

Experion PKS
Procedure and Sequence Custom Display Building
Guide

EPDOC-XX86-en-431A
February 2015

Release 431

Document	Release	Issue	Date
EPDOC-XX86-en-431A	431	0	February 2015

Disclaimer

This document contains Honeywell proprietary information. Information contained herein is to be used solely for the purpose submitted, and no part of this document or its contents shall be reproduced, published, or disclosed to a third party without the express permission of Honeywell International Sàrl.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any direct, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Copyright 2015 - Honeywell International Sàrl

Contents

About this guide	5
Creating and controlling activities on custom displays	7
Adding and configuring toolkit controls	11
Adding and enabling toolkit controls	12
Configuring toolkit controls	14
Initializing the toolkit infrastructure	17
Binding the toolkit controls	18
Dynamic binding use case scenario	19
Procedure and sequence objects	21
CommandButtonControl display object	22
CommandDropDownControl display object	23
CommandPanelControl display object	24
ConfirmationPanelControl display object	25
InformationControl display object	26
InformationPanelControl display object	27
InstructionControl display object	28
InstructionListControl display object	29
ModeAttributeButtonControl display object	30
ModeAttributeDropDownControl display object	31
ModeButtonControl display object	32
ModeDropDownControl display object	33
PhaseControl display object	34
ProcedureControl display object	35
ProcedureListControl display object	36
ProcedurePanelControl display object	37
StepandPhaseListControl display object	38
StepControl display object	39
TransitionControl display object	40
TransitionListControl display object	41
UnitControl display object	42
UnitListControl display object	43
Procedure and sequence object properties	45
General properties	46
Shortcut menu properties	47
CommandButtonControl properties	48
CommandDropDownControl properties	49
CommandPanelControl properties	50
ConfirmationPanelControl properties	51
InformationControl properties	52
InformationPanelControl properties	53
InstructionControl properties	54
InstructionListControl properties	55
ModeAttributeButtonControl properties	56
ModeAttributeDropDownControl properties	57

ModeButtonControl properties	58
ModeDropDownControl properties	59
PhaseControl properties	60
ProcedureControl properties	61
ProcedureListControl properties	62
ProcedurePanelControl properties	63
StepandPhaseListControl properties	64
StepControl properties	65
TransitionControl properties	66
TransitionListControl properties	67
UnitControl properties	68
UnitListControl properties	69
Procedure and sequence object model reference	71
Overview of the procedure and sequence object model	72
Methods	73
Initialize method	73
Pause method	73
Restart method	74
Shutdown method	74
Events	75
ActivitySelectedEvent	75
InstructionSelectedEvent	75
PhaseSelectedEvent	76
ProcedureSelectedEvent	76
StepSelectedEvent	76
TransitionSelectedEvent	77
UnitListChangedEvent	77
UnitSelectedEvent	78
Notices	79
Documentation feedback	80
How to report a security vulnerability	81
Support	82
Training classes	83

About this guide

You use the Procedure and Sequence Toolkit in HMIWeb Display Builder to create procedure and sequence controls on custom displays. This toolkit is also referred to as the *UI Toolkit*.

There are three types of toolkit control:

- Basic controls, which are bound to basic procedure elements such as an Activity, SCM, RCM, or UCM, or a step, phase or transition block.
- List controls, which are designed to organize and present basic controls in a list format.
- Panel controls, which are compound controls consisting of basic controls or list controls grouped together for convenience and enhanced functionality.

 **Attention**

- The toolkit controls have been developed to bind only to SCM/RCM/UCM and Activity points and their sub-elements. They cannot be bound to Master Recipes.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

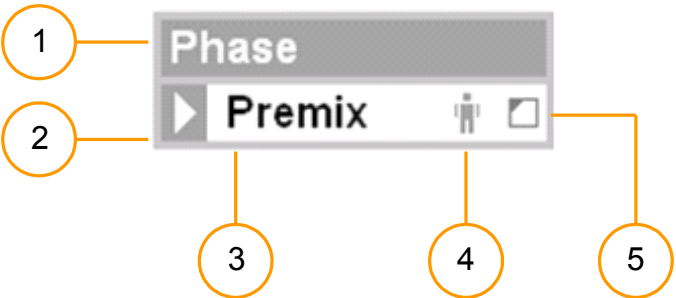
How to use this guide

This guide describes how to add procedure and sequence controls to custom displays.

To learn about	Go to
Procedure and sequence objects	“Procedure and sequence objects” on page 21
Procedure and sequence object properties	“Procedure and sequence object properties” on page 45
The procedure and sequence object model reference	“Procedure and sequence object model reference” on page 71

Introduction to the toolkit controls

The toolkit controls all have the same basic structure. The following image shows the elements of a Procedure and Sequence Control.



Item	Description
1	<p><i>Header</i></p> <p>The header is optional and can be shown or hidden depending on how the control has been configured (ShowHeader property). It indicates the control type — <i>Procedure</i>, <i>Step</i>, <i>Phase</i>, <i>Instruction</i>, <i>Transition</i>, or <i>Unit</i>.</p>
2	<p><i>State icon</i></p> <p>The state icon indicates the <i>state</i> of the control and its associated RCM, SCM or UCM. For details of the possible states, see the State Icon table that follows.</p>
3	<p><i>Name or description</i></p> <p>This property allows the display designer to select whether the name (<i>Public Name</i>) or the description of the procedure element is to be shown (ShowDescription property).</p>
4	<p><i>Status icon</i></p> <p>The status icon indicates the <i>execution status</i> of the control. It is only visible when the control is <i>Executing</i>, (running, restarting, or restarted). For details of the possible status values, see the Status Icon table that follows.</p>
5	<p><i>Procedural level icon</i></p> <p>The Procedure Level icon indicates the S88 procedural level of a phase, step, or unit.</p>

Figure 1: Toolkit control elements

Toolkit control pop-ups

Details pop-up controls are provided for most basic and list controls. These pop-ups can be either a singular version of a control, or they can contain combinations of controls, similar to a panel control. Details pop-ups can be *pinned* to a display by clicking the **Push Pin** (📌). Otherwise, the control will close when you navigate to other displays.

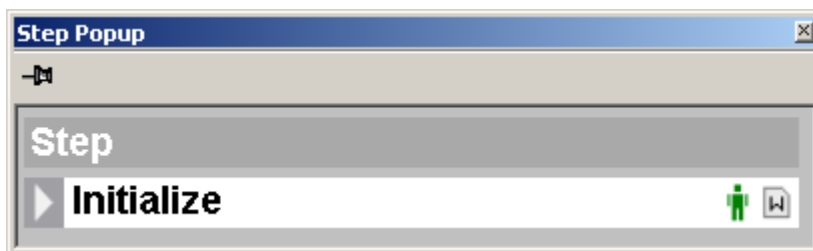


Figure 2: Pop-up control

Creating and controlling activities on custom displays

Experion supports the creation of custom displays containing the same Batch Manager functionality as that provided on the summary pages. You can build a custom display using the HMIWeb Display Builder standard features and add an Activity table object, activity command buttons, and a push button that will launch the **Create Activity** dialog box from within the display.

The **Activity Table** and the **Create Activity** pushbutton can be configured specifically for batch only, procedural operations only, or a mixture of both.

Creating the activity components

- 1 From the HMIWeb Display Builder window, select the **Activity Summary Table** object.
- 2 On the display, position and size the Activity Summary Table and then configure the object's properties.
- 3 Create a set of activity command buttons and command drop down lists to add to your Activity Summary Table by selecting the **Shapelink** object.
- 4 On the display, position and size the Command shapelink object at the bottom left of the Activity Summary table and then configure its properties.
- 5 Lastly, create a **New Batch** push button by selecting the **Pushbutton** object.
- 6 On the display, position and size the Pushbutton object and then configure its properties.

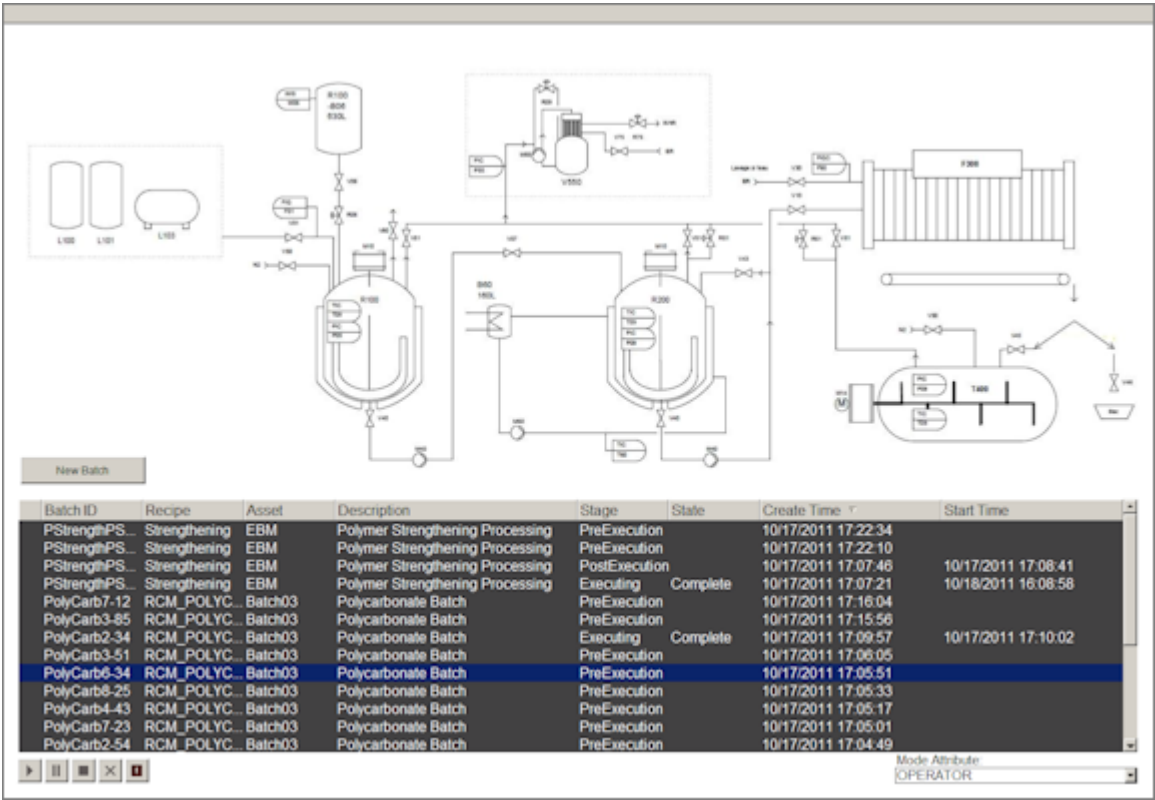


Figure 3: Custom display with activity table, commands, and Create Batch button

Creating an activity on a custom display

- 1 From the custom display, click **New Batch** to display the **Create Batch** dialog box.


 **Tip** Depending on how this button has been configured in HMIWeb Display Builder at your site, it may or may not have Create Batch as its label. This label is used here only to explain the concept.



Figure 4: Custom display with Create Batch dialog box

- 2 Enter a Batch ID for the new batch or procedure and select the recipe on which you would like to base this batch or procedural operation.

Depending on how the **New Batch** button has been configured in *HMIWeb Display Builder*, the asset may have already been assigned. If this is the case, the Location Pane will not be displayed. If the button has not already had the asset configured, the Location pane is displayed and by selecting an asset you can filter the recipes or procedures to those associated with that asset.

You can filter the recipe list further by typing characters into either the **Recipe** or the **Public Name** box.

**Tip**

Batch ID is mandatory for a batch but optional for a procedural operation.

- 3 Select one or both of the available options:
 - a **Automatically Start Activity** will move the batch or procedure into an *Executing* state as soon as it has been created
 - b **Edit Activity Parameters** will launch the **Activity Data** dialog box as soon as the batch or procedure has been created.
 - 4 Click **Create**. The batch or procedure is created using the default parameters.
- If you selected **Edit Activity Parameters**, the batch or procedure is created and the **Activity Data** dialog box is invoked, enabling you to edit its parameters, as shown below.

Description	Master Recipe Value	Control Recipe Value	Units	New Value	Errors
Additive 1 Volume	50	50	litres		
Additive 2 Volume	75	75	litres		
Pre Mix Timer	60	60	sec		
Ester Volume	30	30	millilitres		
Monomer Mass	40	40	kg		
Mix Time 1	60	60	sec		
Mix Time 2	30	30	sec		
Catalyst in phase 1	PlatinumPowder	PlatinumPowder			
Catalyst in phase 2	SilverMatrix	SilverMatrix			
Magenta Pigment	0	0	ppm		
Yellow Pigment	0	0	ppm		
Cyan Pigment	0	0	ppm		
Shade Factor	1	1			

Figure 5: Custom display with Activity Data dialog box

You can monitor the batch or procedure in the Activity Summary Table you added to the custom display earlier. To interact with and control the batch or procedure, you need to add some toolkit controls. See “Adding and enabling toolkit controls” on page 12 for more information, and the remainder of this guide for detailed descriptions of the controls, their properties, and the Procedure and Sequence object model reference.

Adding and configuring toolkit controls

Related topics

“Adding and enabling toolkit controls” on page 12

“Configuring toolkit controls” on page 14

“Initializing the toolkit infrastructure” on page 17

“Binding the toolkit controls” on page 18

Adding and enabling toolkit controls

Inserting controls into HMIWeb displays

- 1 From the HMIWeb Display Builder window, open the **Procedure and Sequence Toolkit** toolbox to display the list of toolkit controls, as shown below.

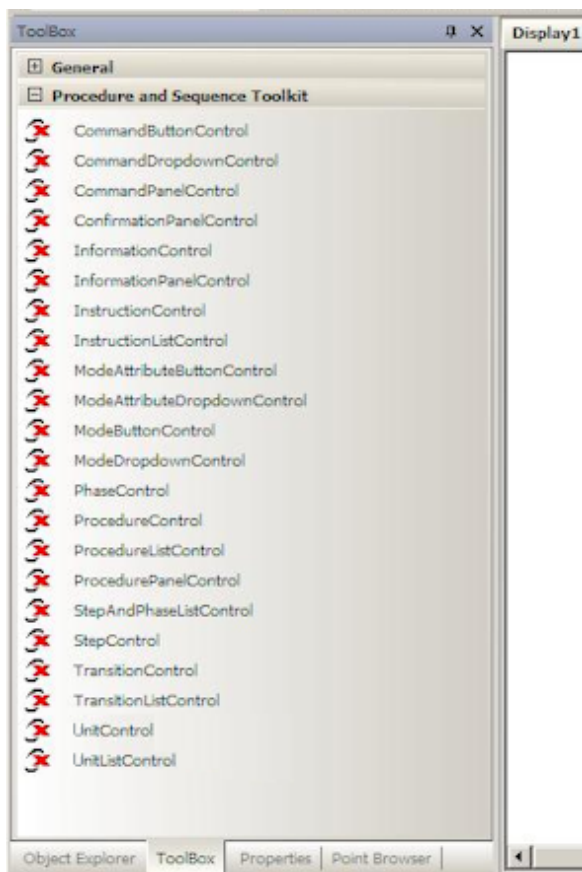


Figure 6: Procedure and Sequence Toolkit controls

- 2 Select a control from the list to insert into the display. The controls can be resized and their height will adjust automatically to work well with the system font size.

Enabling the toolkit context menu

- 1 To enable a context menu for a toolkit control, open the properties dialog for that control.

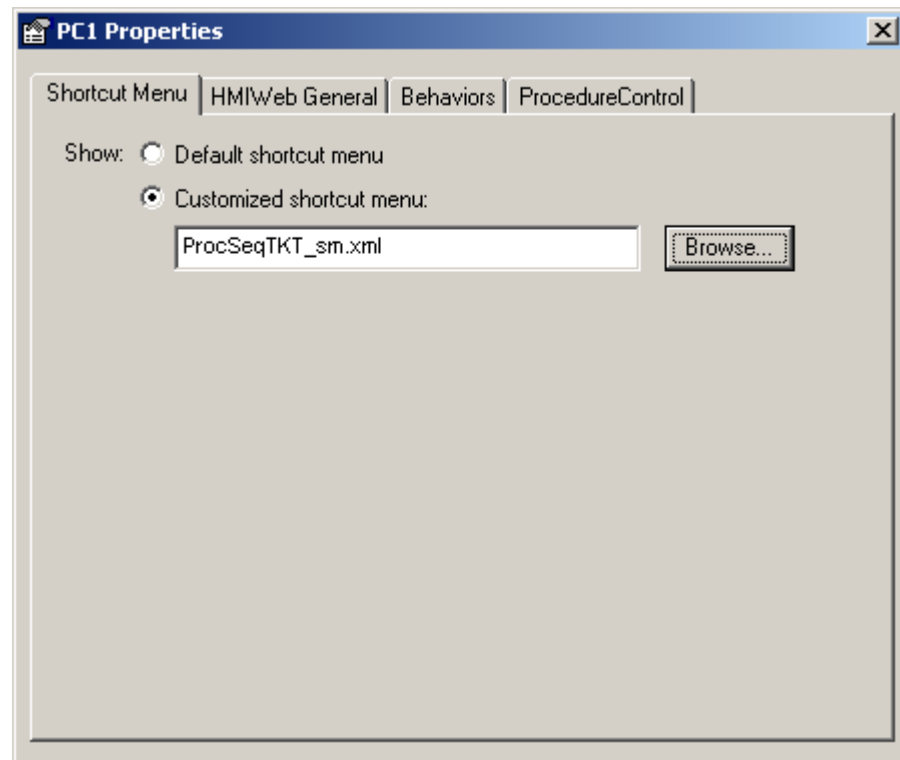


Figure 7: Shortcut menu tab, control properties

- 2 Specify the shortcut menu path as a **customized shortcut menu** option for each control when inserting them into a display. The path for the custom Toolkit shortcut menu is: ProcSeqTKT_sm.xml.

Configuring toolkit controls

The toolkit controls can be configured in two ways, both of which are discussed in this section:

- Using the property grid in HMIWeb Display Builder
- Through scripting, in a similar way to any other native HMIWeb object or ActiveX control.

Toolkit controls have been designed to allow for a very rich set of visualizations for procedures with only a few lines of scripting to initialize the procedure data source and perform the bootstrapping and shutdown of the Toolkit controls in the display. Support for more sophisticated behavior by accessing the controls properties or methods is possible but requires a deeper understanding of how to bind controls to their data source and knowledge of the event sequences during the display instantiation.

Configuring toolkit controls using the HMIWeb property grid

Toolkit control properties can be viewed and configured in the HMIWeb Display Builder property grid. The toolkit controls-specific properties are listed under the ActiveX properties category. These are the same properties found under the tab in the control's property pages.

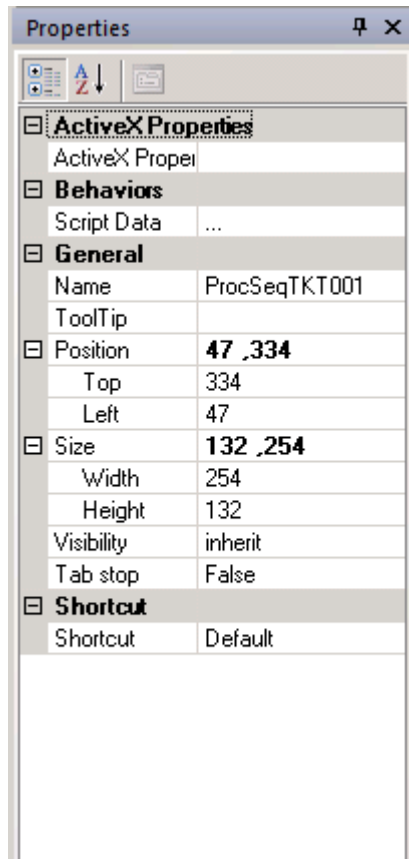


Figure 8: ActiveX Properties

There is a slight difference in the behavior of the properties in the property grid and the property pages. The property pages support the conditional showing/hiding of properties depending on whether they apply based on other property values. This behavior is not supported in the property grid and as a result all properties are always visible. Extra validation code has been added to make sure that only valid entries for the properties in question are accepted. If an invalid value is entered, it is automatically replaced with valid default values.

Configuring toolkit controls using scripting, including dynamic binding

Whether a control's property is changed to bind the control to a new procedure or procedure element, or simply to toggle a configuration option, such modification is considered dynamic when it is done through scripting, as opposed to static when done through the control's Property Grid at design time.

It is important to know whether the script being used to update the control is executed before or after the Toolkit controls have been initialized. If the script is run before the control is running, any of the control's properties can be updated without any additional steps being required to ensure the proper dynamic support.

After a control has been initialized, however, changing its binding or its properties is more difficult as multiple properties may need to be changed before taking effect.

To address this situation, two methods are provided: *Pause()* and *Restart()*. All controls support these methods. As their names indicate, the *Pause()* method pauses the control, allowing its properties to be changed safely. When the properties have been updated, the *Restart()* method can be called to restart the control with its new configuration.



Tip

- The *Pause()* and *Restart()* methods only need to be called after all the Toolkit control have been initialized, and can be called any number of times. In contrast, the *Initialize()* and *Shutdown()* methods should only be called once on the selected ProcedureObject provider.
- When the call to *Initialize()* is made can be significant. For example, if the display is relying on dynamic configuration it may be possible to delay the call and thereby increase the display call-up time until you have defined the control's binding.
- Leaving the Toolkit controls with inconsistent or invalid configuration is easy to do when using scripting as any errors will remain undetected until runtime. Any display using dynamic scripting of Toolkit controls should therefore be thoroughly tested.

For example, following is a script added to handle a button click where a ProcedureListControl (**PLC1**) is dynamically rebound to a new set of procedures (**04_M1_RCM,DCN_RCM_Proc**) and has its ShowDescription option set to **true**:

```
Sub pushbutton001_onclick
PLC1.Pause()
PLC1.ProcedureList = "04_M1_RCM,DCN_RCM_Proc"
PLC1.ShowDescription = true
PLC1.Restart()
End Sub
```

Note that as the button click is expected to occur after all Toolkit controls have already been initialized, the *Pause()* and *Restart()* calls are necessary.

Selection event handling in scripts

The table below lists the events raised by the Toolkit controls and indicates the release that supports these events. Do not confuse these events with the events already raised and handled internally by the Toolkit controls to yoke themselves when some selection is made. These new script events are intended to be used by script event handlers to allow the yoking of non-Toolkit objects to the selection made by some Toolkit controls. For example, a display may need to show different objects based on which Procedure is selected in a ProcedureListControl. The name of the selected procedure is made available as a parameter passed to the ProcedureSelectedEvent callback.

Table 1: Basic controls

Basic controls	Script events raised	R310	R311	R400	R410
InstructionControl	InstructionSelectedEvent(string procedureName, string instructionName)				x
PhaseControl	PhaseSelectedEvent(string procedureName, string phaseName)				x
ProcedureControl	ProcedureSelectedEvent(string procedureName)				x

Basic controls	Script events raised	R310	R311	R400	R410
StepControl	StepSelectedEvent(string procedureName, string stepName)				x
TransitionControl	TransitionSelectedEvent(string procedureName, string transitionName)				x
UnitControl	UnitSelectedEvent(string unitName)	n/a	n/a	n/a	x

Table 2: List controls

List controls	Script events raised	R310	R311	R400	R410
InstructionListControl	InstructionSelectedEvent(string procedureName, string instructionName)	x	x	x	x
ProcedureListControl	ProcedureSelectedEvent(string procedureName)	x	x	x	x
StepAndPhaseListControl	PhaseSelectedEvent(string procedureName, string phaseName) StepSelectedEvent(string procedureName, string stepName)	x	x	x	x
TransitionListControl	TransitionSelectedEvent(string procedureName, string transitionName)	x	x	x	x
UnitListControl	UnitSelectedEvent(string unitName)	n/a	n/a	n/a	x

For example, an event handler **PLC1_ProcedureSelectedEvent** () for the ProcedureSelectedEvent was created for the ProcedureListControl named **PLC1** as shown below. In this example, the name of the selected procedure is displayed it in a message box.

```
Sub PLC1_ProcedureSelectedEvent(ProcedureName)
msgbox ProcedureName
End Sub
```


Initializing the toolkit infrastructure

In addition to inserting toolkit controls and configuring their properties, some additional tasks are required so that the controls are enabled in HMIWeb displays. The toolkit needs to be initialized when the display opens, and shut down when the display is closed.

Initializing the toolkit

To initialize a Toolkit control, the initialization command needs to be invoked in the display **Page.OnPageComplete** event. The Toolkit initialization consists of initializing at least one special control that will discover, bind, and initialize all other Toolkit controls on the page. These special controls, the *ProcedureControl*, *ProcedurePanelControl*, *ProcedureListControl*, *UnitControl* and *UnitListControl* provide the data connection with one or more procedures through a *ProcedureObject* and are known as *ProcedureObject providers*.

A single line of script is required to initialize all the toolkit controls in a display:

```
...
pc1.Initialize()    // pc1 is the name of a ProcedureControl for instance
...
```

See “Binding the toolkit controls” for more information about special binding use cases.

Shutting down the toolkit

The Toolkit shutdown needs to be invoked in the display **Page.onUnload** event handler. The Toolkit shutdown involves shutting down the *ProcedureObject* provider control that was initialized in the *OnPageComplete* script.

The following example shows a *ProcedureControl* being shutdown:

```
...
// Shutdown the Procedure Control...
pc1.Shutdown()    // pc1 is the name of a Procedure Control
...
```



Tip

The HMI Solution pack contains a shape **All_ButtonNmEuHd_lib_01** that performs the initialization and shutdown. The scripts shown above is therefore not required if the **All_ButtonNmEuHd_lib_01** shape is included in the display. The HMIWeb Solution pack library is part of the Experion Media as a separate DVD. For more information, see the *HMIWeb Solution Pack Installation Guide*.

Binding the toolkit controls

The Toolkit controls provide a way for Operators to visualize and interact with procedure and procedure-related items. To do this, they need to be associated, or bound, to procedures and procedure-related items during the display configuration or later, through scripting.

Binding a control can be achieved in several ways:

1. A control needs to be associated with a *ProcedureObject*, an infrastructure component designed to serve all data and events related to a specific procedure. This is done automatically during Toolkit initialization. Each Toolkit control is discovered by the *ProcedureObject* provider during its initialization. The *Procedure Object* is initially created by a *ProcedureObject* provider (*ProcedureControl*, *ProcedurePanelControl*, *ProcedureListControl*, *UnitControl* or *UnitListControl*), and acts as a wrapper to, or modeler of, the procedure bound to the control. *ProcedureControls* and *ProcedurePanelControls* themselves need to be bound to an RCM or an SCM through their *BindTo* property. *ProcedureListControls* will provide a list of procedures, RCM and/or SCM to be displayed in a list. *UnitControls* and *UnitListControls* are designed to visualize UCMs or list of UCMs.
2. Some Toolkit controls are designed to work with procedure elements, such as steps or instructions. Controls are bound to specific steps or instructions through their control properties, where it is possible to specify which step or instruction the control is bound to. Sometimes, it is not desirable to hard code the name of a step or the output index. Instead, the control is configured to follow the currently selected step or instruction, and is dynamically bound when a specific event occurs. For more information, see the details of the specific controls.

As alluded to previously, *ProcedureObject provider* controls are very special Toolkit controls in that they are both regular visualization controls used to show the status of an associated procedure; and they are also critical infrastructure components, essential to data and event flow between Toolkit controls and the procedure and procedure elements they visualize. At least one *ProcedureObject* provider control is required to establish the data source for a specific procedure. Which one is selected depends on the use case scenario. When Toolkit controls are not bound to a unit or procedure and procedure element, a visualization is provided to allow alert the Operator's attention to the uninitialized or otherwise unconfigured controls.

Special initialization/binding use cases

1. Binding the toolkit controls to a Point Detail display

When displays are designed to be Point Detail displays, and be generic in such as way that the name of the Procedure is specified by the *DetailPointID* of the display, the initialization of the *ProcedureControl* and *ProcedurePanelControl* can be modified slightly by calling *InitializeWithDetailPointID()* instead of just *Initialize()* as shown below

```
...
pc1.InitializeWithDetailPointID()    // pc1 is the name of a ProcedureControl for instance
...
```

This causes the *ProcedureControl/ProcedurePanelControl* to initialize all other Toolkit controls bound to *SpecificProcedure* with their *ProcedureName* empty, to be initialized with the *ProcedureName* specified by the *Station.DetailPointID*.



Tip

This should only be used with true Detail displays since the *DetailPointID* is only set for these types of displays.

2. Binding the Toolkit controls to a point specified by logic in script

Similarly, when displays are designed in such as way that the name of the Procedure is to be specified by some logic in the *OnPageComplete* script of the Page, the initialization of the *ProcedureControl* and *ProcedurePanelControl* can be modified slightly by calling *InitializeWithProcedureName(aProcedureName)* instead of just *Initialize()* as shown below:

```
...
Logic to specify aProcedureName goes here...
pc1.InitializeWithProcedureName(aProcedureName)    // pc1 is the name of a ProcedureControl for
```

```
instance
...
```

This causes the ProcedureControl/ProcedurePanelControl to initialize all other Toolkit controls bound to *SpecificProcedure* with their *ProcedureName* empty, to be initialized with the *ProcedureName* passed to *InitializeWithProcedureName()*.

3. Binding a ProcedureListControl to a procedure list specified by logic in script

When a ProcedureListControl is desired but the actual list of procedures is not known at display building time, the *ProcedureList* can be set in the *OnPageComplete* script, just before calling the ProcedureListControl's *Initialize* method as shown below:

```
...
Logic to create aDynamicallyCreatedList goes here...
plc1.ProcedureList = aDynamicallyCreatedList // e.g. "SCM1,RCM13,SCM25"
plc1.Initialize() // plc1 is the name of a ProcedureListControl for instance
...
```

4. Binding a UnitListControl to a unit list specified by logic in script

When a UnitListControl is desired but the actual list of units is not known at display building time, the *UnitList* can be set in the *OnPageComplete* script, just before calling the UnitListControl's *Initialize* method as shown below:

```
...
Logic to create aDynamicallyCreatedList goes here...
ulc1.UnitList = aDynamicallyCreatedList // e.g. "MIXER1,BLENDER1,BLENDER2"
ulc1.Initialize() // ulc1 is the name of a UnitListControl for instance
...
```

Dynamic binding use case scenario

Binding to Phase child procedure via EQUIPNAME parameter

Equipment controlled by a Phase can change dynamically, meaning that the child procedure associated with it can also change therefore making it impossible to hard code the procedure name, especially for a scenario where the procedure is a batch and has an activity associated with it. To expose all confirmable instructions (pending or not) for any procedure associated with the currently selected Phase, the technique shown in the following scenario can be used:

1. An Activity Table is added to the display to provide users with a way to select a batch.
2. A ProcedureControl is added and bound to the SelectedActivity. Calls to the ProcedureControl Initialize() and Shutdown() methods are added to the OnPageComplete and OnUnload event handlers respectively to bootstrap the Toolkit data source.
3. A StepAndPhaseListControl is added and also bound to the SelectedActivity so that users see the progress of the batch.
4. When confirmable instructions exist in the various Phases' child procedures, they are not immediately (or automatically) exposed in an InstructionListControl bound to the SelectedActivity. When it is necessary to show them in the same display, an InstructionListControl can be added and dynamically bound to the procedure associated with the EQUIPNAME parameter of the currently selects Phase in the StepAndPhaseListControl.
5. In order to dynamically bind the InstructionListControl, the SelectedPhase event first needs to have the following script added to it:

```
Sub SPLC_PhaseSelectedEvent(procedureName, phaseName)
On error resume next
if (procedureName) then
  if (phaseName) then
    equipname = me.GetEQUIPNAME(true)
    if (equipname) then
      ILC.Pause()
      ILC.BindTo = "Specific Procedure"
      ILC.procedureName = equipname
      ILC.restart()
    end if
  end if
end if
```

```
end if
End Sub
```

In the script above, **SPLC_PhaseSelectedEvent** is the OnPhaseSelected event handler for SLPC, a StepAndPhaseListControl. When a Phase is selected, the event is fired and the script retrieves the name of the procedure associated with the Phase as determined by the **EQUIPNAME** parameter returned by the GetEQUIPNAME() method. The passed parameter is set to **tur** to indicate that only procedures are of interest, thereby excluding UCMs.

If a procedure name is available, which by the way could be an activity name, the InstructionListControl name **ILC** is then dynamically bound to the procedure. **ILC** was initially bound to a SpecificProcedure but with the ProcedureName left unconfigured.

Procedure and sequence objects

Related topics

- “CommandButtonControl display object” on page 22
- “CommandDropDownControl display object” on page 23
- “CommandPanelControl display object” on page 24
- “ConfirmationPanelControl display object” on page 25
- “InformationControl display object” on page 26
- “InformationPanelControl display object” on page 27
- “InstructionControl display object” on page 28
- “InstructionListControl display object” on page 29
- “ModeAttributeButtonControl display object” on page 30
- “ModeAttributeDropDownControl display object” on page 31
- “ModeButtonControl display object” on page 32
- “ModeDropDownControl display object” on page 33
- “PhaseControl display object” on page 34
- “ProcedureControl display object” on page 35
- “ProcedureListControl display object” on page 36
- “ProcedurePanelControl display object” on page 37
- “StepandPhaseListControl display object” on page 38
- “StepControl display object” on page 39
- “TransitionControl display object” on page 40
- “TransitionListControl display object” on page 41
- “UnitControl display object” on page 42
- “UnitListControl display object” on page 43

CommandButtonControl display object



Figure 9: Command button control

Description

Creates a command button, which can be configured to send a specific command to the bound procedure. Note that the CommandButtonControl does not reflect the current state of the procedure, so when an **Abort** command is sent, for example, the button label does not change. The *Control Confirmation* option can be used to request confirmation before the command is sent.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “CommandButtonControl properties” on page 48

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

CommandDropDownControl display object



Figure 10: Command drop-down control

Description

Creates a command dropdown list, which can be configured to allow selection of a command to send to the bound procedure.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “CommandDropDownControl properties” on page 49

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

CommandPanelControl display object

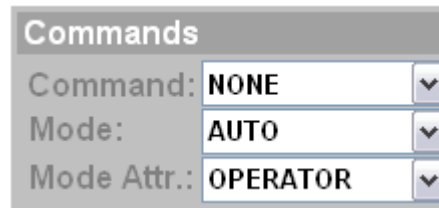


Figure 11: Command panel control

Description

A compound control that groups other basic or list controls for convenience or enhanced functionality. The CommandPanelControl groups a CommandDropDownControl, a ModeDropDownControl, and a ModeAttributeDropDownControl to provide a mini-control panel to send a command to the procedure or to set its Mode or Mode Attribute. You can configure whether the dropdown controls are shown and enabled by setting these properties:

- ShowCommands
- ShowModes
- ShowModeAttributes
- EnableCommands
- EnableModes
- EnableModeAttributes

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “CommandPanelControl properties” on page 50

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

ConfirmationPanelControl display object

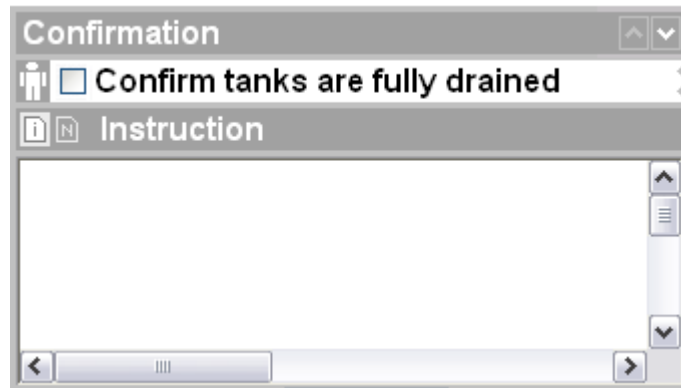


Figure 12: Confirmation panel control

Description

A compound control that groups a set of basic and other compound controls to provide an efficient way to view all relevant information required to confirm a confirmable instruction.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ConfirmationPanelControl properties” on page 51

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “InstructionSelectedEvent” on page 75

InformationControl display object



Figure 13: Information control

Description

Provides a visualization for a Step Warning, a Step or Instruction Note, or a list of instructions, depending on which procedure element and parameter has been bound to it.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “InformationControl properties” on page 52

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “InstructionSelectedEvent” on page 75
- “StepSelectedEvent” on page 76

InformationPanelControl display object



Figure 14: Information panel control

Description

Displays all information for the Step or Instruction bound to the control. The visibility of different information types within this control is determined by whether the associated parameter is configured in Control Builder and by setting the following properties:

- ShowWarning
- ShowInstruction
- ShowNote
- ShowInstructionList

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “InformationPanelControl properties” on page 53

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

InstructionControl display object

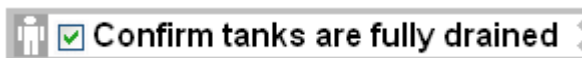


Figure 15: Instruction control

Description

Provides the visualization for an interactive confirmable instruction, its state, and status. This control can be configured to have a blinking Confirmation Pending icon to catch the Operator's attention, and provides a check box for the instruction confirmation. Navigation to related displays can be initiated from this control's shortcut menu.

Properties

- "General properties" on page 46
- "Shortcut menu properties" on page 47
- "InstructionControl properties" on page 54

Methods

- "Pause method" on page 73
- "Restart method" on page 74

Events

- "InstructionSelectedEvent" on page 75

InstructionListControl display object

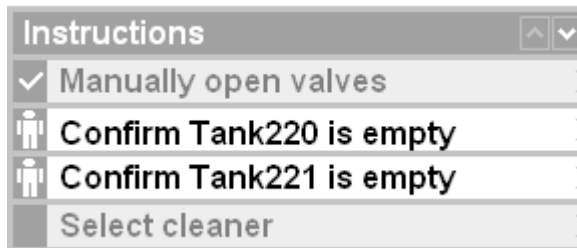


Figure 16: Instruction list control

Description

A list control containing and managing a set of InstructionControls. This control can be bound to all or just the active instructions of a procedure or step. As with other list-type controls, the InstructionListControl raises a selection event which can be used by other controls to dynamically rebind themselves to the currently selected instruction.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “InstructionListControl properties” on page 55

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “InstructionSelectedEvent” on page 75

ModeAttributeButtonControl display object



Figure 17: Mode attribute button control

Description

A button control that can be configured to set the bound procedure to a specific ModeAttribute. Note that the button label does not reflect the current mode, but the mode attribute values that can be selected. The ModeAttributeDropDownControl, however, displays the current mode attribute as well as showing the list of possible mode attribute values. When Operators should only have access to a subset of the mode attributes, it may be necessary to use a dedicated button or set of dedicated buttons to block some mode attributes. Note also that *Control Confirmation* can be used to request confirmation before the ModeAttribute is set.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ModeAttributeButtonControl properties” on page 56

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “ActivitySelectedEvent” on page 75
- “ProcedureSelectedEvent” on page 76
- “UnitListChangedEvent” on page 77
- “UnitSelectedEvent” on page 78

ModeAttributeDropDownControl display object



Figure 18: Mode attribute drop-down control

Description

A button control that allows the user to select and set the mode attribute for a procedure. This control, unlike the ModeAttributeButtonControl shows both the current mode attribute and the possible mode attribute values that can be applied.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ModeAttributeDropDownControl properties” on page 57

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

ModeButtonControl display object



Figure 19: Mode button control

Description

A button control that can be configured to set the bound procedure to a specific mode. Note that the button label does not reflect the current mode but rather the mode that can be set by clicking the control. The ModeDropdownControl, however, displays both the current mode value and a list of possible mode selections to be applied. If operators should only have access to a subset of the modes, it may be necessary to create a dedicated button or set of dedicated buttons to block some modes. Note also that *Control Confirmation* can be used to request confirmation before the mode is set.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ModeButtonControl properties” on page 58

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

ModeDropDownControl display object



Figure 20: Mode drop-down control

Description

A button control that allows the user to select and set the mode for a procedure. This control, unlike the ModeButtonControl shows both the current mode value and a list of possible mode selections.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ModeDropDownControl properties” on page 59

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

None

PhaseControl display object



Figure 21: Phase control

Description

Provides the visualization for a Phase block, its state and status. It also lets the operator know whether there are errors or special Phase states with relation to the child procedure. Through its details popup, the PhaseControl display object enables commands to be sent and mode values to be changed for the child procedure. Navigation to other related displays can also be initiated from its shortcut menu.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “PhaseControl properties” on page 60

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “PhaseSelectedEvent” on page 76

ProcedureControl display object



Figure 22: Procedure control

Description

Provides the visualization for a procedure, SCM, or RCM, as well as its state and status. This control also indicates when there are errors or pending confirmable instructions requiring attention. Through its details popup, the ProcedureControl provides access to commands and mode setting actions. Navigation to related displays can be initiated from its shortcut menu.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ProcedureControl properties” on page 61

Methods

- “Initialize method” on page 73
- “Pause method” on page 73
- “Restart method” on page 74
- “Shutdown method” on page 74

Events

- “ProcedureSelectedEvent” on page 76

ProcedureListControl display object

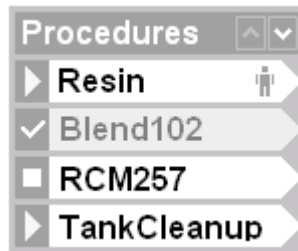


Figure 23: Procedure list control

Description

A list control that groups other basic controls in a list format. The ProcedureListControl contains and manages a set of ProcedureControls. As with all other list-type controls, the ProcedureListControl raises a selection event which can be used by other controls to dynamically rebind themselves to the currently selected procedure.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ProcedureListControl properties” on page 62

Methods

- “Initialize method” on page 73
- “Pause method” on page 73
- “Restart method” on page 74
- “Shutdown method” on page 74

Events

- “ProcedureSelectedEvent” on page 76

ProcedurePanelControl display object

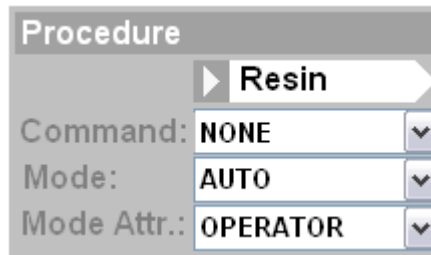


Figure 24: Procedure panel control

Description

A compound control that groups a ProcedureControl and a CommandPanelControl to form a mini-faceplate for a procedure. It can be bound to a procedure or a phase. When bound to a phase it visualizes and provides access to the phase child procedure. In addition, phase state warnings can be displayed when special conditions exist between the phase and its child procedure.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “ProcedurePanelControl properties” on page 63

Methods

- “Initialize method” on page 73
- “Pause method” on page 73
- “Restart method” on page 74
- “Shutdown method” on page 74

Events

- “ProcedureSelectedEvent” on page 76
- “PhaseSelectedEvent” on page 76

StepandPhaseListControl display object



Figure 25: Step and phase list control

Description

A list control that contains and manages a set of StepControls and/or PhaseControls. This control can be bound to all or only the active steps or phases, or a specific list of steps and/or phases in a procedure. As with all other list-type controls, the StepAndPhaseListControl will raise a selection event which can be used by other controls to dynamically rebind themselves to the currently selected item.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “StepandPhaseListControl properties” on page 64

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “ActivitySelectedEvent” on page 75
- “ProcedureSelectedEvent” on page 76
- “PhaseSelectedEvent” on page 76
- “StepSelectedEvent” on page 76

StepControl display object



Figure 26: Step control

Description

Provides the visualization for a Step block, its state and status. The StepControl also indicates whether there are errors or pending confirmable instructions requiring attention. Navigation to related displays can be initiated from its shortcut menu.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “StepControl properties” on page 65

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “PhaseSelectedEvent” on page 76
- “StepSelectedEvent” on page 76

TransitionControl display object

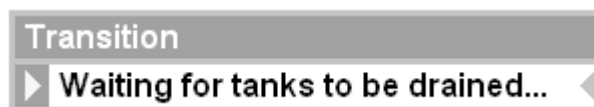


Figure 27: Transition control

Description

Provides the visualization for a Transition block, its state and status. At times, procedures may be spending a significant amount of time between steps waiting for some conditions to be true. This control provides a way to know what the Procedure is doing. Navigation to related displays can be initiated from its shortcut menu.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “TransitionControl properties” on page 66

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “TransitionSelectedEvent” on page 77

TransitionListControl display object



Figure 28: Transition list control

Description

A list control that contains and manages a set of TransitionControls. This control can be bound to all or only the active transitions, or a specific list of transitions in a given procedure. As with other list-type controls, the TransitionListControl will raise a selection event which can be used by other controls to dynamically rebind themselves to the currently selected transition.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “TransitionListControl properties” on page 67

Methods

- “Pause method” on page 73
- “Restart method” on page 74

Events

- “TransitionSelectedEvent” on page 77

UnitControl display object



Figure 29: Unit control

Description

Provides the visualization for a UCM and is somewhat similar to the ProcedureControl display object. Through its Details Popup and ToolTip, it provides additional information about which procedure (Activity or RCM) is currently associated with the Unit. Navigation to related displays can be initiated from its shortcut menu.

The UnitControl display object provides a way for the user to select a given unit and have other toolkit controls display related information when they are bound to the selected unit. In addition, and depending on the applicable use case scenarios, this control can also be used as a ProcedureObject provider, which enables the data access support required by other toolkit controls embedded in the display.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “UnitControl properties” on page 68

Methods

- “Initialize method” on page 73
- “Pause method” on page 73
- “Restart method” on page 74
- “Shutdown method” on page 74

Events

- “UnitSelectedEvent” on page 78

UnitListControl display object

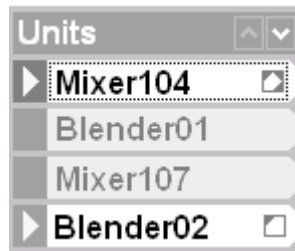


Figure 30: Unit list control

Description

Similar to the ProcedureListControl except that a list of units (UCMs) is visualized instead of a list of RCMs and/or SCMs. As with other list-type controls, the UnitListControl raises a selection event which can be used by other controls to dynamically rebind themselves to the currently selected unit and more precisely to the procedure that is currently using the unit, if any.

Properties

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “UnitListControl properties” on page 69

Methods

- “Initialize method” on page 73
- “Pause method” on page 73
- “Restart method” on page 74
- “Shutdown method” on page 74

Events

- “UnitSelectedEvent” on page 78

Procedure and sequence object properties

Related topics

- “General properties” on page 46
- “Shortcut menu properties” on page 47
- “CommandButtonControl properties” on page 48
- “CommandDropDownControl properties” on page 49
- “CommandPanelControl properties” on page 50
- “ConfirmationPanelControl properties” on page 51
- “InformationControl properties” on page 52
- “InformationPanelControl properties” on page 53
- “InstructionControl properties” on page 54
- “InstructionListControl properties” on page 55
- “ModeAttributeButtonControl properties” on page 56
- “ModeAttributeDropDownControl properties” on page 57
- “ModeButtonControl properties” on page 58
- “ModeDropDownControl properties” on page 59
- “PhaseControl properties” on page 60
- “ProcedureControl properties” on page 61
- “ProcedureListControl properties” on page 62
- “ProcedurePanelControl properties” on page 63
- “StepandPhaseListControl properties” on page 64
- “StepControl properties” on page 65
- “TransitionControl properties” on page 66
- “TransitionListControl properties” on page 67
- “UnitControl properties” on page 68
- “UnitListControl properties” on page 69

General properties

The General properties control the object's basic properties such as its position and size.

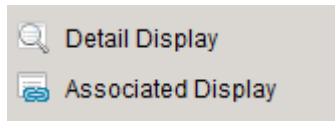
Property	Description
Identification	
Name	The object's name (<i>Public name</i>). Use this name when referring to the object in scripts.
ToolTip	The text that appears when a user hovers the mouse pointer over the object.
Style	
Left Top	The object's position from the top left of the display, in pixels.
Width Height	The object's size, in pixels.
Rotation	This property is not used by the toolkit controls.
Visibility	The object's visibility setting. The values are: <ul style="list-style-type: none"> • Hidden = The object is always hidden • Visible = The object is always visible • Inherit = The object's visibility is determined by its parent (For example, if an object is part of a group, setting the visibility of the group will apply the same setting to the object.)
Tabbing	
Enable tab stop	If selected, users can select the object by pressing the Tab key on their keyboard.
Tab index	Specifies the order in which objects are selected. Note that: <ul style="list-style-type: none"> • Objects with a positive tab index are ordered before object(s) with a tab index of 0 (zero). • Objects with a negative tab index are excluded from the tabbing order. • Objects with the same tab index are ordered according to the order in which they appear in the HTML source for the display.

Shortcut menu properties

The Shortcut menu properties specify the display object's shortcut menu, which appears when a user right-clicks on the display object.

Property	Description
Default shortcut menu	Uses the default shortcut menu for the associated control.
Customized shortcut menu	Uses a specified custom shortcut menu. Provides a box for specifying the path for the customized toolkit shortcut menu (ProcSeqTKT_sm.xml) menu. Click Browse to navigate to a pre-defined path.

The contents of the customized shortcut menu are shown here:



- Detail Display, which invokes the TableView display for this control (rather than the main tab of the Detail display)
- Associated Display, which invokes the display previously configured as the AssociatedDisplay in the CM or the RCM. If no display has been configured and you click this link, a message will display in the message zone stating `no associated display configured`.

CommandButtonControl properties

The CommandButtonControl properties control the object's specific properties data source and binding, appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The name of the procedure. Only applicable if the CommandButtonControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the CommandButtonControl is bound to a specific unit, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Command	Specifies the command to associate with this control. Valid options are: <ul style="list-style-type: none"> • Abort • Active • Cancel • Hold • Inactive • Interrupt • Reset • Restart • Resume • Start • Stop
Appearance and Functionality	
Hide when not bound	When selected, the control will be hidden when it is not bound to a procedure, activity, or unit.
Show label	Specifies whether to show the control label on its left.
Label	Only applicable if Show Label is selected. Allows selection of the default label or provision of a custom label.
Caption	Only applicable if Show Label is selected. Allows selection of the default caption (ToolTip) or provision of a custom caption.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

CommandDropDownControl properties

The CommandDropDownControl properties control the object's specific properties data source and binding, appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The name of the procedure. Only applicable if the CommandDropDownControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the CommandDropDownControl is bound to a specific unit, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Appearance and Functionality	
Hide when not bound	When selected, the control will be hidden when it is not bound to a procedure, activity, or unit.
Show label	Specifies whether to show the control label on its left.
Label	Only applicable if Show Label is selected. Allows selection of the default label or provision of a custom label.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.


CommandPanelControl properties

The CommandPanelControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The procedure name (<i>Public name</i>). Only applicable if the CommandPanelControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The unit name. Only applicable if the CommandPanelControl is bound to a specific unit, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Appearance and Functionality	
Show Header	If selected, show the control header.
Show Labels	If selected, show the label to the left of the control.
Hide when not bound	If selected, hides the control when it is not bound to an activity, procedure, or unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Controls	
Show Command	Specifies whether the Command menu item and drop down list are shown on the control.
Show Mode	Specifies whether the Mode menu item and drop down list are shown on the control.
Show ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are shown on the control.
Enable Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Enable Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
Enable ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

ConfirmationPanelControl properties

The ConfirmationPanelControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The procedure name (<i>Public name</i>) for this step or instruction. If the ConfirmationPanelControl is bound to a specific instruction or step it you can select the procedure for that instruction or step. If the control is bound to a selected step or instruction, Procedure Name , Step Name , and Instruction Index are hidden as this information is provided by the <i>PhaseSelectedEvent</i> , the <i>StepSelectedEvent</i> , or the <i>InstructionSelectedEvent</i> .
Unit Name	The unit name (<i>Public name</i>) for this step or instruction. If the ConfirmationPanelControl is bound to a specific instruction or step it you can select the unit for that instruction or step. If the control is bound to a selected step or instruction, Unit Name , Step Name , and Instruction Index are hidden as this information is provided by the <i>PhaseSelectedEvent</i> , the <i>StepSelectedEvent</i> , or the <i>InstructionSelectedEvent</i> .
Step Name	The step name for this instruction. Only required when the control is bound to a specific step.
Instruction Index	Only required when the control is bound to an instruction. Valid index values are 1 to 16, as this is the maximum number of outputs in a step.
Appearance and Functionality	
Only show when active	If selected, hides the control when the step or instruction is not active or known.
Hide when not bound	If selected, hides the control when it is not bound to a step or instruction.
Allow blinking	Specifies whether the Confirmation Pending  icon should blink or not.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

InformationControl properties

The InformationControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	Specifies the procedure for the step or instruction. Only applicable when the information control is bound to a specific step or instruction. If the control is bound to the currently selected step or instruction, Procedure Name , Step Name , and Instruction Index are hidden, as these details will be provided by the <i>StepSelectedEvent</i> and the <i>InstructionSelectedEvent</i> .
Info Type	Specifies the type of information. Valid values are <i>Warning</i> or <i>Note</i> for a step, <i>Instruction</i> or <i>Note</i> for an instruction.
Step Name	The name of the step. Only applicable when the information control is bound to a specific step or instruction. If the control is bound to the selected step or instruction, Procedure Name , Step Name , and Instruction Index are hidden, as these details will be provided by the <i>StepSelectedEvent</i> and the <i>InstructionSelectedEvent</i> .
Instruction Index	The number for this instruction. Valid values are a number between 1 and 16 as this is the maximum number of outputs in a step.
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show Icon	Specifies whether to show an icon to indicate the information type.
Only show when active	Specifies whether the control should be hidden when the step or instruction is not active or known.
Hide when not bound	If selected, will hide the control when it is not bound to a step or an instruction.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.


InformationPanelControl properties

The InformationPanelControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The procedure name (<i>Public name</i>) for this step or instruction. If the InformationPanelControl is bound to a specific instruction or step it will bind to the related property for that control, such as <i>procedure name</i> , <i>step name</i> . If the control is bound to a selected step or instruction, Procedure Name , Step Name , and Instruction Index are hidden as this information is provided by the <i>PhaseSelectedEvent</i> , the <i>StepSelectedEvent</i> , or the <i>InstructionSelectedEvent</i> .
Step Name	The step name (<i>Public name</i>) for this instruction. If the InformationPanelControl is bound to a specific instruction it will bind to the procedure element specified in the <i>PhaseSelectedEvent</i> , the <i>StepSelectedEvent</i> , or the <i>InstructionSelectedEvent</i> . If the control is bound to a selected instruction, Procedure Name , Step Name , and Instruction Index are hidden as this information is provided by the <i>PhaseSelectedEvent</i> , the <i>StepSelectedEvent</i> , or the <i>InstructionSelectedEvent</i> .
Instruction Index	The index for this instruction. Valid values are 1 to 16, as this is the maximum number of outputs in a step.
Appearance and Functionality	
Only show when active	If selected, hides the control when the step or instruction is not active or known.
Hide when not bound	If selected, hides the control when it is not bound to a step or instruction.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.


InstructionControl properties

The InstructionControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the binding type for this control, such as <i>procedure</i> .
Procedure Name	Specifies the instruction's procedure. Only applicable when the instruction is bound to a specific instruction. If the instruction is bound to the selected instruction Procedure Name , Step Name , and Instruction Index are hidden, as these details will be provided by the <i>InstructionSelectedEvent</i> .
Step Name	The name of the step. Only applicable when the instruction is bound to a specific instruction. If the instruction is bound to the selected instruction, Procedure Name , Step Name , and Instruction Index are hidden, as these details will be provided by the <i>InstructionSelectedEvent</i> .
Instruction Index	The number for this instruction. Valid values are a number between 1 and 16 as this is the maximum number of outputs in a step.
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show Checkbox	If selected, show a check box to confirm the instruction.
Only show when active	Specifies whether the control should be hidden when the instruction is not active or known.
Hide when not bound	If selected, will hide the control when it is not bound to an instruction.
Allow blinking	Specifies whether the Confirmation Pending  icon should blink or not.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

InstructionListControl properties

The InstructionListControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the binding type for this control, such as <i>procedure</i> .
Filter	Specifies whether all instructions, the active instructions, or a specific list of instructions is required. Selecting Instruction List invokes the Step Name box. For specific procedures, steps, or units, you can select either All Instructions or Active Instructions . For selected procedures, activities, steps, or units, you can only select Active Instructions .
Procedure Name	The procedure name (<i>Public name</i>). Only applicable if the InstructionListControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The unit name. Only applicable if the InstructionListControl is bound to a specific unit, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Step Name	Only applicable if you have selected a specific procedure, step, or unit in the Bind to box and Instruction List in the Filter box. Lists the steps you can select to associate with this instruction list, separated by a comma.
Appearance and Functionality	
Show checkbox	If selected, provides a check box beside the control for the user to indicate confirmation that the instruction has been completed.
Allow blinking	Specifies whether the Confirmation Pending  icon should blink or not.
Hide when not bound	If selected, hides the control when it is not bound to an activity, procedure, step, or unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Enable Controls	
Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

ModeAttributeButtonControl properties

The ModeAttributeButtonControl properties control the object's specific properties data source and binding, appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The name of the procedure. Only applicable if the ModeAttributeButtonControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the ModeAttributeButtonControl is bound to a specific unit, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Mode Attr	Specifies the mode to associate and set the procedure, unit, or activity to. Valid options are: <ul style="list-style-type: none"> • Normal • Operator • Program The default value is <i>Operator</i> .
Appearance and Functionality	
Hide when not bound	When selected, the control will be hidden when it is not bound to a procedure, activity, or unit.
Show label	Specifies whether to show the control label on its left.
Label	Only applicable if Show Label is selected. Allows selection of the default label or provision of a custom label.
Caption	Only applicable if Show Label is selected. Allows selection of the default caption (ToolTip) or provision of a custom caption.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

ModeAttributeDropDownControl properties

The ModeAttributeDropDownControl properties control the object's specific properties data source and binding, appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The name of the procedure. Only applicable if the ModeAttributeDropDownControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the ModeAttributeDropDownControl is bound to a specific unit, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Appearance and Functionality	
Hide when not bound	When selected, the control will be hidden when it is not bound to a procedure, activity, or unit.
Show label	Specifies whether to show the control label on its left.
Label	Only applicable if Show Label is selected. Allows selection of the default label or provision of a custom label.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

ModeButtonControl properties

The ModeButtonControl properties control the object's specific properties data source and binding, appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The name of the procedure. Only applicable if the ModeButtonControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the ModeButtonControl is bound to a specific unit, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Mode	Specifies the mode to associate and set the procedure, unit, or activity to. Valid options are: <ul style="list-style-type: none"> • Auto • Manual • Normal • Safestep • Semiauto • Singlestep The default value is <i>Auto</i> .
Appearance and Functionality	
Hide when not bound	When selected, the control will be hidden when it is not bound to a procedure, activity, or unit.
Show label	Specifies whether to show the control label on its left.
Label	Only applicable if Show Label is selected. Allows selection of the default label or provision of a custom label.
Caption	Only applicable if Show Label is selected. Allows selection of the default caption (ToolTip) or provision of a custom caption.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

ModeDropDownControl properties

The ModeDropDownControl properties control the object's specific properties data source and binding, appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The name of the procedure. Only applicable if the ModeDropDownControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the ModeDropDownControl is bound to a specific unit, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Appearance and Functionality	
Hide when not bound	When selected, the control will be hidden when it is not bound to a procedure, activity, or unit.
Show label	Specifies whether to show the control label on its left.
Label	Only applicable if Show Label is selected. Allows selection of the default label or provision of a custom label.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

PhaseControl properties

The PhaseControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	Specifies the phases's procedure. Only applicable when the phase is bound to a specific phase. If the phase is bound to the selected phase, Procedure Name and Phase Name are hidden, as these details will be provided by the <i>PhaseSelectedEvent</i> .
Phase Name	The name of the phase. Only applicable when the phase is bound to a specific phase. If the phase is bound to the selected phase, Procedure Name and Phase Name are hidden, as these details will be provided by the <i>PhaseSelectedEvent</i> .
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show description	If selected, show the description instead of the phase name
Only show when active	Specifies whether the control should be hidden when the phase is not active or known.
Hide when not bound	If selected, will hide the control when it is not bound to a phase.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Enable Controls	
Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

ProcedureControl properties

The ProcedureControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The procedure name (<i>Public name</i>). Only applicable if the ProcedureControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The name of the unit. Only applicable if the ProcedureControl is bound to a specific procedure, in which case, Procedure Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show description	If selected, show the description instead of the procedure name
Hide when not bound	If selected, will hide the control when it is not bound to a procedure, activity or unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Enable Controls	
Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

ProcedureListControl properties

The ProcedureListControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure List	The names of the procedures to be included in the list, separated by a comma.
Appearance and Functionality	
Show description	If selected, show the description instead of the procedure name.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Enable Controls	
Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

ProcedurePanelControl properties

The ProcedurePanelControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	The procedure name (<i>Public name</i>). If the ProcedurePanelControl is bound to a specific activity, procedure, or phase you can specify the procedure and phase names. If the control is bound to a selected activity, procedure, or phase, Procedure Name , and Phase Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>PhaseSelectedEvent</i> , or the <i>ActivitySelectedEvent</i> .
Unit Name	The unit name (<i>Public name</i>). If the ProcedurePanelControl is bound to a specific activity, unit, or phase you can specify the unit and phase names. If the control is bound to a selected activity, unit, or phase, Unit Name , and Phase Name are hidden as this information is provided by the <i>UnitSelectedEvent</i> , the <i>PhaseSelectedEvent</i> , or the <i>ActivitySelectedEvent</i> .
Phase Name	The phase name. Only required when the control is bound to a specific phase.
Appearance and Functionality	
Show Header	If selected, shows the header for this control.
Show Labels	If selected, shows the label to the side of the control.
Show Description	If selected, shows the description in place of the Procedure Name.
Hide when not bound	If selected, hides the control when it is not bound to an activity, phase, procedure, or unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Controls	
Show Command	Specifies whether the Command menu item and drop down list are shown on the control.
Show Mode	Specifies whether the Mode menu item and drop down list are shown on the control.
Show Mode Attribute	Specifies whether the Mode Attribute menu item and drop down list are shown on the control.
Enable Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Enable Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
Enable Mode Attribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

StepandPhaseListControl properties

The StepandPhaseListControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Filter	Specifies whether all steps, phases, or combinations of the two, or just the active ones should be included in the list; or whether a specific list of steps, phases, or combination of the two is required. Selecting specific steps, phases, or combinations of both invokes the Phase List , Step List , or Step/Phase List boxes as appropriate.
Procedure Name	The procedure name (<i>Public name</i>). Only applicable if the StepandPhaseListControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The unit name. Only applicable if the StepandPhaseListControl is bound to a specific unit, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Step List	Only applicable if you have select a specific procedure or unit in the Bind to box and Step List in the Filter box. Lists the steps to be included in the list, separated by a comma.
Phase List	Only applicable if you have select a specific procedure or unit in the Bind to box and Phase List in the Filter box. Lists the phases to be included in the list, separated by a comma.
Step/Phase List	Only applicable if you have select a specific procedure or unit in the Bind to box and Step/Phase List in the Filter box. Lists the steps/phases to be included in the list, separated by a comma.
Appearance and Functionality	
Show description	If selected, show the description instead of the procedure name.
Hide when not bound	If selected, hides the control when it is not bound to an activity, procedure, or unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Enable Controls	
Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

StepControl properties

The StepControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	Specifies the step's procedure. Only applicable when the step is bound to a specific step. If the step is bound to the selected step, Procedure Name and Step Name are hidden, as these details will be provided by the <i>StepSelectedEvent</i> .
Step Name	The name of the step. Only applicable when the step is bound to a specific step. If the step is bound to the selected step, Procedure Name and Step Name are hidden, as these details will be provided by the <i>StepSelectedEvent</i> .
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show description	If selected, show the description instead of the step name
Only show when active	Specifies whether the control should be hidden when the step is not active or known.
Hide when not bound	If selected, will hide the control when it is not bound to a step.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

TransitionControl properties

The TransitionControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Procedure Name	Specifies the transition's procedure. Only applicable when the transition is bound to a specific transition. If the transition is bound to the transition phase, Procedure Name and Transition Name are hidden, as these details will be provided by the <i>TransitionSelectedEvent</i> .
Transition Name	The name of the transition. Only applicable when the transition is bound to a specific transition. If the transition is bound to the selected transition, Procedure Name and Transition Name are hidden, as these details will be provided by the <i>TransitionSelectedEvent</i> .
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show description	If selected, show the description instead of the transition name
Only show when active	Specifies whether the control should be hidden when the transition is not active or known.
Hide when not bound	If selected, will hide the control when it is not bound to a transition.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

TransitionListControl properties

The TransitionListControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Filter	Specifies whether all transitions, the active transitions, or a specific list of transitions is required. Selecting Transition List invokes the Transition List , box.
Procedure Name	The procedure name (<i>Public name</i>). Only applicable if the TransitionListControl is bound to a specific procedure, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Unit Name	The unit name. Only applicable if the TransitionListControl is bound to a specific unit, in which case, Unit Name is hidden. If the control is bound to a selected procedure, activity or unit, both Procedure Name and Unit Name are hidden as this information is provided by the <i>ProcedureSelectedEvent</i> , the <i>ActivitySelectedEvent</i> , or the <i>UnitSelectedEvent</i> .
Transition List	Only applicable if you have select a specific procedure or unit in the Bind to box and Transition List in the Filter box. Lists the transitions to be included in the list, separated by a comma.
Appearance and Functionality	
Show description	If selected, show the description instead of the procedure name.
Hide when not bound	If selected, hides the control when it is not bound to an activity, procedure, or unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.
Enable Controls	
Command	Specifies whether the Command menu item and drop down list are enabled on the control.
Mode	Specifies whether the Mode menu item and drop down list are enabled on the control.
ModeAttribute	Specifies whether the Mode Attribute menu item and drop down list are enabled on the control.

UnitControl properties

The UnitControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control, such as <i>procedure</i> .
Unit Name	Specifies the name of the unit (<i>Public Name</i>).
Appearance and Functionality	
Show Header	Specifies whether to display the control header.
Show Description	When selected, displays the description text in place of the unit name.
Hide when not bound	If selected, will hide the control when it is not bound to a unit.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

UnitListControl properties

The UnitListControl properties control the object's specific properties for data source and binding, and appearance and functionality.

Property	Description
Data Source and Binding	
Bind to	Specifies the type of binding for this control.
Unit List	The names of the units to be included in the list, separated by a comma.
Appearance and Functionality	
Show description	If selected, show the description instead of the procedure name.
Font	Specifies whether to use the default font or a custom font. If necessary, browse to the location of the font to populate the box with the correct path details.

Procedure and sequence object model reference

Related topics

“Overview of the procedure and sequence object model” on page 72

“Methods” on page 73

“Events” on page 75

Overview of the procedure and sequence object model

Procedure and sequence object model

The following figure shows a simplified version of the document object model (DOM) as it applies to displays. The main objects are the document (the display itself) and the children (the objects in the display). The toolkit controls are included as children in the DOM.

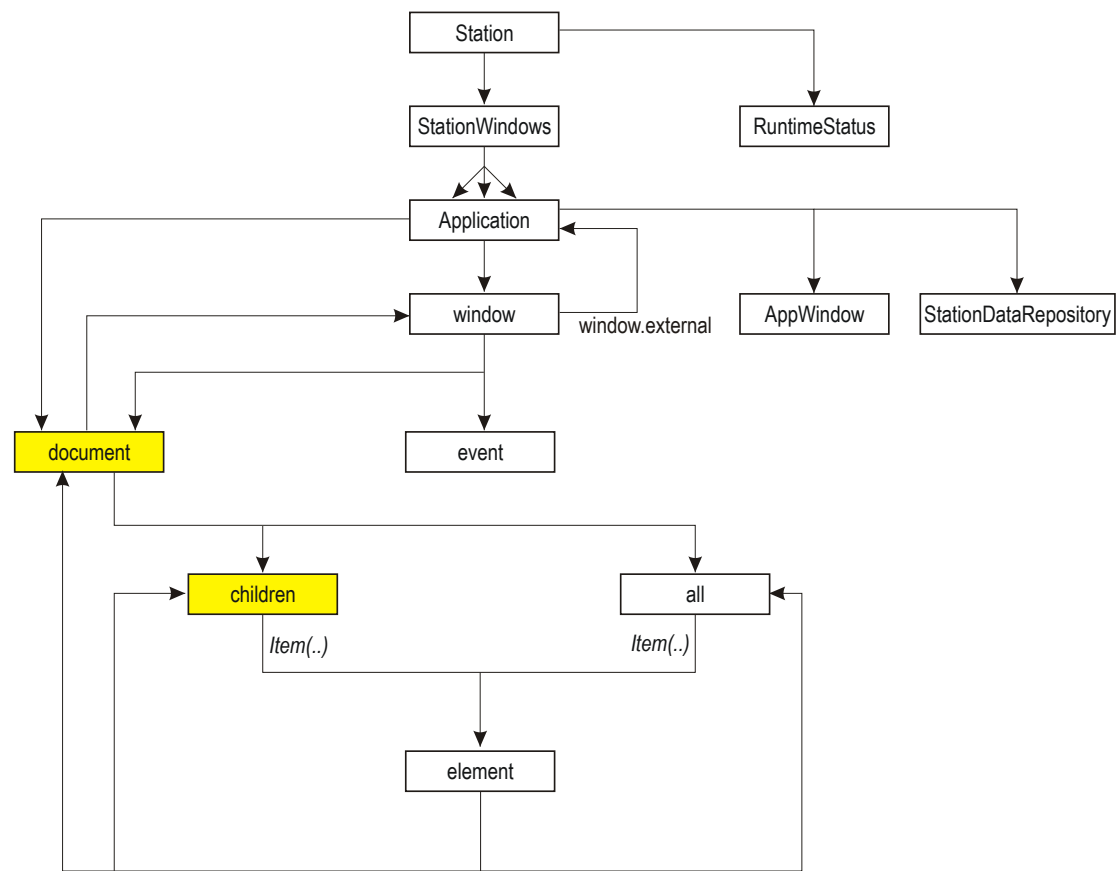


Figure 31: Document object model (DOM)

Each of the toolkit objects has a set of methods and events. These are documented on the following pages.

Methods

Each object has a set of methods and events. The methods are listed here in alphabetical order.

Related topics

“Initialize method” on page 73

“Pause method” on page 73

“Restart method” on page 74

“Shutdown method” on page 74

Initialize method

Applicable to

- “ProcedureControl display object” on page 35
- “ProcedureListControl display object” on page 36
- “ProcedurePanelControl display object” on page 37
- “UnitControl display object” on page 42
- “UnitListControl display object” on page 43

Description

Responsible for initializing the control

Additional initialize variations

InitializeWithDetailPointId(): Procedure control

InitializeWithProcedureName(string procedureName): Procedure control

InitializeWithProcedureList(string procedureList): Procedure list control

InitializeWithUnitList(string unitList): Unit list control

InitializeWithUnitName(string unitName): Unit control

Method name	Description
Initialize	Method responsible for initializing the control. The method needs to be called once per display, and only once using one of the ProcedureObject providers. For all other controls you do not need to make the call.

Pause method

Applicable to

All controls

Description

Responsible for pausing the control

Method name	Description
Pause	Method responsible for pausing the control. You can call this method after the control has been <i>initialized</i> , and it can be called any number of times in combination with the Restart method.

Restart method

Applicable to

All controls

Description

Responsible for restarting the control

Method name	Description
Restart	Method responsible for restarting the control. You can call this method after the control has been <i>initialized</i> , and it can be called any number of times in combination with the Pause method.

Shutdown method

Applicable to

- “ProcedureControl display object” on page 35
- “ProcedureListControl display object” on page 36
- “ProcedurePanelControl display object” on page 37
- “UnitControl display object” on page 42
- “UnitListControl display object” on page 43

Description

Responsible for initializing the control

Method name	Description
Shutdown	Method responsible for shutting down the control. The method needs to be called once per display, and only once using one of the ProcedureObject providers. For all other controls you do not need to make the call.

Events

Each object has a set of methods and events. The events are listed here in alphabetical order.

Related topics

- “ActivitySelectedEvent” on page 75
- “InstructionSelectedEvent” on page 75
- “PhaseSelectedEvent” on page 76
- “ProcedureSelectedEvent” on page 76
- “StepSelectedEvent” on page 76
- “TransitionSelectedEvent” on page 77
- “UnitListChangedEvent” on page 77
- “UnitSelectedEvent” on page 78

ActivitySelectedEvent

Controls that use this event

Action	Event Name
Fired by	The toolkit infrastructure when a selection is made in the Activity Table.

Event description

Action	Description
When consumed	The control listens to this event to keep track of the currently selected activity when bound to that activity.

InstructionSelectedEvent

Controls that use this event

Action	Event Name
Fired by	“ConfirmationPanelControl display object” on page 25 “InstructionControl display object” on page 28 “InstructionListControl display object” on page 29

Event description

Action	Description
When fired	A control will fire this event when a new instruction is selected by the user.

PhaseSelectedEvent

Controls that use this event

Action	Event Name
Fired by	"PhaseControl display object" on page 34 "StepandPhaseListControl display object" on page 38

Event description

Action	Description
When fired	If the control is selected by the user and it is not bound to the selected phase, it will fire this event.

ProcedureSelectedEvent

Controls that use this event

Action	Event Name
Fired by	"ProcedureControl display object" on page 35 "ProcedureListControl display object" on page 36 "ProcedurePanelControl display object" on page 37

Event description

Action	Description
When fired	If the user selects the control and it is not bound to the selected procedure, the control will fire this event.

StepSelectedEvent

Controls that use this event

Action	Event Name
Fired by	"StepControl display object" on page 39 "StepandPhaseListControl display object" on page 38

Event description

Action	Description
When fired	If the control is selected by the user and it is not bound to the selected step, it will fire this event.

TransitionSelectedEvent

Controls that use this event

Action	Event Name
Fired by	"TransitionControl display object" on page 40 "TransitionListControl display object" on page 41

Event description

Action	Description
When fired	If the user selects this control and it is not bound to the selected transition, it fires this event.

UnitListChangedEvent

Controls that use this event

Action	Event Name
Fired by	No controls fire this event

Event description

Action	Description
When consumed	The control listens to this event to keep track of the currently selected unit when bound to a specific unit. When this event is received, the control checks whether the unit it is bound to is in the new unit list. If it is, the event is shown, if not the event is hidden.

UnitSelectedEvent

Controls that use this event

Action	Event Name
Consumed by	“CommandButtonControl display object” on page 22 “CommandDropDownControl display object” on page 23 “CommandPanelControl display object” on page 24 “ConfirmationPanelControl display object” on page 25 “InstructionListControl display object” on page 29 “ModeAttributeButtonControl display object” on page 30 “ModeAttributeDropDownControl display object” on page 31 “ModeButtonControl display object” on page 32 “ModeDropDownControl display object” on page 33 “ProcedureControl display object” on page 35 “ProcedurePanelControl display object” on page 37 “StepandPhaseListControl display object” on page 38 “TransitionListControl display object” on page 41 “UnitControl display object” on page 42 “UnitListControl display object” on page 43
Fired by	“UnitControl display object” on page 42 “UnitListControl display object” on page 43

Event description

Action	Description
When consumed	The control listens to this event to keep track of the currently selected unit when bound to that unit.
When fired	If the control is selected and has not been bound to the selected unit, it will fire this event.

Notices

Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at <http://www.honeywell.com/ps/thirdpartylicenses>.

Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

<http://www.honeywellprocess.com/support>

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

<https://honeywell.com/pages/vulnerabilityreporting.aspx>

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.
- or
- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, <https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx>.

Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see <http://www.automationcollege.com>.

