basically made up of 2 valve objects combined.

**O1.** A standard alarm object

**O2.** A standard control valve object...the only difference is that the numeric is shown to the left of the valve box instead of underneath it. The selected valve will show black text while the non-selected valve text will be grey.

The small triangle beneath the bar graph (O2) will be used to indicate the commanded output. Feedback output will be presented by a filled bar graph, with 0% on the left and 100% on the right.

This depiction is intended for Type 1 & 2 displays only. Type 3 graphics will show each valve in detail. Selection of this object will bring up a split range controller faceplate and show the control relationship line between the valve object and its controller.
VALVE AND DAMPER SYMBOL

MOV, EOV VALVE SYMBOL

Complex MOV/EOV Object Template

DYNAMIC OBJECTS

D1: Valve Position feedback

STATIC OBJECTS

S1: Valve Stem
S2: Valve identifier

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm indication
O3: Multiple equipment object (states 1-8 only)
O4: Status indication for abnormal mode
O5: Handswitch Indication

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
SYMBOL EXAMPLES WITH ALARMS

Closed
MOV 123

Closed
Requested to open
MOV 123

Open 0<PV<100 stopped moving – PV toggled off
?? XXXX
M

Open Requested to open further
?? XXXX
M
51%

Open 0<PV<100 moving – in alarm
?? XXXX
M
35%

Fully Open
MOV 123

Fully Open
Requested to close further
MOV 123

Open 0<PV<100 stopped moving
Tag name and numerical PV toggled off

Open 0<PV<100 stopped moving – in normal mode
Not in normal mode – Handswitch on

Vertical orientation
This object is used for MOV/EOV where the operator has the ability to stop a valve from stroking fully open or closed. If the operator cannot stop valve stem travel, then a simple EOV/MOV valve depiction shall be used (see simple MOV/EOV object template).

The EOV/MOV is the same as a control valve object template, except for two significant differences:

1. The head of the valve is rectangular, rather than semi-circular
2. The exact position of the valve opening is not known. Rather, all that is fed back to the operator is that the valve is fully closed, somewhat open, fully open and that it is transitioning between those states. Hence, the multiple equipment object (OE) is used (states 1–8 only) to identify what the valve is doing.

Alarming is done using the standard alarm object.
SIMPLE MOV EOV VALVE SYMBOL

Simple MOV/EOV Object Template

**DYNAMIC OBJECTS**

- **D1**: Valve Position
- **D2**: Valve open/closed identifier (works with **D1**)
- **D3**: Abnormal mode identifier

**STATIC OBJECTS**

- **S1**: Background line (Works with **D1**)
- **S2**: Valve tag name
- **S3**: Valve stem

**Other OBJECTS used in Graphic**

- **O1**: Status indication for quality of data
- **O2**: Alarm indication
- **O3**: Status indication for abnormal mode
- **O4**: Status indication for abnormal computer control mode

**NOTE:**
- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- \(\pi\) Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
This object is used for MOV/EOV where the operator does not have the ability to stop a valve from stroking fully open or closed. If the operator can stop valve stem travel, then a complex EOV/MOV valve depiction shall be used (see complex MOV/EOV object template).

The simple EOV/MOV has only two major states, open or closed. Alarming shall be depicted using the standard alarm object template.
**MINI BLOCK VALVE SYMBOL**

**Dynamic Objects**

- **D1**: Valve Position
- **D2**: Valve open/closed identifier (works with **D1**)

**Static Objects**

- **S1**: Background line (Works with **D1**)
- **S2**: Valve tag name
- **S3**: Valve stem

**Other Objects used in Graphic**

- **O1**: Status indication for quality of data
- **O2**: Alarm indication
- **O3**: Status indication for abnormal mode

**Note:**

- * Denotes a parameter that is already provided to the operator
- # Denotes a parameter that likely requires a tag to be created
- π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
This object behaves in the same manner as the EOV/MOV shape. The difference is that it is smaller.

The mini block valve has only two major states, open or closed.

Alarming shall be depicted using the standard alarm object template.
DAMPER SYMBOL

DYNAMIC OBJECTS

D1: Damper feedback position (If feedback is available, this object will fill a different shade and directly follow the OP (D3)...see property details)
D2: Normal conditions the fill is gray. When the controller commanded value is > 100 the fill is black when the command is 0% fill is white
D3: OP
D4: Numerical presentation D1 (this can be toggled on/off by the operator)
D5: SP
D6: Tracker
D7: Field Handswitch (option selected at design time, not runtime)

Other notes:
Control Indicator's relationship with damper will only show up when user clicks on the damper or on the control indicator.

STATIC OBJECTS

S1: Background line (Works with D1)
S2: Damper identifier (this can be toggled on/off by the operator)
S3: Damper static line

Other OBJECTS used in Graphic
O1: Status indication for quality of data
O2: Alarm indication
O3: Status indication for mode (Only visible when object in abnormal mode)

NOTE: * Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
**SYMBOL EXAMPLES WITH ALARMS**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Closed</td>
<td><code>DM XXXX</code> with 0%</td>
</tr>
<tr>
<td>Partially Open</td>
<td><code>DM XXXX</code> with 48%</td>
</tr>
<tr>
<td>PID/OUT &gt; 100 or &gt;= upper limit</td>
<td><code>DM XXXX M</code> with 100%</td>
</tr>
</tbody>
</table>

**In Alarm, Damper name toggled off**

- `DM XXXX` with 48%

**Partially Open, Set to manual** (not normal mode)

- `DM XXXX M` with 48%

**Partially Open, no readback from damper** (not normal mode)

- `DM XXXX M` with 48%

**Vertical CV, Set to Manual** (not normal mode)

- `DM XXXX` with 56%

**Actual Size – Handswitch option on**

- `DM XXXX M` with 56%

- `H H H` with question mark

- `?? XXXX M` with 27%

Dampers shall be small and the same color as equipment. The percent the damper is open (AO readback), will be presented by a horizontal bar graph (D1). If feedback is available, this object will fill a different shade and directly follow the OP (D3). If `D1 > 100%` then the damper head (D2) will fill to black. If `D1 <= 0` then the damper head will fill white. If `D1 > 1%` (or an acceptable value) then the damper head (D2) will fill gray to imply flow.

The mode of the damper (D1) shall be displayed only when the damper is not in its “normal” mode.

The damper name (S2) and numerical presentation of output (D4) should be capable of being toggled on and off for L2 and L3 graphics.

Alarming is done using the standard alarm object.
**PARALLEL CONTROL VALVE SYMBOL**

Parallel Control Valves Object Template

```
O1
D3*
D1*

CV XXXX A/B

80%
20%

D2*

O2
```

**DYNAMIC OBJECTS**

D1: Controller output, PID/OUT
D2: Read back position from valve
D3: Numerical presentation of PID/OUT (same as D1)

**Other OBJECTS used in Graphic**

O1: Alarm Object
O2: Control valve object

**NOTE:**

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
Symbol Examples with Alarms

Valve A operating, B closed

Valve A open, B open

Valve A closed, B open

Valve A fully open, B open, non-normal control mode M

Valve A open and selected, B is OOS

Valve A open, and in alarm, B closed, non-normal control mode M

This object is used for split range valves. They are often combined in situations where there are two valves, and where both often need to be open simultaneously.

The object is basically made up of 2 valve objects combined.

01. A standard alarm object
02. A standard control valve object, the only difference is that the numeric is shown to the left of the valve base instead of underneath it. The selected valve will show black text while the non-selected valve text will be grey.

The small triangle beneath the bar graph (02) will be used to indicate the commanded output. Feedback output will be presented by a filled bar graph, with 0% on the left and 100% on the right.

This depiction is intended for Type 1 & 2 displays only. Type 3 graphics will show each valve in detail. Selection of this object will bring up a split range controller faceplate and show the control relationship line between the valve object and it’s controller.
VESSEL SYMBOL

Vessel Object Template

STATIC OBJECTS

S1: Vessel border
S2: Vessel tag name
S3: Vessel tag name background (works with S2)

Other OBJECTS used in Graphic

O1: Status indication for quality of data
O2: Alarm border
O3: Meter object (could be level, pressure, temp, flow, etc)

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The vessels are essentially shells, within which, other meters can be placed. There are three standard vessel widths for vertical vessels (small, medium and large). There is one horizontal vessel and one accumulator vessel. These are all depicted in the vessel Examples page.

The vessels in this object library are standard vessel objects, with a few differences as outlined below:

The tag name ($S2$), which is toggleable on and off is shown in the frame of the vessel.

Whenever $S2$ is toggled off, so is $S3$.

The border of the vessel is alarmed ($O1$) if any alarm within the vessel module alarms. The highest priority alarm active is used for the border. Therefore, the vessel may show that it is in alarm even if the meter(s) shown within the vessel ($O2$) are not in alarm. For example, in the case of the bottom-left vessel on the Examples page, a parameter somewhere on the vessel is in a state of a high priority alarm. Therefore, the vessel alarms, even though the level is fine. As a second example, the fourth vessel from the left on the bottom row shows a state of emergency priority even though the level shows a high priority alarm state.

Again, another parameter associated with that vessel is in a state of emergency alarm.
DIRECTION ARROW SYMBOL

Direction Arrow Object Template

DYNAMIC OBJECTS

D1: Arrow showing direction for equipment

STATIC OBJECTS

S1: Tagname/Nickname Text
S2: Border surrounding arrow

OTHER OBJECTS

O1: Alarm Border
O2: Status Indication

NOTE:

* Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
The direction arrow is used to present the direction of a piece of equipment (e.g., conveyor belt). Its two modes are forward and reverse, represented by the arrow (D). Alarming is presented in the standard format (O1).
## MULTIPLE EQUIPMENT OBJECT

### Mutiple Equipment Object Template

<table>
<thead>
<tr>
<th>Multiple Equipment State</th>
<th>Description</th>
<th>Graphical Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Mnemonic</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Mnemonic (Out Of Service) - Set Manually</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Off</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stopped and Available (Drives)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fully Closed</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Interlocked</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Stopped and Bypassed</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Selected to Out of Service by Upstream Equipment</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Out Of Service - Unavailable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Triangle</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>State changing from Stopped to Running (Single Speed Fwd/Rev Drive)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>State changing from Stopped to Running Slow (Dual Speed Drives)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valve requested to open but still at closed limit (Valves)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>State Changing from Running Slow to Running Fast (Dual Speed Drives)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valve requested to open and has left the closed limit (Valves)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Valve requested to close and has left the open limit (Valves)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>State changing from running to stopped (Single Speed Fwd/Rev Drive)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>State Changing from Running Slow to Stopped (Dual Speed Drives)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Drive running as standby drive in a duty standby arrangement</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Valve open (MOV)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valve Stopped Midway (complex MOV)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running Forward (Fixed Speed Fwd/Rev Drives)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running Forward Slow Speed (Dual Speed Drives)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Running Reverse (Fixed Speed Fwd/Rev Drives)</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>Running Reverse Slow Speed (Dual Speed Drives)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Running and Bypassed</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Valve requested to close but still at open limit (simple MOV)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>State changing from Running Fast to Running Slow (Dual Speed Drives)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Fully Open</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Running Forward at Full Speed (Dual Speed Drive)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Running reverse at Full Speed (Dual Speed Drive)</td>
<td>R</td>
</tr>
<tr>
<td>18</td>
<td>Running Forward at Full Speed (Dual Speed Drive) with an Interlock Bypass</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Active</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Spare</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Spare</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Caustic Cleaning</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Startup Warning Siren</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Fully Open and Interlocked - Fail Open Valves</td>
<td></td>
</tr>
</tbody>
</table>
### Symbol Examples with Alarms

<table>
<thead>
<tr>
<th>1 piece of equipment on</th>
<th>2 pieces of equipment</th>
<th>3 pieces of equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbol Image]</td>
<td>![Symbol Image]</td>
<td>![Symbol Image]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 pieces of equipment</th>
<th>6 pieces of equipment</th>
<th>6 pieces of equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 LOTO, 1 on, 1 in High Priority alarm and tripped</td>
<td>1 LOTO, 2 on, 1 in high priority alarm and running, 1 in emergency priority alarm and tripped, 1 standby</td>
<td>1 changing from on to standby, one changing from standby to online, 2 standby, 1 Low Priority alarm and running, one on</td>
</tr>
<tr>
<td>![Symbol Image]</td>
<td>![Symbol Image]</td>
<td>![Symbol Image]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6 pieces of equipment</th>
<th>1 LOTO, 1 high speed, 1 low speed, 1 standard alarm, 1 in critical alarm and running, 1 changing from high to low speed, 1 standby</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbol Image]</td>
<td>![Symbol Image]</td>
</tr>
</tbody>
</table>
The multiple equipment representations are used whenever there are multiple like pieces of equipment in series or parallel.

The state of the equipment is represented by the fill within the graphic. A solid fill as in State 12, represents a piece of equipment is running. A background fill (State 7) is used to represent the equipment as being off.

State 7 is only possible if the control system is able to determine the out of service (LOTO) status of the equipment, or if provisions are made to allow the operator to manually input that a piece of equipment is LOTO.

This object is used in the following way: Instead of displaying every pump, heat exchanger, etc. one representative icon will be used. The operator will know which is operating by the square indication below the piece of equipment. This object is also used on MOV's that can be stopped part way open/closed.

This object is alarmed using a border. Examples of how it alarms are shown in the examples page. Due to the border, there will be a slight gap between the shapes when not in alarm.
**METER CONNECTOR SYMBOL**

Meter Connector Object Template

D1: Meter connector - current reading
D2: Historical value 1 – only visible if history option is selected
D3: Historical value 2 – only visible if history option is selected
D4: Historical value 3 – only visible if history option is selected

NOTE: * Denotes a parameter that is already provided to the operator
# Denotes a parameter that likely requires a tag to be created
π Denotes a parameter that requires a tag to be created and requires a calculation based on available tags
This object allows the PVs of two meters to be connected, thus giving operators a visual cue as to the relationship between the values of the PVs. Using the meter connector with multiple related meters in series creates a visual profile of the PV values.

It may be drawn together with 1, 2, or 3 historical values (see subpictures). For each historical time step, the connector is drawn using the historical PV values for the designated time in the past performance of the two PVs.

In order to support vertical profiles, a vertical orientation may be selected (see subpictures).