



Your comments and suggestions on the operation of this software are welcome. Please address them to:

ICONICS

100 Foxborough Blvd.

Foxborough, MA 02035

Tel: 508-543-8600

Fax: 508-543-1503

E-mail: support@iconics.com

Web: www.iconics.com

© ICONICS, Inc. All rights reserved.

Contents

Introduction to the AlarmWorX32 Indicator ActiveX

AlarmWorX32 Indicator ActiveX Properties

Runtime Functions

Automation Interfaces

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

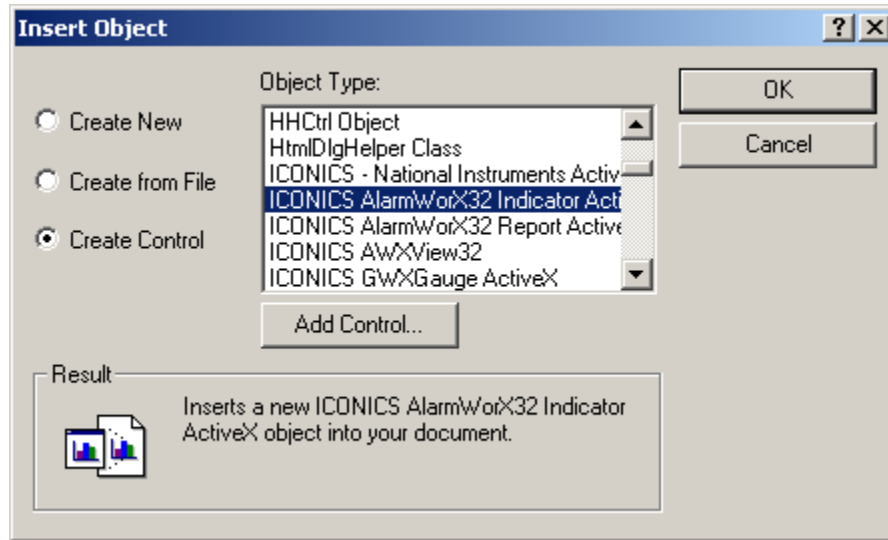
The Alarm Indicator includes the following features:

- 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.



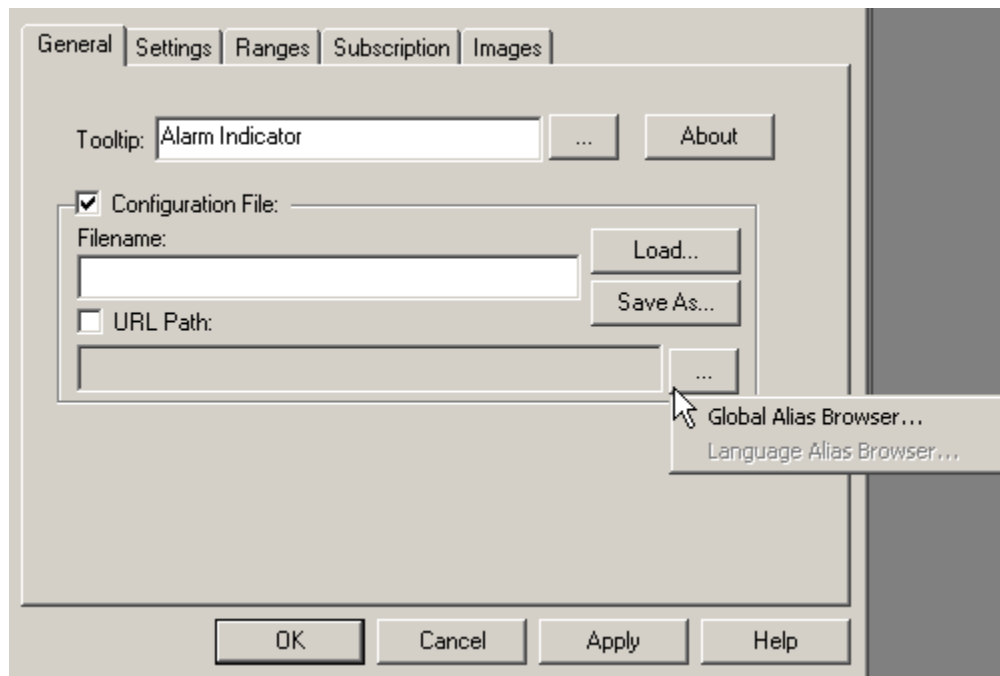
Insert Object Dialog Box

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

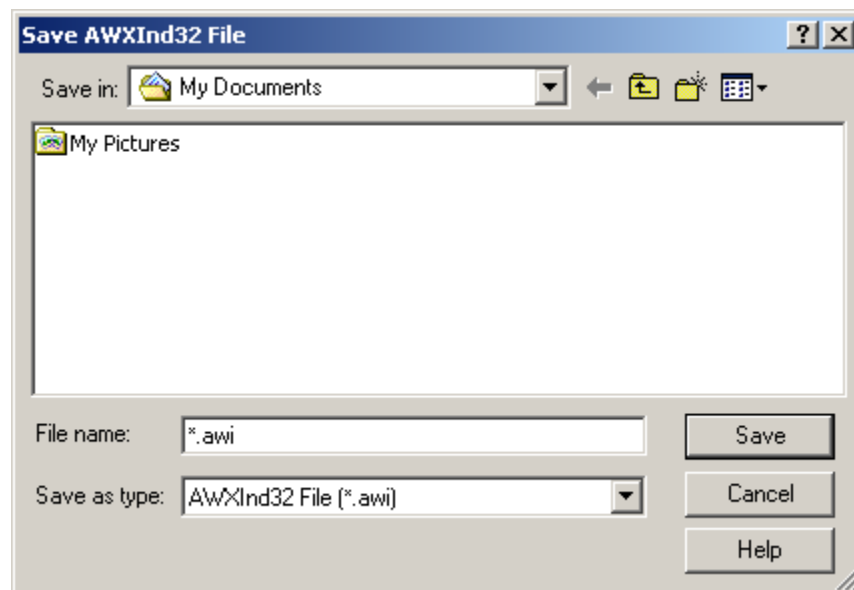
The **General** tab of the **AlarmWorX32 Indicator ActiveX Properties** dialog box, shown below, allows you to specify the text for the ToolTip that appears when the mouse pointer is placed over the Indicator ActiveX during runtime mode. Simply enter the desired text in the **ToolTip** field. In the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

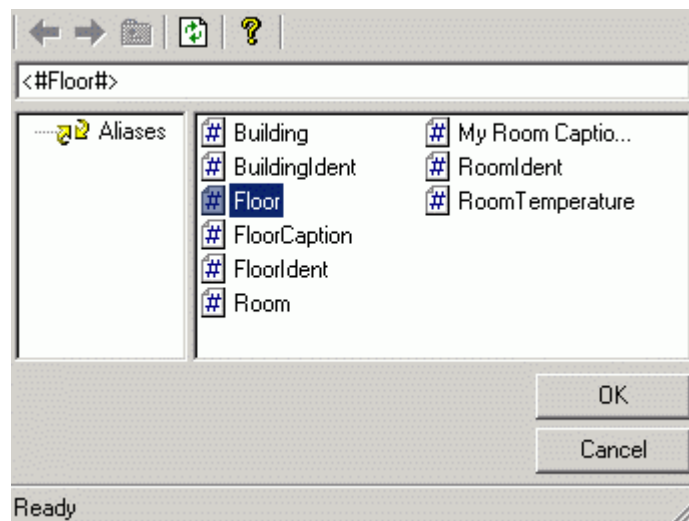
You can also open an existing configuration file by clicking the **Load** button on the **General** tab and browsing for the file. If **Configuration File** is selected, the viewer will attempt to load from the file listed in Filename. If both **Configuration File** and **URL Path** are selected, the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

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 Report ActiveX will attempt to load from the file listed under **File name**. If **URL Path** is checked, the ActiveX will use the specified network 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.

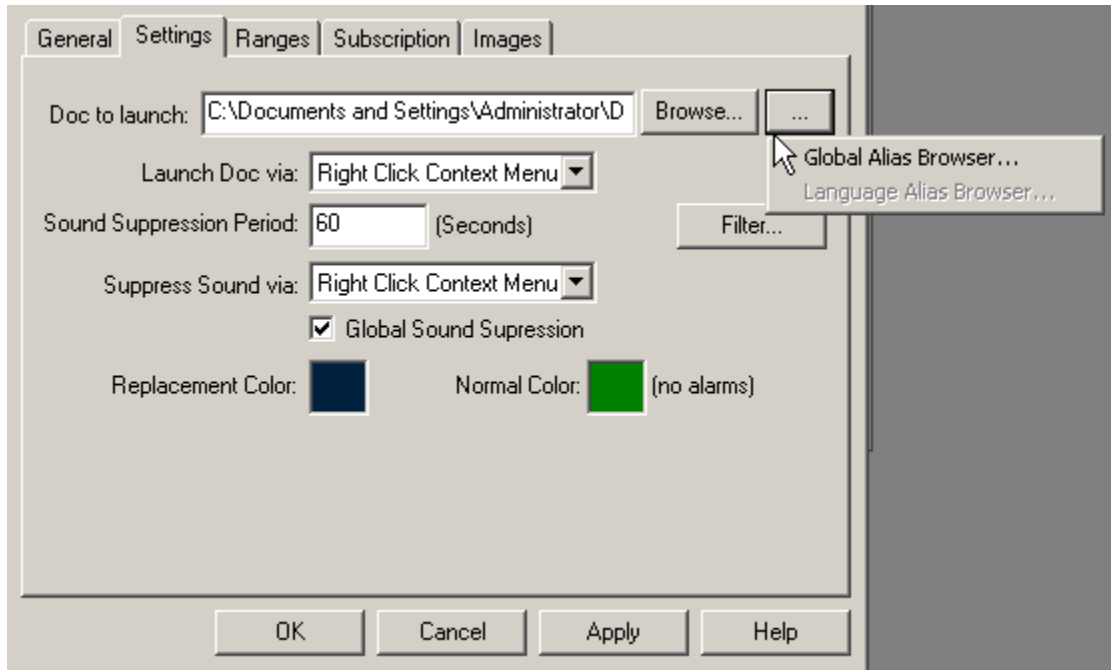
You can also select aliases to use for the configuration file. Clicking the ... button and selecting **Global Alias Browser** from the pop-up menu opens the Global Alias Browser, as shown in the figure below. Select a global alias from the Global Alias Browser, which includes all global aliases in the global aliases 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 Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.



Selecting an Alias From the Global Alias Browser

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

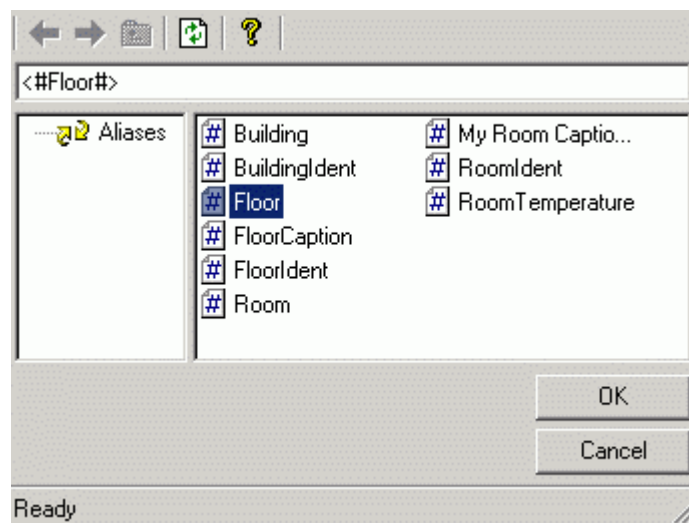
Launching an Application From the Indicator ActiveX

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 Alias Browser, as shown in the figure below. Select a global alias from the Global Alias Browser, which includes all global aliases in the global aliases 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 Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.



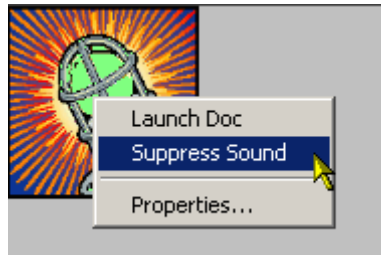
Selecting an Alias From the Global Alias Browser

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.

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)

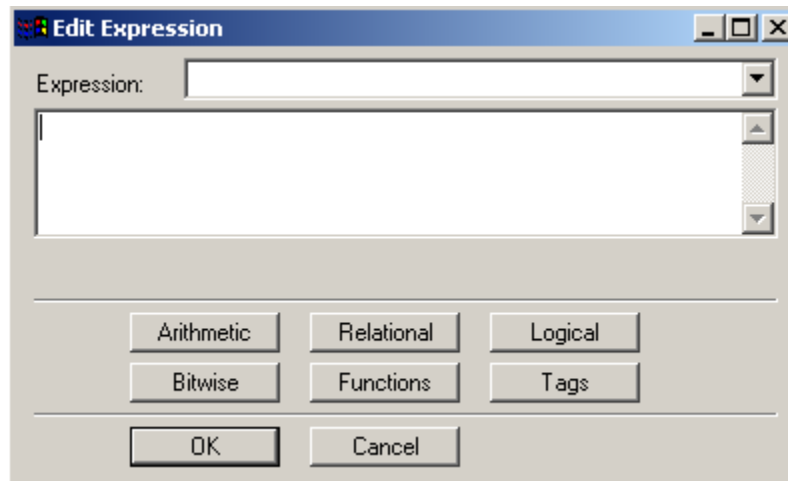
Setting Up Filters

Clicking the **Filter** button on the **Settings** tab opens the **Edit Expression** dialog box. For more information, please see the **Expression Editor** section.

Expression Editor

Clicking the **Filter** button on the **Settings** tab opens the **Edit Expression** dialog box, shown below, which contains the following options that enable you to set up alarm filters:

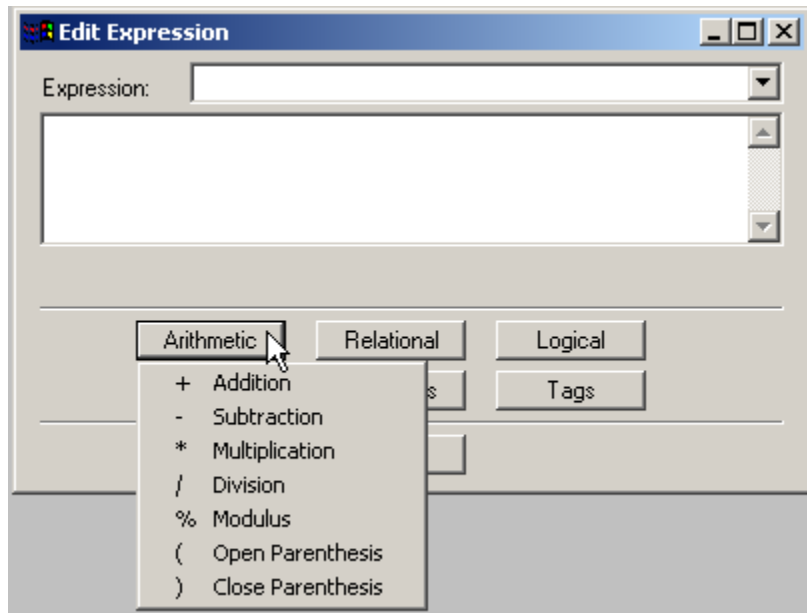
- **Arithmetic**
- **Relational**
- **Logical**
- **Bitwise**
- **Functions**
- **Tags**



Edit Expression Dialog Box

Arithmetic

The **Arithmetic** menu symbols are shown in the figure below.



Arithmetic Symbols

The symbols '+', '-', '*', '/' and '%' use the following format:

expression :: parameter **symbol** parameter

Where

Parameter	A local variable, an OPC tag, a constant, or another expression
Symbol	+ or - or * or / or %

Result

The expression results in a number of any type (float, long, etc.).

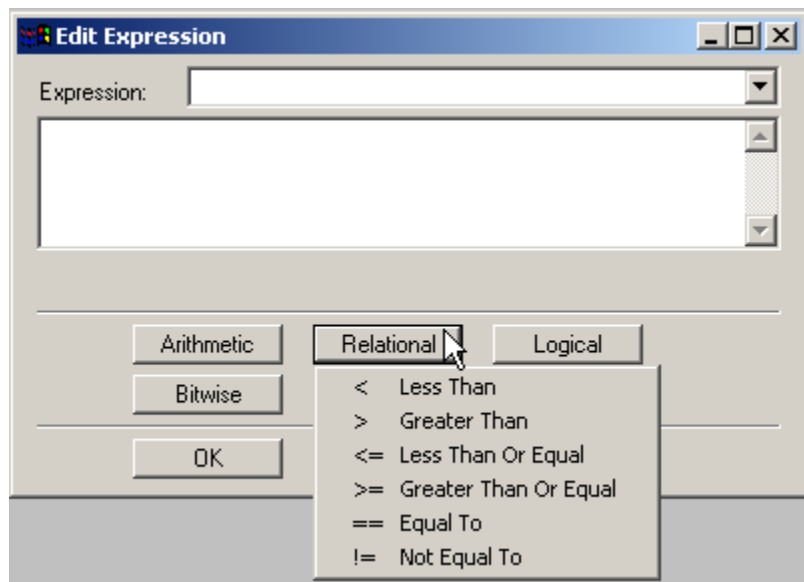
Examples

Symbol	Description	Example	Result
--------	-------------	---------	--------

Symbol	Description	Example	Result
+	Addition	~~var1~~ + ~~var2~~	8+3 = 11
-	Subtraction	~~var1~~ - ~~var2~~	8-3 = 5
*	Multiplication	~~var1~~ * ~~var2~~	8*3 = 24
/	Division	~~var1~~ / ~~var2~~	8/3 = 2.66667
%	Calculates the remainder after division	~~var1~~ % ~~var2~~	8%3 = 2
(and)	Gives precedence to parts of the calculation	~~var1~~ / (~~var2~~ + ~~var3~~)	8/(3+2) = 1.6

Relational

The **Relational** menu symbols are shown in the figure below.



Relational Symbols

The symbols '<', '>', '<=', '>=', '==' and '!=' use the following format:

expression :: parameter **symbol** parameter

Where

Parameter	A local variable, an OPC tag, a constant, or another expression
Symbol	< or > or <= or >= or == or !=

Result

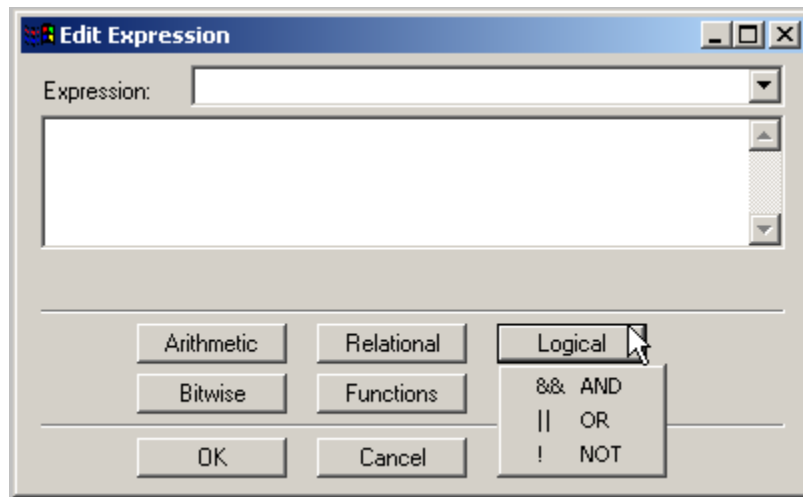
The expression results in a Boolean value (0 or 1).

Examples

Symbol	Description	Example	Result
<	Less than	~~var1~~ < ~~var2~~	8<3 = 0
>	Greater than	~~var1~~ > ~~var2~~	8>3 = 1
<=	Less than or equal to	~~var1~~ <= ~~var2~~	8<=3 = 0
>=	Greater than or equal to	~~var1~~ >= ~~var2~~	8>=3 = 1
==	Equal to	~~var1~~ == ~~var2~~	8==3 = 0
!=	Not equal to	~~var1~~ != ~~var2~~	8!=3 = 1

Logical

The **Logical** menu symbols are shown in the figure below.



Logical Symbols

The symbols '&&' and '||' use the following format:

expression :: parameter **symbol** parameter

The symbol '!' uses the following format:

expression :: **symbol** parameter

Where

Parameter	A local variable, an OPC tag, a constant, or another expression
Symbol	&& or or !

Result

The expression results in a Boolean value (0 or 1).

Truth table

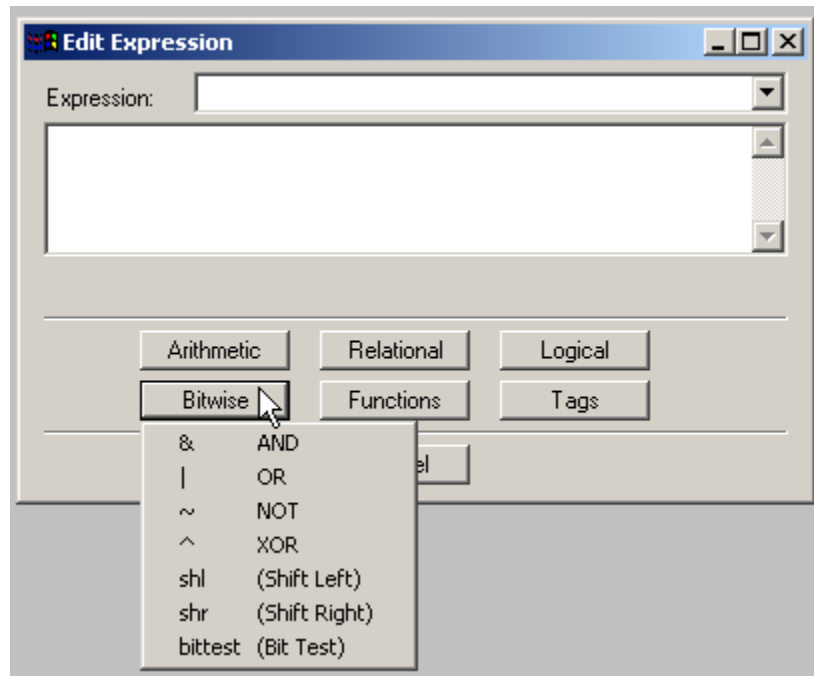
~~var1~~	0		not 0	
~~var2~~	0	not 0	0	not 0
~~var1~~ && ~~var2~~	0	0	0	1
~~var1~~ ~~var2~~	0	1	1	1
!~~var1~~	1	1	0	0

Examples

Symbol	Description	Example	Result
&&	And	~~var1~~ && ~~var2~~	8 && 3 = 1
	Or	~~var1~~ ~~var2~~	8 3 = 1
!	Not	!~~var1~~	!8 = 0

Bitwise

The **Bitwise** menu symbols are shown in the figure below.



Bitwise Symbols

The symbols '&', '|', and '^' of the bitwise group use the following format:

expression :: parameter **symbol** parameter

The symbol '~' of the logical group uses the following format:

expression :: **symbol** parameter

The symbols 'shl' and 'shr' of the bitwise group use the following format:

expression :: **symbol** (value, shift by)

Where

Parameter	A local variable, an OPC tag, a constant, or another expression
Symbol	&& or or ^ or shl or shr or ~

Result

The expression results in a number when the parameters used contain numbers.

Bit Table

	Binary (Decimal)	Binary (Decimal)
~~var1~~	0000.0000.0000.1000 (8)	0000.0000.0110.0000 (96)
~~var2~~	0000.0000.0000.1010 (10)	0000.0000.0000.1000 (8)
~~var1~~ & ~~var2~~	0000.0000.0000.1000 (8)	0000.0000.0000.0000 (0)
~~var1~~ ~~var2~~	0000.0000.0000.1010 (10)	0000.0000.0110.1000 (104)
~~var1~~ ^ ~~var2~~	0000.0000.0000.0010 (2)	0000.0000.0110.1000 (104)
shl (~~var1~~,3)	0000.0000.0100.0000 (64)	0000.0011.0000.0000 (768)
shr (~~var1~~,3)	0000.0000.0000.0001 (1)	0000.0000.0000.1100 (12)
~(~~var1~~)	1111.1111.1111.0111 (-9)	1111.1100.1111.1111 (-97)
bittest(~~var1~~,3)	0000.0000.0000.0001 (1)	0000.0000.0000.0000 (0)

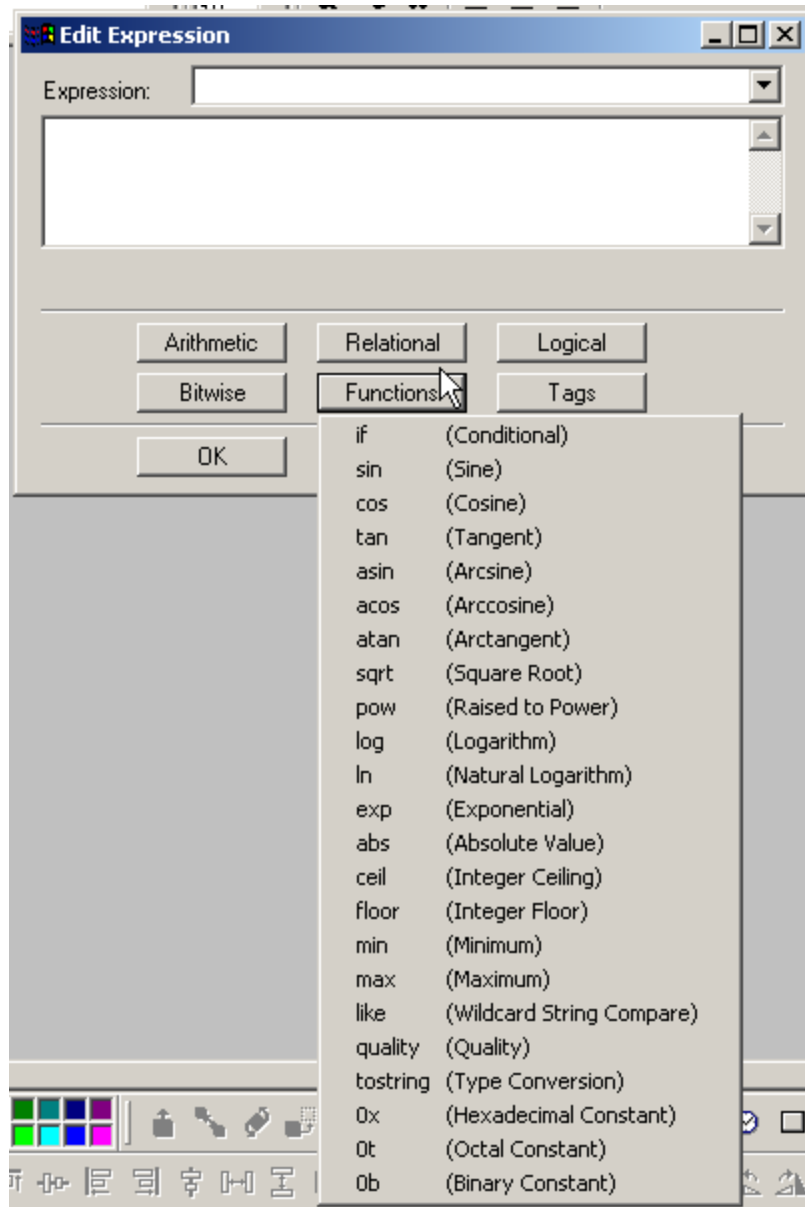
Examples

Symbol	Description	Example	Result
&	Bit And	~~var1~~ & ~~var2~~	8 & 3 = 0
	Bit Or	~~var1~~ ~~var2~~	8 3 = 11
^	Bit eXclusive Or	~~var1~~ ^ ~~var2~~	8^3=11
shl	Bit shift left	shl(~~var1~~,3)	8<<3=64
shr	Bit shift right	shr(~~var1~~,3)	8>>3=1
~	Not (two's complement)	~(~~var1~~)	!8 = -9
bittest	Bit Test	bittest (5 , 0)	1

Note: The bittest function requires you to specify the position of the bit to be tested. You must indicate that it starts from 0. In other words, a bit position of "0" indicates the "less significant" bit.

Functions

The **Functions** menu options are shown in the figure below.



Functions Menu Options

The symbols 'sin', 'asin', 'cos', 'acos', 'tan', 'atan', 'log', 'ln', 'exp', 'sqrt', 'abs', 'ceil', and 'floor' use the following format:

expression :: **symbol** (parameter)

The symbols 'pow', 'min', and 'max' use the following format:

expression :: **symbol** (parameter,parameter)

The symbol 'if' uses the following format:

expression :: **symbol** (parameter,parameter,parameter)

Where

Parameter	A local variable, an OPC tag, a constant, or another expression
Symbol	sin, asin, cos, acos, tan, atan, log, ln, exp, sqrt, abs, ceil, floor, min, max, pow, or if

Result

The expression results in a number.

Examples

Symbol	Description	Example	Result
sin	sine of an angle in radians	sin(~var1~)	sin(0.785)=0.71
cos	cosine of an angle in radians	cos(~var1~)	cos(0.785)=0.71
tan	tangent of an angle in radians	tan(~var1~)	tan(0.785)=1.0
asin	arc sine returns an angle in radians	asin(~var1~)	asin(0.5)=0.52
acos	arc cosine returns an angle in radians	acos(~var1~)	acos(0.5)=1.05
atan	arc tangent returns an angle in radians	atan(~var1~)	atan(1)=0.785
sqrt	Returns the square root	sqrt(~var1~)	sqrt(100)=10

Symbol	Description	Example	Result
pow	Returns value 1 raised to the power value 2	pow(~~var1~~,~~var2~~)	pow(100,1.5)=1000
log	10 based logarithm	log(~~var1~~)	log(100)=2
ln	e based logarithm	ln(~~var1~~)	ln(7.389)=2
exp	Exponential	exp(~~var1~~)	exp(2)=7.389
abs	Absolute value	abs(~~var1~~)	abs(-1)=1
ceil	Integer ceiling	ceil(~~var1~~)	ceil(7.39)=8
floor	Integer floor	floor(~~var1~~)	floor(7.39)=7
min	Lowest value of two	min(~~var1~~,~~var2~~)	min(10,5)=5
max	Highest value of two	max(~~var1~~,~~var2~~)	min(10,5)=10
if	Conditional statement	if(~~var1~~<~~var2~~,~~var1~~,~~var2~~)	if(5<8,5,8)=5
like	Wildcard string compare	Like(string, pattern, casesensitive')	
quality	Quality of tag or expression	See below.	See below.
tostring	Type conversion	See below.	See below.
0x	Hexadecimal constant	x=0x11	17
0t	Octal constant	x=0t11	9
0b	Binary constant	x=0b11	3

Note: For the like operator: "string" equals the string to search in; "pattern" equals the string to search for (can include wildcards); nonzero for case-sensitive search; zero for case-insensitive search. String syntax is \$"string"\$.

You can use these special characters in pattern matches in string:

- ? Any single character.
- Zero or more characters.

- # Any single digit (0-9).
- [charlist] Any single character in charlist.
- [!charlist] Any single character not in charlist.

Quality

The **quality** option on the **Functions** menu of the **Expression Editor** is used to evaluate the quality of an OPC tag or an expression.

The following general syntax is used for quality expressions:

x=quality(expression)

Note: The "(expression)" can also be a simple expression composed of a single tag.

The **quality** function returns the OPC quality of the string between parentheses as one of the following results:

- 192: quality is GOOD
- 64: quality UNCERTAIN
- 0: quality BAD

Note: The OPC Foundation establishes the value ranges for quality. There are actually varying degrees of quality:

- GOOD: 192-252
- UNCERTAIN: 64-191
- BAD: 0-63

For more information, refer to the *OPC Data Access Custom Interface Standard* available for download at the OPC Foundation's Web site, www.opcfoundation.org/.

Example Quality Expression

Expression	Result
x=quality({{ICONICS.Simulator.1\SimulatePLC.PumpStatus}})	192 (Quality GOOD)

The quality of an expression is determined through the evaluation of each single tag in the expression. Thus, if you have multiple tags in an expression (and each tag has a different quality), the result of the expression (i.e. 192 [GOOD], 64 [BAD], or 0 [UNCERTAIN]) corresponds to the quality of the tag with the lowest quality. If an expression contains a conditional statement (e.g. if, then, or else), then the result of the expression is affected only by the quality of the branch being executed.

Consider the following sample expression:

x= if (quality({Tag1}) == 192, {Tag1}, {Tag2})

This expression can be read as follows:

"If the quality of Tag1 is GOOD (i.e. 192), then the expression result (x) is the value of Tag1. In all other cases (i.e. the quality of Tag1 is UNCERTAIN or BAD), the expression result (x) is the value of Tag2."

We can calculate the results for this expression using different qualities for Tag1 and Tag2, as shown in the figure below.

Case	Tag1 quality	Tag2 quality	Result	Result quality
1	GOOD	GOOD	Tag1	192 (GOOD)
2	GOOD	UNCERTAIN	Tag1	192 (GOOD)
3	GOOD	BAD	Tag1	192 (GOOD)
4	UNCERTAIN	GOOD	Tag2	192 (GOOD)
5	UNCERTAIN	UNCERTAIN	Tag2	64 (UNCERTAIN)
6	UNCERTAIN	BAD	Tag2	0 (BAD)
7	BAD	GOOD	Tag2	192 (GOOD)
8	BAD	UNCERTAIN	Tag2	64 (UNCERTAIN)
9	BAD	BAD	Tag2	0 (BAD)

In cases 1-3 above, the quality of Tag1 is GOOD, and therefore the result of the expression is GOOD. Thus, the result of the expression is not affected by the quality of Tag2 (the "else" branch of the expression), which is ignored.

In cases 4-6, the quality of Tag1 is UNCERTAIN, and therefore the result of the expression is the quality of Tag2.

In cases 7-9, the quality of Tag1 is BAD, and therefore the result of the expression is the quality of Tag2.

Note: The "quality()" function returns a value that represents the quality of the expression within the parentheses but is always GOOD_QUALITY. For example, if Tag1 is BAD_QUALITY then the expression "x=quality({Tag1})" will return 0 with GOOD_QUALITY.

The result of an expression is the minimum quality of the evaluated tag in the expression and is affected only by the quality of the conditional (if, then, or else) branch that is executed.

Consider the following sample expression:

x= if ({{TAG_01}}>0,{{TAG_02}},{{TAG_03}})

This expression can be read as follows:

"If the value of TAG_01 is greater than 0, then the expression result (x) is TAG_02. If the value of TAG_01 is less than or equal to 0, then the expression result (x) is TAG_03."

Let's assume that the following values and qualities for these tags:

TAG_01=5 with quality GOOD

TAG_02=6 with quality UNCERTAIN

TAG_03=7 with quality BAD

Because the value of TAG_01 is 5 (greater than 0), the expression result is TAG_02. Thus, the final expression result is 6, and the final expression quality is UNCERTAIN.

Type Conversion

The **tostring** option on the **Functions** menu of the **Expression Editor** takes the value of whatever item is in parentheses and converts it into a string as follows:

The value is +(value)+unit

It can be used to convert from number to string, and it can be very useful for string concatenation.

The proper syntax for the **tostring** option is:

x="\$The value is "\$ + tostring(value) + \$" unit"\$

Note: In the expression above, the word "unit" is placeholder text for a user-specified unit of measurement or variable (e.g. Watt, inches, meters, etc.).

Example Expressions Type Conversion

Expression	Result
x="\$The value is "\$ + tostring({{gfwsim.ramp.float}}) + \$" Watt"\$	"The value is 543.2345152 Watt"

Constants

The **Functions** menu of the **Expression Editor** supports constant values, including hexadecimal, octal, and binary formats.

Example Expressions Using Constants

Expression	Result
x=0x11	17
x=0t11	9
x=0b11	3

The **Expression Editor** conveniently inserts the 0x and 0t and 0b prefixes for you so do not have to recall them.

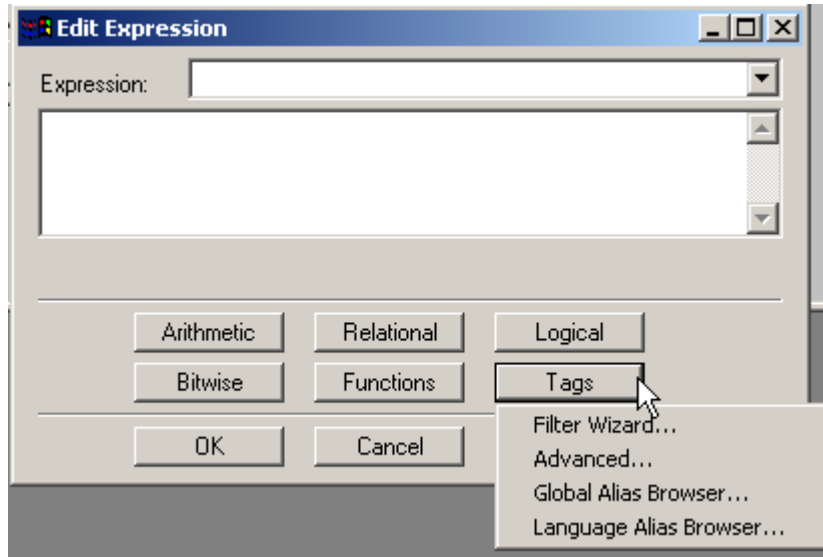
Interpreting and Translating Constants

The examples below show how values are calculated for each type of constant.

- **Hexadecimal:** $0x20A = 2 * (16^2) + 0 * (16^1) + 10 * (16^0) = 2*256 + 0*16 + 10 * 1 = 512 + 0 + 10 = 522$
- **Octal:** $0t36 = 3 * (7^1) + 6 * (7^0) = 3*7 + 6*1 = 21 + 6 = 27$
- **Binary:** $0b110 = 1 * (2^2) + 1 * (2^1) + 0 * (2^0) = 1 * 4 + 1 * 2 + 0 * 1 = 4+2+0 = 6$

Tags

The menu options available under the **Tags** button of the Expression Editor are shown in the figure below. The **Expression Editor** dialog box can also be used to create and 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.

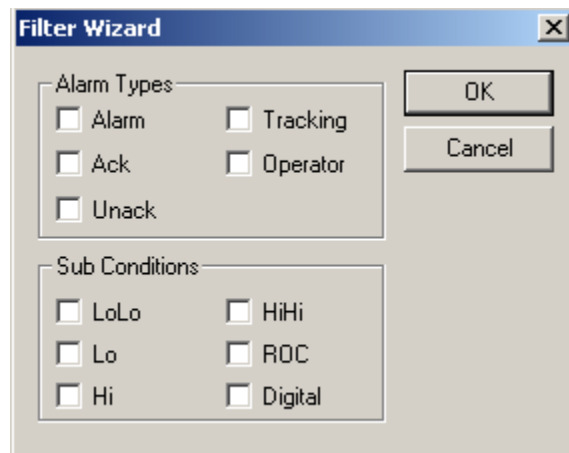


Creating Alarm Filters Using the Expression Editor

Filter Wizard

The **Filter Wizard**, shown in the figure below, allows you to choose from the following to items enter in your expression. Select one or more items, and then click **OK**. The filter string is automatically inserted into the **Edit Expression** dialog box.

- **Alarm Types:** Alarm, Ack, Unack, Tracking, and Operator
- **Subconditions:** LoLo, Lo, Hi, HiHi, ROC, and Digital



Filter Wizard

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**.



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
X = {{Severity}} > 500.	Only alarm messages with a severity greater than 500 will be visible.
X = Like({{Source}}, \$"Tag"\$,0)	Only messages with the tag in the source name will be displayed.
X = 1.	Filter displays all messages.
X = 0.	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.

Global Aliases

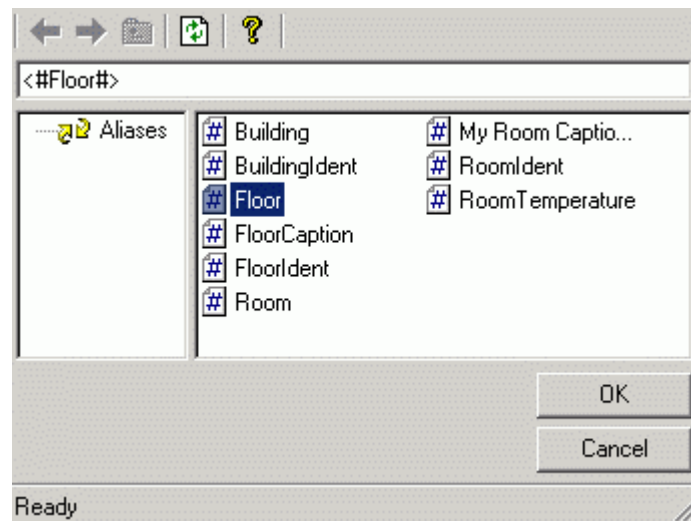
For global aliases within the expression, use the following syntax:

`<#global_alias_name#>`

Example:

`x=<#RoomTemperature#>`

Selecting **Global Alias Browser** opens the Global Alias Browser, as shown in the figure below. Select a global alias from the Global Alias 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 Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the `<#` and `#>` delimiters to the alias name. Click the **OK** button.



Selecting an Alias From the Global Alias Browser

Language Aliases

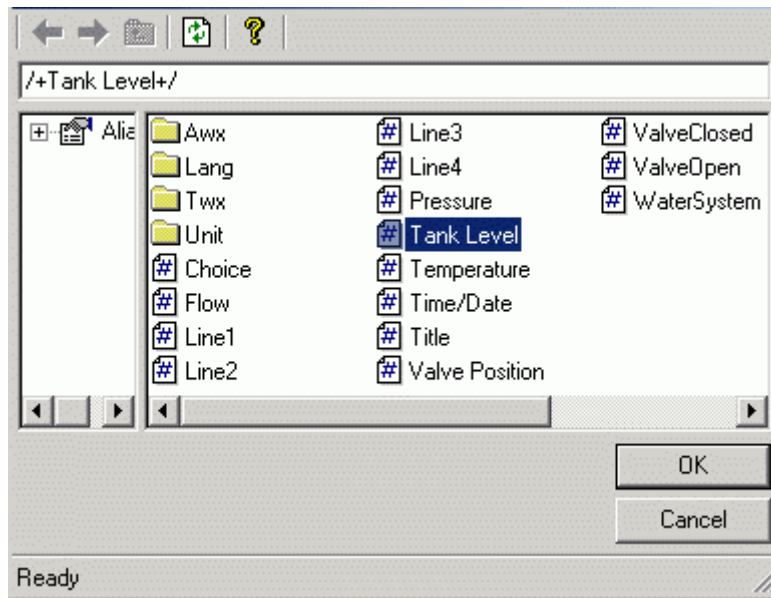
For language aliases within the expression, use the following syntax:

`/+language_alias_name+/-`

Example:

x=/+WaterSystem+/-

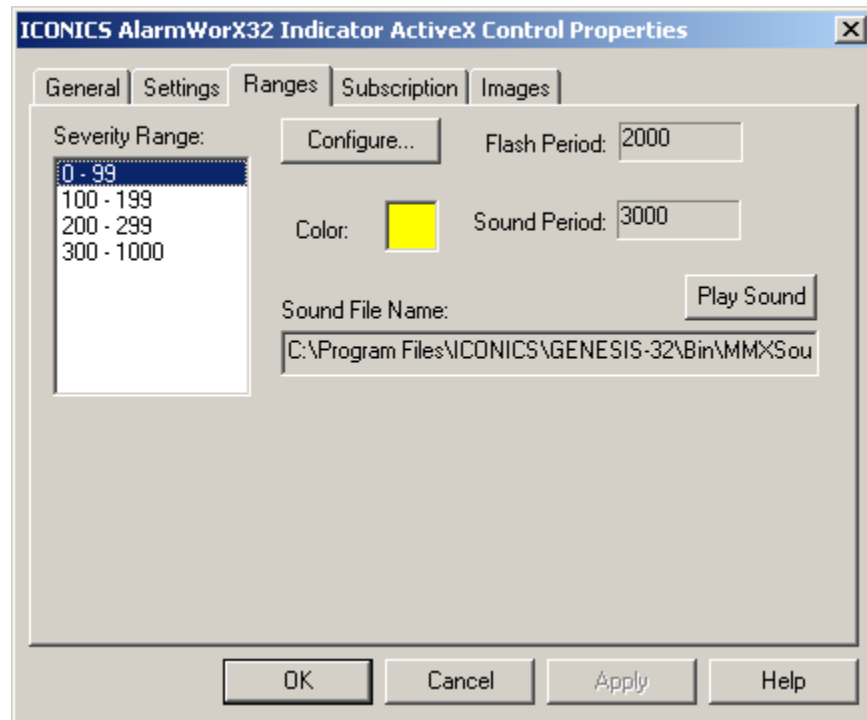
Selecting **Language Alias Browser** from the pop-up menu opens the Language Alias Browser, as shown in the figure below. The browser includes all language aliases in the language database. All language aliases that are configured in the Language Configurator are conveniently available to choose from inside the browser. The tree control of the Language Configurator is mimicked in the tree control of the Language Alias Browser. Select a language 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.



Selecting an Alias From the Language Alias Browser

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.



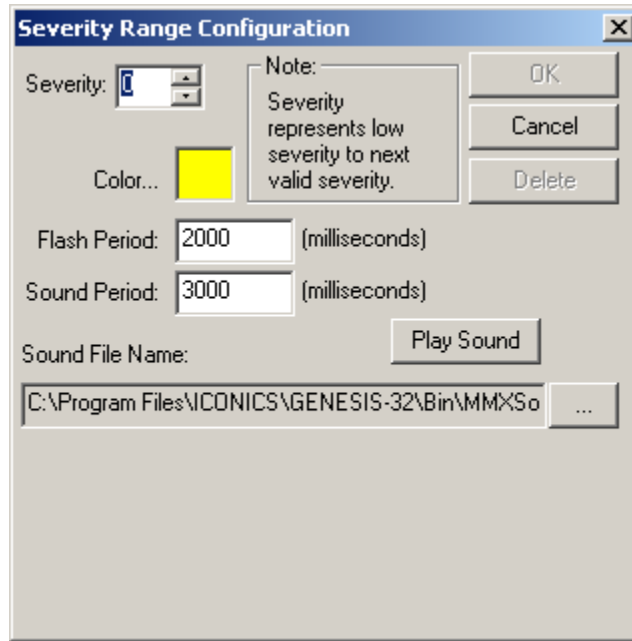
AlarmWorX32 Indicator ActiveX Control Properties: Ranges Tab

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

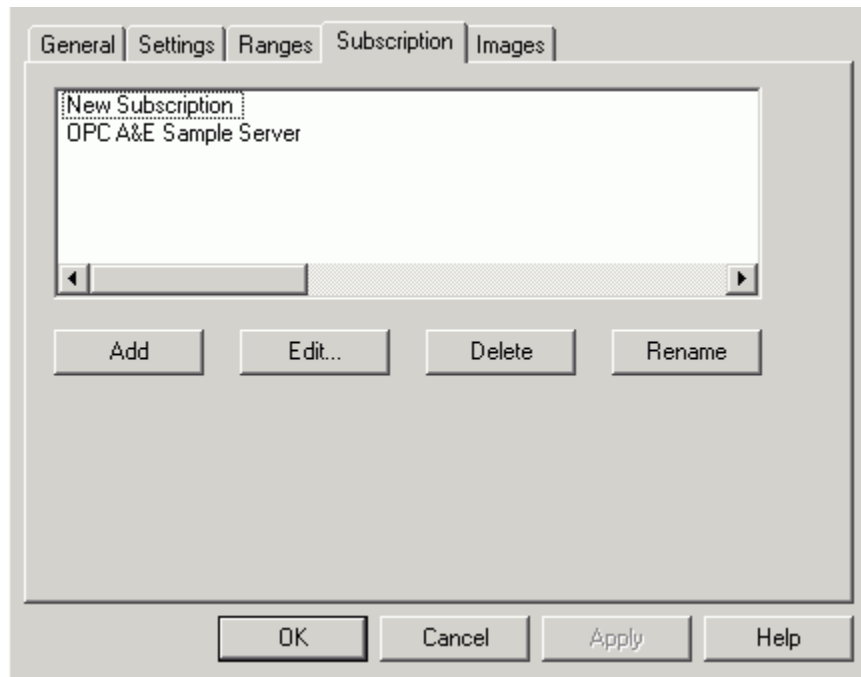
To delete a severity range configuration:

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

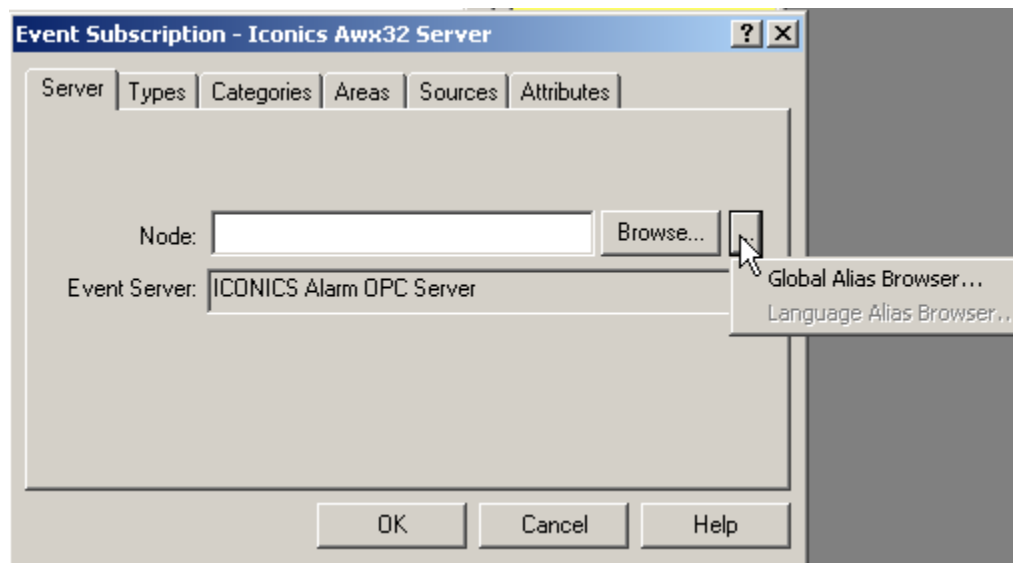
The **Subscription** tab of the **Alarm Indicator ActiveX Properties** dialog box, shown below, determines what type of OPC connection will be made. It allows you to add, rename, delete, or edit a subscription to a particular alarm. To add a new subscription, click the **Add** button.



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.

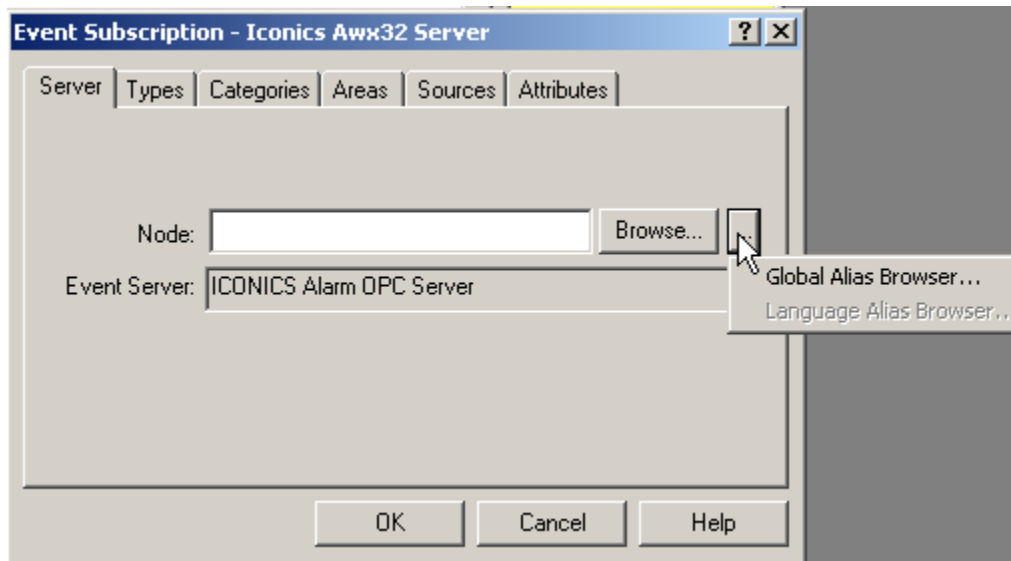


Creating an Event Subscription

Server
Types
Categories
Areas
Source
Attributes

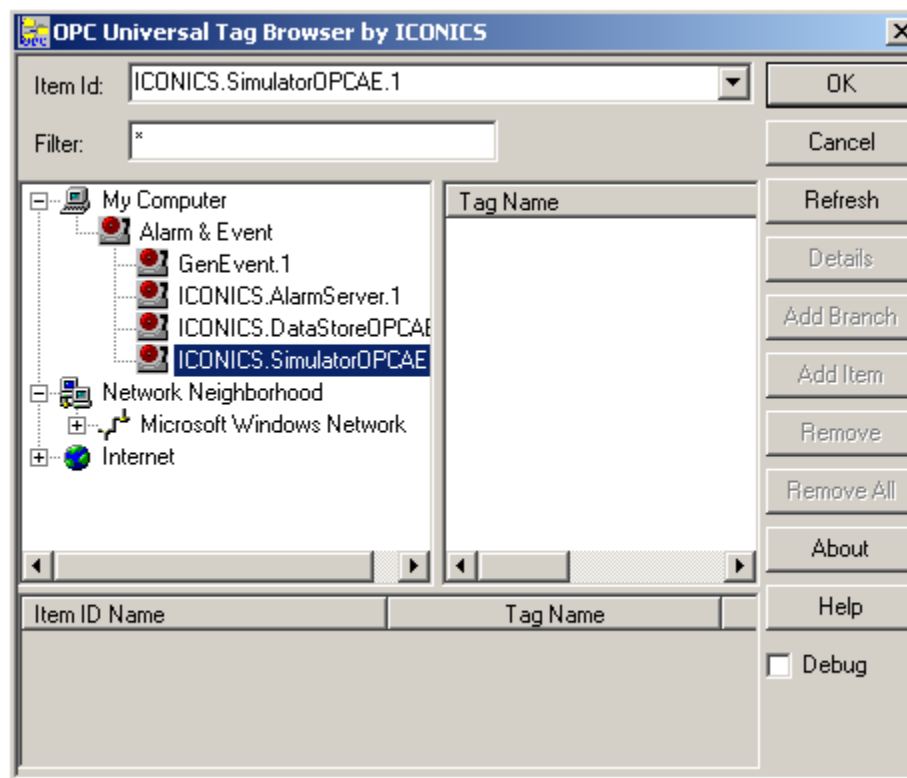
Server

The **Server** tab of the **Event Subscription** dialog box, shown in the figure below, allows you to select an **Event Server** and **Node** for each subscription. To select the event server, click the **Browse** button.



Event Subscription Dialog Box: Server Tab

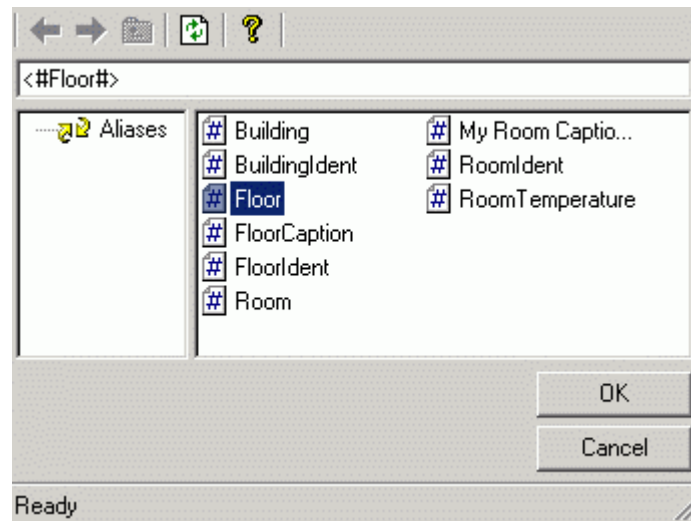
This opens the **OPC Universal Tag Browser**, shown in the figure below, which lists all available Alarm and Event OPC servers. Select the desired server, and click **OK**. For local servers, it is not necessary to fill in the **Node** field.



Selecting an Alarm and Events Server From the OPC Universal Tag Browser

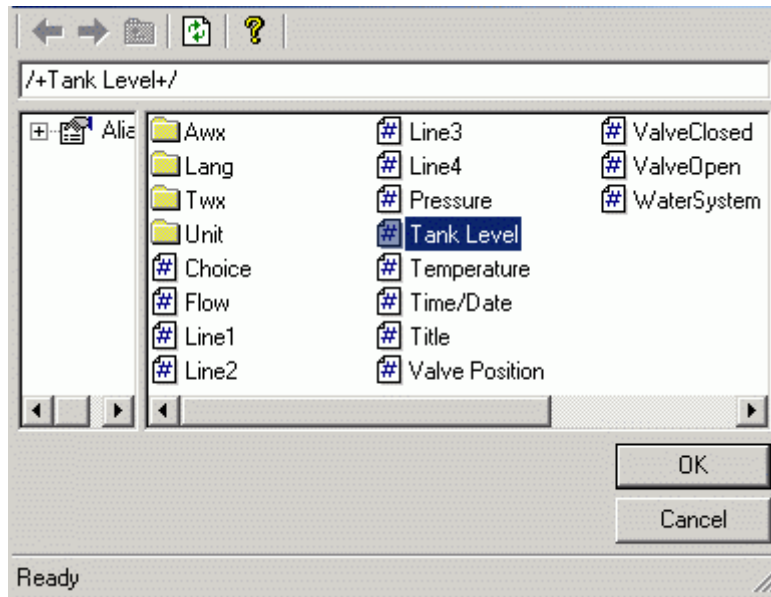
You can also select global aliases to use in the **Node** field. Clicking the ... button and selecting **Global Alias Browser** from the pop-up menu opens the

Global Alias Browser, as shown in the figure below. Select a global alias from the Global Alias Browser, which includes all global aliases in the global aliases 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 Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.



Selecting an Alias From the Global Alias Browser

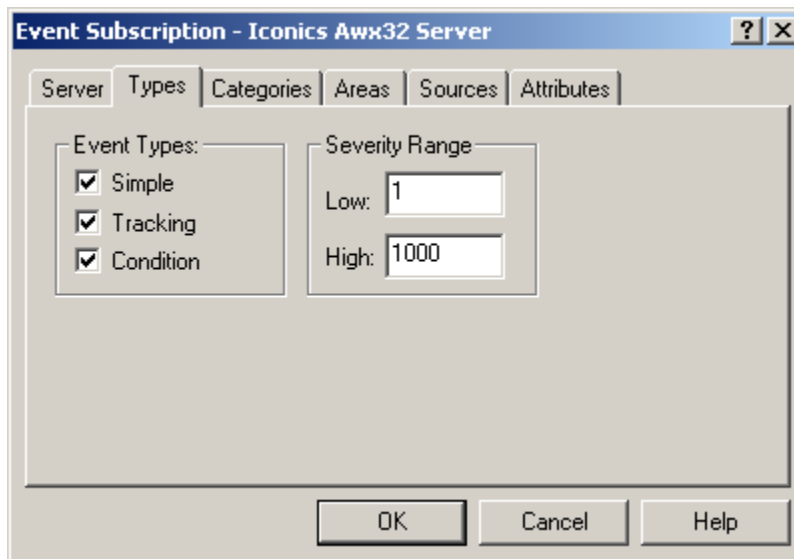
Clicking the ... button and selecting **Language Alias Browser** from the pop-up menu opens the Language Alias Browser, as shown in the figure below. The browser includes all languages aliases in the language database. All language aliases that are configured in the Language Configurator are conveniently available to choose from inside the browser. The tree control of the Language Configurator is mimicked in the tree control of the Language Alias Browser. Select a language 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.



Selecting an Alias From the Language Alias Browser

Types

The **Types** tab of the **Event Subscription** dialog box, shown in the figure below, allows you to configure which OPC-defined event types each subscription should have, as well as to set the ranges for severity (priority). A value of "0" represents the low severity value, and "1000" represents the high severity value. Please note that OPC Alarm and Event (AE) servers are required to scale severity values to the OPC ranges (i.e. an AE server that contained two severity ranges would convert these to "0" and "1000").



Event Subscription Dialog Box: Types Tab

Simple: These messages state information but do not have alarm status, nor do they contain information on what initiated the message. This includes the following information: Source, Time, Type, EventCategory, Severity, Message, Cookie, and server-specific items.

Example: "FIC101, 12:0:0 1/1/99, Simple, Category1, 100, 'Shift Change', 1"

Simple messages would be similar to an event.

Tracking: These messages contain the additional information of the client that initiated the event. This includes the following information: Source, Time, Type, EventCategory, Severity, Message, Cookie, ActorID, and server-specific items.

Example: "FIC101, 12:0:1 1/1/99, Tracking, Category1, 300, 'Pump pressure Set to 10 psi', 1, Station 12"

Tracking messages are similar to event messages in that the cause of the event is important. An example would be an operator changing a setpoint value. This type of message does not include acknowledge capability.

Note: Simple and tracking messages are removed from the alarm viewer via the acknowledge mechanism.

Condition: These messages contain all of the above information but also include an acknowledgement portion. This includes the following information: Source, Time, Type, EventCategory, Severity, Message, Cookie, ConditionName, SubConditionName, ChangeMask, NewState, ConditionQuality, AckRequired, ActiveTime, ActorID and server-specific items.

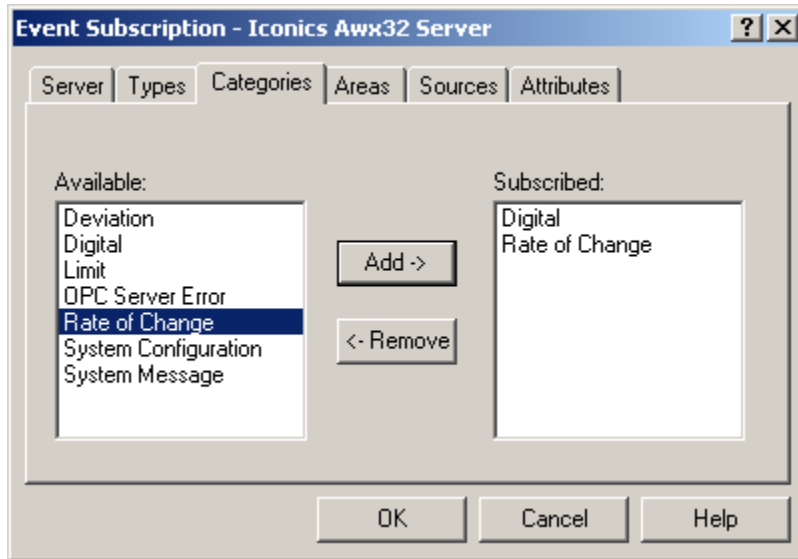
Example: "FIC101, 12:0:3 1/1/99, Condition, Category1, 700, 'Pump pressure to high', 1, Limit, HiHi, 1, Active Enabled, Good, TRUE, 12:0:2 1/1/99"

Condition messages would be considered a "typical" alarm message with acknowledge capability.

For further details on any of the included information, please refer to the OPC Alarm and Events specification.

Categories

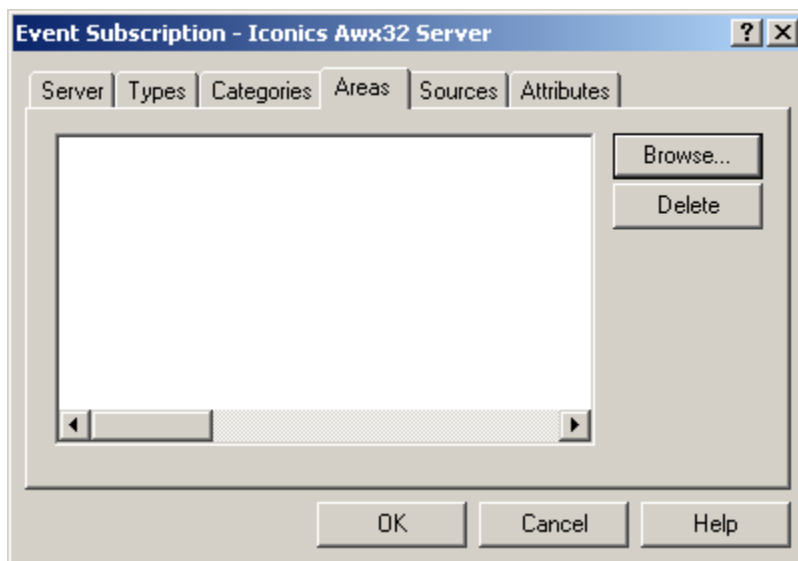
The **Categories** tab of the **Event Subscription** dialog box, shown in the figure below, allows you to select the categories for a subscription. Select the category from the list of available categories, and then click the **Add ->** button. The category will appear in the **Subscribed** list. To remove a category from this list, select it in the **Subscribed** list and click the **<- Remove** button. If no categories are listed in the **Subscribed** list, then all categories are selected by default.



Event Subscription Dialog Box: Categories Tab

Areas

Use the **Areas** tab of the **Event Subscription** dialog box, shown in the figure below, to select an area or a group of areas for filtering in a subscription.



Event Subscription Dialog Box: Areas Tab

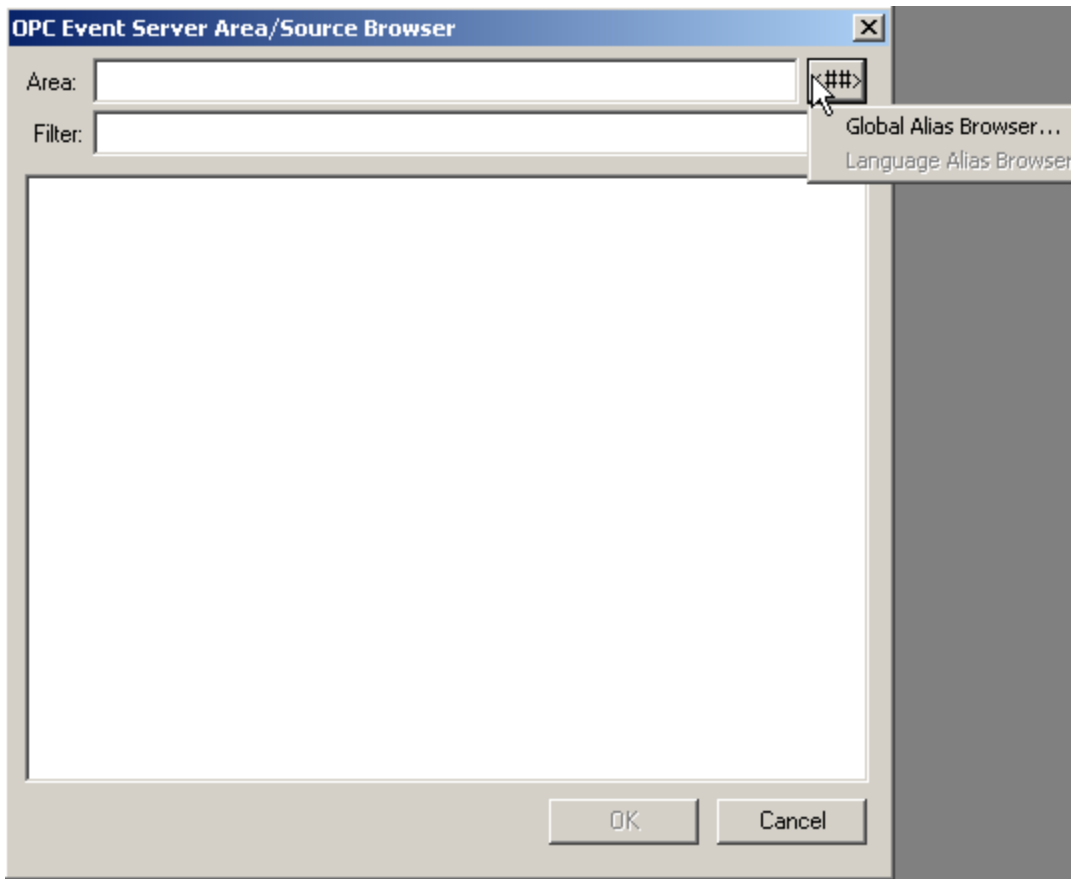
Clicking the **Browse** button opens the **OPC Event Server Area / Source Browser**, shown in the figure below, which lists all available areas for your event server. Area subscription supports wildcards, which subscribe to the format of the Microsoft Visual Basic 'like' command. For example, "Area1*" will subscribe to all alarm areas that contain strings beginning with "Area1". "Area1\" will subscribe to the root area and its "child" areas. A detailed explanation of the wildcard support can be found in the OPC Alarm and Events

documentation. It is recommended that you thoroughly read the wildcard documentation before attempting to use complicated expressions.

It is also possible to delete an area from the **Areas** tab. To delete an area, select it from the list of areas for this particular subscription, and then click the **Delete** button. Not all OPC Alarm and Event servers support area filtering as part of the subscription.

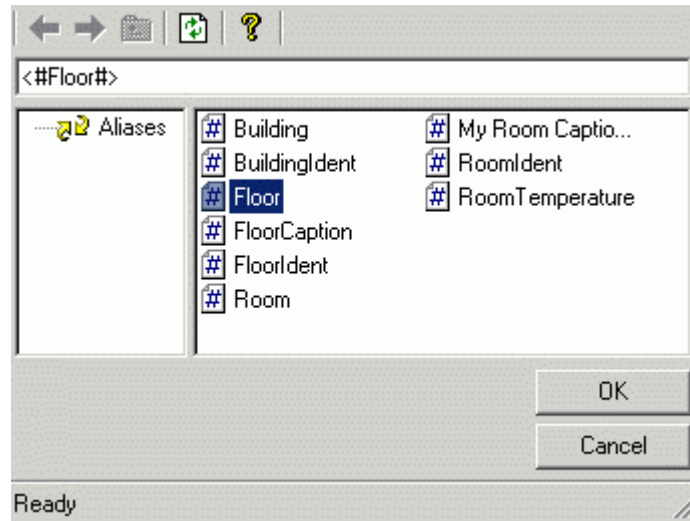
Note: If no areas are listed, then all areas are selected by default.

Note: It is recommended that you have an area in the alarm server to which you subscribe; this ensures that the relevant multimedia alarms are received.



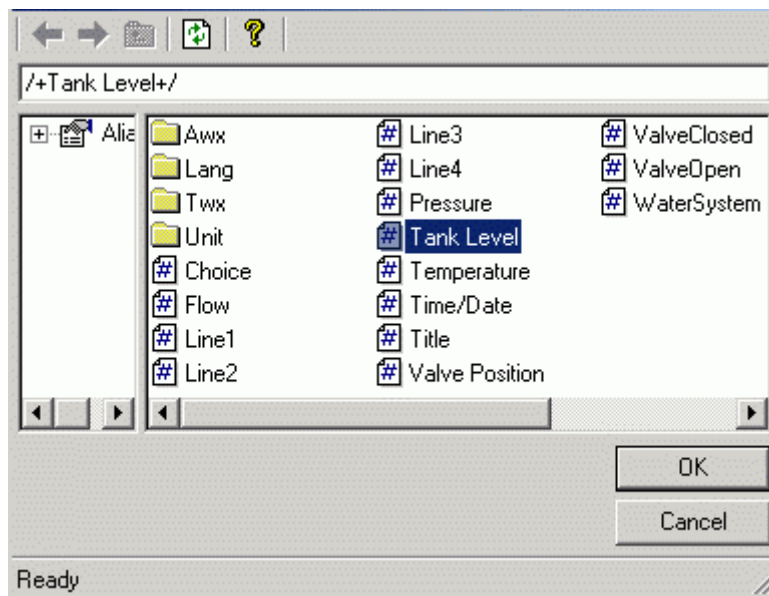
OPC Event Server Area/Source Browser

You can also select global aliases to use in the **Area** field. Clicking the ... button and selecting **Global Alias Browser** from the pop-up menu opens the Global Alias Browser, as shown in the figure below. Select a global alias from the Global Alias 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 Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.



Selecting an Alias From the Global Alias Browser

Clicking the ... button and selecting **Language Alias Browser** from the pop-up menu opens the Language Alias Browser, as shown in the figure below. The browser includes all languages aliases in the language database. All language aliases that are configured in the Language Configurator are conveniently available to choose from inside the browser. The tree control of the Language Configurator is mimicked in the tree control of the Language Alias Browser. Select a language 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.

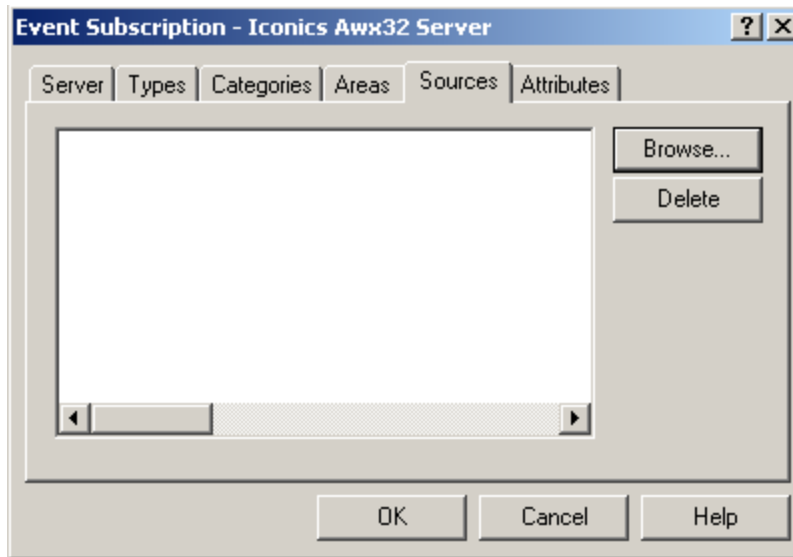


Selecting an Alias From the Language Alias Browser

Sources

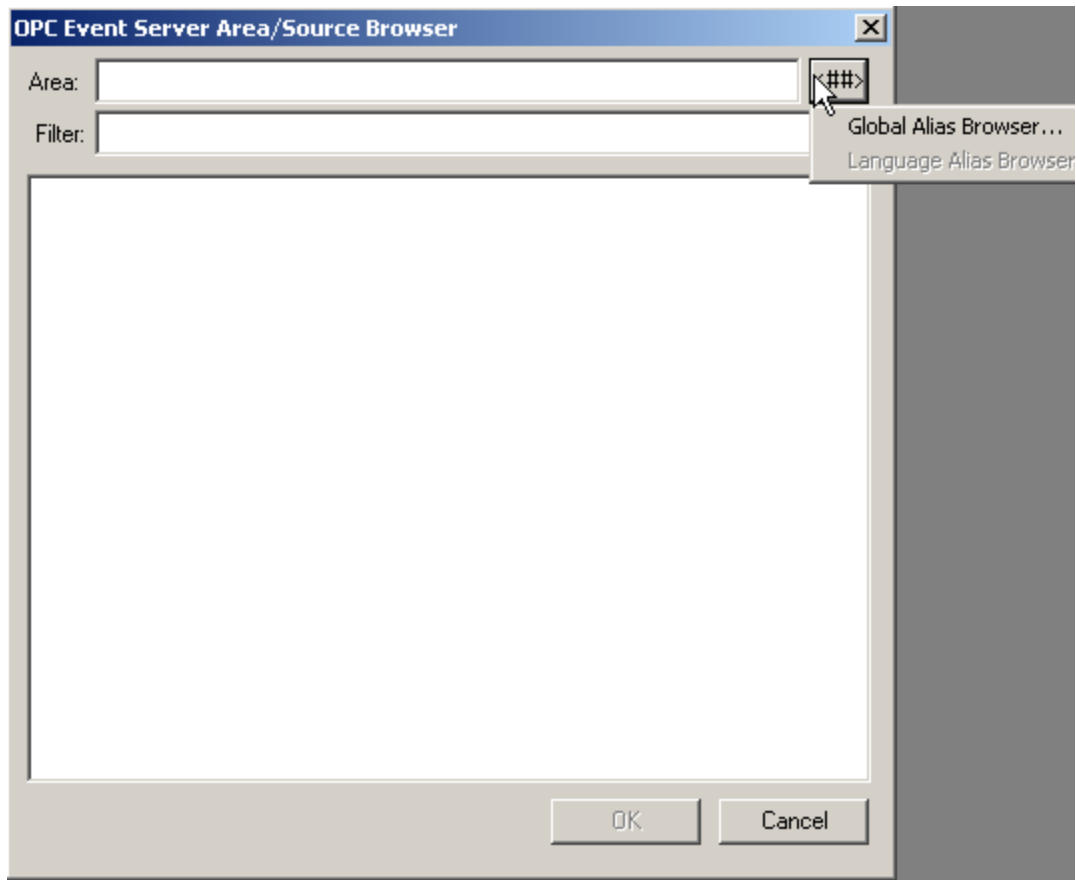
The **Sources** tab of the **Event Subscription** dialog box, shown in the figure below, allows you to filter on a source or group of sources for a particular subscription. It is also possible to delete a source for a particular subscription. To delete a source, select it from the list of sources for this particular subscription and click the **Delete** button. Not all OPC Alarm and Event servers support source filtering as part of the subscription.

Note: If no sources are listed, then all sources are selected by default.



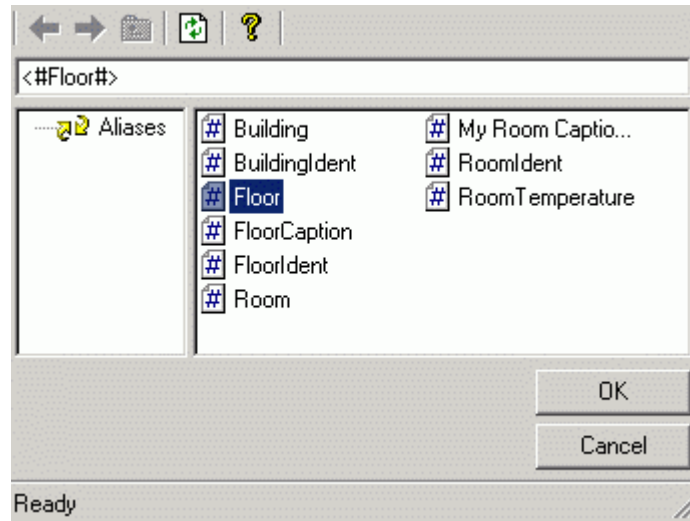
Event Subscription Dialog Box: Sources Tab

To select a source, click the **Browse** button and select one from the **OPC Event Server Area / Source Browser**, shown in the figure below. Source subscription provides the wildcard support found in the area subscription.



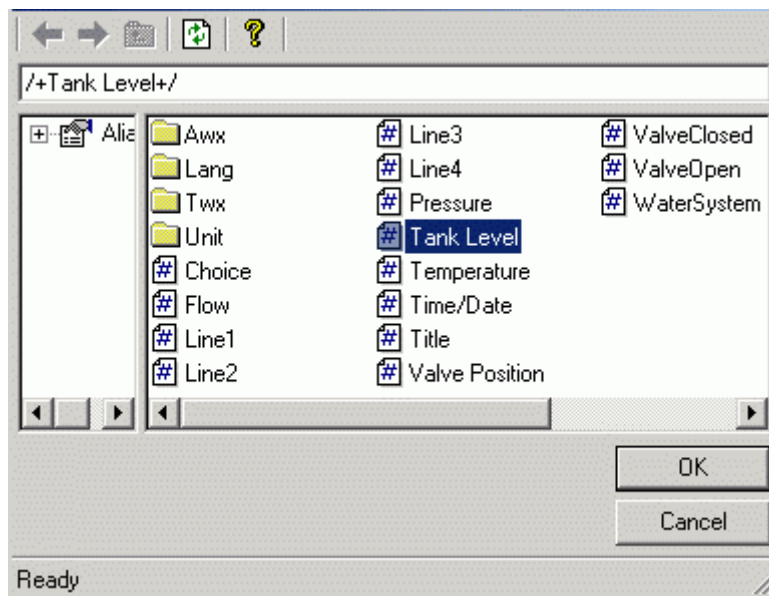
OPC Event Server Area/Source Browser

You can also select global aliases to use in the **Area** field. Clicking the ... button and selecting **Global Alias Browser** from the pop-up menu opens the Global Alias Browser, as shown in the figure below. Select a global alias from the Global Alias 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 Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the <# and #> delimiters to the alias name. Click the **OK** button.



Selecting an Alias From the Global Alias Browser

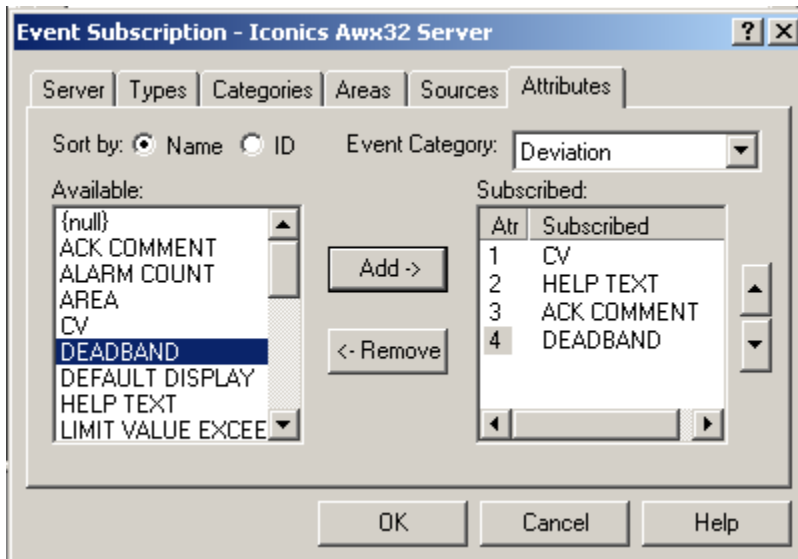
Clicking the ... button and selecting **Language Alias Browser** from the pop-up menu opens the Language Alias Browser, as shown in the figure below. The browser includes all languages aliases in the language database. All language aliases that are configured in the Language Configurator are conveniently available to choose from inside the browser. The tree control of the Language Configurator is mimicked in the tree control of the Language Alias Browser. Select a language 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.



Selecting an Alias From the Language Alias Browser

Attributes

The **Attributes** tab of the **Event Subscription** dialog box, shown in the figure below, allows you to add extra attributes to a particular event category within a subscription. Select the event category from the corresponding drop-down list. Choose the desired items from the available attributes list, and click the **Add** button. To remove an attribute from the subscribed list, select that particular attribute and click **Remove**.



Event Subscription Dialog Box: Attributes Tab

It is important to note that the order of the subscribed attributes does matter. The order of the attributes determines the order they will be selected from the server, and also determines with which viewer attribute column they are associated. To change the order of the attributes in the subscribed field, simply select an attribute and click the "up" and "down" arrow buttons. Server-specific information will be displayed in the Attributes1-n columns. The attributes columns are added in the **Column** tab of the **Alarm Viewer ActiveX Properties** dialog box.

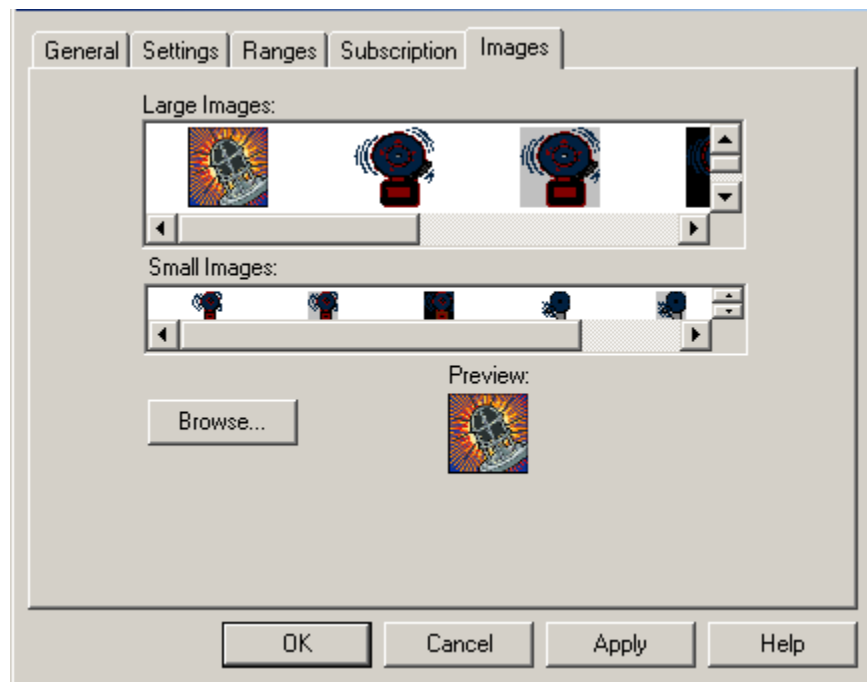
Note: To receive extra attributes, you must request them per event category.

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. This opens a dialog box that allows you to select .gif, .jpg, or .bmp image files.

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.

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 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

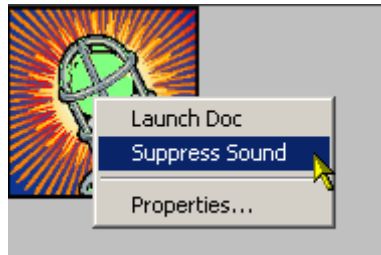
You can launch an application from the Alarm Indicator ActiveX during runtime mode. The type of application, the application file, and the method for launching the application have already been specified in the **Settings** configuration tab. Launch the application using the method you selected in the 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.



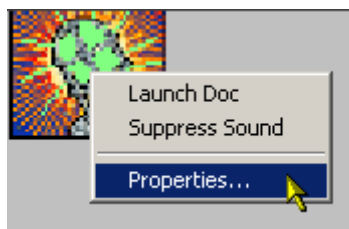
Turning off the Sound During Runtime

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

Automation Interfaces

The AlarmWorX32 Indicator ActiveX provides a COM interface that allows external programs to manipulate the indicator as it is running. The interface is available to all programming languages that support COM including Visual Basic (VB), Visual Basic for Applications (VBA) and Microsoft Visual C++.

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':

OLE_COLOR BackColor

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)
```

boolean CanUserConfigure

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

boolean CanUserLaunchDoc

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

boolean CanUserSuppressSound

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

string Caption

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"
```

boolean FileEnabled

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
```

string FileName

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"
```

OLE_COLOR FillColor

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)
```

boolean GlobalSoundSuppression

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
```

string LaunchDoc

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"
```

tagINVOKE_VIA LaunchDocVia

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
```

IPictureDisp Picture

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
```

object RangeCollection

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
```

long SuppressSoundPeriod

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
```

tagINVOKE_VIA SuppressSoundVia

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
```

string URLPath

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':

object Clone

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
```

long Color

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)
```

long FlashPeriod

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
```

boolean IsWav

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
```

long PlayPeriod

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
```

long Severity

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
```

string WavFileName

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  
AwxIndicator1.RangeCollection.Item(0).WavFileName = "C:\wav_file.wav"
```

Members of 'RangeCollection':

object Clone

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 = AwxIndicator1.RangeCollection.Clone
```

long Count

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 = AwxIndicator1.RangeCollection.Count
```

object Item

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 = AwxIndicator1.RangeCollection.Item(0)  
'This grabs the Range object of highest severity
```

Available Control Methods

Members of 'AwxIndicator':

void 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
```

void DoLaunchDoc()

Description

Launches the document specified in the LaunchDoc property.

Example

This example shows how to perform the LaunchDoc action.

AwxIndicator1.DoLaunchDoc

void 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

void 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

void 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

void 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
```

void 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
```

void 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)
```

long GetAlarmOleBlob()

void SetAlarmOleBlob(pBlob As Long)

Description

These above two methods are part of AlarmOLE.

long ReplaceFilePath(OldSubstring As String, NewSubstring As String)

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
```

long ReplaceHost(OldHostName As String, NewHostName As String)

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
```

long ReplaceHost(OldHostNameSubstring As String, NewHostNameSubstring As String, MatchCase As Long, MatchWholeWord 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
```

string ReplaceStringInString(StringToReplace As String, OldSubstring As String, NewSubstring As String)

Description

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

long ReplaceTag(OldSubstring As String, NewSubstring As String)

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':

void 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.

```
AwxIndicator1.RangeCollection.Item(0).WavFileName = "C:\My  
Directory\MyWav.wav"  
AwxIndicator1.LoadWavFile  
'Loads the file MyWav.wav
```

void 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.

```
AwxIndicator1.RangeCollection.Item(0).WavFileName = "C:\My  
Directory\MyWav.wav"  
AwxIndicator1.PlayWav  
'Plays the file MyWav.wav
```

Members of 'RangeCollection':

void 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 = AwxIndicator1.RangeCollection.Item(0).Clone
newRange.Color = vbBlue
newRange.FlashPeriod = 300
newRange.PlayPeriod = 6000
newRange.Severity = 500
AwxIndicator1.RangeCollection.AddItem(newRange)
```

void DeleteItem(ToDelete As Object)