



GraphWorX64 Scripting

Interacting with AlarmWorX64 Viewer



APPLICATION NOTE

September 2013

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

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

General Requirement: General knowledge of GraphWorX64 and its scripting capabilities.

Introduction

This guide will show you how to perform a simple task like acknowledging all the alarms in an AlarmWorX64 Viewer control through GraphWorX64 Scripting.

Although the example is very simple, you may want to read "GraphWorX64 Scripting – Quickstart" first, to introduce yourself to GraphWorX64 Scripting.

Preparing the display

To create this example we will need a display containing an AlarmWorX64 Viewer control and a button to trigger our script. So, just start GraphWorX64 and add those two elements to the display. It is very important that the alarm viewer control has a name, so we will name it "AwvViewer1".

In our case, the display looks like this:

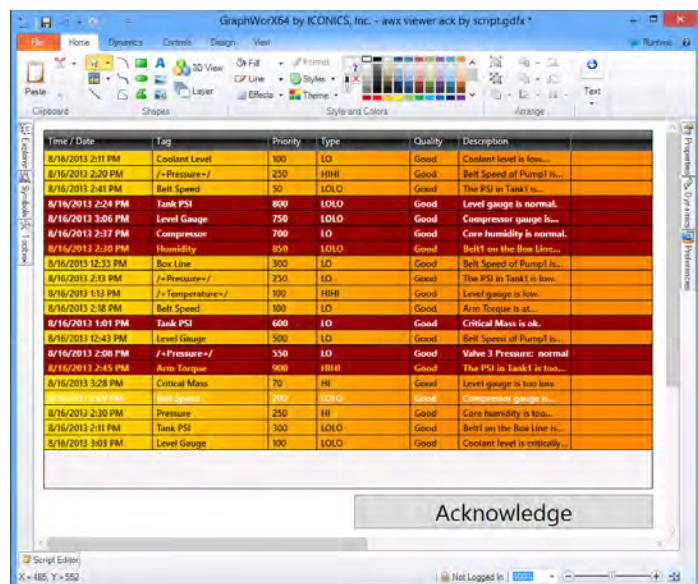


Figure 1 – The display with the alarm viewer and the button

As we have already done in the "GraphWorX64 Scripting – Quickstart" Application Note we will now set the button to run a

script, and create the empty script function that we will use through the rest of this 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.

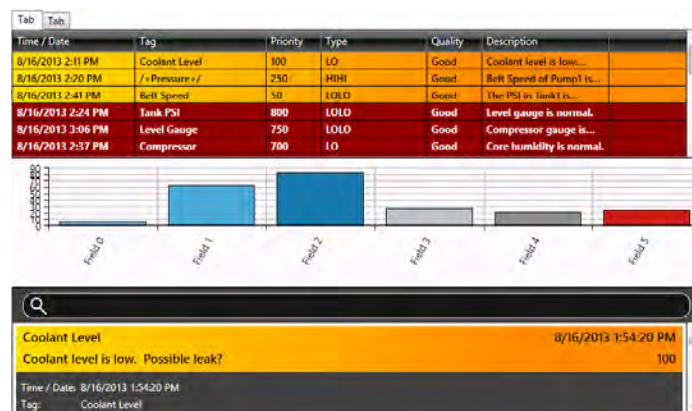


Figure 2 – An alarm viewer with some elements in it

To keep this example easier, we will use an AlarmWorX64 Viewer control with just one tab and one grid, but it is important for you to know that in your 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.



GraphWorX64 Scripting

Interacting with AlarmWorX64 Viewer



APPLICATION NOTE

September 2013

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").ToDe  
pendencyObject();
```

Please note that you have to name AlarmWorX64 Viewer control using same name as you are specifying in parameter of `GetObjectByName` method. To do so just select the control and in Properties tab fill the name in very first property called '(Name)'.

Now we have our reference to the alarm viewer. We will now get the reference to the first (and only) tab. AlarmWorX64 Viewer tabs have type `ItemsControl`. Please note that we are accessing the first element of the `Items` property of the viewer control.

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

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

```
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. This will be little bit more complicated as we have to fill all parameters of `Acknowledge` method for proper functionality. This involves using of `AlarmType` type which is located in different assembly that AWX64 Viewer itself thus we have to bind it manually:

1. In Script Editor click 'Edit References' button , a dialog will open
2. Click to last free row, column 'Assembly Codebase' to get '...' button displayed, click it to open browser
3. Browse for file 'AwxCClientCommon.dll' located (by default) in
C:\Program Files\ICONICS\GENESIS64\Components
folder, confirm the selection
4. Confirm the References dialog

Now we can start writing a code for `AcknowledgeByFilter` method which has following syntax:

```
StatusCode AwxGridView.AcknowledgeByFilter(AckType  
ackType, AlarmType type, Boolean visibleOnly,  
Int32 minSeverity, Int32 maxSeverity, String comment,  
String key, String value)
```

Code for acknowledging of all alarms in the grid is:

```
grid.AcknowledgeByFilter(Ico.Awx.AckType.All,  
Ico.Awx.Common.AlarmType.None, false, 0, 100, "My  
Comment", "", "");
```

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 enough permission to perform this operation, of course).

Conclusion

This simple example shows some important concepts. For example, you've learned how to get a reference to an alarm viewer control. This is very similar to getting a reference to other controls too. You've also learned about how the AlarmWorX64 Viewer is structured and how to interact with its elements. Even if calling the `Acknowledge` method is one of the simplest operation available, feel free to experiment with the other methods available. Intellisense will help you in browsing through them.

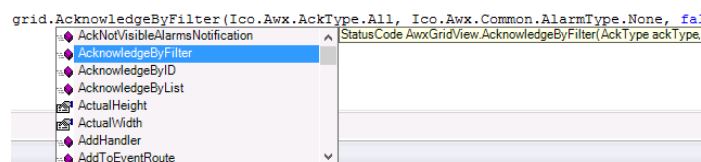


Figure 3 – IntelliSense showing available methods and properties



GraphWorX64 Scripting

Interacting with AlarmWorX64 Viewer



APPLICATION NOTE

September 2013

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 : System.Windows.Controls.ItemsControl =
        System.Windows.Controls.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.AcknowledgeByFilter(Ico.Awx.AckType.All, Ico.Awx.Common.AlarmType.None, false, 0, 1000,
        "My Comment", "", "");
}
```