

Description: A scripting guide showing interaction to the AlarmWorX64 Viewer control.

OS Requirement: Windows Server 2003 x64/Vista x64/ Server 2008 x64/Windows 7 x64/ Server 2008 R2 x64

General Requirement: General knowledge of GraphWorX64 and its scripting capabilities; familiarity with the AlarmWorX64 Viewer control.

Introduction

This guide will introduce you to writing scripts that interface with the AlarmWorX64 Viewer. It includes an example of how to acknowledge all the alarms in an AlarmWorX64 Viewer using GraphWorX64 Scripting.

This document assumes that you have read the *GraphWorX64 Scripting – Quick Start* application note.

Preparing the display

To create this example we will need a display containing an AlarmWorX64 Viewer control, plus one button to trigger our script. Start GraphWorX64, create a new display and add those two elements to the display.

It is very important that the alarm viewer control has a name. Our scripts will use this name to identify which alarm viewer to interact with (in case there is more than one on a display). To name your alarm viewer, double-click on it to open the properties then go to the Advanced tab. (You could also click once on the alarm viewer and look at the Properties panel.) In the Common section fill in “AwvViewer1” for the “(Name)” field.

Your display should look something to the one in Figure 1.

Time / Date	Tag	Priority	Type	Quality	Description
1/8/2010 8:51 AM	Coolant Level	100	LO	Good	Coolant level is low
1/8/2010 10:57 AM	/+Pressure+/-	250	HIHI	Good	Belt Speed of Pump1 is normal
1/8/2010 10:26 AM	Belt Speed	50	LOLO	Good	The PSI in Tank1 is normal
1/8/2010 9:17 AM	Tank PSI	800	LOLO	Good	Level gauge is normal
1/8/2010 8:39 AM	Level Gauge	750	LOLO	Good	Compressor gauge is reading normal
1/8/2010 9:29 AM	Compressor	700	LO	Good	Core humidity is normal
1/8/2010 9:44 AM	Humidity	850	LOLO	Good	Belt1 on the Box Line is slow
1/8/2010 10:33 AM	Box Line	300	LO	Good	Belt Speed of Pump1 is low
1/8/2010 9:09 AM	/+Pressure+/-	250	LO	Good	The PSI in Tank1 is normal
1/8/2010 10:56 AM	/+Temperature+/-	100	HIHI	Good	Level gauge is low
1/8/2010 10:33 AM	Belt Speed	100	LO	Good	Arm Torque is at normal
1/8/2010 9:42 AM	Tank PSI	600	LO	Good	Critical Mass is ok
1/8/2010 8:56 AM	Level Gauge	500	LO	Good	Belt Speed of Pump1 is normal
1/8/2010 11:19 AM	/+Pressure+/-	550	LO	Good	Valve 3 Pressure: normal
1/8/2010 10:51 AM	Arm Torque	900	HIHI	Good	The PSI in Tank1 is normal
1/8/2010 8:53 AM	Critical Mass	70	HI	Good	Level gauge is too low
1/8/2010 9:10 AM	Belt Speed	200	LOLO	Good	Compressor gauge is normal

Acknowledge All Alarms

Figure 1 - Display with Alarm Viewer and Button

Configure the button to perform the Run Script command and create a new function from the Events page. (The procedure for this is described in the *GraphWorX64 Scripting – Quick Start* application note.)

Designing the Script

To acknowledge all the alarms we will use a specific method exposed by the AlarmWorX64 Viewer grid. To do this, we first need to understand a little bit better how the AlarmWorX64 Viewer control is organized.

The AlarmWorX64 Viewer control can have several tabs, and each tab can have grids or charts in it. For example, in the next figure we have a control with two tabs, with the first tab containing a grid, then a chart, then another grid.

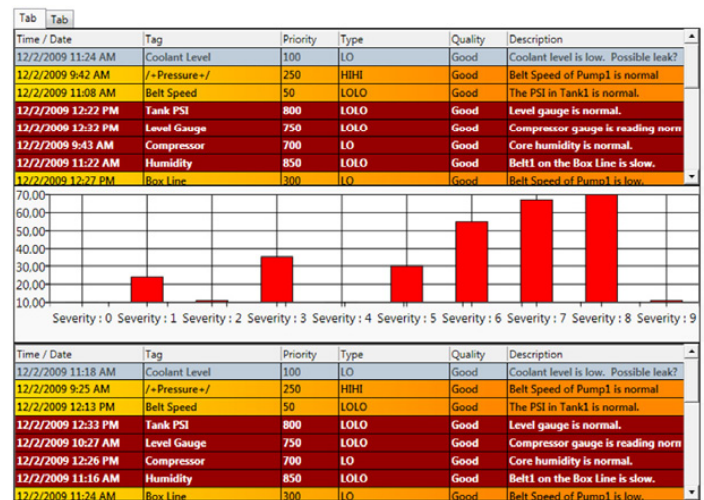


Figure 2 - A More Complicated Alarm Viewer

For this example we will use an AlarmWorX64 Viewer control with just one tab and one grid, but it is important for you to understand that in a real application you may have a more complex scenario to work with.

Before writing the actual code, let's review what we need to do in our script:

1. First, we need to get a reference to the AlarmWorX64 Viewer control.
2. Then, we need to get a reference to its first tab.
3. From the tab, we take a reference to the grid.
4. Once we have the grid's reference, we can invoke the method to acknowledge all the alarms.

Writing the script

Now that we know what needs to be done we can start writing the code to implement the script.

First we need to know how to get a reference to the AlarmWorX64 Viewer control. We will declare a variable of the correct type, then use the `GetObjectByName` method to retrieve the reference to it.

```
var awxViewer : Ico.Awx.AwxViewControl =
ThisConfiguration.GetObjectByName("AwxViewer1").ToDependencyObject();
```

NOTE: We are using the AlarmWorX64 Viewer named "AwxViewer1".

Now we will get the reference to the first (and only) available tab. With this next line we are accessing the first element of the `Items` property of the viewer control.

```
var tab : ItemsControl =
ItemsControl(awxViewer.Items[0]);
```

Using a similar approach we get a reference to the first grid of the chosen tab.

```
var grid : Ico.Awx.DataGrid.AwxGridView =
Ico.Awx.DataGrid.AwxGridView(tab.Items[0]);
```

We are now ready to invoke the method to acknowledge all the alarms.

```
grid.Acknowledge(Ico.Awx.DataGrid.AckType.All);
```

The code is complete. We can save our display, go to runtime mode, wait for some alarms to come, then press the button to acknowledge them all (provided we have permission to perform this operation, of course).

Conclusion

This simple example shows some important concepts, like how to get a reference to an alarm viewer control. This is very similar to the procedure for getting a reference to other controls as well. You've also learned about how the AlarmWorX64 Viewer is structured and how to interact with its elements.

Feel free to experiment with the other available methods. You can find these by using the Intelli-sense built into the Script Editor, or by browsing the GENESIS64 API Reference Help.

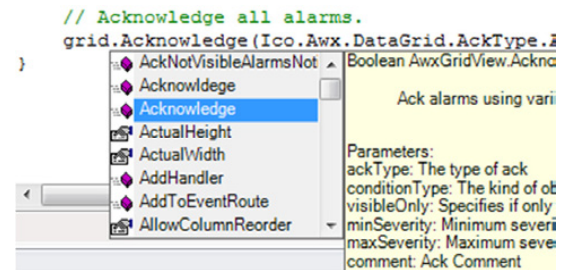


Figure 3 - Intelli-sense Showing Available Methods and Properties

ICONICS Customer Connection Portal hosts a library of automation examples that are very handy when you start scripting.

Complete code

Here is the full code of the function we created in the previous section, along with some comments.

```
function Pick1_CommandExecuted(sender : System.Object, cmdArgs : Ico.Gwx.CommandExecutionEventArgs)
{
    // Take a reference to the AwxViewer control.
    var awxViewer : Ico.Awx.AwxViewControl =
        ThisConfiguration.GetObjectByName("AwxViewer1").ToDependencyObject();

    // Get a reference to the first (and only) tab available.
    var tab : ItemsControl = ItemsControl(awxViewer.Items[0]);

    // Get a reference to the first (and only) grid.
    var grid : Ico.Awx.DataGrid.AwxGridView = Ico.Awx.DataGrid.AwxGridView(tab.Items[0]);

    // Acknowledge all alarms.
    grid.Acknowledge(Ico.Awx.DataGrid.AckType.All);
}
```