



ICONICS, Inc.
100 Foxborough Blvd.
Foxborough, MA 02035
Phone: (508) 543-8600
Fax: (508) 543-1503
[Worldwide Support Centers](#)
www.iconics.com



© ICONICS, Inc. All rights reserved.
Specifications are subject to change without notice.
GENESIS, GENESIS32, Pocket GENESIS, BizViz
and their respective modules, OPC-To-The-Core,
and Visualize Your Enterprise are trademarks of
ICONICS, Inc. Other product and company names
mentioned herein may be trademarks of their
respective owners.

Introduction to the AlarmWorX32 Indicator ActiveX

The AlarmWorX32 Indicator ActiveX, shown below, notifies you of alarms that are occurring in your GENESIS32 system. The flexible Alarm Indicator allows you to monitor alarm activity from virtually anywhere on your computer display, including various container applications as well as the Windows taskbar.



AlarmWorX32 Indicator ActiveX

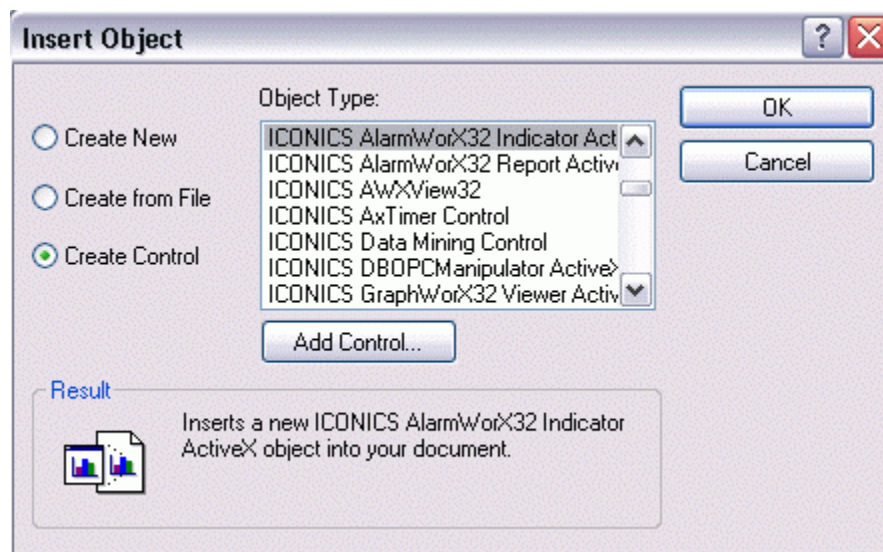
The Alarm Indicator includes the following features:

- Monitors the alarm status of the aggregate system
- Configurable color, flashing, and sound parameters
- Configuration for multiple alarm severity ranges
- Sound suppression
- Sizing and scaling for increased flexibility
- Support for multiple concurrent instances
- Global sound suppression for multiple instances
- Ability to launch applications from Alarm Indicator
- Configuration of alarm filters and subscriptions
- Support for custom image files
- Support for custom sound files

The AlarmWorX32 Indicator ActiveX is included in the GENESIS32 installation. It can be inserted into any container (such as GraphWorX32, AlarmWorX32, TrendWorX32, Microsoft Visual Basic forms, Microsoft Word, and Microsoft Excel) that has the capability to embed ActiveX objects.

AlarmWorX32 Indicator ActiveX Properties

To insert the AlarmWorX32 Indicator ActiveX, choose **Insert New Object** from the **Edit** menu in any of the GENESIS32 containers, including GraphWorX32, TrendWorX32, and AlarmWorX32. This opens the **Insert Object** dialog box, shown below. Select **AlarmWorX32 Indicator ActiveX** from the list of object types in the **Insert Object** dialog box. Alternatively, you can click the **AlarmWorX32 Indicator ActiveX** button on the **ActiveX** toolbar.



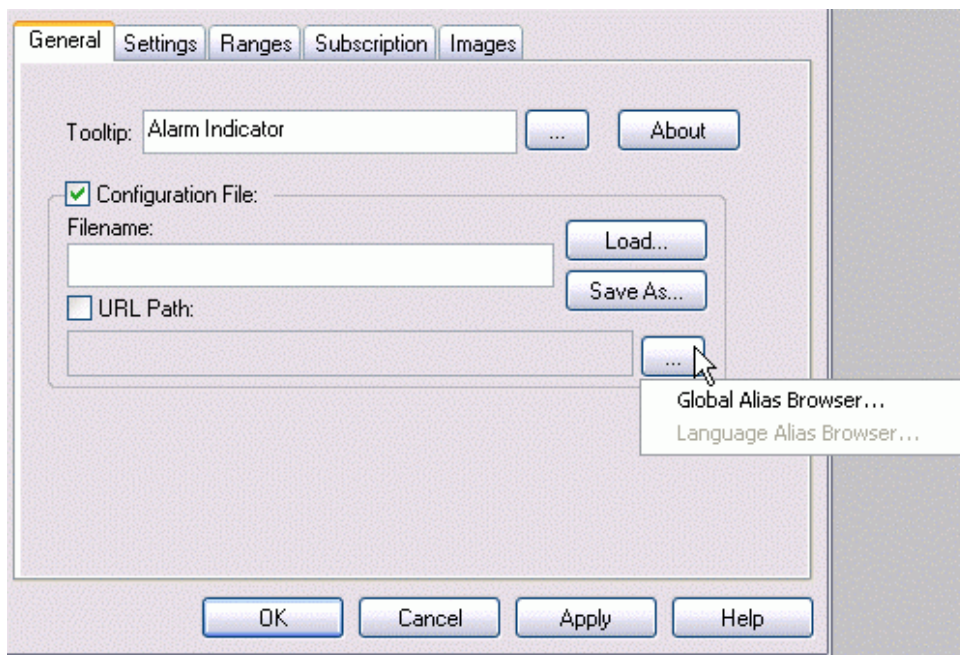
Insert Object Dialog Box

Double-clicking on the Indicator ActiveX (or right-clicking and then selecting **AlarmWorX32 Indicator ActiveX Object** from the pop-up menu) opens the **AlarmWorX32 Indicator ActiveX Properties** dialog box, which contains the following tabs:

- General Tab
- Settings Tab
- Ranges Tab
- Subscription Tab
- Images

General Tab

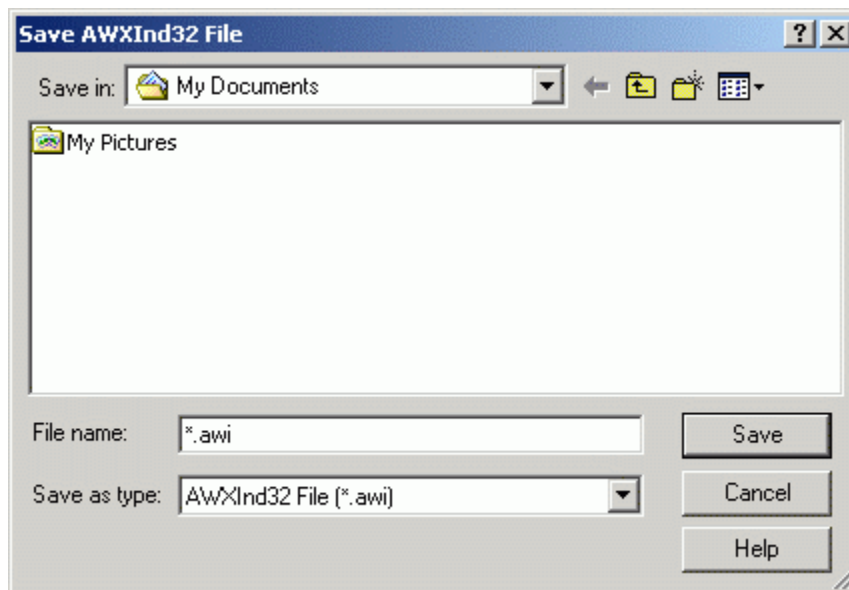
The **General** tab of the **AlarmWorX32 Indicator ActiveX Properties** dialog box, shown below, allows you to specify the text for the ToolTip that appears when the mouse pointer is placed over the Indicator ActiveX during runtime mode. Simply enter the desired text in the **ToolTip** field. In the example below, the text "Alarm Indicator" will appear in the ToolTip during runtime mode.



AlarmWorX32 Indicator ActiveX Control Properties: General Tab

Configuration File

By default, information is stored as part of the container application's document file. If the **Configuration File** check box is checked, as shown above, information is stored as part of a separate (.awi) file. To save the .awi file, click **Save As** to open the **Save AWXInd32 File** dialog box, shown below. Enter the name for the .awi file in the **File Name** field.



Saving an .awi File

You can also open an existing configuration file by clicking the **Load** button on the **General** tab and browsing for the file. If **Configuration File** is selected, the viewer will attempt to load from the file listed in Filename. If both **Configuration File** and **URL Path** are selected, the ActiveX will use the URL path upon loading the file.

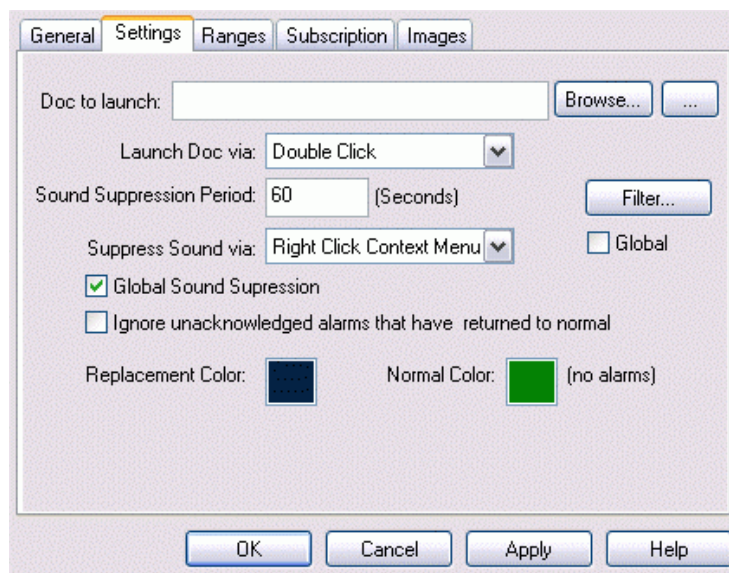
Note: The ActiveX cannot be saved to a URL path. To create a URL file, save a report configuration to a local file and copy the file to the desired network location.

Clicking the **About** button on the **General** tab opens the **About Box**, which contains information about this product.

You can also select aliases to use for the configuration file. Clicking the ... button and selecting **Global Alias Browser** from the pop-up menu opens the Global Aliases tab of the Unified Browser. Select a global alias from the Unified Browser, which includes all global aliases in the global alias database. This eliminates the need to manually type in the alias name. All global aliases that are configured in the Global Alias Engine Configurator are conveniently available to choose from inside the browser. The tree control of the Global Alias Engine Configurator is mimicked in the tree control of the Browser. Select a global alias by double-clicking the alias name. The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.

Settings Tab

The **Settings** tab of the **AlarmWorX32 Indicator ActiveX Properties** dialog box, shown below, allows you to configure the runtime settings for the Alarm Indicator ActiveX. Runtime options for the Indicator ActiveX include launching an application and suppressing the sound (if a sound is used to indicate an alarm) when you click on the ActiveX.



AlarmWorX32 Indicator ActiveX Control Properties: Settings Tab

Launching an Application From the Indicator ActiveX

The **Doc to Launch** field enables you to specify an application to be launched from the Indicator ActiveX. Click the **Browse** button to select an application file. In the **Launch Doc Via** field, choose the method by which you want the application to be launched (**Single-Click**, **Double-Click**, or **Right-Click Context Menu**). For example, suppose you want to launch a Microsoft Excel document by right-clicking the Alarm Indicator ActiveX when an alarm is indicated in runtime mode. First select your Excel document in the **Doc to Launch** field, and then select **Right-Click Context Menu** from the **Launch Doc Via** field. In runtime mode, right-click the Indicator ActiveX and then select **Launch Doc** from the pop-up menu, as shown below.



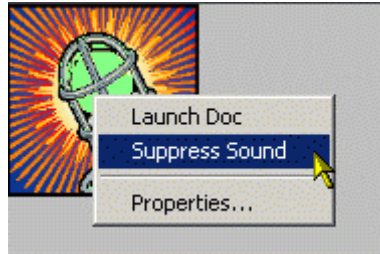
Launching an Application During Runtime

You can also select aliases to use for the **Doc to Launch** field. Clicking the ... button and selecting **Global Alias Browser** from the pop-up menu opens the Global Aliases tab of the Unified Browser. Select a global alias from the Unified Browser, which includes all global aliases in the global alias database. This eliminates the need to manually type in the alias name. All global aliases that are configured in the Global Alias Engine Configurator are conveniently available to choose from inside the browser. The tree control of the Global Alias Engine Configurator is mimicked in the tree control of the Browser. Select a global alias by double-clicking the alias name. The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.

Turning off the Sound of an Alarm Indicator

If you have configured a sound to be used when an alarm is indicated, you can turn off the sound during runtime mode. The **Settings** tab allows you to configure the parameters for sound suppression. The **Sound Suppression Period** is the duration of sound suppression (in seconds). In the **Suppress Sound Via** field, choose the method by which you want turn off the sound (**Single-Click**, **Double-Click**, or **Right-Click Context Menu**). For example, if you want to turn off the sound by right-clicking the Alarm Indicator ActiveX when an alarm is indicated in runtime mode, select **Right-Click Context Menu** from the **Suppress Sound Via** field. In runtime mode, right-click the Indicator ActiveX and then select **Suppress Sound** from the pop-up menu, as shown below.

Note: The "suppress sound" method can be coupled with the "launch doc" method. For example, suppose **Double-Click** is selected in both the **Suppress Sound Via** field and the **Launch Doc Via** field. When the Alarm indicator is double-clicked during runtime mode, the selected application will be launched, *and* the sound will be turned off.



Turning off Sound During Runtime

Global Sound Suppression

If multiple instances of the Alarm Indicator ActiveX are running concurrently and each instance is configured to produce a sound indicating an alarm, you can use the **Global Sound Suppression** function in the **Settings** tab to monitor the sound status of all instances. For example, suppose you have two Alarm Indicators open (Alarm Indicator 1 and Alarm Indicator 2). If Alarm Indicator 1 sounds an alarm, and the **Global Sound Suppression** check box is checked, the sound of Alarm Indicator 2 will be temporarily suppressed until Alarm Indicator 1 stops making a sound.

Note: The **Launch Doc**, **Suppress Sound**, and **Properties** commands are generally available on the right-click menu of the Alarm Indicator ActiveX during runtime mode (depending on the Security server settings). The **Properties** command opens the **AlarmWorX32 Indicator ActiveX Properties** dialog box.

If the **Ignore unacknowledged alarms that have returned to normal** check box is checked, all alarms that return to normal are automatically treated as acknowledged, even when they are not acknowledged.

Choosing Colors for the Alarm Indicator

When an alarm is triggered, the Alarm Indicator ActiveX will flash repeatedly to visually indicate the alarm. In the **Settings** tab, you can set the color of the flash. First select the **Normal Color** by clicking on the color palette. The normal color is the runtime color of the ActiveX when there is no alarm.

The flashing of the Alarm Indicator during runtime mode is made possible by simply replacing the normal color with a replacement color. Select the **Replacement Color** by clicking on the color palette. For example, if the normal color is green and the replacement color is red, the light bulb of the Alarm Indicator will be green when no alarms are occurring, as shown below.



Example of Normal Color (No Alarm)

If an alarm occurs, the bulb will flash red, as shown below.



Example of Replacement Color (Alarm Occurring)

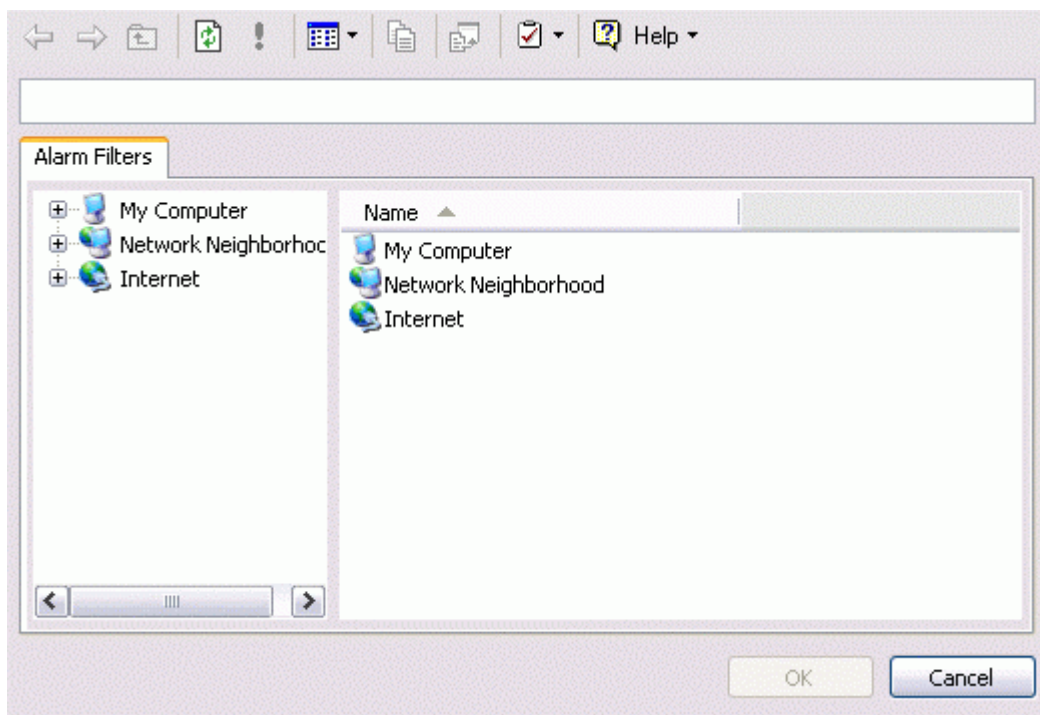
Setting Up Filters

The Alarm Indicator uses local filters by default. You have the option of using both global alarm filters and local alarm filters. To create a global filter, check the **Global** check box on the **Settings** tab.

Note: When using global filters, if you switch to any global filter, then there will be a delay for the global filter to update. During this delay period, the previous filter is still active. If a global filter cannot be resolved, then all alarms will be passed through.

If you use a global filter, the **Unified Data Browser** dialog box opens, as shown in the figure below. where you can select a filter from the **Alarm Filters** tab, which lists all global filters configured in the Unified Data Manager. Global filters are prefixed by "awxf".

For more information about filters, please see the Unified Data Manager help documentation and the AlarmWorX32 Viewer ActiveX help documentation.



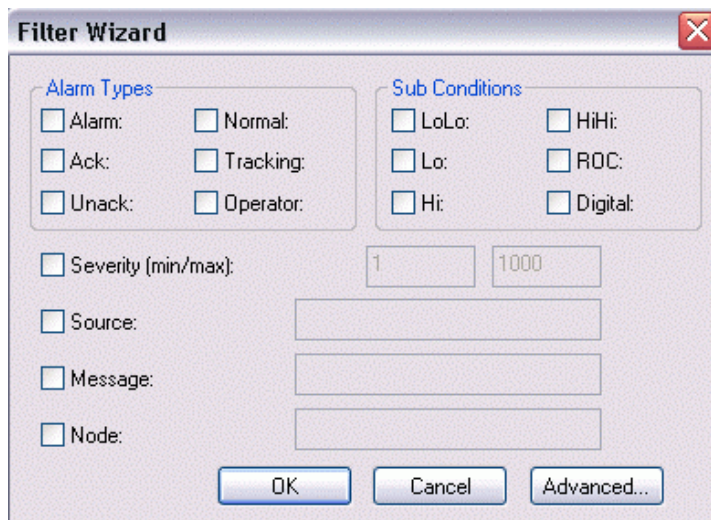
Unified Data Browser (Global Filter)

Clicking the **Filter** button on the **Settings** tab opens the **Filter Wizard**. The **Filter Wizard**, shown in the figure below, allows you to choose from the following items to enter in your expression. Select one or more items, and then click **OK**.

- **Alarm Types:** Alarm, Ack, Unack, Normal, Tracking, and Operator
- **Subconditions:** LoLo, Lo, Hi, HiHi, ROC, and Digital
- **Severity:** This is the OPC-defined value for alarm priority. The valid OPC severity range is 1 (lowest) to 1000 (highest). The first number is the low limit, and the second number is the high limit.
- **Source:** The tag name.
- **Message:** Description of the tag.
- **Node:** Name of the PC from which the tag originates.

Clicking the **Advanced** button opens the Expression Editor.

Note: Alarm Types are logically "or" with Alarm Types. Subconditions are logically "or" with Subconditions. Alarm Types, Subconditions, Severity, Source, Message, and Node are logically combined ("and") to create the expression string.



The **Filter Wizard** dialog box is used to configure alarm filters. It contains the following sections:

- Alarm Types:**
 - ☐ Alarm: ☐ Normal:
 - ☐ Ack: ☐ Tracking:
 - ☐ Unack: ☐ Operator:
- Sub Conditions:**
 - ☐ LoLo: ☐ HiHi:
 - ☐ Lo: ☐ ROC:
 - ☐ Hi: ☐ Digital:
- Severity (min/max):** Two input fields with values 1 and 1000.
- Source:** An empty text input field.
- Message:** An empty text input field.
- Node:** An empty text input field.

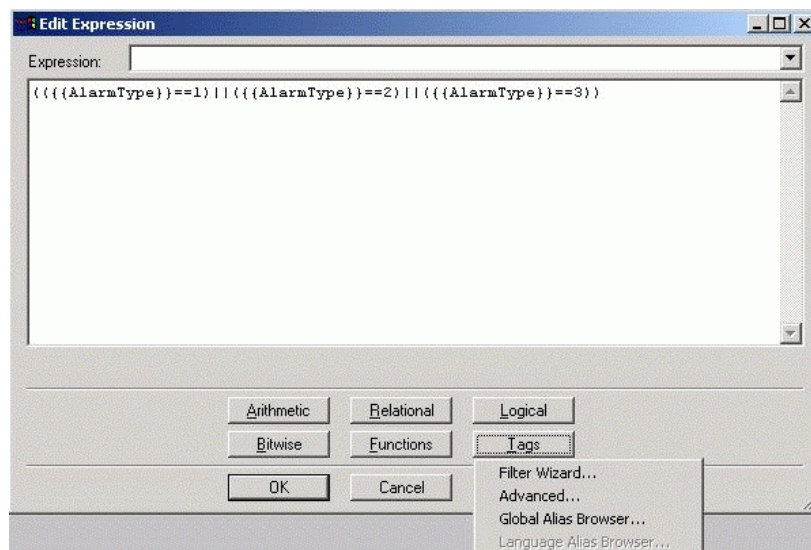
Buttons at the bottom: **OK**, **Cancel**, and **Advanced...**

Filter Wizard

Expression Editor

The filter string is automatically inserted into the **Edit Expression** dialog box. To edit the expression, click the **Edit** button on the **Client Filters** dialog box. This opens the **Edit Expression** dialog box, shown below, which enables you to edit alarm filters. The Expression Editor provides a Filter Wizard and an Alarm Tag list to help you create simple alarm filters. If you want to customize your alarm filters, you can use the other functions in the Expression Editor to set up your alarm filters manually.

Note: For more information about the Expression Editor, please see the Expression Editor help documentation.



The **Edit Expression** dialog box shows the current filter expression in a text area. Below the text area are buttons for **Arithmetic**, **Relational**, **Logical**, **Bitwise**, **Functions**, and **Tags**. The **Tags** button is currently selected, opening a dropdown menu with the following options:

- Filter Wizard...
- Advanced...
- Global Alias Browser...
- Language Alias Browser...

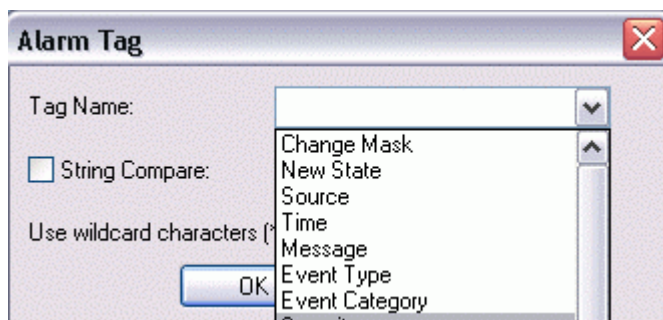
Buttons at the bottom: **OK** and **Cancel**.

Edit Expression Dialog Box

Selecting Alarm Attributes

Selecting **Advanced** from the **Tags** menu of the Expression Editor opens the **Alarm Tag** list, shown in the figure below, which allows you to choose alarm attributes for your alarm filter. Select the attribute that you want to include in the filter expression and click **OK**.

Note: You can also perform a wildcard string compare using the "like" operator syntax. Check the **String Compare** check box and type the desired string in the text field.



Alarm Attributes List

There are two additional attributes available for use in filtering: **Alarm Type** and **Current Time**. The Alarm Type attribute allows you to filter alarms according to ALARM 1, ACK 2, UNACK 3, OPER 4, TRACK 5 or NORM 6. For example, you can set up a filter with the condition:

`x={{AlarmType}}`

If the **Alarm Type** is true, then the alarms are displayed. If they are false then, the alarms are not displayed.

The **Current Time** attribute allows you to filter according to the current time. Only alarms occurring around the current time will be displayed.

Example Alarm Filters

Expression	Result
<code>x={{Severity}} > 500.</code>	Only alarm messages with a severity greater than 500 will be visible.
<code>x=like({{Source}}, "Tag",0)</code>	Only messages with the tag in the source name will be displayed.
<code>x=like({{Message}}, "Boiler Pump not*",0)</code>	Only messages with a description beginning with "Boiler Pump not" will be displayed.

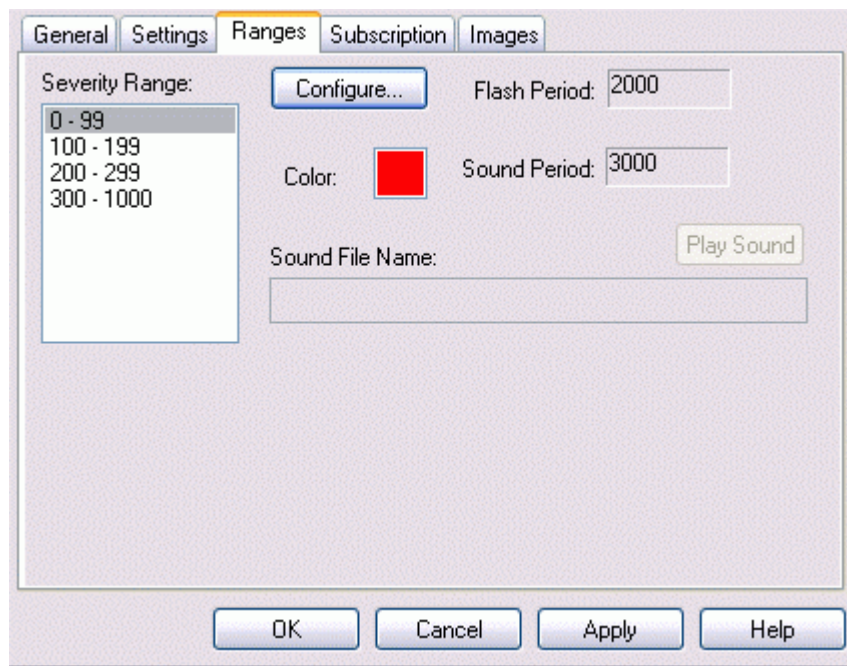
<code>x=like({{Node}}, "Alpha",0)</code>	Only messages from Node Alpha will be displayed.
<code>x=like({{Attribute 1}}, "Plant Area 1",0)</code> <code>x={{Attribute 2}} == 40.</code>	Tag values Attribute 1 through Attribute 20 are server specific. They can be strings, numeric values, or arrays. The most common array usage is for Areas. When dealing with areas, use the like function.
<code>x=1.</code>	Filter displays all messages.
<code>x=0.</code>	Filter does not display any messages.

All filters resolve to TRUE or FALSE. All nonzero values resolve to TRUE.

For more information, please see the AlarmWorX32 Server documentation.

Ranges Tab

The **Ranges** tab of the **AlarmWorX32 Indicator ActiveX Properties** dialog box, shown below, allows you to configure colors, flash periods, and sounds for the Alarm Indicator. The **Severity Range** field enables you to set Alarm Indicator parameters based on the severity level of alarms. The severity values range from 0-1000. Each severity range is configured separately. For example, the figure below shows the default configuration for the severity range 0-99. Each severity range has a different configuration. Thus, the next severity range (e.g. 100-199), will have a different configuration.



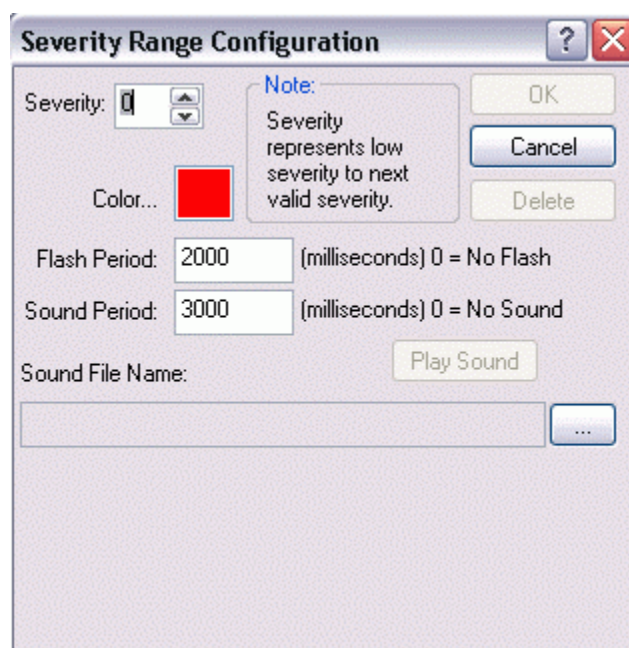
AlarmWorX32 Indicator ActiveX Control Properties: Ranges Tab

Configuring Severity Ranges for the Alarm Indicator

To configure a severity range:

1. Under the **Severity Range** field of the **Ranges** tab, select a severity range to configure.
2. Click the **Configure** button to open the **Severity Range Configuration** dialog box, shown below. The **Severity** field displays the lower limit for the selected severity range.

Note: You can customize the severity ranges by changing the "low limit" value in the **Severity** field. The new range will be displayed in the **Severity Range** field on the **Ranges** tab. The default severity value '0' cannot be deleted.



Severity Range Configuration Dialog Box

3. Choose a color that will indicate the severity level when the Alarm Indicator flashes. Click on the color palette to select a color.
4. Set the value for the flash period. The **Flash Period** is the frequency at which the Alarm Indicator will flash (in milliseconds) when an alarm occurs within the specified severity range.
5. Set the value for the sound period. The **Sound Period** is the frequency at which the Alarm Indicator will make a sound (in milliseconds) when an alarm occurs within the specified severity range.
6. Select a sound that will indicate alarms that fall under the severity range. Click the ... button under the **Sound File Name** field to browse for a sound (.wav) file. To test the selected sound file, click the **Play Sound** button.

Note: Flashing and sound will occur for unacknowledged alarms in order of priority from the highest to lowest severity level.

7. Click **OK** to apply the changes. The new configuration settings will be shown on the **Ranges** tab.

Repeat the steps above for each severity range you wish to configure for the Alarm Indicator.

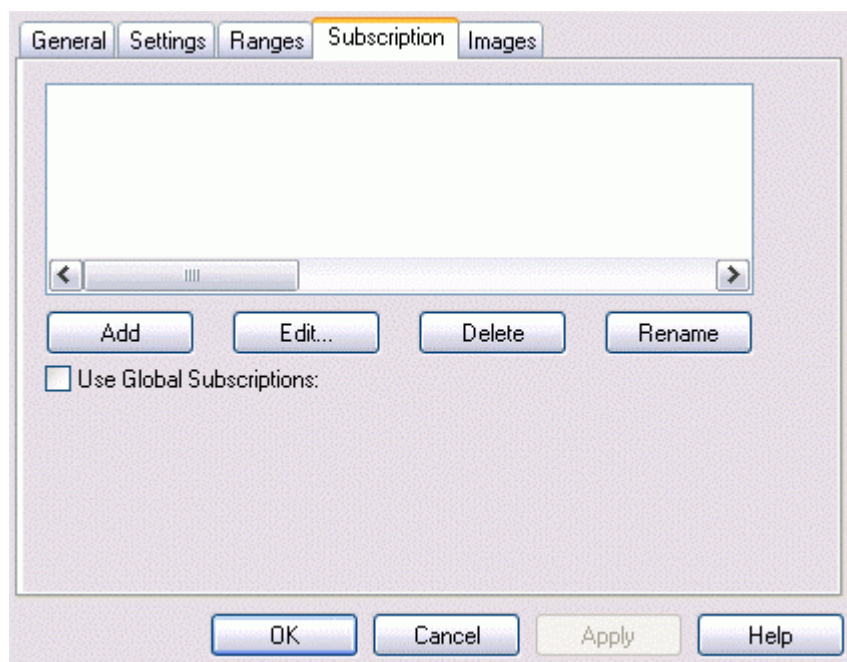
Deleting a Severity Range Configuration

To delete a severity range configuration:

1. Select the severity range for the configuration you wish to delete in the **Ranges** tab.
2. Click the **Configure** button to open the **Severity Range Configuration** dialog box.
3. Click the **Delete** button.

Subscription Tab

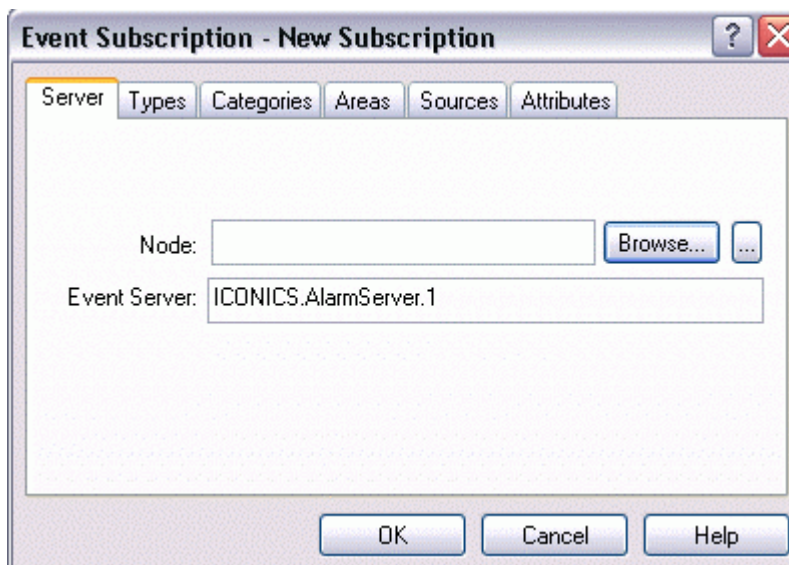
The **Subscription** tab of the **Alarm Indicator ActiveX Properties** dialog box, shown below, determines what type of OPC connection will be made. It allows you to add, rename, delete, or edit a subscription to a particular alarm. To add a new subscription, click the **Add** button.



Alarm Viewer ActiveX Properties: Subscription Tab

The subscription named "New Subscription" appears. This subscription does not contain any data, so it is necessary to immediately edit the new subscription. To do so, click the **Edit** button to open the **Event Subscription** dialog box, as shown in the figure below. It is only possible to add or edit subscriptions that are connected to active OPC Alarm and Event servers.

Note: It is possible for a server to have more than one subscription. In fact, it is a very effective way to achieve filtering.



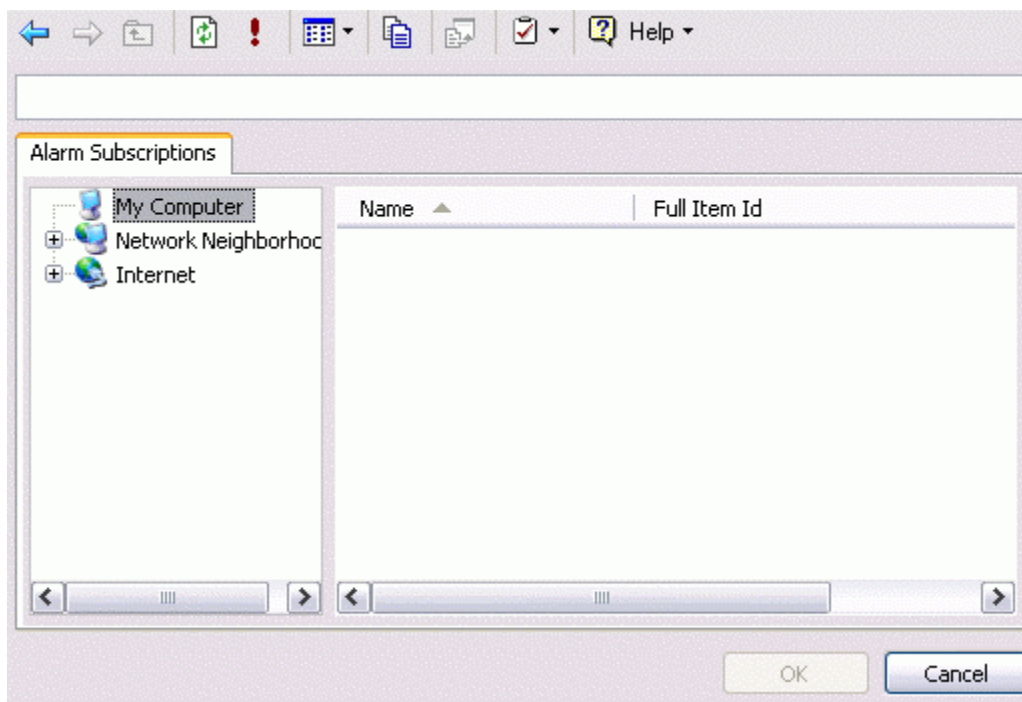
Creating an Event Subscription

The Alarm Indicator uses local subscriptions by default. You have the option of creating a global alarm subscription or a local alarm subscription. To create a global subscription, first remove all existing local subscriptions, and then check the **Use Global Subscriptions** check box on the **Subscription** tab.

Note: You must use either all global subscriptions or all local subscriptions; global and local subscriptions cannot be mixed.

If you use a global subscription, the **Unified Data Browser** dialog box opens, as shown in the figure below, where you can select a subscription from the **Alarm Subscriptions** tab, which lists all global filters configured in the Unified Data Manager. Global subscriptions are prefixed by "awxs".

For more information about subscriptions, please see the Unified Data Manager help documentation and the AlarmWorX32 Viewer ActiveX help documentation.



Unified Data Browser (Global Subscription)

Images Tab

The **Images** tab of the **AlarmWorX32 Indicator ActiveX Properties** dialog box, shown below, allows you to choose which image to use for the Alarm Indicator ActiveX. For your convenience, the Alarm Indicator provides several default large and small images to use for the ActiveX. The small images are particularly well-suited for placement of the Indicator in an area of your display where space is limited, such as the Windows taskbar.

Alternatively, you can select your own images to use for the Alarm Indicator by clicking the **Browse** button under **Preview**. This opens a dialog box that allows you to select .gif, .jpg, or .bmp image files. The **Mask Preview** is new support for the new drawing code. The mask preview is intended to show you which areas will be colored in the Alarm Indicator ActiveX. You can select your own images to use for the mask preview by clicking the **Browse** button.

Note: Although you are free to use your own images for the Alarm Indicator, it is recommended that you use the default images provided because the default images contain predefined flashing regions. If you use your own images, you will have to manually specify the flashing regions for the images.



AlarmWorX32 Indicator ActiveX Control Properties: Images Tab

Runtime Functions

When you have finished configuring the AlarmWorX32 Indicator ActiveX, you are ready to enter the Alarm Indicator into runtime mode. During runtime mode, the Alarm Indicator notifies you of alarms that are occurring in your GENESIS32 system by flashing and making sounds based on the color, sound, flashing, and severity parameters that you have set in your configuration.

To enter the ActiveX display into runtime mode from the AlarmWorX32 or TrendWorX32 container applications, select **Runtime Mode** from the **Actions** menu. To enter the display into runtime mode from GraphWorX32, click **Runtime** on the menu bar. When runtime mode starts, the **Indicator Tray Alert** in the Windows task tray flashes, as shown in the figure below.



Indicator Tray Alert

The Alarm Indicator ActiveX includes the following runtime functions:

- ToolTips
- Colors, Flashing, and Sounds
- Launching an Application During Runtime
- Turning off the Sound During Runtime
- Changing Properties During Runtime

ToolTips

When you place the mouse pointer over the Alarm Indicator ActiveX during runtime mode, you can view the ToolTip that you specified in the **General** configuration tab. In the example below, the text "Control Room" appears in the ToolTip during runtime mode.



Viewing the ToolTip in Runtime

Colors, Flashing, and Sounds

When an alarm is triggered during runtime mode, the Alarm Indicator ActiveX will flash repeatedly to visually indicate the alarm. The normal color (no alarm) and the replacement (flashing alarm) color have been specified in the **Settings** configuration tab. In the example below, the Alarm Indicator is flashing red to indicate that an alarm is occurring.



Alarm Occurring in Runtime

When an alarm occurs during runtime, the Alarm Indicator flashes and/or sounds according to the parameters that have been set in the **Ranges** configuration tab, in which the colors, flash period, and sound period have been specified for the Alarm Indicator. The sound, color, and flashing of the Alarm Indicator during runtime vary depending on the severity range of each alarm. The severity ranges have been specified in the **Severity Range Configuration** dialog box.

Each severity range has a different configuration, so the Alarm Indicator may look and sound different for each occurring alarm. For example, the following configuration might be used. For each alarm that occurs in the severity range 100-199, the Alarm Indicator will beep once every two seconds and flash blue once every two seconds. For each alarm that occurs in the severity range 200-299, the Alarm Indicator will sound a horn one time per second and flash red one time per second.

Note: Flashing and sound will occur for unacknowledged alarms in order of priority from the highest severity level to the lowest severity level.

Launching an Application During Runtime

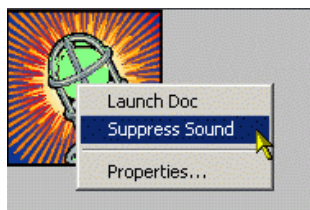
You can launch an application from the Alarm Indicator ActiveX during runtime mode. The type of application, the application file, and the method for launching the application have already been specified in the **Settings** configuration tab. Launch the application using the method you selected in the configuration (single-click, double-click, or right-click context menu). For example, suppose you have configured the Alarm Indicator to launch a document when the Alarm Indicator is right-clicked during runtime mode. In runtime mode, right-click the Indicator ActiveX, and then select **Launch Doc** from the pop-up menu, as shown below.



Launching an Application During Runtime

Turning off the Sound During Runtime

If you have configured the Alarm Indicator to make a sound when an alarm is indicated, you can turn off the sound during runtime mode using the sound suppression feature. The sound suppression period is specified in the **Settings** configuration tab. Turn off the sound using the method you selected in the configuration (single-click, double-click, or right-click context menu). For example, suppose you have configured the Alarm Indicator to suppress the sound when the Alarm Indicator is right-clicked during runtime mode. In runtime mode, right-click the Indicator ActiveX, and then select **Suppress Sound** from the pop-up menu, as shown below. The sound will be suppressed for the duration of the sound-suppression period specified in the configuration. When the sound-suppression period has elapsed, the Alarm Indicator will resume making the sound.



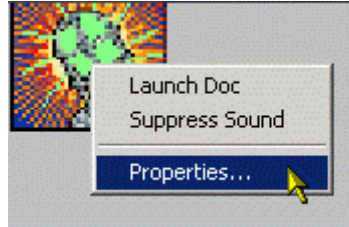
Turning off the Sound During Runtime

Global Sound Suppression

If multiple instances of the Alarm Indicator ActiveX are running concurrently and each instance is configured to produce a sound indicating an alarm, you can use the global sound suppression function to monitor the sound status of all instances. For example, suppose you have two Alarm Indicators open (Alarm Indicator 1 and Alarm Indicator 2). If Alarm Indicator 1 sounds an alarm, and global sound suppression is enabled in the **Settings** configuration tab, the sound of Alarm Indicator 2 will be temporarily suppressed until Alarm Indicator 1 stops making a sound.

Changing Properties During Runtime

During runtime mode, you can quickly and easily access the Alarm Indicator ActiveX Properties configuration dialog box by right-clicking the Alarm Indicator and selecting **Properties** from the pop-up menu, as shown below.



Accessing Configuration Properties During Runtime

AlarmWorX32 Indicator ActiveX OLE Automation Reference

To access the Automation interface from VB and VBA, AlarmWorX32 Indicator must be made available by choosing **Project - References** from the main menu in the VB or VBA development environment and selecting **AwxInd32 1.0 Type Library** in the list of available references.

Available Control Properties

Members of 'AwxIndicator':

BackColor As OLE_COLOR

Description

This is the "replacement" color of the Alarm Indicator. It is the color that you see behind the flashing color when there are alarms coming in. Property is read/write.

Remarks

Please note, the replacement color can only be a valid color that already exists in the bitmap, otherwise you will not see any change.

OLE_COLORS can be set in more than one way. You can use a VB color constant such as vbWhite, an RGB color value, or a Long integer.

Example

This example shows how to read and write the background color.

```
Dim backColor As Long
backColor = AwxIndicator1.BackColor
AwxIndicator1.BackColor = 16777215
'Or:
AwxIndicator1.BackColor = vbWhite
'Or:
AwxIndicator1.BackColor = RGB(255, 255, 255)
'(All examples make the background color white)
```

CanUserConfigure As Boolean

Description

True if configuration is enabled for the currently logged in user. Property is read-only.

Example

This example shows how to determine if user configuration is enabled.

```
MsgBox AwxIndicator1.CanUserConfigure
```

CanUserLaunchDoc As Boolean**Description**

True if launching a document is enabled for the currently logged in user. Property is read-only.

Example

This example shows how to determine if launching a document is enabled.

```
MsgBox AwxIndicator1.CanUserLaunchDoc
```

CanUserSuppressSound As Boolean**Description**

True if sound suppression is enabled for the currently logged in user. Property is read-only.

Example

This example shows how to determine if sound suppression is enabled.

```
MsgBox AwxIndicator1.CanUserSuppressSound
```

Caption As String**Description**

This is the text that appears when you hover over the Alarm Indicator with your mouse. Property is read/write.

Example

This example shows how to read and write the caption.

```
Dim caption As String  
caption = AwxIndicator1.Caption  
AwxIndicator1.Caption = "My Caption"
```

FileEnabled As Boolean**Description**

True if a configuration file is enabled. Property is read/write.

Example

This example shows how to determine if a configuration file is enabled.

```
Dim bEnabled As Boolean  
bEnabled = AwxIndicator1.FileEnabled
```

This example shows how to turn on/off a configuration file.

```
AwxIndicator1.FileEnabled = True
```

FileName As String**Description**

This is the name of the configuration file. Property is read/write.

Remarks

Use this property to load and save configuration files. First set the property to the correct path and file name, and then call either FileLoad() or FileSave().

Example

This example shows how to read and write the configuration file name.

```
Dim fileName As String  
fileName = AwxIndicator1.FileName  
AwxIndicator1.FileName = "C:\My Directory\AwxIndCfg.awi"
```

FillColor As OLE_COLOR**Description**

This is the "normal" color of the Alarm Indicator. It is the color that flashes when there are no alarms coming in. This can be any color. Property is read/write.

Remarks

OLE_COLORS can be set in more than one way. You can use a VB color constant such as vbWhite, an RGB color value, or a Long integer.

Example

This example shows how to read and write the fill color.

```
Dim fillColor As Long  
fillColor = AwxIndicator1.FillColor  
AwxIndicator1.FillColor = 16777215  
'Or:  
AwxIndicator1.FillColor = vbWhite  
'Or:  
AwxIndicator1.FillColor = RGB(255, 255, 255)  
'(All make the fill color white)
```

Filter As String

GlobalSoundSuppression As Boolean**Description**

True if global sound suppression is enabled. Property is read/write.

Example

This example shows how to determine if global sound suppression is enabled.

```
Dim bEnabled As Boolean  
bEnabled = AwxIndicator1.GlobalSoundSuppression
```

This example shows how to turn on/off global sound suppression.

```
AwxIndicator1.GlobalSoundSuppression = True
```

IgnoreNormalUnacked As Boolean**LaunchDoc As String****Description**

This is the file name of the Word document to be launched from the Alarm Indicator. Property is read/write.

Example

This example shows how to read and write the launch document file name.

```
Dim launchDoc As String  
launchDoc = AwxIndicator1.LaunchDoc  
AwxIndicator1.LaunchDoc = "C:\My Directory\MyWordDoc.doc"
```

LaunchDocVia As tagINVOKE_VIA**Description**

This property corresponds to a constant that determines which method can be used to launch a document. Property is read/write.

Remarks

The tagINVOKE_VIA constant values are:

```
SingleClick = 1  
DoubleClick = 2  
ContextMenu = 3
```

Example

This example shows how to read and write the document launching method.

```
Dim n As tagINVOKE_VIA  
n = AwxIndicator1.LaunchDocVia  
AwxIndicator1.LaunchDocVia = ContextMenu
```

Mouselcon As IPictureDisp**Picture As IPictureDisp**

Description

This is the bitmap object that is displayed on the Alarm Indicator.

Example

This example shows how to grab the Picture object.

```
Dim pic As IPictureDisp  
Set pic = AwxIndicator1.Picture
```

RangeCollection As Object**Description**

This object returns a RangeCollection, which is a collection of Range objects that determine the Alarm Indicator's color depending on the severity of the alarm. There is only one RangeCollection per Alarm Indicator instance; however, there can be many Range objects in each RangeCollection.

Remarks

The RangeCollection object has 3 properties and 2 methods:

Clone As Object
Count As Long
Item As Object

AddItem(NewItem As Object)
DeleteItem(ToDelete As Object)

Example

This example shows how to grab the RangeCollection object.

```
Dim rc As RangeCollection  
Set rc = AwxIndicator1.RangeCollection
```

SuppressSoundPeriod As Long**Description**

This specifies the number of seconds the sound will be suppressed when you invoke the "Suppress Sound" action. Property is read/write.

Example

This example shows how to read and write the sound suppression period.

```
Dim n As Long  
n = AwxIndicator1.SuppressSoundPeriod  
AwxIndicator1.SuppressSoundPeriod = n + 1
```

SuppressSoundVia As tagINVOKE_VIA**Description**

This property corresponds to a constant that determines which method can be used to suppress a sound.

Remarks

The tagINVOKE_VIA constant values are:

SingleClick = 1
DoubleClick = 2
ContextMenu = 3

Example

This example shows how to read and write the sound suppression method.

```
Dim n As tagINVOKE_VIA  
n = AwxIndicator1.SuppressSoundVia  
AwxIndicator1.SuppressSoundVia = SingleClick
```

URLPath As String**Description**

This specifies the URL path of a configuration file. Property is read/write.

Example

This example shows how to read and write the URL path.

```
Dim path As String  
path = AwxIndicator1.URLPath  
AwxIndicator1.URLPath = "http://www.iconics.com/awxindcfg.awi"
```

Members of 'Range':**Clone As Object****Description**

This is a copy of the Range object. Property is read-only.

Example

This example shows how to grab a copy of the Range object.

```
Dim range As Range  
range = AwxIndicator1.RangeCollection.Item(0).Clone
```

Color As Long

Description

This is the color that will flash when alarms within this particular severity range are coming in. Property is read/write.

Remarks

This property accepts all valid OLE_COLORS.

Example

This example shows how to read and write the range color.

```
Dim color As Long
color = AwxIndicator1.RangeCollection.Item(0).Color
AwxIndicator1.RangeCollection.Item(0).Color = 16777215
'Or:
AwxIndicator1.RangeCollection.Item(0).Color = vbWhite
'Or:
AwxIndicator1.RangeCollection.Item(0).Color = RGB(255, 255, 255)
'(All make the range's color white)
```

FlashPeriod As Long

Description

This is the duration (in milliseconds) of each flash for that particular severity range. Property is read/write.

Example

This example shows how to read and write the flash period.

```
Dim n As Long
n = AwxIndicator1.RangeCollection.Item(0).FlashPeriod
AwxIndicator1.RangeCollection.Item(0).FlashPeriod = 200
```

IsWav As Boolean

Description

Returns true if the current Range object has a valid file associated with it. Returns false if it cannot locate the file on the system.

Example

This example shows how to read the IsWav property.

```
Dim bEnabled As Boolean
bEnabled = AwxIndicator1.RangeCollection.Item(0).IsWav
```

PlayPeriod As Long

Description

This specifies the number of milliseconds between each time the .wav file is played. Property is read/write.

Remarks

If the .wav file that you are using is longer than the play period, the sound will not cut off; it will immediately restart once the .wav file is done playing.

Example

This example shows how to read and write the play period.

```
Dim n As Long  
n = AwxIndicator1.RangeCollection.Item(0).PlayPeriod  
AwxIndicator1.RangeCollection.Item(0).PlayPeriod = 10000
```

Severity As Long

Description

This is the severity level of the current range object. Property is read/write.

Remarks

If you change the severity level of a range to be lower than the severity level of any ranges below it, it will delete those ranges and your current range will encompass the entire severity level of the previous ranges combined. It will never create a brand new severity level.

Example

This example shows how to read and write the severity level.

```
Dim n As Long  
n = AwxIndicator1.RangeCollection.Item(0).Severity  
AwxIndicator1.RangeCollection.Item(0).Severity = 500
```

WavFileName As String

Description

This is a property that you set before loading or playing a .wav file for a particular range. Property is read/write.

Remarks

Use this property to load and play .wav files. First set the property to the correct path and file name, and then call either LoadWavFile() or PlayWav().

Example

This example shows how to read and write the .wav file name.

```
Dim wavFile As String  
wavFile = AwxIndicator1.RangeCollection.Item(0).WavFileName
```

```
AwIndicator1.RangeCollection.Item(0).WavFileName = "C:\wav_file.wav"
```

Members of 'RangeCollection':

Clone As Object

Description

This is a copy of the RangeCollection object. Property is read-only.

Example

This example shows how to grab a copy of the RangeCollection object.

```
Dim rc As RangeCollection  
rc = AwIndicator1.RangeCollection.Clone
```

Count As Long

Description

This is the number of Range objects in the RangeCollection. (Read-only).

Example

This example shows how to read the count.

```
Dim n As Long  
n = AwIndicator1.RangeCollection.Count
```

Item(Index As Long) As Object

Description

This corresponds to each Range object within the RangeCollection. Item is an array of objects. Property is read-only.

Remarks

The Item array corresponds to the Range objects in reverse order (or by decreasing severity). The first array index is 0 and it goes up to the RangeCollection Count – 1.

Example

This example shows how to grab a Range object from the RangeCollection Item array.

```
Dim r As Range  
Set r = AwIndicator1.RangeCollection.Item(0)  
'This grabs the Range object of highest severity
```

Available Control Methods

Members of 'AwxIndicator':

Sub DoContextMenu()

Description

Launches the context menu so that a user can then click on an option.

Example

This example shows how to launch the context menu.

AwxIndicator1.DoContextMenu

Sub DoLaunchDoc()

Description

Launches the document specified in the LaunchDoc property.

Example

This example shows how to perform the LaunchDoc action.

AwxIndicator1.DoLaunchDoc

Sub DoLeftMouseButton()

Description

Simulates a left mouse button click, thus performing whichever action (if any) is associated with the left mouse button.

Example

This example shows how to simulate a left mouse button click.

AwxIndicator1.DoLeftMouseButton

Sub DoLeftMouseDoubleClick()

Description

Simulates a left mouse button double-click, thus performing whichever action (if any) is associated with double-clicking the left mouse button.

Example

This example shows how to simulate a left mouse button double-click.

AwxIndicator1.DoLeftMouseDoubleClick

Sub DoSuppressSound()**Description**

Performs the sound suppression action. This will silence the Alarm Indicator for the number of seconds specified in the SoundSuppressionPeriod property.

Example

This example shows how to suppress the sound.

```
AwxIndicator1.DoSuppressSound
```

Sub FileLoad()**Description**

Loads the configuration file that is currently in the FileName property.

Remarks

You must set the FileName property and make sure that FileEnabled is set to "True" before calling this method or it will have no effect.

Example

This example shows how to load a configuration file.

```
AwxIndicator1.FileName = "C:\My Directory\AwxIndCfg.awi"  
AwxIndicator1.FileEnabled = True  
AwxIndicator1.FileLoad  
'Loads the file AwxIndCfg.awi
```

Sub FileSave()**Description**

Saves the configuration file that is currently in the FileName property.

Remarks

You must set the FileName property and make sure that FileEnabled is set to "True" before calling this method or it will have no effect.

Example

This example shows how to save a configuration file.

```
AwxIndicator1.FileName = "C:\My Directory\AwxIndCfg1.awi"  
AwxIndicator1.FileEnabled = True  
AwxIndicator1.FileSave  
'Saves the current configuration out to the file AwxIndCfg1.awi
```

Sub ForceConfigMode(Force As Boolean)**Description**

Allows the Alarm Indicator to exit runtime and enter configuration mode while the container remains in runtime mode.

Example

This example shows how to force configuration mode.

```
AwxIndicator1.ForceConfigMode(True)
```

GetAlarmOleBlob() As Long**Sub SetAlarmOleBlob(pBlob As Long)****Description**

These above two methods are part of AlarmOLE.

ReplaceFilePath(OldSubstring As String, NewSubstring As String) As Long**Description**

Works over path name parameters of pick dynamics. Returns 0 on success and HRESULT when something fails.

Example

```
'similar to ReplaceTag
'VBA example, works only in configure mode
'replaces file path in whole display

Dim Status As Long
Status = ThisDisplay.ReplaceFilePath("C:\Windows\Temp",
"D:\Temp")
'if Status <> 0 then there was no replacement performed or
an error occurs
If Status <> 0 Then
    MsgBox "No replacements"
Else
    MsgBox "Tags replaced"
End If
```

ReplaceHost(OldHostName As String, NewHostName As String) As Long

Description

Works over both data source tags and path name attributes and replaces node name **substring** within URL path only. Returns 0 on success and HRESULT when something fails.

Example

```
'similar to ReplaceTag
'VBA example, works only in configure mode
'replaces host name in whole display

Dim Status As Long
Status = ThisDisplay.ReplaceHost("Host1", "Host2")
'if Status <> 0 then there was no replacement performed or
an error occurs
If Status <> 0 Then
    MsgBox "No replacements"
Else
    MsgBox "Tags replaced"
End If
```

ReplaceHostEx(OldHostNameSubstring As String, NewHostNameSubstring As String, MatchCase As Long, MatchWholeWord As Long) As Long

Description

Works over both data source tags and path name attributes and replaces node name **substring** within URL path only, and supports case-sensitivity, wildcard strings and MatchWholeWord flag. Returns 0 on success and HRESULT when something fails.

Example

```
'similar to ReplaceTag
'VBA example, works only in configure mode
'replaces host name in whole display, regarding "case" and
"whole words" options

Dim Status As Long
Status = ThisDisplay.ReplaceHostEx("Host1", "Host2", True,
True)
'if Status <> 0 then there was no replacement performed or
an error occurs
If Status <> 0 Then
    MsgBox "No replacements"
Else
    MsgBox "Tags replaced"
End If
```

ReplaceStringInString(StingToReplace As String, OldSubstring As String, NewSubstring As String) As String

Description

Replaces **OldSubstring** with **NewSubstring** inside (within) StingToReplace. Returns 0 on success and HRESULT when something fails.

ReplaceTag(OldSubstring As String, NewSubstring As String) As Long

Description

Works over data source tags in dynamic actions of all display objects. Replaces **OldSubstring** with **NewSubstring** in certain situations. Returns 0 on success and HRESULT when something fails.

Typical use of this method:

Example

```
'VBA example, works only in configure mode
'replaces tag name in whole display

Dim Status As Long
Status = ThisDisplay.ReplaceTag("gfwsim.ramp.float",
"gfwsim.sine.double")
'if Status <> 0 then there was no replacement
'performed or an error occurs
If Status <> 0 Then
    MsgBox "No replacements"
Else
    MsgBox "Tags replaced"
End If
```

Members of 'Range':

Sub LoadWavFile()

Description

Sets the .wav file to be played when that specific range is in alarm condition. Uses the .wav file that is currently residing in the WavFileName property.

Remarks

You must set the WavFileName property before calling this method or it may return an error.

Example

This example shows how to load a .wav file.

```
ThisDisplay.AwxIndicator1.RangeCollection.Item(0).WavFileName =
"C:\Program Files\ICONICS\GENESIS32\Bin\MMXSounds\DRUMS.WAV"
ThisDisplay.AwxIndicator1.RangeCollection.Item(0).LoadWavFile
ThisDisplay.AwxIndicator1.RangeCollection.Item(0).PlayWav
```

Sub PlayWav()**Description**

Plays the .wav file that is currently residing in the WavFileName property.

Remarks

You must set the WavFileName property before calling this method or it may return an error.

Example

This example shows how to play a .wav file.

```
ThisDisplay.AwxIndicator1.RangeCollection.Item(0).WavFileName =
"C:\Program Files\ICONICS\GENESIS32\Bin\MMXSounds\DRUMS.WAV"
ThisDisplay.AwxIndicator1.RangeCollection.Item(0).LoadWavFile
ThisDisplay.AwxIndicator1.RangeCollection.Item(0).PlayWav
```

Members of 'RangeCollection':**Sub AddItem(NewItem As Object)****Description**

Adds a new item into the RangeCollection of the Alarm Indicator.

Remarks

The easiest way to do this is to clone an already existing Range object and then modify its properties to fit your new Range needs.

Example

This example shows how to clone an existing Range object and add a modified version of it to the RangeCollection.

```
Dim newRange As Range
Set newRange = ThisDisplay.AwxIndicator1.RangeCollection.Item(0).Clone
newRange.Color = vbBlue
newRange.FlashPeriod = 300
newRange.PlayPeriod = 6000
newRange.Severity = 500
ThisDisplay.AwxIndicator1.RangeCollection.AddItem newRange
ThisDisplay.AwxIndicator1.RangeCollection.DeleteItem newRange
```

Sub DeleteItem(ToDelete As Object)

Description

Deletes an item from the RangeCollection of the Alarm Indicator.

Example

```
Dim newRange As Range
Set newRange = ThisDisplay.AwxIndicator1.RangeCollection.Item(0).Clone
newRange.Color = vbBlue
newRange.FlashPeriod = 300
newRange.PlayPeriod = 6000
newRange.Severity = 500
ThisDisplay.AwxIndicator1.RangeCollection.AddItem newRange
ThisDisplay.AwxIndicator1.RangeCollection.DeleteItem newRange
```