AlarmWorX64 Viewer Commanding

APPLICATION NOTE

April 2017

Description: Guide to using Commands for the AlarmWorX64 Viewer.
General Requirement: Familiarity with the AlarmWorX64 Viewer and Commanding in general.

Introduction

Before the introduction of Commanding it used to require scripting to interact with controls like the AlarmWorX64 Viewer. Commanding provides a simpler, more intuitive, and easier to maintain way to send these commands to controls.

This document will go over the commands that can be sent to or from an AlarmWorX64 Viewer. More information about Commanding in general or with respect to other controls can be found in other application notes or in the help files.

If you are unfamiliar with any aspects of Commanding, please refer to the application note entitled, *GENESIS64 – Commanding Introduction* to get started.

> **NOTE:** All properties named **Index** are zero-based, meaning the first object always has an Index of 0.

Commands with an AlarmWorX64 Viewer Target

Acknowledge

The Acknowledge command will acknowledge some or all of the alarms in the viewer. The **Alarm** property filters the types of alarms to acknowledge. "None" means no type filter (all alarm types). **Acknowledge** specifies the scope of the acknowledgement.

- **Acknowledge: All** – Every alarm will be acknowledged.
- **Acknowledge: Comparison** – Allows you to specify a particular field, and only acknowledge alarms that match that value in that field. See the next section for more information on how to configure this type of acknowledge command.
- **Acknowledge: Filter** – Only acknowledged alarms that match the current filter.
- **Acknowledge: Selection** – Only acknowledge alarms that are selected.
- **Acknowledge: Inactive** – Only acknowledge alarms that are inactive (i.e. have returned to normal).

LowSeverity and **HighSeverity** limit the severities of the alarms to be acknowledged. These numbers are inclusive.

VisibleOnly determines whether only currently visible alarms will be acknowledged or all alarms in the viewer. (Note, if just one pixel of an alarm row is visible, the alarm will be considered visible.)

An alarm must meet all of these properties to be acknowledged. For example, if you have Acknowledge set to Selection and LowSeverity set to 500, then you select a group of alarms that includes one with a severity of 100, that one severity 100 alarm will not be acknowledged when the command is executed.

The **Comment** property allows you to optionally add a comment to every alarm that gets acknowledged with this command.

Acknowledge: Comparison

The "Comparison" form of the Acknowledge command, as mentioned above, is a way to acknowledge alarms that only match a particular value in a particular field. This is a very powerful form of acknowledgement, but is a bit more complex to configure.

This type of acknowledge command can be used to acknowledge one single alarm, for example, or only alarms with a particular text in the message.

To configure a comparison acknowledge you first need to decide what field you want to compare, and then find the proper name for that field. For example, let's say we want to acknowledge one particular alarm, so we are looking for the Tag or Source field.

Edit your target AlarmWorX32 Viewer, select the Grid object, and go to the Column tab. Hover your mouse over the column you want and observe the tooltip. For the Tag column, the Field Name is "OriginalSource" and the Field Type is "BaseConditionType".

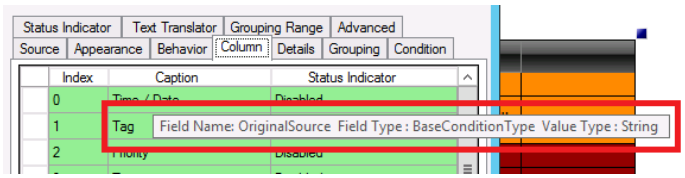

Figure 1 - Finding the proper field name of a column

The proper syntax for the **Field** property of the command is the Field Name combined with the Field Type, so for Tag it would be "OriginalSourceBaseConditionType". (Note that there are no spaces.)

The **Comparison** property is the string you'd like to match to that field. You can use an asterisk as a wild card. So, for example, to acknowledge all alarms with a source name that starts with "Tank" you would put "Tank*" in the Comparison property.

Command	**Acknowledge**
⊟ CommandParame	
DragDrop	False
Alarm	None
Acknowledge	**Comparison**
LowSeverity	0
HighSeverity	1000
VisibleOnly	False
Comparison	**Tank***
Field	**OriginalSourceBaseConditionType**
Comment	
Target	**AWX**

Figure 2 - Example Comparison Acknowledge command

Expand Collapse

When grouping is enabled the Expand Collapse command will expand or collapse those groups. The **ExpandCollapse** property determines whether the command will cause the groups to expand or collapse. With regards to the AlarmWorX64 Viewer, Collapse does the same as CollapseAll and Expand does the same as ExpandAll.

The **ItemName** and **Index** properties of this command are not applicable to the AlarmWorX64 Viewer.

Export Image

The Export Image command will export an image of the AlarmWorX64 Viewer based on either the current view or the whole alarm list. The **Area** property defines what region of the viewer you'd like to export.

- **Area: EntireControl** – Every historical alarm subscription in the control will be exported.
- **Area: CurrentView** – Only the selected view will be exported. Note, you can have multiple views (grids, lists, or charts) on a tab. The "current" view is the one that has focus. Usually it was the one that was last clicked. If you are showing the runtime toolbar, the current view will be the same one that the runtime toolbar interacts with.

The **FileName** property is the name and path of the exported file. Accepted file extensions are: png, jpg, jpe, jpeg, gif, tif, tiff, jfx, bmp, and ico.

Group

The Group command adds or removes grouping for the AlarmWorX64 Viewer. The **Action** property determines whether groups are added or removed.

- **Action: New** – Removes any existing grouping and groups only by the specified column.
- **Action: Add** – Adds a new grouping level for the specified column.
- **Action: Reset** – Removes all grouping.

Use either the **ItemName** or **Index** properties to determine the column to group. ItemName is the name of the column you would like to group. If no ItemName is specified, the Index is used.

Load Alarm Configuration

The Load alarm Configuration command will load a pre-created AlarmWorX64 Viewer configuration into the targeted AlarmWorX64 Viewer.

Print

The Print command will print some or all of the AlarmWorX64 Viewer. Similar to the Export Image command, the **Area** property defines what region of the display you'd like to print.

- **Area: EntireControl** – Every historical alarm subscription in the control will be printed.
- **Area: CurrentView** – Only the selected view will be printed. Note, you can have multiple views (grids, lists, or charts) on a tab. The "current" view is the one that has focus. Usually it was the one that was last clicked. If you are showing the runtime toolbar, the current view will be the same one that the runtime toolbar interacts with.

The **WhiteBackground** property does not apply to AlarmWorX64 Viewer targets.

ShowPrintDialog will determine whether the printer dialog is shown, otherwise the default or specified printer and settings will be used and the print will happen immediately. **PrinterName** specifies a specific printer to use instead of the default.

> **NOTE:** Both the ShowPrintDialog and the PrinterName parameters are ignored in Silverlight environments. The runtime print dialog will always be shown.

> **NOTE:** The Print command is not supported in Windows RT or Windows Phone environments.

Refresh

The Refresh command will refresh historical alarm data. It does not apply to live alarm data. This is the same as hitting the "Refresh" in the AlarmWorX Viewer runtime ribbon or right-click context menu. The **Area** property determines how much of the viewer is refreshed.

- **Area: EntireControl** – Every historical alarm subscription in the control will be refreshed.

- **Area: CurrentView** – Only the selected view will be refreshed. Note, you can have multiple views (grids, lists, or charts) on a tab. The "current" view is the one that has focus. Usually it was the one that was last clicked. If you are showing the runtime toolbar, the current view will be the same one that the runtime toolbar interacts with.

Save Configuration

The Save Configuration command will save the AlarmWorX64 Viewer configuration to the specified **FileName**. When **Overwrite** is true, an existing file with the specified name will be overwritten.

> **NOTE:** No message will be shown if the specified file already exists and Overwrite is set to False. The save action will simply not do anything in that case.

> **NOTE:** The Save Configuration command is not supported in Silverlight, Windows RT, or Windows Phone environments.

Select Element

The Select Element command allows you to change the tab or select a row of the AlarmWorX64 Viewer. **Element** specifies the whether you are selecting a tab, grid, or row. **ElementName** or **Index** are used to specify the particular tab, grid, or row to be selected.

- **Element: Tab** – Activates the specified tab.
- **Element**: **View** – Activates the specified grid. This command is most useful if you have more than one grid object within a AlarmWorX64 Viewer tab. The selected grid will receive any subsequent commands.
- **Element: Row** – Selects the specified row. Use Index to determine which row is selected.
- **Element**: **Object** – This Element option is not applicable to the AlarmWorX64 Viewer.

Set Filter

The Filter command will apply the selected filter or filter expression to the AlarmWorX64 Viewer, or remove all filters. The **Action** property determines what action is taken.

- **Action: Set** – Applies the configured filter or filter expression.
- **Action**: **Reset** – Remove all filters.

FilterName is the name of a pre-existing filter to apply. **FilterExpression** is the expression of a new filter to apply.

> **NOTE:** Because this command can be applied to many different types of targets, the FilterExpression field provides no help and has no error checking for expressions. Unless you are very familiar with the syntax it's recommended that you use the filter editor within the AlarmWorX64 Viewer to compose your filter expressions, then copy and paste them into the FilterExpression field.

Set Time Range

The Set Time Range command filters the alarm viewer to a particular range between **StartTime** and **EndTime**. This command only applies to an AlarmWorX64 Viewer displaying historical data.

Set Visibility

The Set Visibility command allows you to hide or show columns in the AlarmWorX64 Viewer. For an AlarmWorX64 Viewer target, the **ObjectType** must be set to Column. Visual and Pen ObjectTypes do not apply to the AlarmWorX64 Viewer.

ObjectName is the name of the column whose visibility you want to change. If ObjectName is not specified, **Index** is used.

The **Visibility** property specifies whether you want the command to show the column, hide it, or toggle between hidden and shown.

Sort

The Sort command applies sorting to the specified column of the AlarmWorX64 Viewer. The **Action** property determines whether the command will replace existing sorting, add to it, or remove it.

- **Action: New** – Removes any existing sorting and sorts only by the specified column.
- **Action: Add** – Adds a new sorting level for the specified column.
- **Action: Reset** – Removes all sorting.

Use either the **ItemName** or **Index** properties to determine the column to sort. ItemName is the name of the column you would like to sort. If no ItemName is specified, the Index is used.

The **Direction** property specifies whether to sort Ascending or Descending.

Commands Sent From AlarmWorX64 Viewer

The AlarmWorX64 Viewer can actually be configured to send any command from a given alarm.

First, we need to create a virtual column in the AlarmWorX64 Viewer that the user can click to execute a command.

1. Configure your AlarmWorX64 Viewer control, select the grid object, and go to the **Source** tab.
2. Select the "Edit the fields of the subscription" button. (You must not be using one of the pre-configured read-only subscriptions, or this button will be grayed out.)

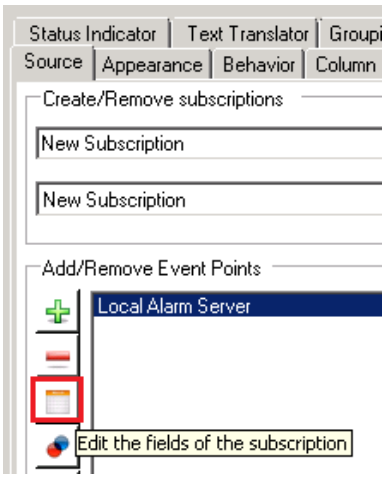

Figure 3 - "Edit the fields of the subscription" button

3. Go to the **Virtual Fields** tab.
4. Add a new virtual field. Name it "WriteValue".
5. Click the "…" in the Expression column to bring up the expression editor.
6. Click the **Insert Command** button in the bottom left.

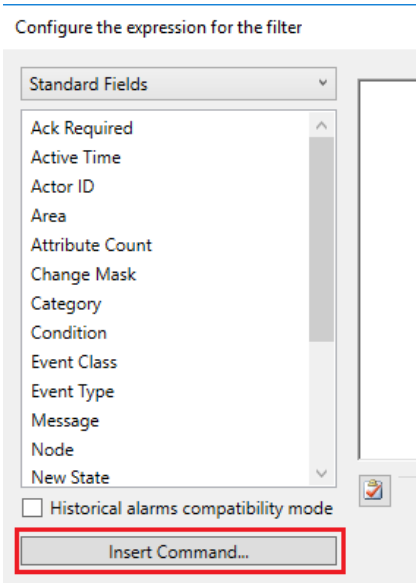

Figure 4 - "Insert Command" button

7. Configure the Write Value command, as shown in Figure 5, then click **OK**. You should now have a serialized command in the expression editor.

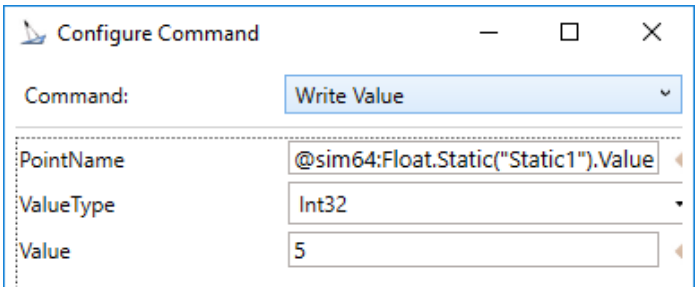

Figure 5 - Configuring a Write Value command

8. If the OPC tag that you chose contained any double-quotation marks (like @sim64:Float.Static("Static1").Value) it will make an invalid expression – either add dollar signs to the outside of the whole string or replace the double-quotation marks in the tag name with single quotation marks. The strings below are both valid ways to correct the expression.

```
$"command:Type=WriteValueCommandData&Command=<cmd><PointName>@sim64:Float.Static("Static1").Value</PointName><Value>5</Value></cmd>"$
```

```
"command:Type=WriteValueCommandData&Command=<cmd><PointName>@sim64:Float.Static('Static1').Value</PointName><Value>5</Value></cmd>"
```

9. Click **OK** to the expression editor. Your final virtual field should look something like this:

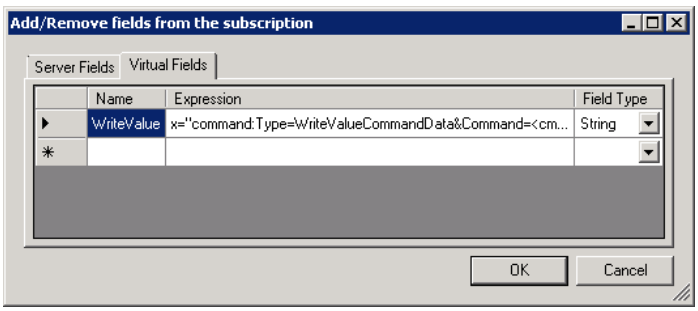

Figure 6 - Example Virtual Field

10. Click **OK** when done.
11. Go to the **Column** tab.
12. Select the new virtual field we added, which should be at the bottom, and should be named "WriteValue".
13. Check the box for **Contains a clickable link**. If desired, change the "Link Text".
14. Also if desired, change the position of this new column and make any other changes you feel necessary. Click the green checkmark button when done.

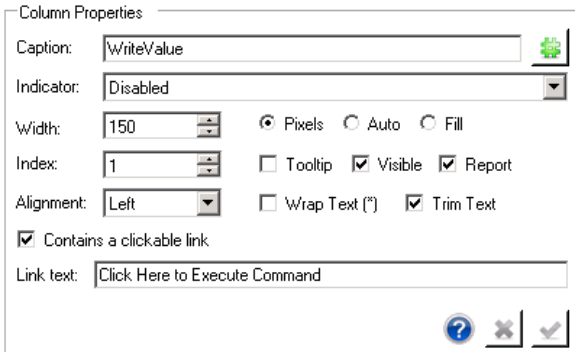

Figure 7 - Clickable column properties

15. Close the configuration dialog and put your viewer into runtime. You should now have a new column named "WriteValue", and if you click on this column it should execute your command.

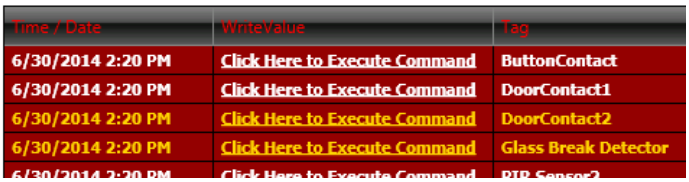

Figure 8 - Clickable column

To make this feature even more powerful, you can edit the expression to use fields from the alarm, like the source name, message, priority, etc.

1. Configure the AlarmWorX64 Viewer control again, go back to the "Edit the fields of the subscription" button on the Source tab of the grid object, go back to the Virtual Fields tab, and click the "**...**" button for the expression of your WriteValue field.
2. Find the place in the string that represents the value to write (between the <Value> tags) and replace it with the Severity of the alarm. For best results, choose the "Severity (Priority)" column from the left rather than typing it in. We will also need to wrap it in the "toString" function, if the value is not already a string. The final expression should look something like this (edited parts highlighted in yellow):

```
"command:Type=WriteValueCommandData&Command=<cmd><PointName>@sim64:Float.Static('Static1').Value</PointName><Value>"+tostring({{BaseEventType.Severity}})+"</Value></cmd>"
```

3. Click **OK** or **Close** to all dialogs when finished, then put your alarm viewer back into runtime. Now when you click on the column it should write the severity of the alarm for that row.