



Vykon Pro User Guide

VykonPro Service.....	2
Abs Time To Status String.....	2
Active Level To Int.....	2
Alarm Data Filter.....	3
Boolean Bql.....	3
Block Demand Calc.....	3
Boolean Bql.....	5
Boolean Delay With Timer.....	6
Boolean Switch.....	6
Boolean To Balancer Cmd.....	7
Boolean To Sequencer Cmd.....	7
Connection Failover.....	8
Differential Compare.....	9
Enhanced Bound Label.....	12
Enhanced Bound Label Binding.....	13
Enhanced Hyperlink Binding.....	14
Enhanced Value Binding.....	15
File Servlet.....	16
Format To Ord Resolver.....	16
Hyperlink List.....	16
Hyperlink List Selector.....	17
Min Max Avg Bql.....	19
Numeric Cov Totalizer Ext.....	20
Numeric Delta.....	20
Numeric History Analysis.....	21
Opt Start Stop.....	22
Peak Tracker.....	25
Rel Time Command.....	26
Rotational Load Sequencer.....	26
Runtime Balancer.....	27
Status Boolean To Trigger.....	28
Status String To Ord.....	28
Time Latch.....	28
Tool Tip Binding.....	29
User Name Binding.....	30
Vykon Pro Report Source.....	31
WB Messenger.....	31
Weekly Trigger.....	33
Document Change Log.....	34

Vykon Pro User Guide

The vykonPro module is a collection of station components and px widgets that are provided as a value add to the Vykon channel partners. The station components and px widgets only function in a station whose license is branded Vykon (brandId="vykon" or brandId="vykon.security"). There are no extra features required in the license file for the components to function.

A demonstration station called vykonProTest is available from Niagara Central. The station demonstrates how each of the components in the vykonPro module can be implemented.

VykonPro Service

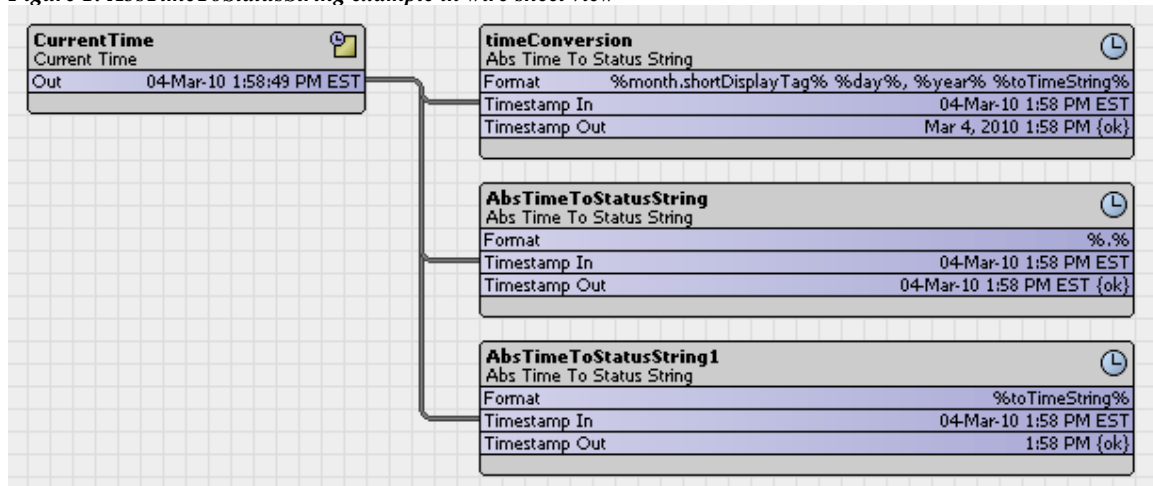
The VykonPro Service is located in the root of the vykonPro palette. The service must be located in the station's services container and is required to be present in the station for some of the components in the palette to function.

The service maintains a worker which various components from the vykonPro module post their work to. This insures that lengthy actions are posted on the VykonPro Service thread instead of the main engine thread.

Abs Time To Status String

The AbsTimeToStatusString component is used to enter a configurable BFormat text string which is used to format a BAbsTime property and set the value of a StatusString output slot.

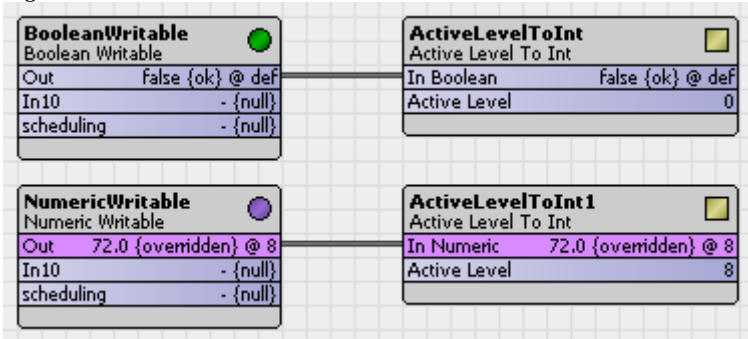
Figure 1: AbsTimeToStatusString example in wire sheet view



Active Level To Int

The Active Level To Int component is used to break out the active level from a control point's status. Extracting the active level from the control point can be useful in px views for animating widget properties based on the point's active level. The Active Level To Int component is located in the vykonPro palette under the conversion sub folder.

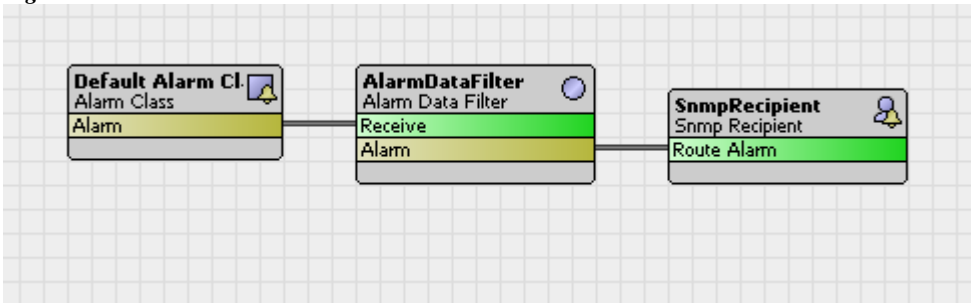
Figure 3: Active Level To Int in wire sheet view



Alarm Data Filter

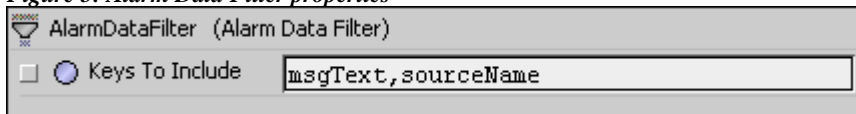
The Alarm Data Filter is designed to be used between an Alarm Class component and a SNMP Recipient to limit the alarmData keys which are sent to the SNMP alarm recipient.

Figure 4: Alarm Data Filter in wire sheet view



The alarmDataFilter component has one property named 'Keys To Include' which is used to configure which alarmData keys from the alarm record are included when the Alarm topic is fired. The order of the keys in the filtered alarm record alarmData will be in the order which they are entered in the 'Keys To Include' property.

Figure 5: Alarm Data Filter properties



Block Demand Calc

The Block Demand Calc component is used to calculate a variety of consumption and demand values based on accumulated pulse counts and peakMode. The component calculates block demand every 5 minutes. Consumption is calculated on change of value of the current pulse count slot. The Block Demand Calc component is located in the vykonPro palette under the energy sub folder.

Properties

- **Status** - BStatus property which reflects the current status of the component
- **Last Fail Cause** - String property which displays the current fault cause.
- **Last Fault Time** - BAbsTime property which displays the time of the last fault occurrence.
- **Consumption Facets** - BFacets property which is applied to all consumption type properties of the component.

- **Demand Facets** - BFacets property which is applied to all demand type properties of the component.
- **Current Pulse Count** - BStatusNumeric input property which is linked to the source pulse counts. Note that the source point could be pulse counts or actual consumption.
- **Kwh Per Pulse** - Double property which configures the ratio used to convert from pulses to consumption. If linking consumption to currentPulseCount property, this value should be 1.
- **Kwh Per Pulse Facets** - BFacets property which is applied to the kwhPerPulse slot.
- **Max Pulse Frequency** - Double property used to configure the maximum allowable pulse frequency for the currentPulseCount slot. Pulse counts which accumulate at a frequency higher than the configured maxPulseFrequency will be ignored in the calculation.
- **Actual Pulse Frequency** - Double property which displays the calculated pulse frequency.
- **Meter Rollover** - Long property which configures the maximum expected pulse count value. This value will vary depending on the device being used to count the consumption pulses. Check the device's literature to determine the value.
- **Peak Mode** - BStatusBoolean input which determines the calculation mode of the component. When true (onPeak) the calculated consumption and demand values are stored in the appropriate slots for onPeak values. When false (offPeak) the calculated consumption and demand values are stored in the appropriate slots for offPeak values.
- **Kwh Total** - BStatusNumeric slot which stores the total accumulated consumption regardless of peakMode.
- **Kwh Off Peak Total** - BStatusNumeric slot which stores the total accumulated consumption during the offPeak time.
- **Kwh On Peak Total** - BStatusNumeric slot which stores the total accumulated consumption during the onPeak time.
- **Monthly Kwh** - BStatusNumeric slot which stores the total accumulated consumption regardless of peakMode since the last monthlyReset action was executed.
- **Monthly Kwh Off Peak** - BStatusNumeric slot which stores the total accumulated consumption during the offPeak time since the last monthlyReset action was executed.
- **Monthly Kwh On Peak** - BStatusNumeric slot which stores the total accumulated consumption during the onPeak time since the last monthlyReset action was executed.
- **Kwh5** - BStatusNumeric slot which stores the consumption from the previous 5 minute period.
- **Kwh15** - BStatusNumeric slot which stores the consumption from the previous 15 minute period.
- **Kwh30** - BStatusNumeric slot which stores the consumption from the previous 30 minute period.
- **Max Kw** - BStatusNumeric slot which stores the maximum calculated Kw value. This is either the maximum Kw15 or Kw30 value depending on the peakKwInterval property selection.
- **Max Kw Timestamp** - BAbsTime slot which stores the timestamp at which the maxKw value was calculated.
- **Max Kw Off Peak** - BStatusNumeric slot which stores the maximum calculated Kw value during the offPeak time. This is either the maximum Kw15 or Kw30 value depending on the peakInterval property selection.
- **Max Kw Off Peak Timestamp** - BAbsTime slot which stores the timestamp at which the maxKwOffPeak value was calculated.
- **Max Kw On Peak** - BStatusNumeric slot which stores the maximum calculated Kw value during the onPeak time. This is either the maximum Kw15 or Kw30 value depending on the peakInterval property selection.
- **Max Kw On Peak Timestamp** - BAbsTime slot which stores the timestamp at which the maxKwOnPeak value was calculated.
- **Peak Kw Interval** - Enum property which configures the property used for calculating the peak Kw values.
- **Kw5** - BStatusNumeric slot which stores the calculated demand for the previous 5 minute period.
- **Kw15** - BStatusNumeric slot which stores the calculated demand for the previous 15 minute period.
- **Kw30** - BStatusNumeric slot which stores the calculated demand for the previous 30 minute period.

Actions

- **Preset Kwh** - The action prompts the user for values to preset the kwhTotal, kwhOffPeakTotal, kwhOnPeakTotal, monthlyKwh, monthlyKwhOffPeak, and monthlyKwhOnPeak properties.
- **Preset Kw** - The action prompts the user for values to preset the maxKw, maxKwTimestamp, maxKwOffPeak, maxKwOffPeakTimestamp, maxKwOnPeak, and maxKwOnPeakTimestamp properties.
- **Monthly Reset** - The action resets the values to zero or null for each of the properties that store monthly values.
- **Reset** - The action resets all values to zero or null.

Boolean Bql

The Boolean Bql component is used to perform logical Boolean analysis against a group of Boolean components in the station. The component is located in the vykonPro palette in the hvac subfolder.

Figure 6: Boolean Bql properties

Property	Value
Status	{ok}
Fault Cause	
Enabled	enabled
Point Query	station: slot:/Zones bql:select * from control:BooleanPoint
Number Of Points	99.0 {ok}
Facets	>> [refresh icon]
Or Out	true {ok}
And Out	false {ok}
Update Values Interval	000000h 00m 05s
Refresh Query Interval	+00012h 00m 00s

The group of Boolean components is determined by entering a BQL query in the 'Point Query' field. The results of the Bql query are stored as an array of components. The array of components is updated automatically at the frequency specified by the 'Refresh Query Interval' property.

There is also an action available on the component named 'Initialize List' which forces the query to be performed and the array to be updated. This may be useful if performing station engineering work such as adding components.

In either case the number of components which are found by the Bql query is stored in the 'Number Of Points' property.

The Boolean Bql component evaluates the state of each component in the array at a frequency determined by the 'Update Values Interval'. The 'Or Out' slot is the result of a logical or analysis, and the 'And Out' slot is the result of a logical and analysis.

Boolean Delay With Timer

The Boolean Delay With Timer component provides a way to delay the status change of a Boolean status "out" property value by configuring an associated "Delay" property. Delay properties are provided for on (true) and off (false) statuses and are labeled "On Delay" and "Off Delay", respectively. The delay applies to any transition (status change from on to off or off to on) at the component's status Boolean input. Both delay times are configurable in terms of hours, minutes and seconds.

Figure 7: Boolean Delay With Timer properties

The screenshot shows the configuration panel for the Boolean Delay With Timer component. It includes the following properties:

- Update Time:** +00000h 00m 00.500s
- Time Facets:** showSeconds=true, showMilliseconds=false
- Time Remaining:** +00000h 00m 59s
- Out Facets:** trueText=true, falseText=false
- In:** false {ok} @ def
- On Delay:** +00000h 02m 10s
- Off Delay:** +00000h 01m 10s
- On Delay Active:** false
- Off Delay Active:** true
- Out:** true {ok}
- Out Not:** false {ok}

The 'Time Remaining' property indicates the amount of time remaining before the two output slots (Out and Out Not) change state. The 'Time Remaining' slot is used for both the on and off delay remaining time.

Boolean Switch

The Boolean Switch component selects between two StatusBoolean inputs based upon the Boolean value at the StatusBoolean input 'In Switch'. The Boolean Switch is located in the vykonPro palette under the util sub folder.

Figure 8: Boolean Switch properties

The screenshot shows the configuration panel for the Boolean Switch component. It includes the following properties:

- In Switch:** true {ok} @ def
- False Output:** - {null} @ def
- True Output:** false {ok} @ def
- Null Output:** false {ok}
- Out:** false {ok}

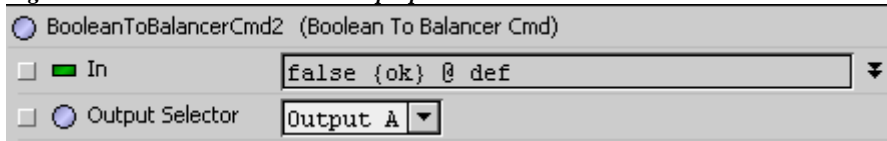
- **In Switch** – StatusBoolean input used in determining what the 'Out' slot value is set to.
- **False Output** – StatusBoolean input value which the 'Out' slot is set to when the 'In Switch' value is false.

- **True Output** – StatusBoolean input value which the ‘Out’ slot is set to when the ‘In Switch’ value is true.
- **Null Output** – StatusBoolean input value which the ‘Out’ slot is set to when the ‘In Switch’ status is null.

Boolean To Balancer Cmd

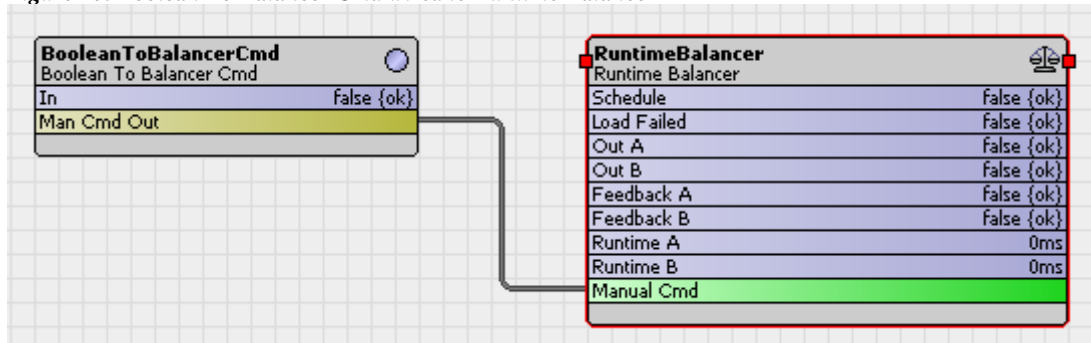
The Boolean To Balancer Cmd component provides an object for connecting control logic to the ‘ManualCmd’ action slot of the Runtime Balancer. When the ‘In’ property changes state the component fires a trigger via the ‘ManCmdOut’ slot, thereby commanding the load specified by the Outputs elector property. The component is located in the vykonPro palette in the hvac sub folder.

Figure 9: Boolean To Balancer Cmd properties



The ‘ManCmdOut’ topic on the Boolean To Balancer Cmd component is linked to the ManualCmd action on the Runtime Balancer component. This topic is fired on a change of value of the In property which passes the values of the ‘In’ and ‘OutputSelector’ properties as arguments to the Runtime Balancer. On a transition from false to true (In property) the trigger causes the specified load to be disabled in the Runtime Balancer components. On a transition from true to false (In property) the trigger causes the specified load to be enabled in the Runtime Balancer.

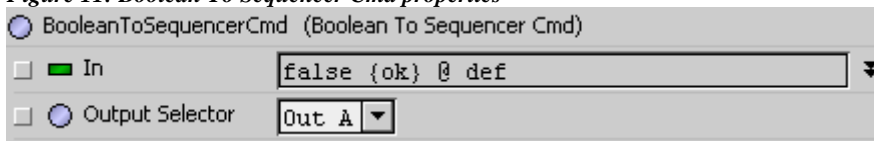
Figure 10: Boolean To Balancer Cmd wired to Runtime Balancer



Boolean To Sequencer Cmd

The Boolean To Sequencer Cmd component provides an object for connecting control logic to the ‘ManualCmd’ action slot of the Rotational Load Sequencer. When the ‘In’ property changes state the component fires a trigger via the ‘ManCmdOut’ slot, thereby commanding the load specified by the Outputs elector property. The component is located in the vykonPro palette in the hvac sub folder.

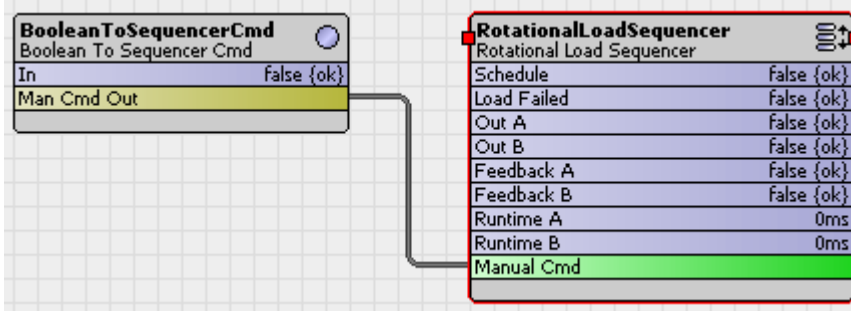
Figure 11: Boolean To Sequencer Cmd properties



The ‘ManCmdOut’ topic on the Boolean To Sequencer Cmd component is linked to the ManualCmd action on the Rotational Load Sequencer component. This topic is fired on a change of value of the In property

which passes the values of the 'In' and 'OutputSelector' properties as arguments to the Rotational Load Sequencer. On a transition from false to true (In property) the trigger causes the specified load to be disabled in the Rotational Load Sequencer components. On a transition from true to false (In property) the trigger causes the specified load to be enabled in the Rotational Load Sequencer.

Figure 12: Boolean To Sequencer Cmd wired to Rotational Load Sequencer



Connection Failover

The Connection Failover component is designed to ping an IP host at the configured frequency and provides a Boolean output slot which can be used to trigger an event or an alarm. The component is located in the vykonPro palette under the util sub folder.

Figure 13: Connection Failover properties

ConnectionFailover (Connection Failover)

<input type="checkbox"/> Enabled	enabled
<input type="checkbox"/> Restore Mode	automatic
<input type="checkbox"/> Ping Frequency	+00000h 05m 00s
<input type="checkbox"/> Ping Host	192.168.1.25
<input type="checkbox"/> Ping Timeout	5 s [0 - max]
<input type="checkbox"/> Ping Failure Limit	5 [1 - max]
<input type="checkbox"/> Number Of Ping Fails	0
<input type="checkbox"/> Ping Status	ping successfull 192.168.1.25 {ok}
<input type="checkbox"/> Ping Fail	false {ok}
<input type="checkbox"/> Last Ping Time	18-Mar-2010 11:53 AM EDT

The maximum allowable number of consecutive ping failures is configured using the 'Ping Failure Limit' property. If the 'Number Of Ping Fails' exceeds the 'Ping Failure Limit', the 'Ping Fail' property is set to true.

The 'Restore Mode' property determines whether the ping failure condition is latched or not. In 'manual' mode, the 'Ping Fail' property remains in the true (failed) state until a user executes the 'Reset Fail' action of the connectionFailover component.

In 'automatic' mode, the component continues to ping the configured host at the configured 'Ping Frequency'. When a response is received from the host, the 'Ping Fail' property is set to false (normal).

The connectionFailover component may be used to monitor the connectivity of the primary broadband connection by pinging a host on the internet. If the host cannot be pinged that might trigger logic to activate a backup dialup networking connection.

Differential Compare

The Differential Compare component is used to compare two analog values and outputs a boolean value which is determined by evaluating the analog values, logic mode and dead band.

The Differential Compare component is located in the vykonPro palette under the logic sub folder.

Types of Differential Compare component properties include the following:

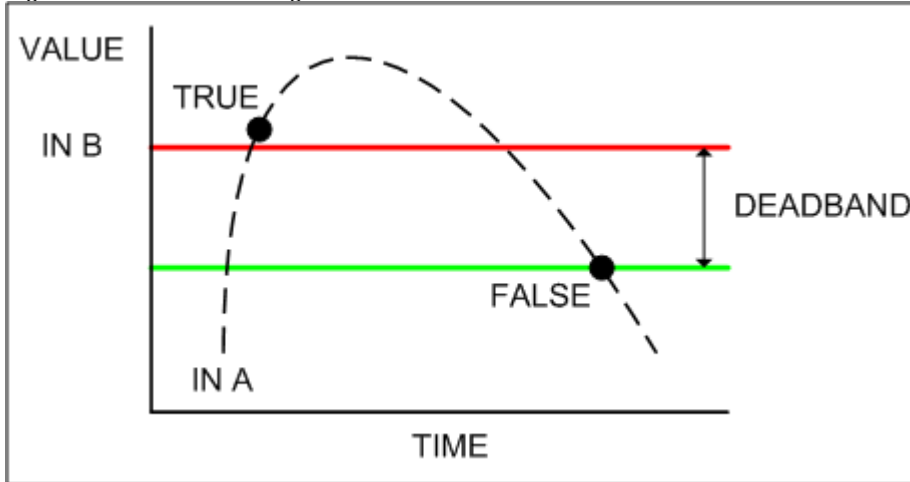
Properties

- **Facets** - Use this property to set the trueText and falseText for the Out property values. For example, you might want to set the facet trueText to display "ON" and the facet falseText to display "OFF".
- **Out** - This property has true, false, or null options available. The value is set based on evaluating the In A, In B, Logic Mode and Deadband values.
- **Null On Inactive** - If this property is set to true, when the logic evaluates a false value the out slot is set to a null status.
- **Propagate Flags** - Property controls which Flags bits are propagated from the In A or In B slots to the out slot.
- **Status** - BStatus slot which reflects the current status of the component.
- **In A** - StatusNumeric input slot which is evaluated against In B based on logic mode and deadband.
- **In B** - StatusNumeric input slot which is evaluated against In A based on logic mode and deadband.
- **Fault Cause** - String property which stores the fault cause if the component's status is in fault.
- **Deadband** - StatusNumeric slot which is the deadband value used in evaluating the logic.
- **Logic Mode** - Enum property used to select the relational operator used when evaluating the numeric inputs. Relational operator choices include 'Less Than', 'Less Than Or Equal To', 'Greater Than', 'Greater Than Or Equal To' and 'Equal To'.

Logic Mode Operation

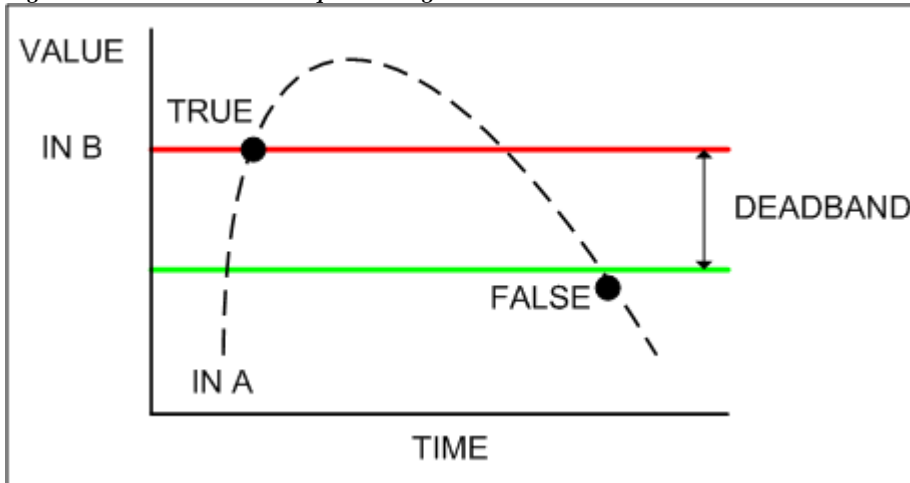
Greater Than – The out is true when $In A > In B$. As $In A$ value decreases the out slot remains true through the deadband. When $In A \leq In B - \text{deadband}$, the out slot is set to false.

Figure 14 - Greater Than Logic Mode



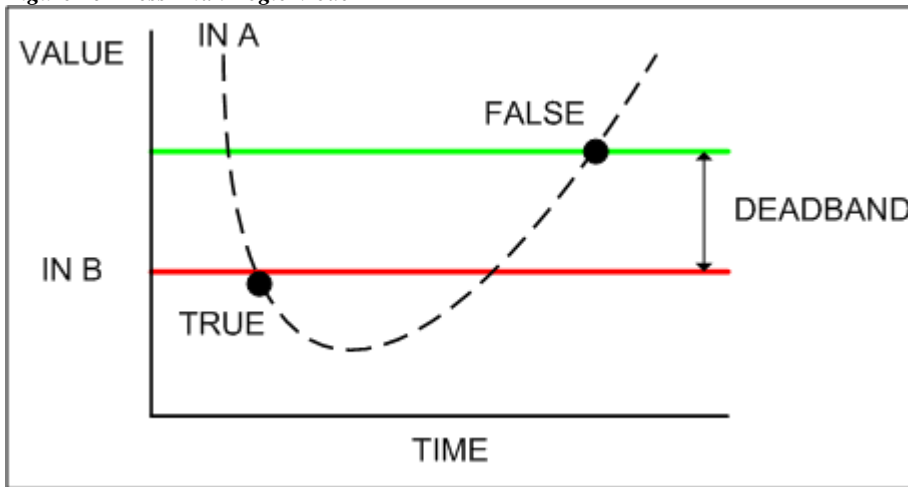
Greater Than Or Equal To - The out is true when $In A \geq In B$. As $In A$ value decreases the out slot remains true through the deadband. When $In A < In B - \text{deadband}$, the out slot is set to false.

Figure 15 - Greater Than Or Equal To Logic Mode



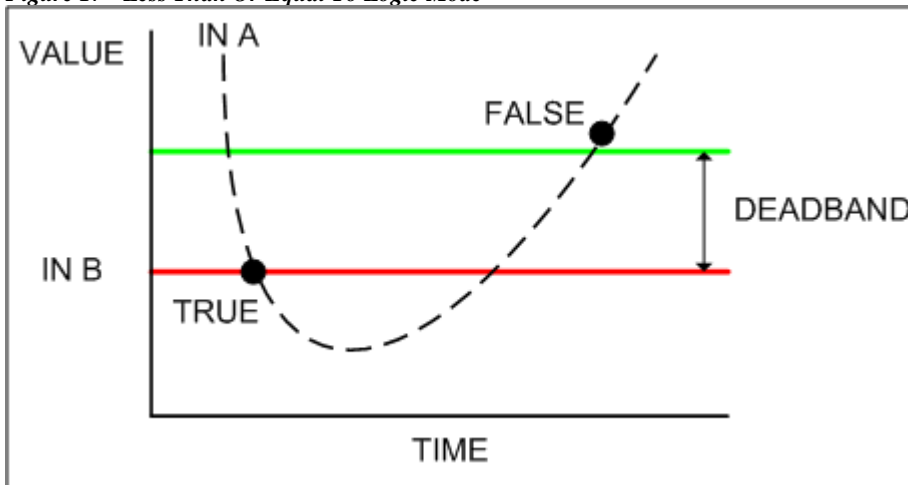
Less Than - The out is true when $In A < In B$. As $In A$ value increases the out slot remains true through the deadband. When $In A \geq In B + \text{deadband}$, the out slot is set to false.

Figure 16 - Less Than Logic Mode



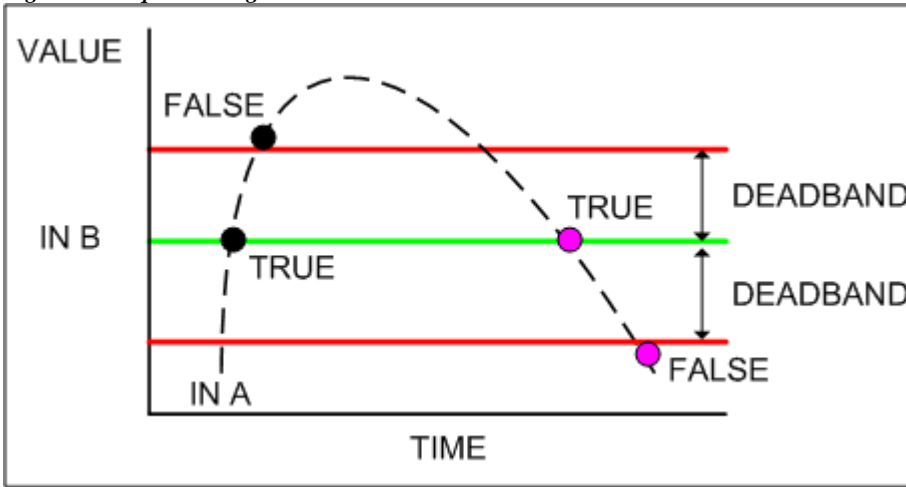
Less Than Or Equal To - The out is true when $In A \leq In B$. As $In A$ value increases the out slot remains true through the deadband. When $In A > In B + \text{deadband}$, the out slot is set to false.

Figure 17 - Less Than Or Equal To Logic Mode



Equal To - The out is true when $In A = In B$. As $In A$ value either increases or decreases the out slot remains true through the deadband. When $In A < In B - \text{deadband}$ or $In A > In B + \text{deadband}$, the out slot is set to false.

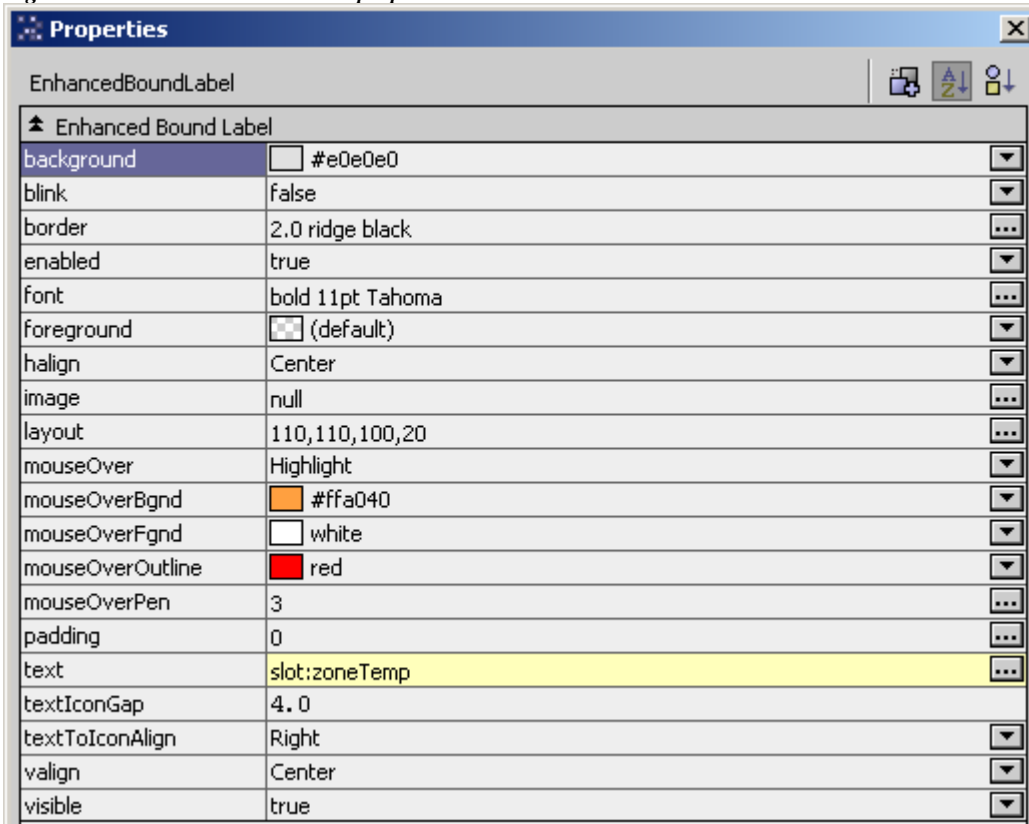
Figure 18 - Equal To Logic Mode



Enhanced Bound Label

The Enhanced Bound Label is an extension of the standard Bound Label found in the kitPx Module. The inherited properties all function identically. Only the additional properties are covered in this document. The Enhanced Bound Label is located in the vykonPro palette in the px sub folder.

Figure 19: Enhanced Bound Label properties

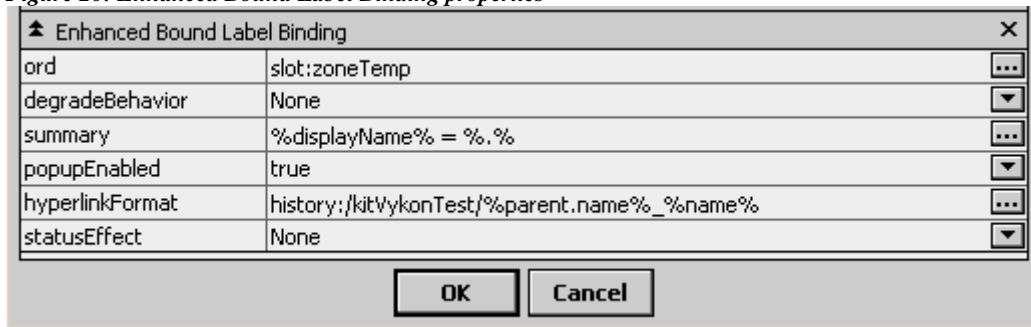


- **mouseOverBgnd** - Brush selector for configuring the background color to be displayed when the mouseOver property is configured for "highlight" and the mouse is hovered over the px widget.
- **mouseOverFgnd** - Brush selector for configuring the foreground color to be displayed when the mouseOver property is configured for "highlight" and the mouse is hovered over the px widget.
- **mouseOverOutline** - Brush selector for configuring the line color to be displayed when the mouseOver property is configured for "outline" and the mouse is hovered over the px widget.
- **mouseOverPen** - The pen configuration used for the line to be displayed when the mouseOver property is configured for "outline" and the mouse is hovered over the px widget.

Enhanced Bound Label Binding

The Enhanced Bound Label Binding is an extension of the standard Bound Label Binding found in the kitPx module. See the documentation for the Bound Label Binding for an explanation of the inherited properties. The hyperlink property from the standard Bound Label Binding is replaced with a 'hyperlinkFormat' property whose usage is detailed below.

Figure 20: Enhanced Bound Label Binding properties



- **hyperlinkFormat** - BFormat property used to resolve the hyperlink ord. The BFormat text is resolved using the binding's ord property as the base ord.

Examples

- **Default %.%** - The default format is "%.%" which will simply use the ord specified by the ord property.
- **Relativized ord hyperlink to history** - Assuming that the Niagara Network was used to bring multiple VAV devices into a supervisor station and the points for a specific device are located under a folder called "VAVx", where "x" is the zone number. There is a point called zoneTemp which is logged in the source Jace station with a history name format of "VAVx_zoneTemp". The history is imported to the supervisor station via a Niagara Network history import.

Set the ord property to "slot:". This configures the ord property as a relative ord which points to the root of the VAVx points.

Set the hyperlinkFormat to "history:/%parent.parent.name%/%name%_%zoneTemp.name%".

The "%parent.parent.name%" syntax walks up the tree to resolve the name of the Niagara station. The "%name%" syntax resolves the name of the device folder, such as VAV1. The "%zoneTemp.name%" walks down the tree to resolve the name of the point.

- **Relativized ord hyperlink to filtered audit history** - Assuming that the px view is assigned to the pointDeviceExt of a driver device. There is a writable proxy point called zoneSpt

which is typically commanded by the operator. An Enhanced Bound Label Binding is being used to display the current zoneStpt value and provide right click action for changing the set point.

Set the ord property to "slot:zoneStpt". This configures the ord property as a relative ord which points to the zoneStpt.

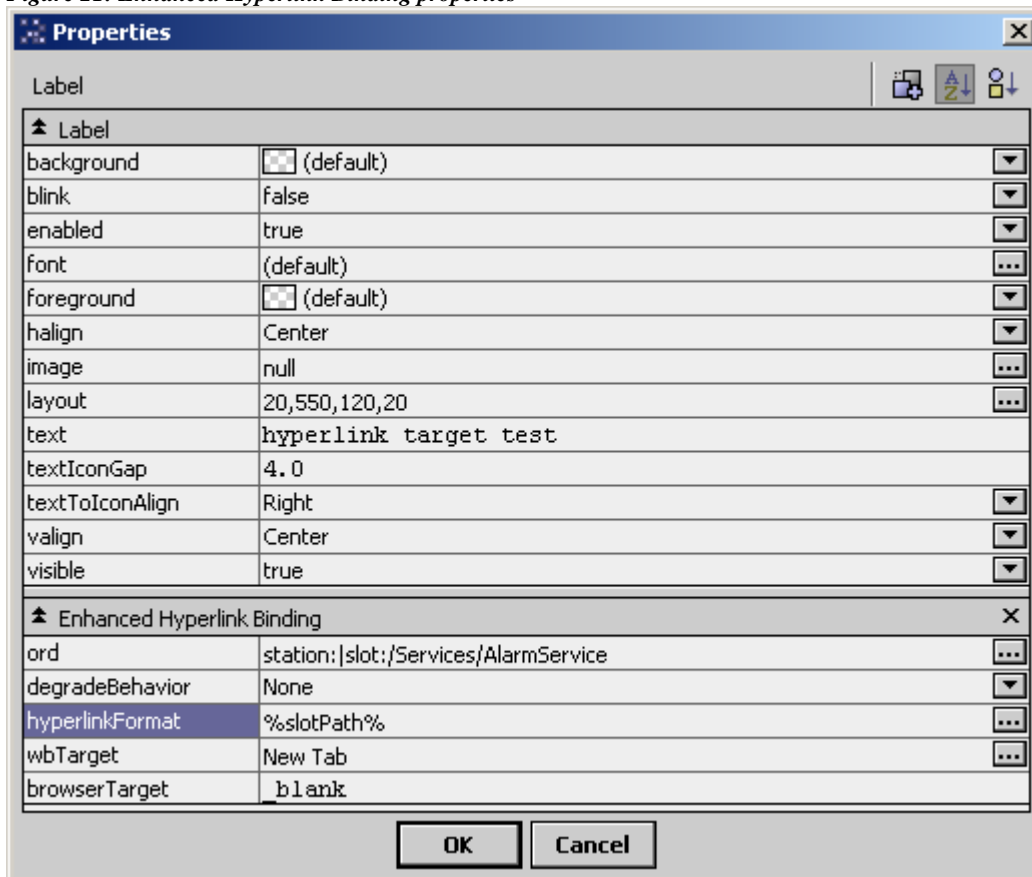
Set the hyperlinkFormat to "history:/vykonProTest/AuditHistory|bql:select * where target like '*%parent.name%/%%name%*' and operation='Invoked'".

The "%parent.name%" syntax will resolve the devices name. The "%name%" syntax will resolve the point's name. The where statement will filter the bql query return to only the audit history records for this specific point for this specific device. The return is further filtered by checking the operation type for "Invoked" which will only display actions invoked on the point.

Enhanced Hyperlink Binding

The Enhanced Hyperlink Binding creates a hyperlink which is resolved from BFormat text to provide flexibility using a single graphic for multiple components. The binding also includes properties for configuring the target behavior in both workbench and a browser environment.

Figure 21: Enhanced Hyperlink Binding properties



- **hyperlinkFormat** - BFormat property which is resolved against the base ord to construct the hyperlink. See the above section on Enhanced Bound Label Binding for a detailed explanation.

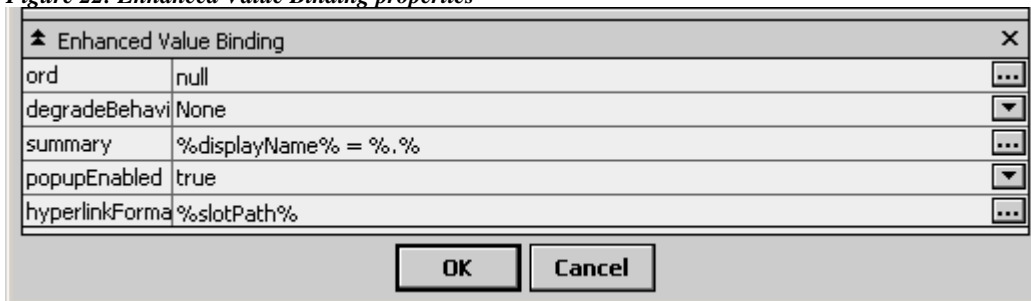
- **wbTarget** - Enumerated selector for the BHyperlinkMode.
 - Replace - Clicking the hyperlink will cause the content of the current view to be replaced with the new content of the configured hyperlink. This setting allows the user to also utilize the standard key functions such as 'ctrl' to override the replace mode by opening a new tab, or 'ctrl+shift' to override the replace mode by opening a new shell.
 - New Tab - Clicking the hyperlink will cause the new content of the configured hyperlink to be loaded in a new workbench tab. This overrides the key functions such as 'ctrl+shift' to open a new shell.
 - New Shell - Clicking the hyperlink will cause the new content of the configured hyperlink to be loaded in a new workbench shell.
- **browserTarget** - String property which is used to formulate the hyperlink and specifies the target used in the browser client.
 - `_self` - loads in the window and frame that contains the applet
 - `_parent` - loads in the applet's parent frame. If the applet's frame has no parent frame, acts the same as "`_self`".
 - `_top` - loads in the top-level frame of the applet's window. If the applet's frame is the top-level frame, acts the same as "`_self`".
 - `_blank` - loads in a new, unnamed top-level window.
 - `name` - loads in the frame or window named `name`. If a target named `name` does not already exist, a new top-level window with the specified name is created, and the document is shown there.

Enhanced Value Binding

The Enhanced Value Binding is an extension of the standard BValueBinding and implements all of the inherited properties. The only difference in functionality is the usage of the `hyperlinkFormat` property instead of the `hyperlink` property.

See the Enhanced Bound Label Binding section of this document for an explanation of how to use the `hyperlinkFormat` property.

Figure 22: Enhanced Value Binding properties



File Servlet

The file servlet is designed to provide simplified file access to resources on the station outside of the standard ord scheme which utilizes the html snoopier. The file servlet is located in the vykonPro palette in the web sub folder.

Figure 23: File Servlet properties

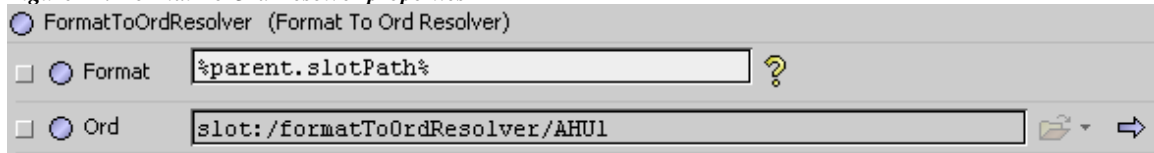


- **Servlet Name** - The syntax which is used in a url to access the servlet. The default value is "filesrv". A typical url would be <http://x.x.x.x/filesrv/myFileName.ext>
- **Base Path** - The specified location is the base path for where file resources are located. The default value is "file:^temp", which would be a folder named "temp" located under the root file system of the station.

Format To Ord Resolver

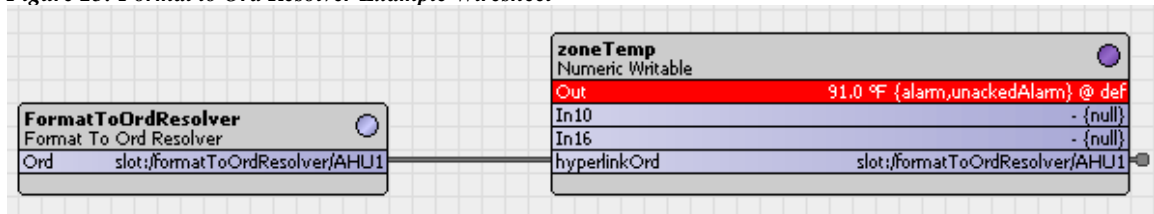
The Format To Ord Resolver is used to enter a configurable BFormat text string which is then converted to a standard BOrd value. The Format To Ord Resolver component is located in the vykonPro palette under the util sub folder.

Figure 24: Format To Ord Resolver properties



A typical usage of this component might be to programmatically set the hyperlinkOrd property of an alarm source extension.

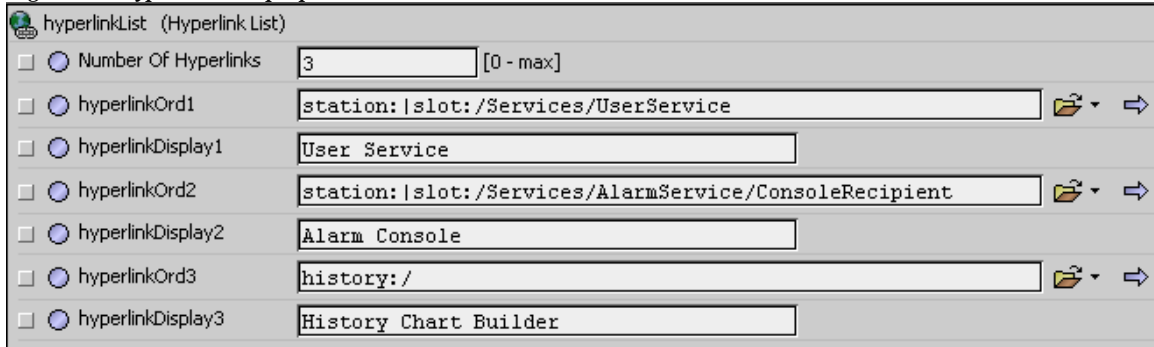
Figure 25: Format to Ord Resolver Example Wiresheet



Hyperlink List

The Hyperlink List is used to define a group of links. Links are configured using a pair of slots, one slot of type BOrd which specifies the link and a second slot of type BString which specifies the text to display in the link selector. The Hyperlink List component is located in the vykonPro palette in the util sub folder.

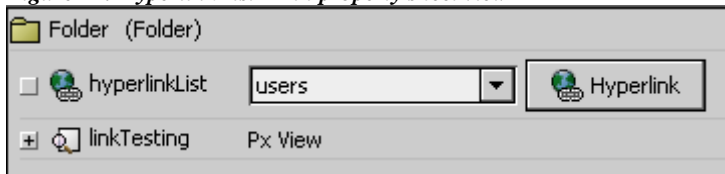
Figure 26: Hyperlink List properties



- **Number Of Hyperlinks** - Integer property used to configure the number of hyperlinks which may be configured with the component. Changing this value either adds or removes pairs of components. The pairs consist of a BOrd slot named 'hyperlinkOrdX' and a BString slot named 'hyperlinkDisplayX'. Where 'X' is a unique numeric identifier.
- **Hyperlink Ord X** - BOrd property used to specify the ord of the hyperlink item.
- **Hyperlink Display X** - BString property used to configure the name of the hyperlink in the list selector.

When viewing the property sheet of the parent component, the hyperlinkList component is displayed using a hyperlinkListFE. The field editor allows selection of defined links using a drop down selector and the link may be followed by clicking the button.

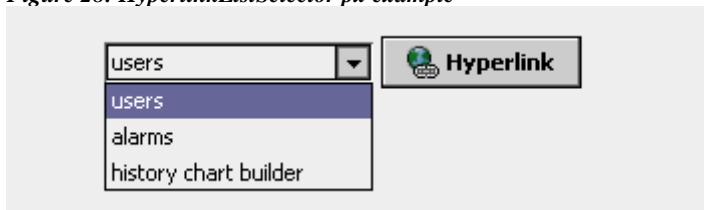
Figure 27: HyperlinkListFE in property sheet view



Hyperlink List Selector

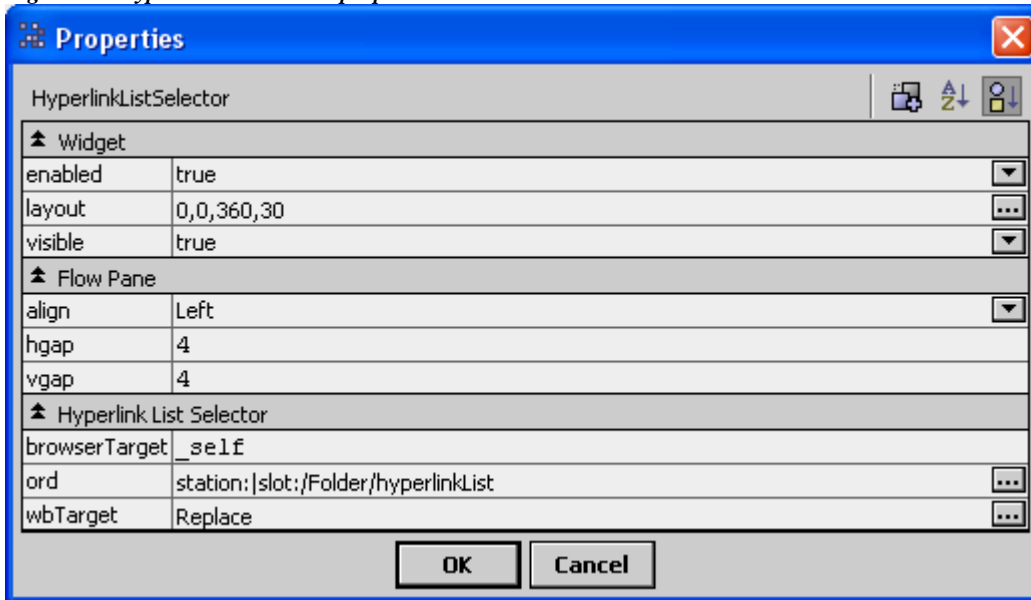
The Hyperlink List Selector is a px widget used to provide access to a hyperlinkList component. The px widget consists of a drop down list selector and a button for following the selected link.

Figure 28: HyperlinkListSelector px example



The Hyperlink List Selector component is located in the vykonPro palette under the px sub folder.

Figure 29: HyperlinkListSelector properties



- **enabled** - Boolean property which controls whether the px widget functions.
- **layout** - BLayout property which controls the size and location of the px widget in the view.
- **visible** - Boolean property which controls the visibility of the px widget in the view.
- **align** - Property controls the alignment of the drop down list and the button in the flow pane.
- **hgap** - Integer property which configures the number of horizontal pixels between components in the flow pane.
- **vgap** - Integer property which configures the number of vertical pixels between components in the flow pane.
- **browserTarget** - String property which is used to formulate the hyperlink and specifies the target used in the browser client.
 - **_self** - loads in the window and frame that contains the applet
 - **_parent** - loads in the applet's parent frame. If the applet's frame has no parent frame, acts the same as "_self".
 - **_top** - loads in the top-level frame of the applet's window. If the applet's frame is the top-level frame, acts the same as "_self".
 - **_blank** - loads in a new, unnamed top-level window.
 - **name** - loads in the frame or window named name. If a target named name does not already exist, a new top-level window with the specified name is created, and the document is shown there.
- **ord** - BOrd property which points to the hyperlinkList component in the station used to populate the drop down selector list.
- **wbTarget** - Enumerated selector for the BHyperlinkMode.
 - **Replace** - Clicking the hyperlink will cause the content of the current view to be replaced with the new content of the configured hyperlink. This setting allows the user to also utilize the standard key functions such as 'ctrl' to override the replace mode by opening a new tab, or 'ctrl+shift' to override the replace mode by opening a new shell.
 - **New Tab** - Clicking the hyperlink will cause the new content of the configured hyperlink to be loaded in a new workbench tab.
 - **New Shell** - Clicking the hyperlink will cause the new content of the configured hyperlink to be loaded in a new workbench shell.

Min Max Avg Bql

The Min Max Avg Bql component is used to perform mathematical analysis against a group of numeric components in the station. The Min Max Avg Bql component is located in the vykonPro palette under the hvac sub folder.

Figure 30: Min Max Avg Bql properties

<input type="checkbox"/> Status	{ok}
<input type="checkbox"/> Fault Cause	
<input type="checkbox"/> Enabled	enabled
<input type="checkbox"/> Point Query	station: slot:/Zones bql:select * from control:NumericPoint
<input type="checkbox"/> Number Of Points	10.0 {ok}
<input type="checkbox"/> Facets	>> [icon]
<input type="checkbox"/> Avg	73.0 {ok}
<input type="checkbox"/> Sum	730.0 {ok}
<input type="checkbox"/> Min	73.0 {ok}
<input type="checkbox"/> Min Zone Identifier	VAV10 {ok}
<input type="checkbox"/> Max	73.0 {ok}
<input type="checkbox"/> Max Zone Identifier	VAV10 {ok}
<input type="checkbox"/> Zone Identifier Format	%parent.displayName% ?
<input type="checkbox"/> Update Values Interval	00000h 00m 05s [1sec - +inf]
<input type="checkbox"/> Refresh Query Interval	00012h 00m 00s [1min - +inf]

The group of numeric components is determined by entering a BQL query in the 'Point Query' field. The results of the Bql query are stored as an array of components. The array of components is updated automatically at the frequency specified by the 'Refresh Query Interval' property.

There is also an action available on the component named 'Initialize List' which forces the query to be performed and the array to be updated. This may be useful if performing station engineering work such as adding components.

In either case the number of components which are found by the Bql query is stored in the 'Number Of Points' property.

The component evaluates the state of each component in the array at a frequency determined by the 'Update Values Interval'.

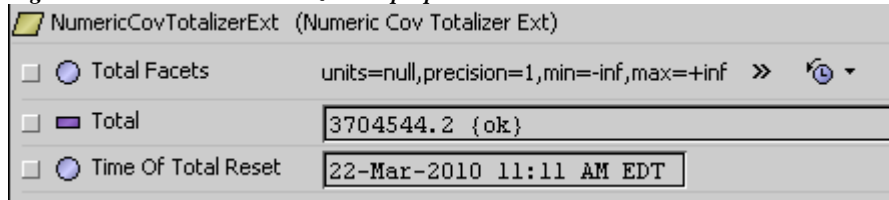
- **Avg** – BStatusNumeric slot which stores the average value of all components found by processing the query.
- **Sum** – BStatusNumeric slot which stores the sum of all values of all components found by processing the query.
- **Min** – BStatusNumeric slot which stores the minimum value of all components found by processing the query.
- **Min Zone Identifier** – String property which indicates the specific component whose value is displayed in the Min property. The string is resolved by applying the 'Zone Identifier Format' value against the component.
- **Max** – BStatusNumeric slot which stores the maximum value of all components found by processing the query.

- **Max Zone Identifier** – String property which indicates the specific component whose value is displayed in the Max property. The string is resolved by applying the ‘Zone Identifier Format’ value against the component.

Numeric Cov Totalizer Ext

The Numeric Cov Totalizer Ext is a control point extension used for calculating a total value for a numeric point. The Numeric Cov Totalizer Ext is located in the vykonPro palette under the point sub folder.

Figure 31: Numeric Cov Totalizer Ext properties



The total value is updated based on a change of value (COV) of the connected numeric input.

For example, suppose that a proxy point for a Modbus electric meter which represents interval consumption is connected to the input of the parent point. Whenever the proxy point value updates with the new consumption value for the interval, the total property of the numericCovTotalzierExt updates with the new total.

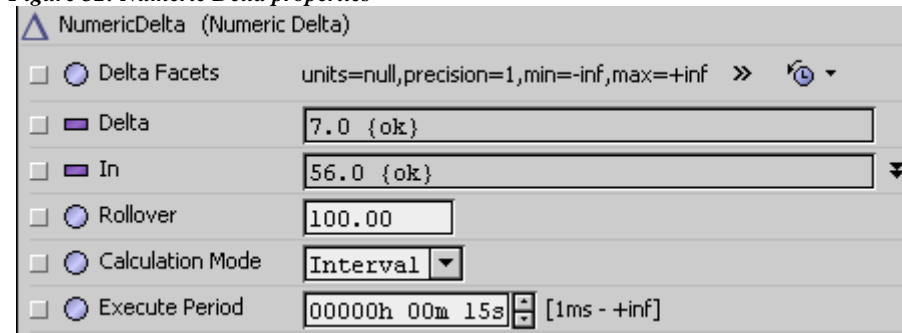
The ‘Total’ slot persistently stores the accumulated total value.

Note that this is different than the BNumericTotalizerExt found in the Control palette. The BNumericTotalizerExt is designed to convert rate to consumption.

Numeric Delta

The Numeric Delta component calculates the change in value of the connected input at a frequency based on the calculationMode property. The purpose of the component is to convert a totalized (ever increasing) value to an interval total. The Numeric Delta component is located in the vykonPro palette under the energy sub folder.

Figure 32: Numeric Delta properties



- **Rollover** – Defines the maximum value that the connected input may reach before rolling over to a zero value. For example if the source represents the total value as a 16 bit integer (unsigned), then the maximum rollover value would be 65,535.
- **Calculation Mode** – Configures when the component performs the calculation and updates the ‘Delta’ property value.

- **Interval** – Calculation is performed at the frequency specified by the ‘Execute Period’ property.
- **Cov** – Calculation is performed any time the ‘In’ property value changes.
- **Manual** – Calculation is performed when the components ‘Manual Calculate’ action is executed.
- **Execute Period** – Configures the interval at which the calculation is performed when the ‘Calculation Mode’ is set to ‘Interval’.

Numeric History Analysis

The Numeric History Analysis component calculates the total, minimum and maximum values for specific intervals of logged values from a Niagara history. The component is located in the vykonPro palette in the history sub folder.

Figure 33: Numeric History Analysis properties

The screenshot shows the configuration window for the 'arlingtonKw' component. The properties are as follows:

Status	{ok}
Fault Cause	
History Name	%componentSpace.rootComponent.stationName
History Ord	history:/vykonProTest/arlingtonKw
Facets	units=kW,precision=1,min=-inf,max=+inf
Delta Logged	true
Execute Period	+00001h 00m 00s
Current Day Total	54193.3 kW {ok}
Current Day Avg	564.5 {ok}
Current Day Min	263.3 kW {ok}
Current Day Max	861.1 kW {ok}

- **History Name** – BFormat property used to specify the history id to be used in the total calculations. This can be a combination of static text and BFormat calls. The intent is to allow flexibility in specifying the target history without having to manually code the property for duplicated devices.

For example assuming the numericHistoryAnalysis object was the child of a numeric point named 'mainElectricConsumption' which is configured to be logged as a history with a history id of 'vykonProTest/mainElectricConsumption'. The 'History Name' property could be configured as 'vykonProTest/%parent.name%'.

Note that the default component uses a special BFormat syntax to resolve the station name without having to hard code it.

%componentSpace.rootComponent.stationName%

- **History Ord** – Read-only BOrd property that displays the ord which is resolved from the history name property.
- **Delta Logged** – Boolean property used to configure whether the historical data is stored in an interval or totalized format. If the historical data is stored in an interval method, then set delta logged to true.
- **Execute Period** – BRelTime property which configures the frequency at which the component performs the mathematical calculations against the history.

- **Xxxx Total** – BStatusNumeric slot which stores the total of the history records found for the specified period.
- **Xxxx Avg** – BstatusNumeric slot which stores the average of the history records found for the specified period.
- **Xxxx Min** – BStatusNumeric slot which stores the minimum value of the history records found for the specified period.
- **Xxxx Max** – BStatusNumeric slot which stores the maximum value of the history records found for the specified period.

Note that there are groupings of slots to represent the total, average, minimum and maximum values for various periods. The component includes slots for current day, yesterday, current week, last week, current month, last month, current year and last year.

Opt Start Stop

The OptStartStop component allows a system to use Start Time Optimization and Stop Time Optimization to save energy. This component uses a space temperature input and area characteristics to calculate an optimal amount of lead-time before a scheduled event. It can analyze area temperature changes and adjust the optimization parameters based on the actual temperature change rates after an optimized start or stop.

The OptStartStop component is available in the vykonPro energy folder.

The two basic optimization types are described, as follows:

- **Start time optimization** - This type of optimization reduces energy consumption by turning on equipment at the latest possible time that still allows for providing a comfortable temperature by occupancy time.
- **Stop time optimization** - This type of optimization turns equipment off at the earliest possible time that allows the building to drift and stay within a temperature comfort range until the end of occupancy time.

The OptStartStop calculation is performed at 15 seconds after the beginning of every minute, when the appropriate Start Enable or Stop Enable properties are set to true, a valid schedule event is linked to the component, and the next scheduled event value is not already set

For example, if a value is scheduled to be set to "true" in 1 hour but is already set to "true", no calculation is performed, even if the Start Enable or Stop Enable properties are set to "true".

The product of this calculation is the "Calculated Start Time" and the "Calculated Stop Time". The "Calculated Start Time" defines an early start command to achieve a specified temperature range by occupancy time. The "Calculated Stop Time" defines an early stop command without sacrificing the temperature range by the scheduled occupancy change time. After an optimized event is invoked, the actual area response (temperature change rate) is analyzed and weighted adjustments are made to the calculation parameters based on the detected values so that subsequent calculations might be more accurate.

Start time and stop time operations are described below:

- **Calculated Start Time** - Only one optimized start sequence is performed per day. The following factors affect the Calculated Start Time calculation.
 - **Temperature differential** - If the space temperature is outside the range defined by the lower and upper comfort limits, the difference between the space temperature and the closer limit represents the number of degrees the mechanical equipment must make up during the prestart optimized period.

- **Run-time minutes** - The run-time heating or cooling factors (depending on the direction the space temperature must move) are multiplied by the temperature differential to determine the number of run-time minutes required to achieve the comfort limit at occupancy time, as defined by the schedule's start time.
- **Optimum start time** - When the system's time is later than the schedule's time offset by the calculated lead-time, the optimum start outputs are enabled.

The calculated lead-time is limited by either the "Earliest Start Time" or the "Max Earliest Start Hours" depending on the current setting of the "Start Limit Mode" property. If the "Calculated Start Time" is prior to the configured limit, the optimum start occurs at configured limit. An optimum start is performed only for the first scheduled start for the day.

- **Calculated Stop Time** - You can perform multiple stop operations but no optimized stop can occur before the time specified by the Earliest Stop Time property.
 - **Temperature differential** - If the space temperature is inside the range defined by the lower and upper comfort limits and the schedule's status is active, the difference between the space temperature and one of the limits (depending on the mode) represents the number of degrees the temperature can drift between the time the mechanical equipment is stopped and the schedule's inactive event time.
 - **Drift time** - The drift (lead-time) calculation is similar to the one for Start Time but using the drift-time heating and cooling factors.
 - **Optimum stop time** - Optimum stop time is invoked for each of the schedule's inactive events and is based on the drift time and Next Event Time value.

The OptStartStop component includes the following properties:

- **Status** - BStatus slot which reflects the current status of the component.
- **Fault Cause** - String property which stores the fault cause if the component's status is in fault.
- **Stop Calc Mode** - This Boolean property allows you to select either "Heat and Cool", "Cool Only" or the "Heat Only". The selected option applies only to optimized stop calculations which means that optimized stop calculations are performed only for the selected mode.
- **Parameter Reset Time** - This property displays the time when any of the four runtime or drift time properties change to the User Defined values. **Start Enable** - This property allows you to manually or automatically enable or disable the optimized start function.
- **Stop Enable** - This property allows you to manually or automatically enable or disable the optimized stop function.
- **Schedule Status** - This Boolean property monitors and displays the status of the schedule that is linked to it.
- **Next Event Time** - This property is linked to a schedule for the time of the next scheduled event.
- **Next Event Value** - This property is linked to a schedule and reflects the value of the action for next scheduled event.
- **Outside Temp** - This property is linked to outside temperature and displays the value for information only.
- **Space Temp** - This property is linked to a space temperature output and displays the temperature of the area affected by equipment associated with the OSS component.
- **Start Time Command** - This Boolean property is an output that you link to a control for invoking an equipment start command. For example, it can be linked to a prioritized input of a Boolean writable - or directly to the equipment Start control.
- **Stop Time Command** - This Boolean property is an output that you link to a control for invoking an equipment stop command. For example, it can be linked to a prioritized input of a Boolean writable - or directly to the equipment Stop control.
- **Message** - This field provides information that indicates the results of the latest start or stop command, the status of an optimized start analysis, or other possible messages. For example, the

following message is displayed to indicate that an optimized stop has occurred: "Optimized stop for 14-Jun-07 5:18 PM EDT schedule time. Space temp is 75.0."

- **Upper Comfort Limit** - This property value is the Cooling mode target temperature.
- **Lower Comfort Limit** - This property value is the Heating mode target temperature.
- **Comfort Deadband** - This property value is a dead band applied to the upper and lower comfort limits when evaluating the effectiveness of the optimized start event.
- **Dynamic Parameter Adjust** - This controls whether or not calculation parameters are programmatically adjusted after an execution. After the OSS component completes a start or stop control, if this property value is set to true, the component evaluates the actual recovery rate (degrees/hour) and automatically adjusts the Runtime and Drift time properties values so that they are influenced by actual drift time and run time.
- **Old Parameter Multiplier** - This property is used to weight the dynamic parameter adjustment calculation. The value that you specify in this field affects how much weighting you assign to the previous runtime property value when it is used in the dynamic parameter adjustment calculation. A larger value increases the amount of weighting given to the previous runtime and a smaller value decreases the weighting.
- **Earliest Stop Time** - This property allows you to specify a time, before which, no stop command may be issued. If this value is set earlier than the Calculated Stop Time, the Calculated Stop Time is adjusted to equal this time.
- **Start Limit Mode** - This property defines whether the "Earliest Start Time" property or the "Max Earliest Start Hours" is used in the optimal start time calculation.
- **Earliest Start Time** - The property allows you to specify a time, before which, no start command may be issued. If the Calculated Start Time is prior to this time, the Calculated Start Time is adjusted to equal this time. This prevents an optimal start event from occurring prior to this configured time.
- **Max Earliest Start Hours** - The property allows you to specify the maximum number of hours prior to the scheduled occupancy which a start command may be issued. If the difference between the actual Next Event Time and the Calculated Start Time exceeds the Max Earliest Start Hours, then the Calculated Start Time is adjusted to match. If the Calculated Start Time results in a time prior to midnight, then the Calculated Start Time is set to midnight.
- **Drift time Per degree Cooling** - This property displays the actual value that is used for calculating an optimized stop time when the equipment is in cooling mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- **Drift time Per degree Heating** - This property displays the actual value that is used for calculating an optimized stop time when the equipment is in heating mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- **Runtime Per degree Cooling** - This property displays the actual value that is used for calculating an optimized start time when the equipment is in cooling mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- **Runtime Per degree Heating** - This property displays the actual value that is used for calculating an optimized start time when the equipment is in heating mode. This value is adjusted automatically if the Dynamic Parameter Adjust value is set to true.
- **Last Start Time** - This is a record of the last Start Time that was used for calculating an optimized start time. Since only one optimized start per day is allowed, this value does not display Start Times (restarts) that are subsequent to the initial Start Time for a day.
- **Last Stop Time** - This is a record of the last Stop Time that was used for calculating an optimized stop time. Since multiple Optimized Stops are allowed in a day, this value changes to reflect the latest Optimized Stop time.
- **Outside Temp At Beginning** - This is a record of what the outside air temperature was at the time of the last start or stop command. This is the temperature that was used in calculations for dynamic parameter adjustment.
- **Space Temp At Beginning** - This is a record of what the space temperature was at the time of the last start or stop command. This is the temperature that was used in calculations for dynamic parameter adjustment.
- **Calculated Start Time** - This field shows the calculated time for the next start command.

- **Calculated Stop Time** - This field shows the calculated time for the next stop command.
- **Comfort Alarm** - StatusBoolean property which indicates the effectiveness of the last optimized start event. If the space temperature at the scheduled time of occupancy was outside of the comfort band, the value is set to "alarm". The comfort band is defined as lowerComfortLimit - comfortDeadband and upperComfortLimit + dead band.
- **Program Mode** - As part of the logic that the OSS component uses, there are five "program mode" states. These states serve primarily in logic control, however, they may be informative to the system engineer, as well. The Program Mode value displays the current heating or cooling state for optimized start or stop. The following list describes the possible display values and meanings.
 - **0 ("No" Calculation)** - This value indicates that no calculation is being made.
 - **1 ("Start" Calculation)** - This value indicates that the optimized start calculation process is ongoing but that an optimized start or stop is not yet in progress.
 - **2 ("Start" in Process)** - This value indicates that an optimized start has been initiated.
 - **3 ("Stop" Calculation)** - This value indicates that an optimized stop calculation process is ongoing but that an optimized start or stop is not yet in progress.
 - **4 ("Stop" in Process)** - This value indicates that an optimized stop has been initiated.
- **Mode** – String representation of the Program Mode property.

Peak Tracker

The Peak Tracker component stores the peak value and timestamp of the peak for the connected input in a specific slot for the current month. There are individual slots for each month of the year for both the peak value and the timestamp of the peak. The Peak Tracker component is located in the vykonPro palette under the history sub folder.

The component has a single action 'Reset Peaks' which when executed clears all peak values and timestamps.

There is a workbench view assigned to the Peak Tracker component which displays the monthly peak values and timestamps in a standard table format. The table includes the standard built in helper icon in the top right corner which allows exporting the table.

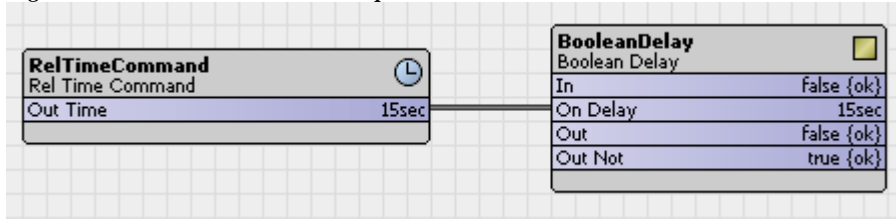
Figure 34: Peak Tracker View

PeakTracker - Monthly Peak Values			
Month	Timestamp	Value	⌵
January	null	0.0 kW	
February	null	0.0 kW	
March	null	0.0 kW	
April	null	0.0 kW	
May	null	0.0 kW	
June	null	0.0 kW	
July	null	0.0 kW	
August	null	0.0 kW	
September	21-Sep-09 4:20 PM EDT	98.0 kW	
October	null	0.0 kW	
November	null	0.0 kW	
December	null	0.0 kW	

Rel Time Command

The Rel Time Command component has a single action 'Change Time' which sets the 'Out Time' property. The component can easily be linked to other components with BRelTime property slots to facilitate using a [px](#) action instead of a set point field editor. The Rel Time Component can also be proxied through the Niagara network to allow issuing the action across the network connection.

Figure 35: Rel Time Command example



Rotational Load Sequencer

Rotational Load Sequencer provides lead-lag control of from 2 to 10 Boolean Writables based upon a fixed sequence. The number of active loads is determined by the Schedule and Required Loads inputs.

Rotational Load Sequencer is available in the HVAC folder of the vykonPro palette

Setup of the object involves the following properties:

- **Status** – BStatus property that indicates the status of the component.
- **Fault Cause** – String property that indicates the current fault cause if any.
- **Facets** - Facets applied to the Out (A-J) slots.
- **Schedule** - A StatusBoolean input that controls whether any control device should be on. If this input is true, at least one of the outputs will be active.
- **Load Failed** – A StatusBoolean output which indicates if any controlled loads have failed.
- **Number Outputs** - Specifies the number of devices (outputs) that are controlled. Changing this property causes various slots to be displayed or hidden as applicable.
- **Required Loads** - Specifies the number of devices (outputs) that should be active provided the Schedule input is true. This input can be linked to logic which evaluates the need for multiple loads to be energized such as in parallel pumping applications.
- **Rotate Mode** – Enum property which determines the mode of operation. When set to interval, rotation occurs after the station elapsed time exceeds the rotationInterval value regardless of load runtime. When set to rotate, rotation occurs when the lead load's accumulated runtime (since being designated as the lead load) exceeds the rotateInterval value. When set to manual, rotation occurs only when manually commanded via the component's action.
- **Rotate Interval** - Specifies the maximum amount of time that must elapse based on the rotate mode before the lead load is shifted automatically.
- **Feedback Delay Time** - Specifies the delay time used to evaluate the feedback links (if any). If an active load does not indicate a running status via its feedback, then an additional load is started to maintain the number of running loads equal to the required number of loads.
- **Feedback A-J** - A StatusBoolean input, to provide positive feedback that a controlled device actually started. If the feedback value does not show true within the Feedback Delay time, the output will show alarm, and the Runtime Balancer switches to the next output with the least accumulated runtime. Setting this value to true (and not linking) disables this alarm feature.
- **Out A-J** - StatusBoolean outputs, each typically linked to a Boolean Writable control point with a DiscreteTotalizerExt. Outputs are typically used to control loads of some type.
- **Runtime A-J** - These are RelTime inputs that are used for runtime feedback for the corresponding Out A - J. These inputs will typically be linked to the ElapsedActiveTime property of the DiscreteTotalizerExt that is measuring the runtime of the corresponding Out A - J.

Individual loads may be removed from the rotation logic by executing the 'Disable Output' action on the Rotational Load Sequencer component. The user is prompted with a dialog box to select which output is to be disabled. There is an additional action 'Enable Output' which is used to restore a previously disabled load to the rotation logic.

Individual loads may be disabled or enabled through control logic by utilizing a Boolean To Sequencer Cmd component, which is also located in the vykonPro palette in the hvac sub folder. See the section in this document regarding the Boolean To Sequencer Cmd.

Runtime Balancer

Runtime Balancer provides lead-lag control of from 2 to 10 Boolean Writables based upon their accumulated runtimes (elapsed active time). This object balances the active runtime of each of the devices. The number of active loads is determined by the Schedule and Required Loads inputs.

Runtime Balancer is available in the HVAC folder of the vykonPro palette

Setup of the object involves the following properties:

- **Facets** - Facets applied to the Out (A-J) slots.
- **Schedule** - A StatusBoolean input that controls whether any control device should be on. If this input is true, at least one of the outputs will be active based on runtime.
- **Load Failed** - A StatusBoolean output which indicates if any controlled loads have failed.
- **Number Outputs** - Specifies the number of devices (outputs) that are controlled. Changing this property causes various slots to be displayed or hidden as applicable.
- **Required Loads** - Specifies the number of devices (outputs) that should be active provided the Schedule input is true. This input can be linked to logic which evaluates the need for multiple loads to be energized such as in parallel pumping applications.
- **Rotate Mode** - Boolean property which determines the mode of operation. When set to interval the component evaluates load runtimes and potentially switches loads at the frequency defined by the rotate interval. When set to manual load runtime evaluations and potential switching only occurs when manually triggered via the 'Manual Rotate' action.
- **Rotate Interval** - Specifies the maximum amount of time that a given output will be continuously active before triggering the component to evaluate the runtimes and potentially switch running loads.
- **Feedback Delay Time** - Specifies the delay time used to evaluate the feedback links (if any). If an active load does not indicate a running status via its feedback, then an additional load is started to maintain the number of running loads equal to the required number of loads.
- **Feedback A-J** - A StatusBoolean input, to provide positive feedback that a controlled device actually started. If the feedback value does not show true within the Feedback Delay time, the output will show alarm, and the Runtime Balancer switches to the next output with the least accumulated runtime. Setting this value to true (and not linking) disables this alarm feature.
- **Out A-J** - StatusBoolean outputs, each typically linked to a Boolean Writable control point with a DiscreteTotalizerExt. Outputs are typically used to control loads of some type.
- **Runtime A-J** - These are RelTime inputs that are used for runtime feedback for the corresponding Out A - J. These inputs will typically be linked to the ElapsedActiveTime property of the DiscreteTotalizerExt that is measuring the runtime of the corresponding Out A - J.

Individual loads may be removed from the rotation logic by executing the 'Disable Output' action on the Runtime Balancer component. The user is prompted with a dialog box to select which output is to be disabled. There is an additional action 'Enable Output' which is used to restore a previously disabled load to the rotation logic.

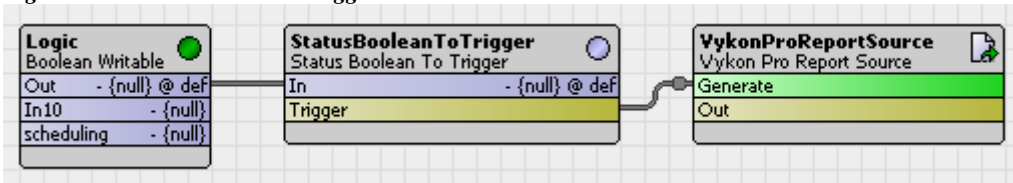
Individual loads may be disabled or enabled through control logic by utilizing a Boolean To Balancer Cmd component, which is also located in the vykonPro palette in the hvac sub folder. See the section in this document regarding the Boolean To Balancer Cmd.

Status Boolean To Trigger

The StatusBooleanToTrigger component fires a trigger upon transition of the in property based on the fireOn property value. If fireOn equals true, when the in property value transitions from false to true the trigger is fired. If fireOn equals false, when the in property value transitions from true to false the trigger is fired. The StatusBooleanToTrigger component is located in the vykonPro palette under the conversion sub folder.

The purpose of using this component is to isolate the boolean logic transition which typically yields two transitions from the action being linked to. When linking with a boolean component to an action it typically results in the action being fired twice, both on the transition to true and back to false.

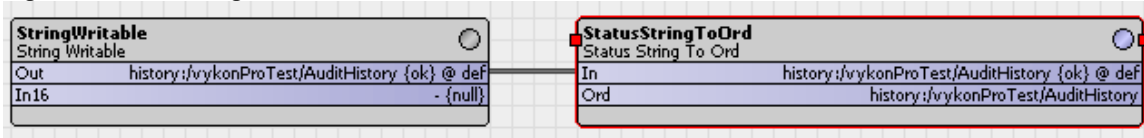
Figure 36: Status Boolean To Trigger wiresheet



Status String To Ord

The Status String To Ord converts In slots string value to a BFormat and stores it in the Ord slot. The Status String To Ord component is located in the vykonPro palette under the conversion sub folder.

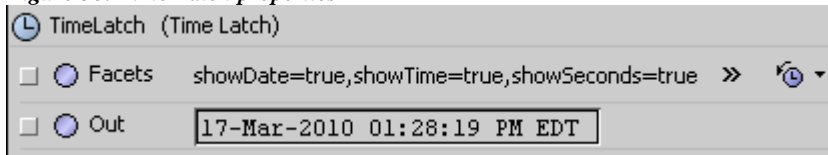
Figure 37: Status String To Ord wiresheet



Time Latch

The Time Latch component latches the current station time when the 'updateTime' action is called, and stores the time in the 'Out' property. The Time Latch component is located in the vykonPro palette under the timer sub folder.

Figure 38: Time Latch properties



The 'Facets' property is used to configure the display format of the time displayed in the 'Out' slot.

Tool Tip Binding

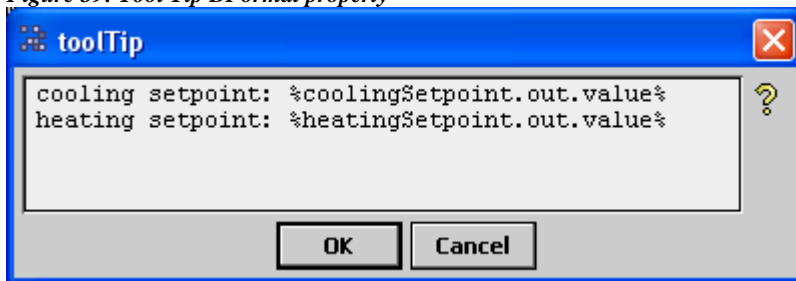
The Tool Tip Binding is a binding which can be added to a px widget, which invokes a popup tool tip when the mouse is hovered over the widget.

Figure 39: Tool Tip Binding properties

Tool Tip Binding	
ord	\$(zone) ...
degradeBehavi	None ▼
toolTip	cooling setpoint: %coolingSetpoint.out.value%heating setpoint: %heatingSetpoint.out.value% ...
leaseDepth	1
xOffset	10
yOffset	10

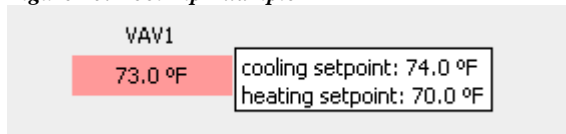
- **ord** – BOrd property which specifies the base which the tool tip text is resolved against.
- **toolTip** – BFormat property used to specify the text which is displayed in the popup tool tip window. This can be a combination of station and BFormat text syntax.
- **leaseDepth** – Integer property used to configure the depth to which components are lease to populate updated values in the tool tip window. The default value is 1, which will lease all of the direct child components of the specified ord.
- **xOffset** – int property which controls the horizontal distance from the mouse event where the popup tool tip window is located.
- **yOffset** – int property which controls the vertical distance from the mouse event where the popup tool tip window is located.

Figure 39: Tool Tip BFormat property



The 'toolTip' property is a multi-line BFormat property. The enter key is used to generate multiple lines in the property editor, which will display in the popup tool tip window using the same line breaks.

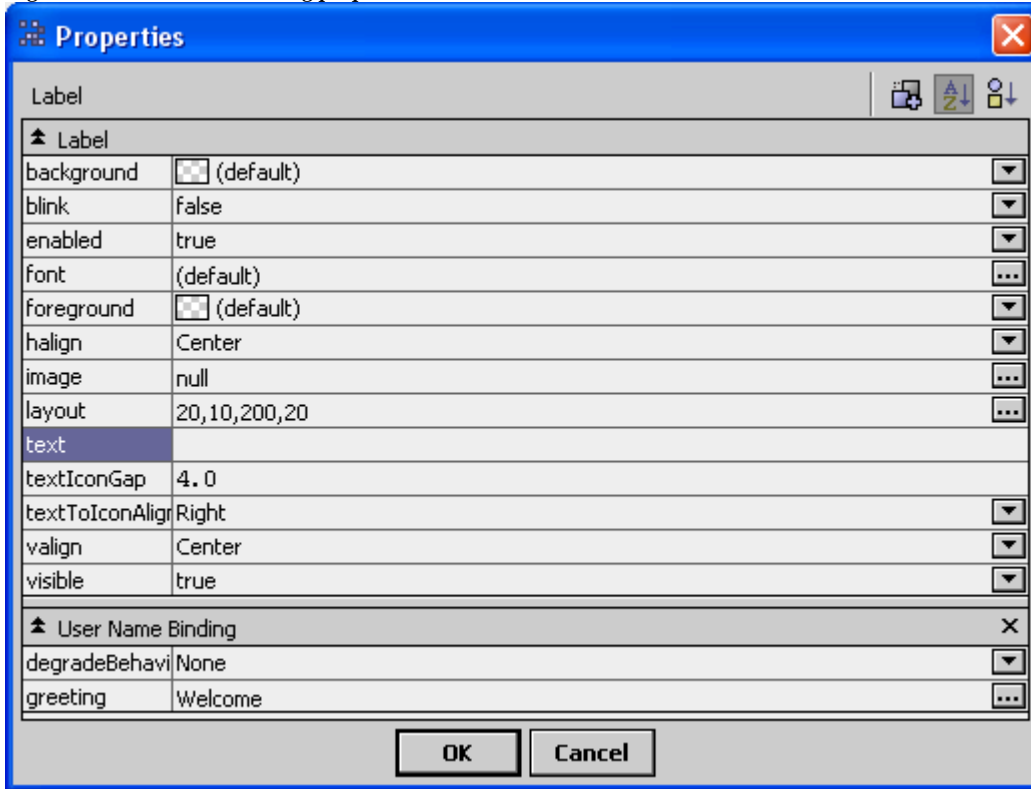
Figure 40: Tool Tip Example



User Name Binding

The User Name Binding is a binding which can be added to a px widget, which displays the username from the context facets for the currently logged in user.

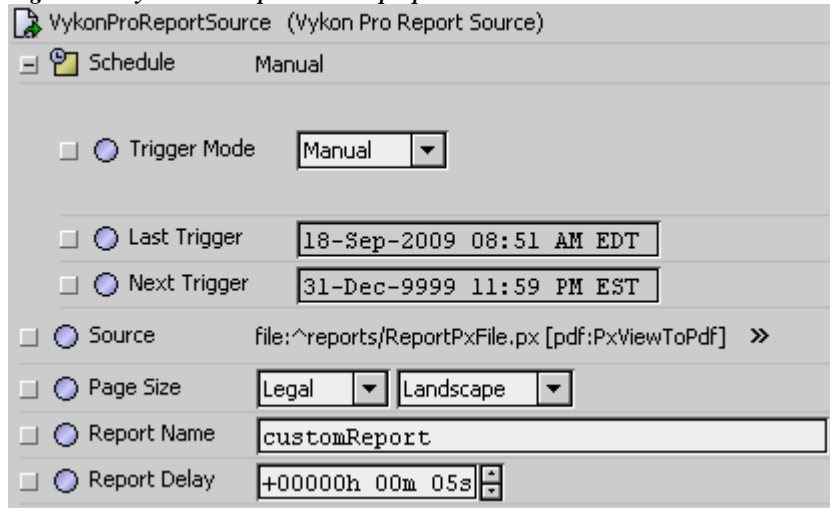
Figure 41: User Name Binding properties



Vykon Pro Report Source

The Vykon Pro Report Source is used to schedule the export of a specified px view to a PDF file. The 'out' slot must be linked to a report recipient (file or email) which is used to actually route the generated PDF. The Vykon Pro Report Source component is located in the vykonPro palette under the report sub folder.

Figure 42: Vykon Pro Report Source properties

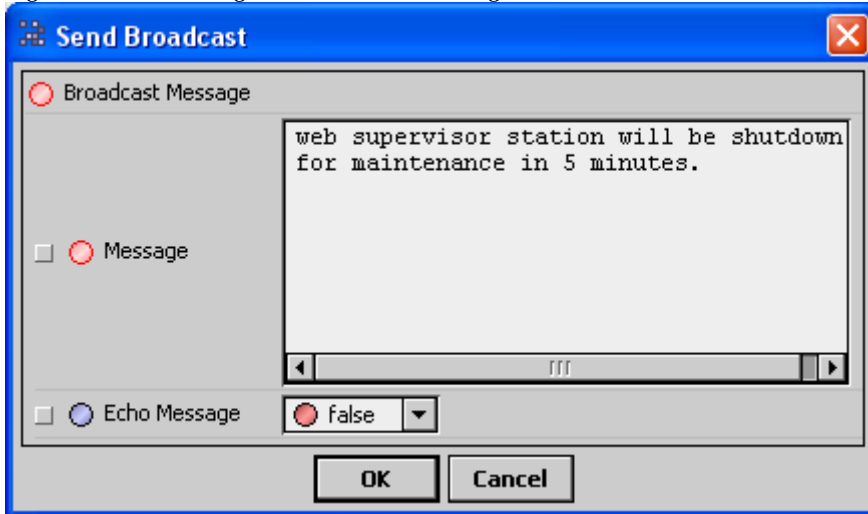


- **Schedule** - This property contains standard scheduling options that allow you to choose different options for timing the delivery of your report. It also displays a "Last Trigger" and "Next Scheduled Trigger" field.
- **Source** - This property specifies the report display that you want to email. An arrow button located at the right end of the property display opens the Export Source Wizard dialog box. This dialog box provides two steps that assist you in assigning source location and selecting a PDF exporter for your report.
- **Page Size** - This property specifies the page size and orientation (landscape or portrait) for the PDF to be generated
- **Report Name** - String property which specifies the name of the report which is passed to the report recipient component.
- **Report Delay** - BRelTime property used to configure the amount of delay time between instantiating the px view and generating the PDF file. The default time is 5 seconds, but may need to be adjusted to allow time for components to be leased.

WB Messenger

The Wb Messenger component makes use of the workbench notification API to send a message to either a specific or all connected workbench clients. The message is sent by either invoking the "Send Broadcast" or "Send To Single User" action, which provides a popup window for entering a text message. The Wb Messenger component is located in the vykonPro palette in the util sub folder.

Figure 43: WB Messenger Send Broadcast Dialog



Clicking the 'OK' button sends the entered message out as a broadcast message to all users who are connected with either workbench or a browser using a workbench profile.

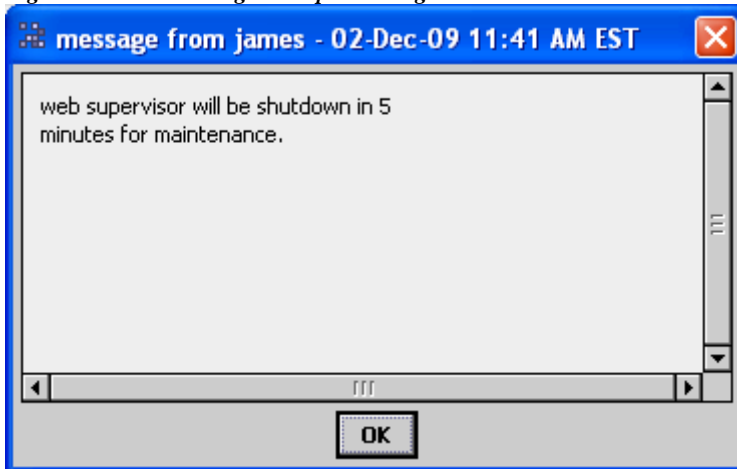
Figure 44: WB Messenger Send To Single User Dialog



Clicking the 'OK' button sends the entered message to the selected user. The user drop down list is populated based on the currently connected workbench sessions.

Setting the "Echo Message" property to true causes the popup message to be delivered to the sender.

Figure 45: WB Messenger Sample Message



Weekly Trigger

The weekly trigger component is used to configure a trigger which is fired on a specified day of the week at a specified time of day and at a specified weekly interval. The component is located in the vykonPro palette in the timer sub folder.

Properties

- **Trigger Time** – BTime property used to configure the time of day that the trigger will fire.
- **Trigger Day Of Week** – Bweekday property used to configure the day of the week that the trigger will fire.
- **Trigger Week Interval** – Integer property used to configure the number of weeks that must elapse between trigger events. If set to a value of one, the trigger will fire every week. If set to a value of two, the trigger will fire every other week on the trigger day of week at the trigger time.
- **Last Trigger** – BAbsTime property which displays the timestamp at which the trigger was last fired.
- **Next Trigger** – BAbsTime property which displays the calculated timestamp at which the trigger will next fire the trigger.

Actions

- **Preset Next Trigger** – Executing the action prompts the user with a dialog to enter the time and date at which the component will next fire the trigger. The Next Trigger property will be updated with the entered time and date.

Document Change Log

- 20 October 2009 – Added content for Differential Compare component.
- 5 November 2009 – Enhancements to OptStartStop component. Fixed issue where optimal stop analysis in cooling mode calculated a negative drifttime.
- 2 December 2009 – Enhancements to WbMessenger and TooltipBinding.
 - WbMessenger enhanced to allow sending a message to a single user. The popup message dialog was modified and the message no longer displays to the originating user when sending a broadcast message.
 - TooltipBinding added property “leaseDepth” to allow configuring the depth of components leased prior to displaying the values in the tooltip window.
- 4 March 2010
 - Added AbsTimeToStatusString component
 - Added BRelTimeCommand component
 - Modified WbMessenger to include “Echo Message” property.
- 16 April 2010
 - Added BBlockDemand component
 - Added BRotationalLoadSequencer, BBooleanToSequencerCmd
 - Added BWeeklyTrigger
 - Corrected issue with WbMessenger that prevented message from being properly echoed.
- 21 May 2010
 - Added BActiveLevelToInt component.