



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

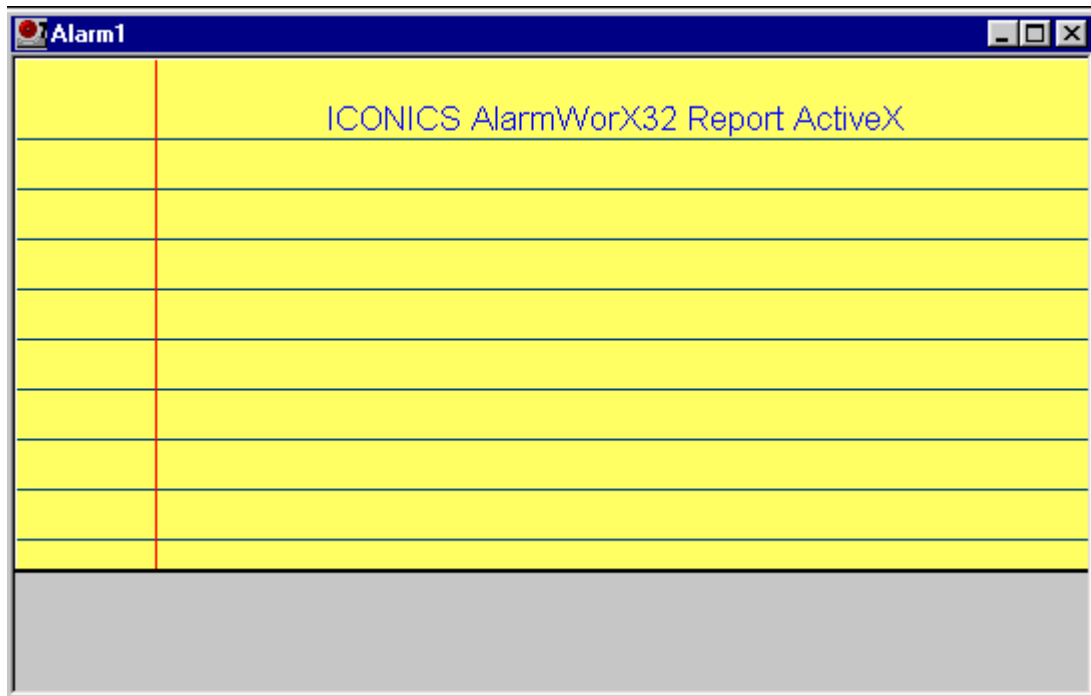
Web: www.iconics.com

© ICONICS, Inc. All rights reserved.

AlarmWorX32 Report

The AlarmWorX32 Report ActiveX allows reporting (user-configured or pre-configured) and graphing of alarms. The source of the Alarm data is alarms previously logged by the Alarm Logger. The Alarm Report is an ActiveX, allowing it to be placed in any ActiveX container application, including GraphWorX32, Visual Basic, or a Web page.

The AlarmWorX32 Report ActiveX is accessible from the ActiveX toolbar in the AlarmWorX32 Container, or from the **Edit** menu. Selecting **Insert New Object** from the **Edit** menu opens the **Insert Object** dialog box. Select the ICONICS AlarmWorX32 Report ActiveX item, and then click **OK**. The Report ActiveX is inserted into the display, as shown in the figure below.



Alarm Container Window With the AlarmWorX32 Report ActiveX

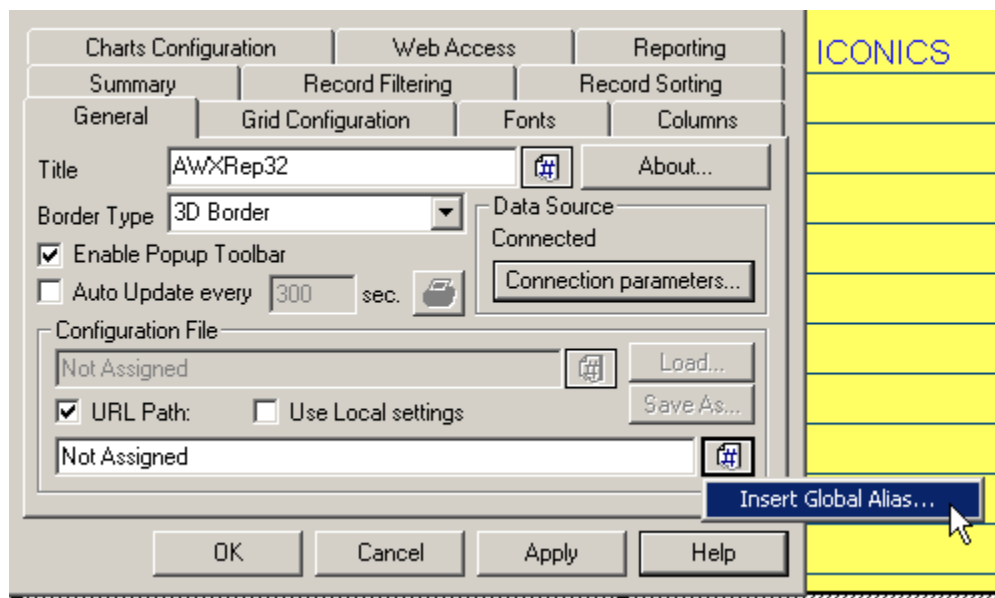
Configuration

To start the configuration of the Report ActiveX, double-click the ActiveX (the yellow notebook type object). This opens the **Alarm Report ActiveX Properties** dialog box for the ActiveX. Use these property pages to configure how alarms will be reported (appearance, organization, etc.) and which report type will be used.

General
Columns
Grid Configuration
Charts Configuration
Fonts
Records Sorting
Record Filtering
Reporting
Summary
Web Access

General

The **General** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, allows you to specify the name of the report file, select the border type and connect to a data source.



Report ActiveX General Tab

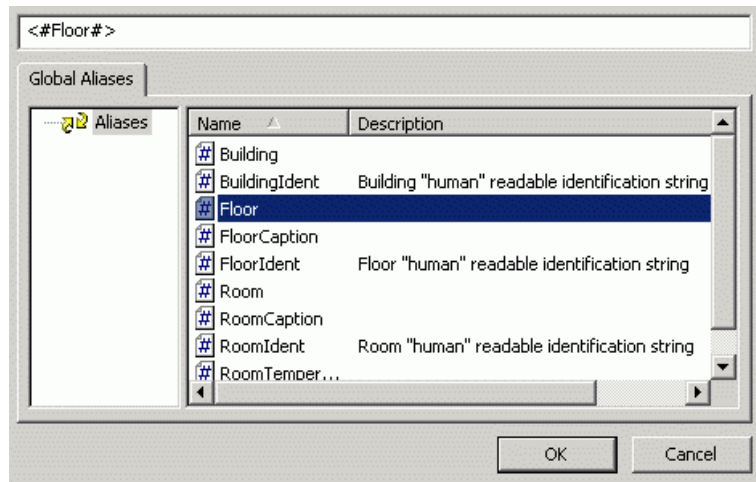
Configure the following **General** tab settings:

- **Title:** Enter a title for the report.
- **About:** Clicking the **About** button displays both technical support information and the ICONICS About Box, which contains registration and serial number information as well as system resource information, such as memory and disk space available.
- **Border Type:** Select one of the following border types for the report from the drop-down list: None, Flat, or 3-dimensional.
- **Enable Popup Toolbar:** When this option is checked, the Report toolbar is available during runtime mode.
- **Auto Update:** Specifies the frequency (in seconds) at which the alarm report updates data from the Alarm Logger database.
- **Configuration File:** By default, information is stored as part of the container application's document file. You can also store information as part of a separate (.awr) file. To save the .awr file, click **Save As** to open the **Save AWXReport32 Configuration Info** dialog box. Enter a name for the .awr file in the **File Name** field, and then click **Save**.

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.

You can also select aliases to use for the configuration file. Clicking the **#** button and selecting **Insert Global Alias** from the pop-up menu opens the Global Aliases tab of the Unified Data Browser, as shown in the figure below. 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 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

- **Data Source:** Clicking the **Connection parameters** button opens the **Database Connection** dialog box, where you can specify an alarm logger database. See "Connecting to a Data Source" below.
- **Auto Print:** The Auto Print feature is available only when the **Auto Update** feature is enabled. Reports are printed automatically on the same time interval specified for the auto update. Thus, if the Auto Update is set for 300 seconds, then reports will be automatically printed every 300 seconds.

Clicking the printer button opens **the Auto Print Settings** dialog box, as shown in the figure below. Check the Enable Auto Print feature check box. If you use the **Default Printer** for your system, you can specify the page orientation and number of copies. If you choose an alternate device under **Selected Devices**, you must click on the printer buttons to the right to configure the settings for each device.

The options available under **What to Print?** depend on your database connection and whether you have configured any Crystal Report Designs in the **Reporting** tab of the **Alarm Report ActiveX Properties** dialog box.

Note: If the Crystal Report Engine is not installed on the local machine, then auto print for Crystal Reports will not work on that machine.

Auto print settings

☒ Enable Auto Print feature

Print to devices:

☒ Default Printer Orientation: ☒ Portrait Number of Copies: 1

☐ Selected Devices:

<unspecified> [Printer Icon]

☒ <unspecified> [Printer Icon]

☒ <unspecified> [Printer Icon]

☐ <unspecified> [Printer Icon]

☐ <unspecified> [Printer Icon]

What to print?

☒ Current Grid / Chart View ☐ Crystal Report - Design #3

☐ Condensed Alarm Report ☐ Crystal Report - Design #4

☐ Detailed Alarm Report ☐ Crystal Report - Design #5

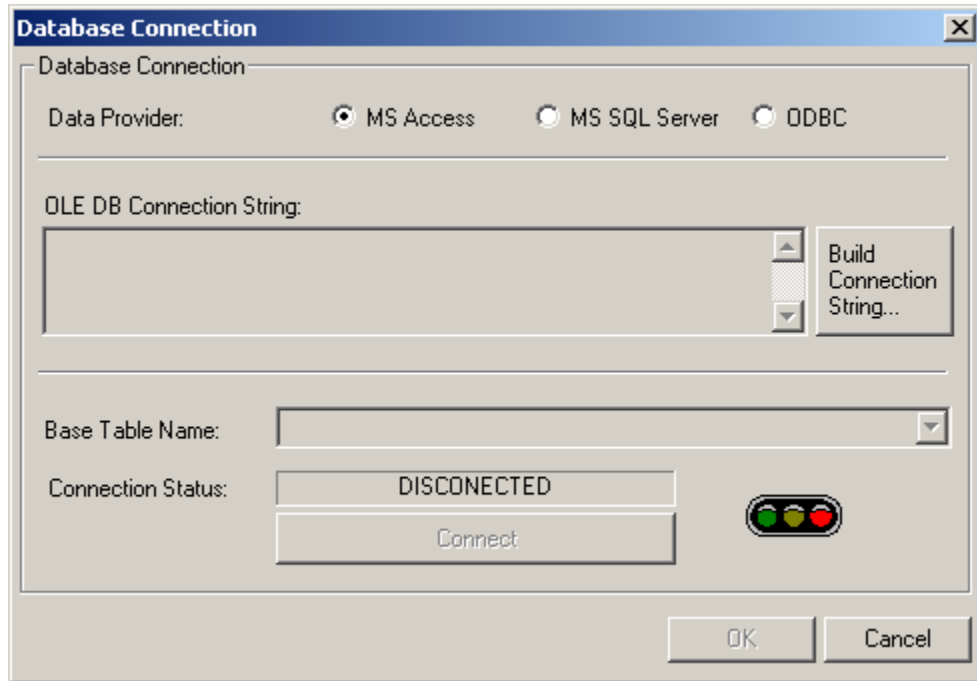
OK Cancel

Configuring Auto Print Settings

Connecting to a Data Source

To connect to a data source:

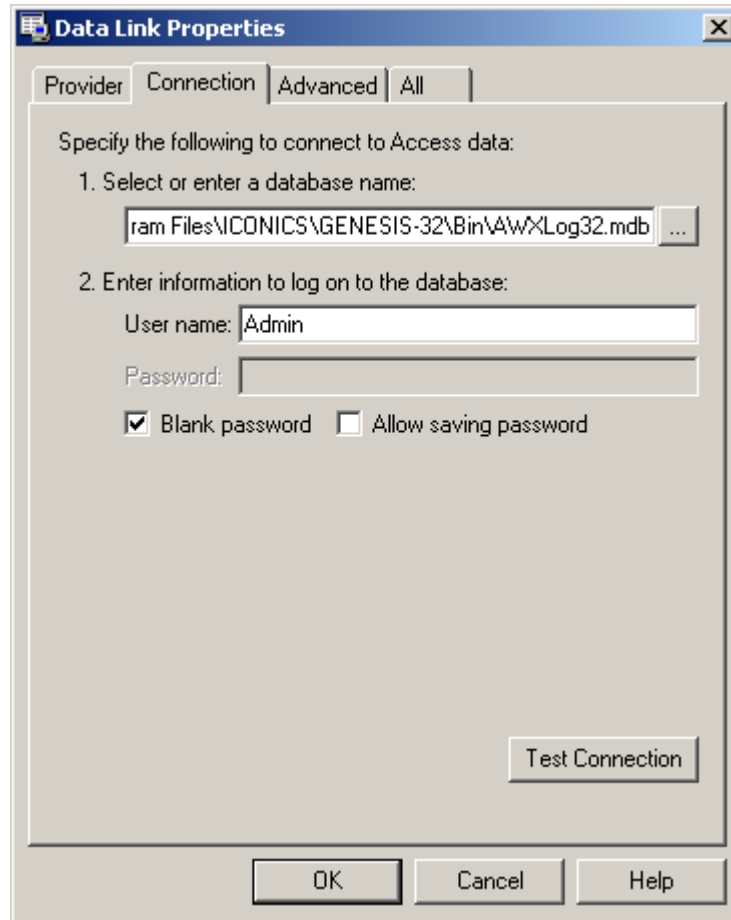
1. Click the **Connection parameters** button. This opens the **Database Connection** dialog box, as shown below.
2. Under **Data provider**, select from Microsoft Access, Microsoft SQL Server, or ODBC (Open Data Base Connectivity), as shown in the figure below.



Connecting to a Database

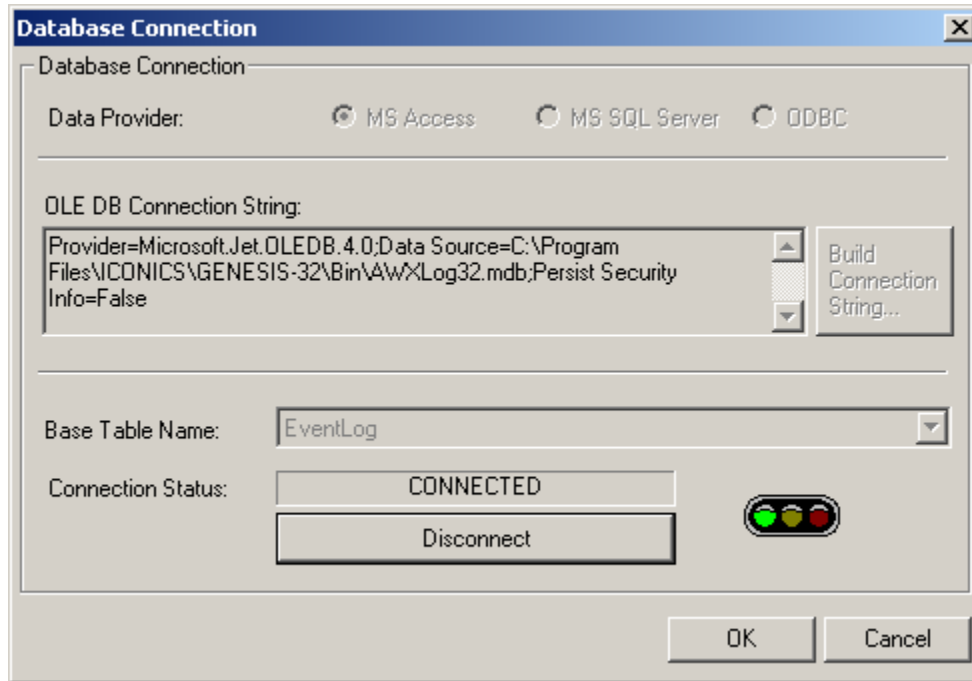
3. Click the **Build Connection String** button. This opens the Microsoft **Data Link Properties** dialog box, as shown in the figure below. In the **Connection** tab specify the data source and then click **OK**.

Note: All data source connections are made through the **Data Link Properties** dialog box. The **Connection** tab settings may vary depending on which data provider you selected. Click the **OK** button. Click the **Help** button at any time to view the Microsoft Data Link help documentation.



Specifying a Data Source

4. The data source reference appears in the **Database Connection** dialog box in the **OLE DB Connection String** field, as shown below. The **Base Table Name** field lists the tables in the database. You can select a specific table from the drop-down list to use as the reference table for the database.
5. Click the **Connect** button to connect to the database. The traffic light icon changes to green when the connection is successful, as shown in the figure below.

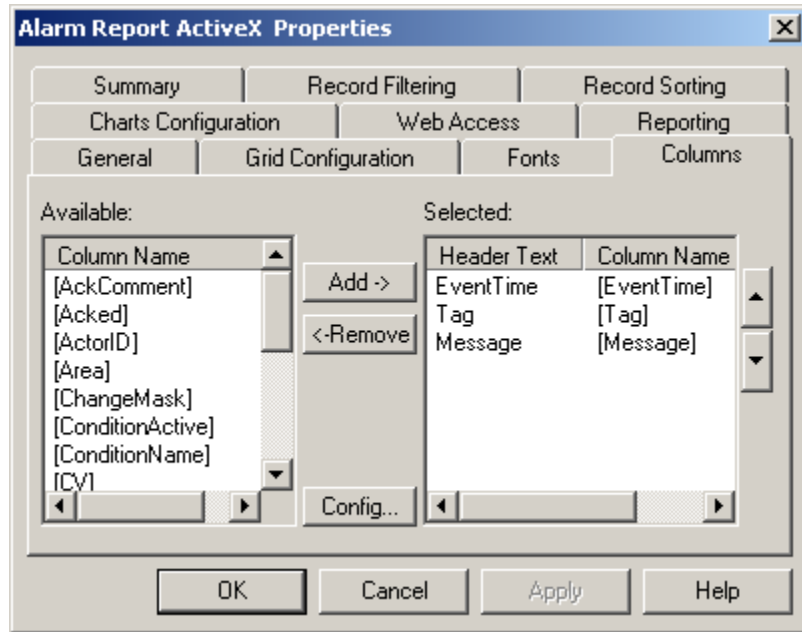


Database Connection

Columns

In the **Columns** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, you can specify which columns will appear in your alarm report. By default, the following three items are included in the report: EventTime, Tag, and Message, as shown in the **Selected** list in the figure below.

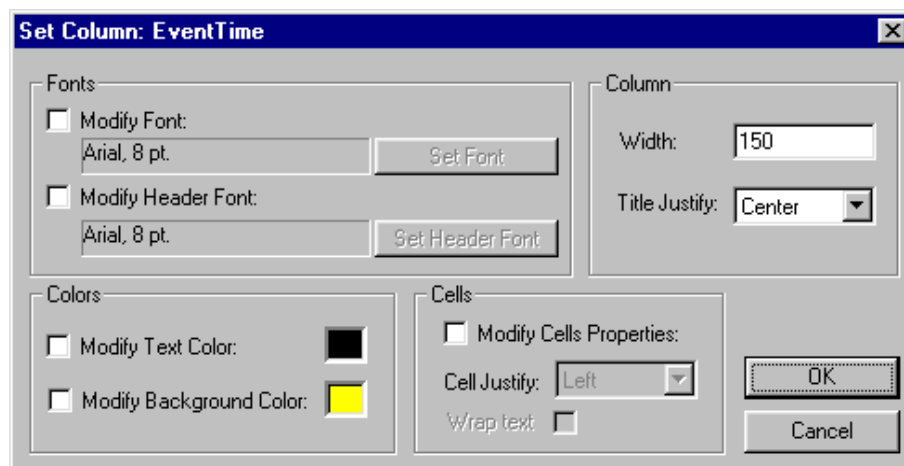
Note: For more information about column types, event types and attributes, please see the AlarmWorX32 Viewer, AlarmWorX32 Server, and AlarmWorX32 Logger documentation.



Report ActiveX Columns Tab

By clicking the **Add** and **Remove** buttons in the middle of and the "up" and "down" arrows on the right side of this the Columns tab, you can select which columns to include in the report as well as indicate the order of their appearance. To move an item from the **Available** list to the **Selected** list, simply highlight the item and click the **Add** button. To move an item back from the **Selected** list to the **Available** list, highlight the item and click the **Remove** button. To change the order of an item in the **Selected** list, simply select the item and click the "up" and "down" arrows located on the right-hand side.

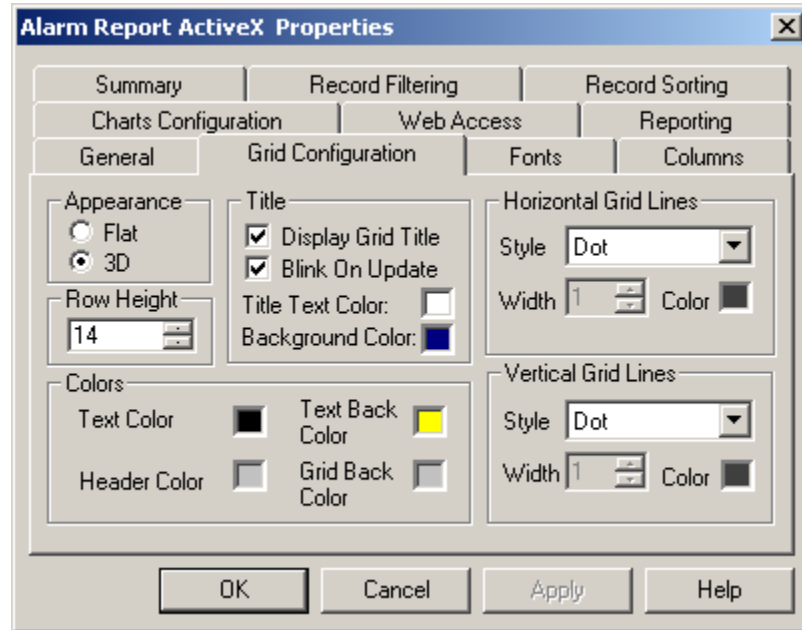
You can also change the appearance of a column by either double-clicking on an item in the **Selected** list or selecting the item and clicking the **Config...** button. The **Set Column** configuration dialog for the item opens, as shown below. From this dialog, you can set the font, color, width, and cell properties by selecting and changing the various options available.



Columns Configuration Dialog

Grid Configuration

The **Grid Configuration** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, determines how the grids on your alarm report will look and act.



Report ActiveX Grid Configuration Tab

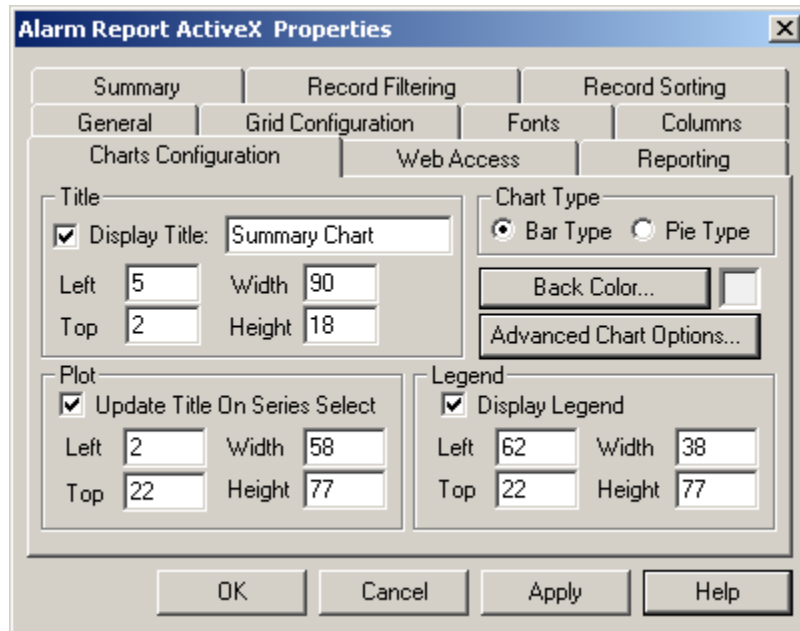
Configure the following grid settings:

- **Appearance:** Choose either a 3-dimensional grid or a flat grid.
- **Row Height:** Specifies the space between horizontal grid lines.
- **Colors:** Specifies colors for the grid text, headers, and background.
- **Title:** To show the grid title on the report, check the Display Grid Title check box. When **Blink on Update** is checked, the grid title will blink each time the report is updated with new data from the Alarm Logger database. You can also specify the color for both the title text and background.
- **Grid Lines:** Both **Horizontal** and **Vertical** grid lines appear on the alarm report. You can specify the **Width**, **Style** (e.g. Solid, Dashed, etc.), and **Color** of the grid lines. To change the color of the grid lines, simply click the **Color** check box and select a color from the color palette.

Note: It is not necessary to have grid lines, but it is helpful in distinguishing between different records.

Charts Configuration

The **Charts Configuration** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, refers to the chart that will be generated from the report data. The settings in the **Charts Configuration** tab determine how the chart will appear in runtime mode. To access the report chart, enter runtime and click the **Chart Summary** button. For more details on this runtime feature, please see the **Runtime - Chart Summary** section.



Report ActiveX Charts Configuration Tab

Configure the following chart settings:

- **Chart Type:** Select one of two types of chart: **Bar** or **Pie**.
- **Background Color:** Set the background color for the chart (the color that will surround the chart) and the chart's title.
- **Title/Plot/Legend:** You can toggle the visibility of chart title and/or legend by checking or unchecking **Display Title** or **Display Legend**, respectively.

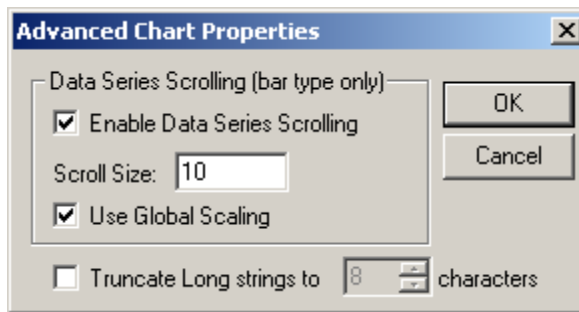
You can also set the exact positions and sizes of chart elements (title, plot, and legend). The positions are determined by values in the **Left** and **Top** fields, and sizes for particular chart elements are set in the **Height** and **Width** fields. The positions and sizes are given in the percentage of full width and height of chart area, assuming that the coordinate origin (0,0) is located in the top left corner of the chart. For example, setting Left=50, Top=50, Height=50, Width=50 for the chart legend will position legend in the lower right quadrant of the chart area.

If **Update Title On Series Select** is checked, the chart's title text will be updated when you click on any data series (either bar or pie segment) of the plot object. A name of the selected data series will be included

with the title text. This mode is particularly useful when the chart legend is hidden.

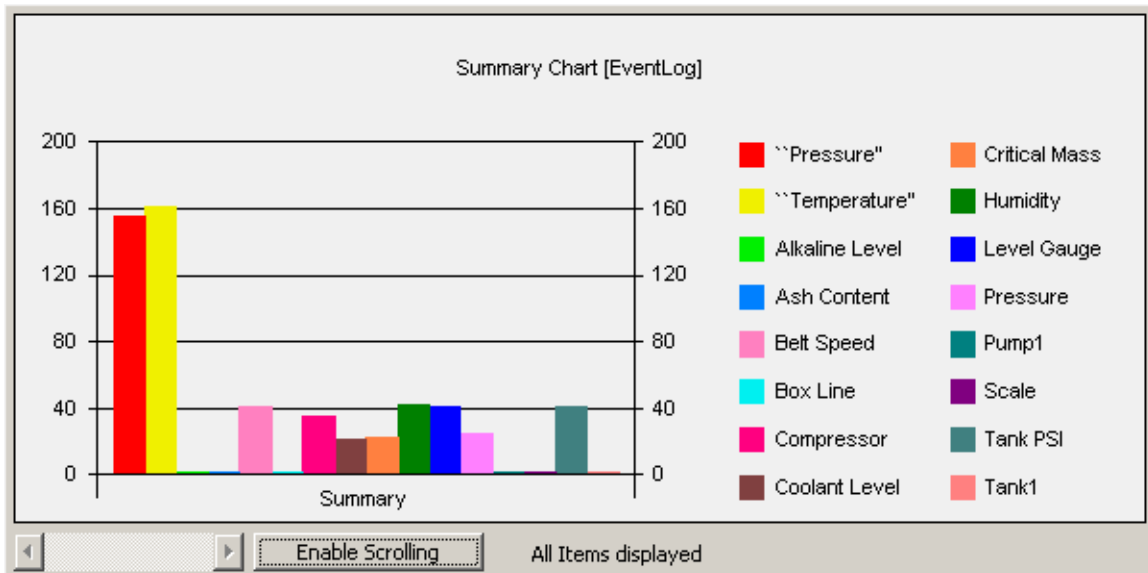
- **Advanced Chart Options:** Clicking the **Advanced Chart Options** button opens the **Advanced Chart Properties** dialog box, as shown in the figure below. If your alarm records contain long strings of data, you can abbreviate the strings by checking **Truncate Long strings to "X" characters**. Then specify the maximum number of string characters to be displayed (in place of the "X").

For bar charts, checking **Enable Data Series Scrolling** allows you to scroll across the bars in the chart. The number of bars in the scrolling region is determined by the **Scroll Size**. When scrolling is enabled for bar charts, you have the option of using global scaling.



Advanced Chart Properties

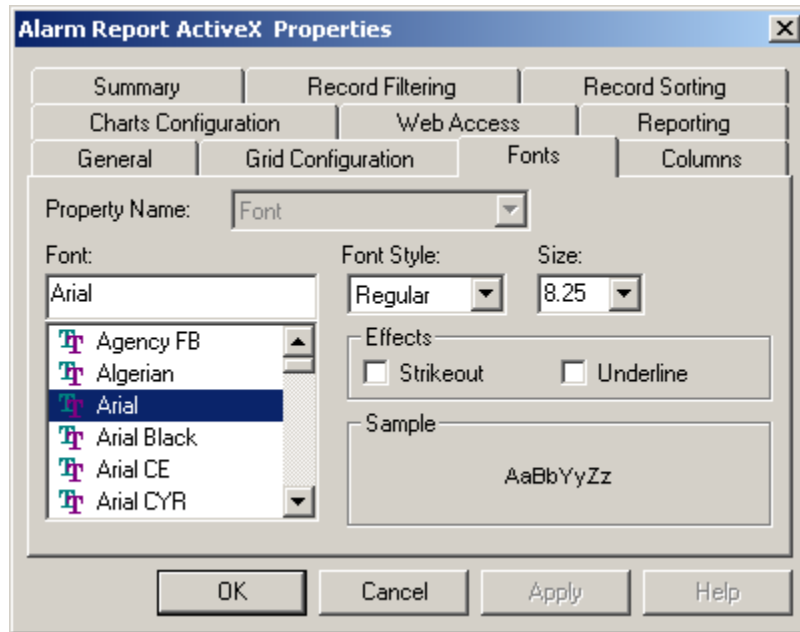
If scrolling is enabled, an **Enable Scrolling** button appears at the base of the bar chart summary in runtime mode, as shown below. Click this button to scroll left or right between bars on the chart using the arrows.



Bar Chart Summary

Fonts

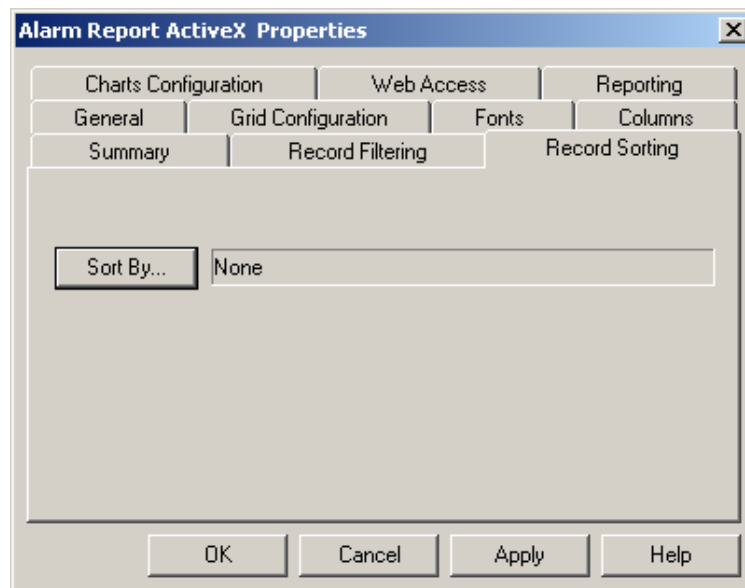
The **Fonts** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, specifies the font size, style, and text effects (i.e. underline and strikethrough) for the alarm report.



Report ActiveX Fonts Tab

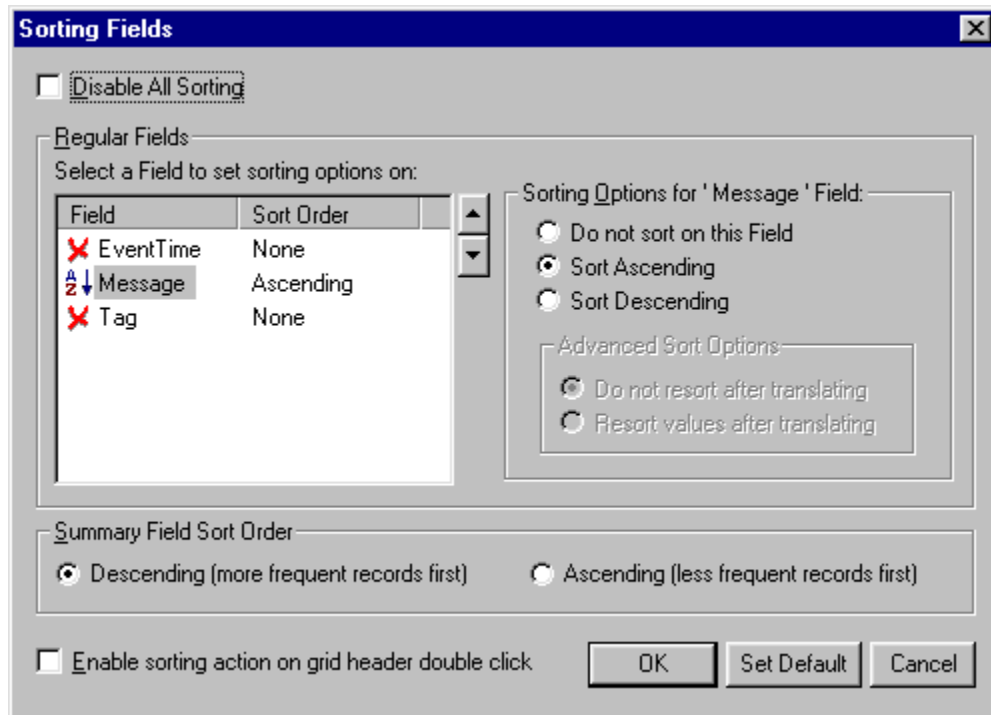
Records Sorting

The **Records Sorting** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, has just one button: the **Sort By** button.



Report ActiveX Records Sorting Tab

Clicking the **Sort By** button opens the **Sorting Fields** dialog box, as shown in the figure below. You can choose whether to disable or enable sorting by leaving the **Disable All Sorting** check box checked or unchecked. If you choose to leave this box unchecked, then several sorting options are available. Here you will specify the priority by which the fields in the alarm report are sorted.



Sorting Fields Dialog Box

To change the priority by which the fields will be displayed, highlight a field and then click the **Move Up/Move Down** buttons. The order of the columns under **Field** indicates their priority level.

To determine the order in which the data will be displayed within in each column, highlight that column and select one of the following sorting options:

- **Do Not Sort on This Field:** Leaves the data unsorted.
- **Sort Ascending:** Sorts the data in the column in ascending order (i.e. from least to greatest)
- **Sort Descending:** Sorts the data in the column in ascending order (i.e. from greatest to least)

All elements will be sorted by the first category in the list. In the event that two elements in the first category are identical, the reported alarms are sorted by the criteria configured for the next item in the list.

If you are dealing with language translations, you have the option of resorting values after translating them. This option is for advanced users and is not recommended for scenarios where there is a large set of records after filtering.

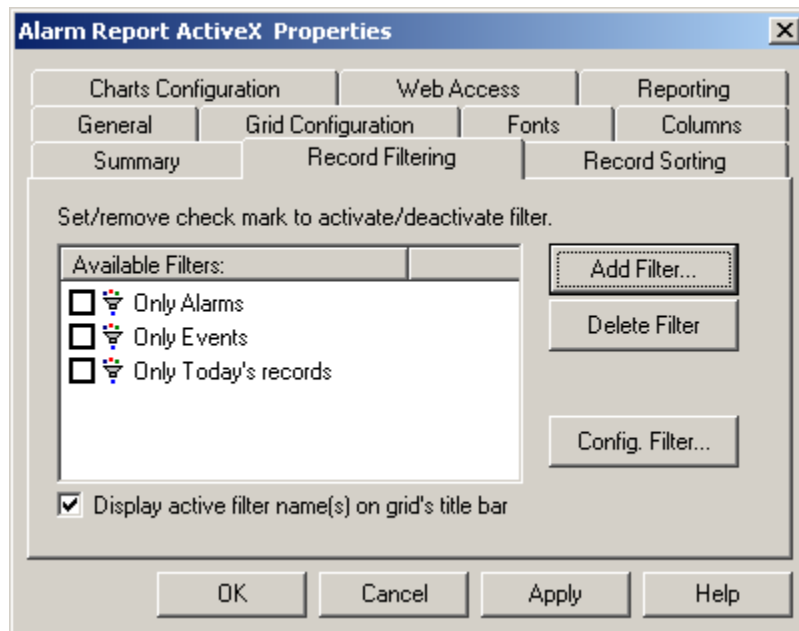
You can also set the **Summary Field Sort Order** (see **Runtime - Data Grid** and **Chart Summary**) for more information on the summary field) by selecting **Descending** or **Ascending**.

If the **Enable sorting action on grid header double-click** check box is selected, then you have the option of double-clicking the header of each column in the report while in runtime to toggle between ascending and descending sorting for the selected column.

Note: You can also access the **Sorting Options** dialog box via the **Sort Alarms** button on the **AWXRep32** toolbar while in runtime mode. (Double-click anywhere in the ActiveX client box to show the **AWXRep32** toolbar.)

Record Filtering

The **Records Filtering** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, offers simple filtering options for the alarm report. You can filter by date, by tag, or by custom SQL statements.



Reports ActiveX Record Filtering Tab

All of the items checked in the **Available Filters** list box are used as criteria for determining which alarms/events are displayed on the grid in runtime. If the **Display active filter name(s) on grid's title bar** check box is selected, then the names of the filters that are checked will appear on the grid's title bar while in runtime. To configure a filter, either click the **Add Filter...** button to set a new filter or select an existing filter and click the **Config Filter...** button to modify the filter. Either action opens the **Record Filter** dialog box, shown below.

Record Filter

General

Filter Name:

Column Name:

Filter Type

☐ Inclusive Filter
☒ Exclusive Filter
☐ Custom Filter

Excluded String/Number Values

Available Values

- ☒ 'Condition'

Check All UnCheck All

Excluded DateTime Values

☒ Today
☐ Single Day:
☐ Date Range: From: To:
☐ Last: Days
 Hours
 Minutes

SQL WHERE statement

OK Cancel

Record Filter Dialog

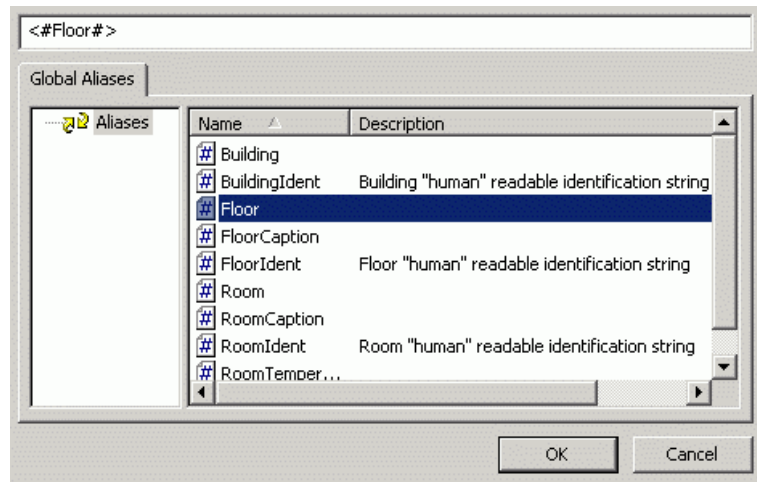
The decision of whether to use string/number values or date/time ranges for filtering depends on the column selected in the **Column Name** drop-down list. If **[EventTime]** is selected from this list, then a date/time range specified in the **Included Date/Time Values** field is used for the filtering criteria. For any other **Column Name** selection, the string/number values that are checked in the **Available Values** list box are used for the filtering criteria. You will notice, as you select different values for **Column Name** that the list of available values in the list box changes.

Three types of filtering are available:

Inclusive Filter: If an inclusive filter is used, then the alarm conditions that meet the criteria set by the items selected in the **Available Values** list box or the alarms appearing within the time range specified in the **Included Date/Time Values** field pass through the filter and are displayed in the alarm report grid in runtime.

Exclusive Filter: If an exclusive filter is used, then the matching alarms do not pass through the filter and are not displayed in runtime.

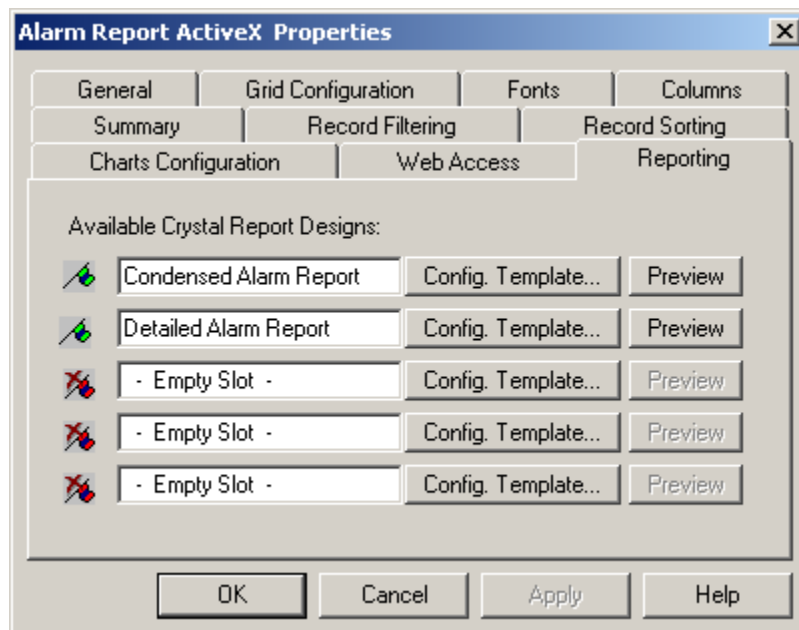
Custom Filter: If custom filtering is used, then you have the option of typing a **SQL WHERE statement** in the text box at the bottom of the dialog box to specify what type of values can pass through the filter. You can also select a global alias to use in the SQL WHERE statement. Clicking the **#** button opens the **Global Alias Browser**, shown below, which allows you to choose from a list of aliases in the Global Aliasing System Engine.



Selecting Global Aliases From the Unified Browser

Reporting

The Report ActiveX Control uses Seagate Crystal Reports for versatile, neat, and easy-to-manage alarm reports. From the **Reporting** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, you can configure templates for Crystal Reports. Clicking the **Preview** button next to any configured template launches Crystal Reports and displays a sample report matching the criteria set in the template configuration.



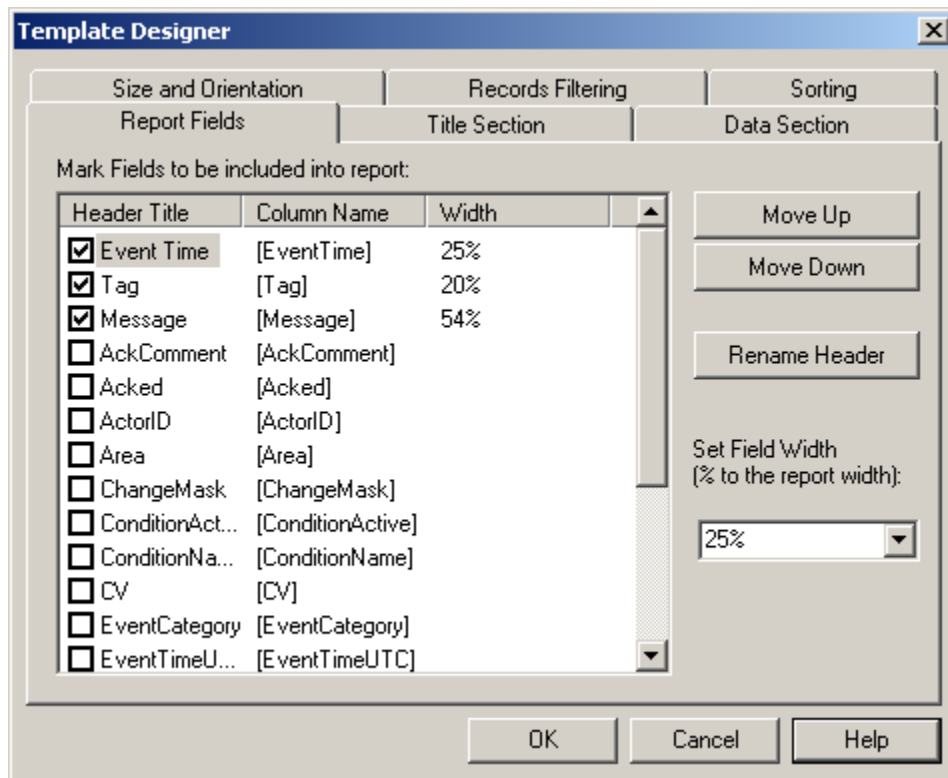
Alarm Report ActiveX Reporting Tab

Clicking on any of the **Config. Template...** buttons opens the **Template Designer** dialog box, which contains the following tabs:

- **Report Fields**
- **Title Section**
- **Data Section**
- **Size and Orientation**
- **Records Filtering**
- **Sorting**

Report Fields

As the **Template Designer** dialog box opens, it defaults to the **Report Fields** tab, shown below. Here you can configure which alarm columns to report, the order in which they are to be reported, and the amount of space each category will take on the report.



Report Fields Tab of Template Designer Dialog Box

To select columns to be reported, simply check the appropriate items in the list box. To change the order of an item, highlight the item and click the **Move Up** and **Move Down** buttons. You will see the item move up and down the list as you do so. When you select an item, you can also specify a width as a percentage of the width of the entire report. You can rename the header title by clicking the **Rename Header** button and filling in the appropriate text, or you can simply click on the header text itself and wait for the cursor to appear.

Title Section

The **Title Section** tab of the **Template Designer** dialog box, shown below, allows you to configure a title for your report. Simply adjust the text, colors, borders, font, shadowing, and alignment of your title and subtitle to match the desired appearance.

The screenshot shows the 'Template Designer' dialog box with the 'Title Section' tab selected. The dialog has three main tabs: 'Size and Orientation', 'Records Filtering', and 'Sorting'. Below these are three sub-tabs: 'Report Fields', 'Title Section', and 'Data Section'. The 'Title Section' sub-tab is active, showing settings for the 'Report Title' and 'Report SubTitle'. For the 'Report Title', the font is 'Arial, 32 pt.', text is 'ICONICS Alarm Report', alignment is 'Center', and it has a 'Drop Shadow' checked. The borders are set to 'Single' on all four sides. For the 'Report SubTitle', the font is 'Arial, 14 pt.', text is 'Condensed Alarm Report', alignment is 'Left', and 'Include Current Table Name' is checked. The borders are set to 'No Line' on all four sides. The dialog also includes 'OK', 'Cancel', and 'Help' buttons at the bottom.

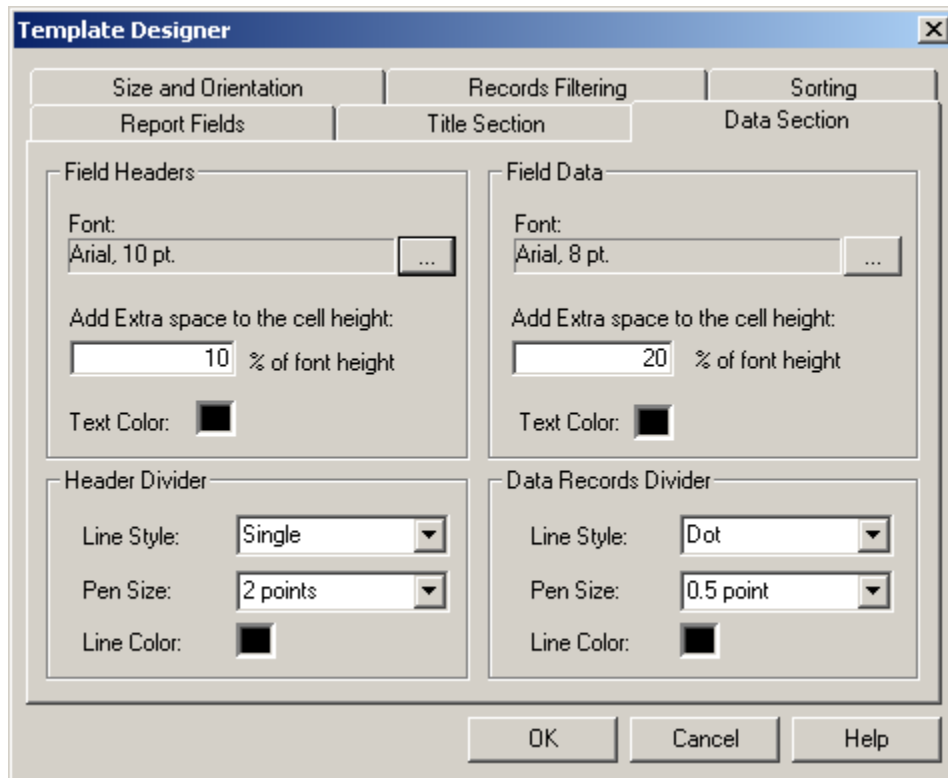
Report Title		
Font: Arial, 32 pt.	Fore Color: [Black]	Alignment: Center
Text: ICONICS Alarm Report	Back Color: [White]	Left: 15 %
☒ Drop Shadow	Border Color: [Black]	Top: 5 %
Left Border: Single	Top Border: Single	Width: 70 %
Right Border: Single	Bottom Border: Single	Height: 50 %

Report SubTitle		
Font: Arial, 14 pt.	Fore Color: [Black]	Alignment: Left
Text: Condensed Alarm Report	Back Color: [White]	Left: 0 %
☒ Include Current Table Name	☐ Drop Shadow	Top: 0 %
Left Border: No Line	Top Border: No Line	Width: 70 %
Right Border: No Line	Bottom Border: No Line	Height: 100 %

Title Section Tab of Template Designer Dialog

Data Section

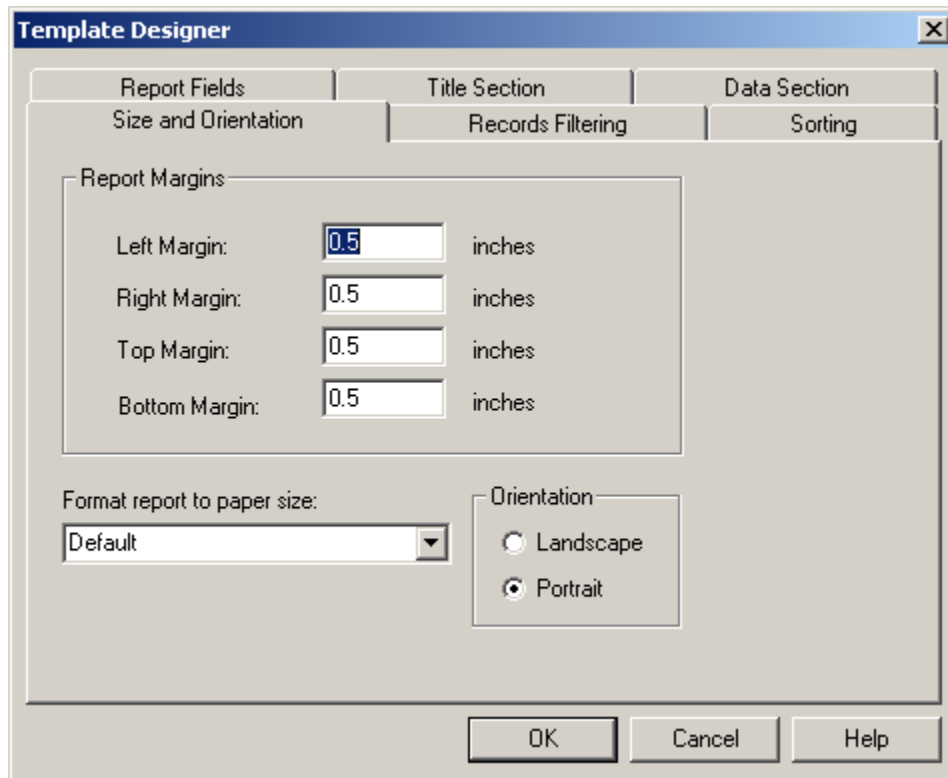
The **Data Section** tab of the **Template Designer** dialog box, shown below, configures the appearance of the headers and fields of your reported data. Adjust the font, cell padding, and text color of the column headers in the **Field Headers** frame. You can change the style, thickness, and color of the header dividers by adjusting the **Line Style**, **Pen Size** and **Line Color** items in the **Header Divider** frame. To adjust the appearance of the reported data and dividers, make the appropriate modifications to the **Field Data** and **Data Records Divider** frames.



Data Section Tab of Template Designer Dialog Box

Size and Orientation

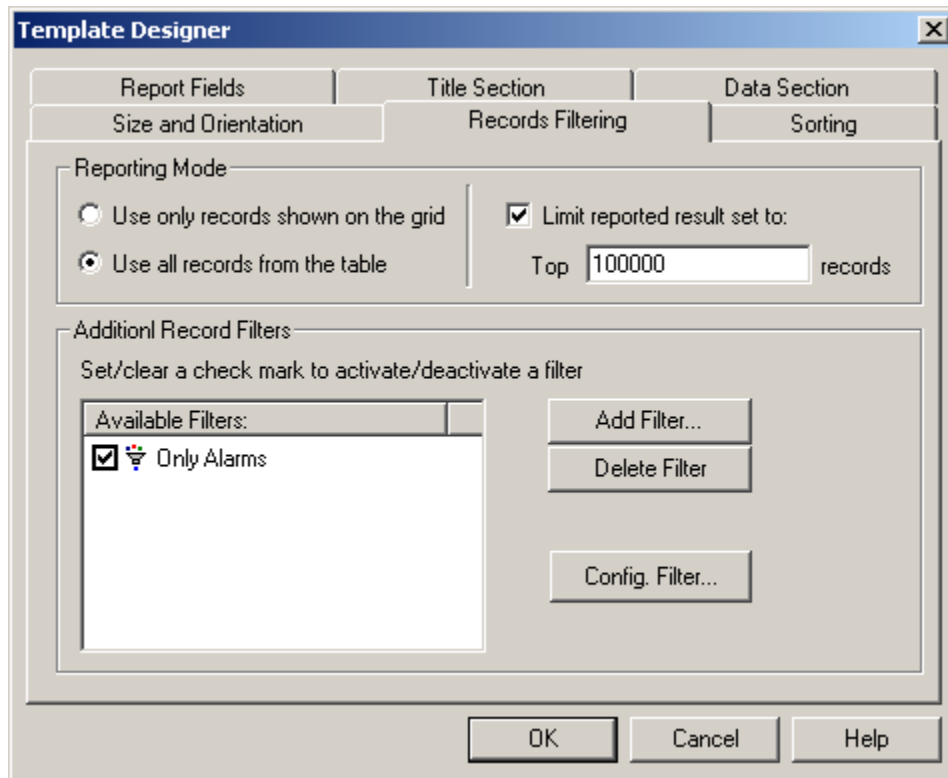
The **Size and Orientation** tab of the **Template Designer** dialog box, shown below, adjusts the margins for the final report. Simply fill in the margin fields as needed. Adjust the paper size by selecting an item from the **Format report to paper size** drop-down list. (The default is 8.5 x 11 inches.) You can also specify a **Portrait** (vertical) or **Landscape** (horizontal) orientation for the report.



Size Tab of Template Designer Dialog Box

Records Filtering

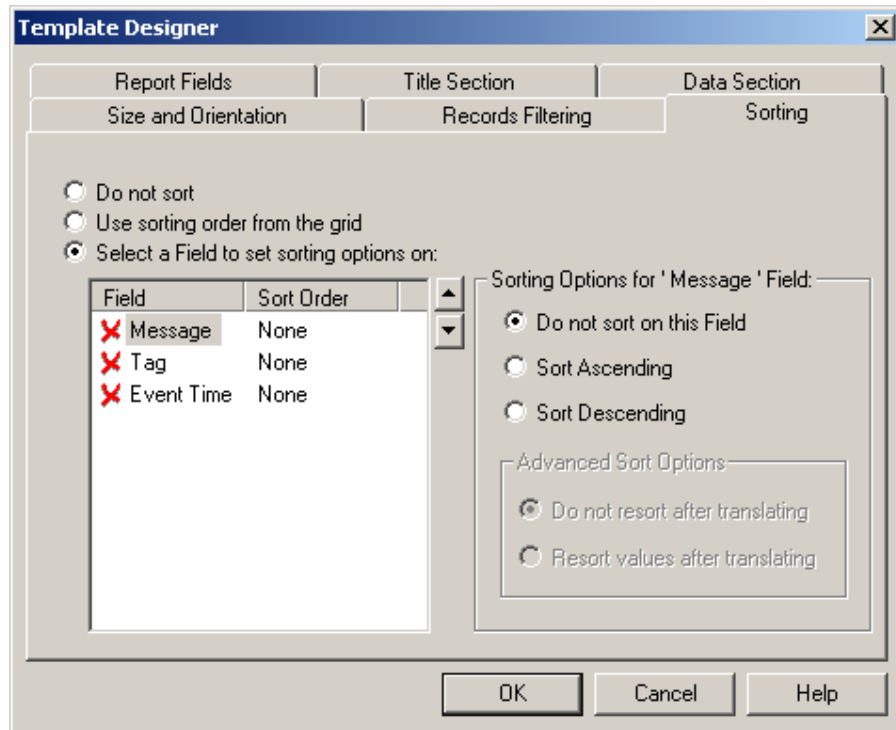
You can configure filters for your reports on the **Records Filtering** tab of the **Template Designer** dialog box, shown below. Records can be reported either from the grid or from the alarm table. One level of filtering marks the difference between reporting alarms from the grid and reporting alarms from the table. The alarms that make it from the table to the grid are the ones that pass through the filter defined in the **Record Filtering** tab of the **Alarm Report ActiveX Properties** dialog box. The records chosen, whether from the alarm table or from the grid, can be filtered for reporting in the **Available Filters** list. (In case you are reporting only records shown on the grid, this is a second level of filtering from the alarm table to the report) The adding, deleting, and configuring of filters is handled the exact same way as on the **Record Filtering** property tab.



Records Filtering Tab of Template Designer Dialog Box

Sorting

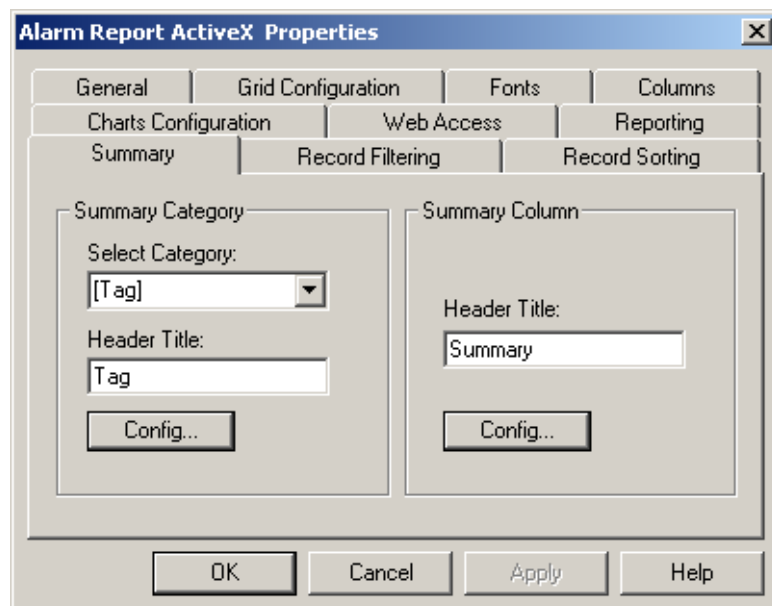
The **Sorting** tab of the **Template Designer** dialog box, shown below, provides three options for sorting alarm records in your report. You can choose not to sort at all by selecting the **Do not sort** option. If the **Use sorting order from the grid** option is selected, then the report sorting defaults to the sorting defined on the **Records Sorting** tab of the **Alarm Report ActiveX Properties** dialog box. You can override the grid sorting by choosing **Select a Field to set sorting options on**. Set your sorting parameters with the list box and options below. For detailed information on configuring your custom sorting, see the **Records Sorting** section.



Sorting Tab of Template Designer Dialog Box

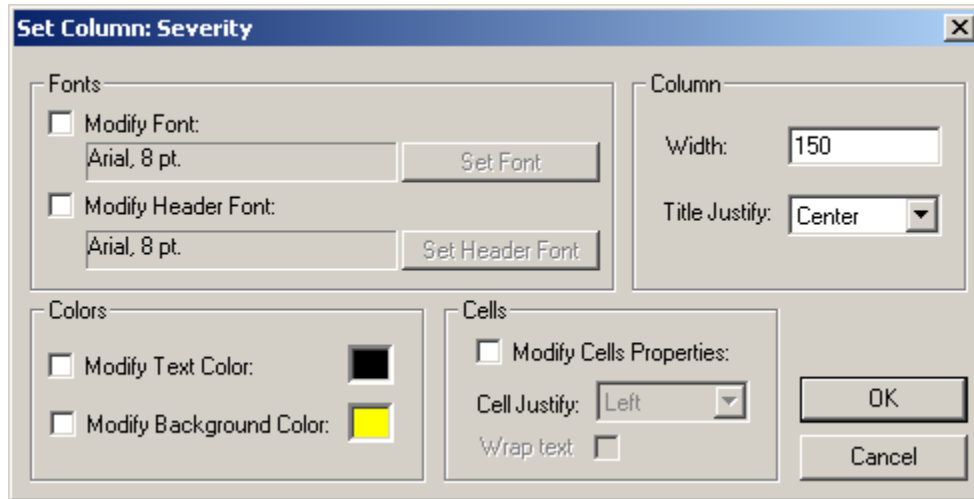
Summary

The **Summary** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, configures the content and appearance of summary information to be displayed on the alarm report grid in runtime mode. The **Select Category** drop-down list specifies the category of alarm to be displayed in the summary.



Alarm Report ActiveX Summary Tab

You can also select a title and click the **Config...** button. This opens the **Set Column** dialog box for the selected category, as shown below. Here you can adjust the font, color, and appearance of the selected alarm category. The title and configuration on under **Summary Column** are used to modify the appearance of the values associated with the **Summary Category**.



Setting the Summary Column/Category Characteristics

Web Access

The **Web Access** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, allows you to access the currently connected Alarm Logger database over the Internet via a **Remote Data Service (RDS)**, or **Remote Database Agent (RDA)**. The RDS, which is hosted by an Internet Information Server (IIS), enables the downloading of report data from the logger database to a client over the Internet. Simply specify the URL name or IP address of the primary IIS Web server in the **Primary Internet Information Server for RDA** field, as shown in the figure below. The **Remote Recordset Page Size** field specifies the size of the recordset portion in the database records.

Note: You also may designate a redundant IIS Web server by specifying the URL name or IP address of the redundant IIS Web server in the **Backup Internet Information Server for RDA** field, as shown in the figure below.

Summary	Record Filtering	Record Sorting	
General	Grid Configuration	Fonts	Columns
Charts Configuration	Web Access	Reporting	

☐ Access database over the Internet via Remote Database Agent

Primary Internet Information Server for RDA

Backup Internet Information Server for RDA

 (Example: http://WebHMI.SomeCompany.com)

Remote Recordset Page Size (database records) :

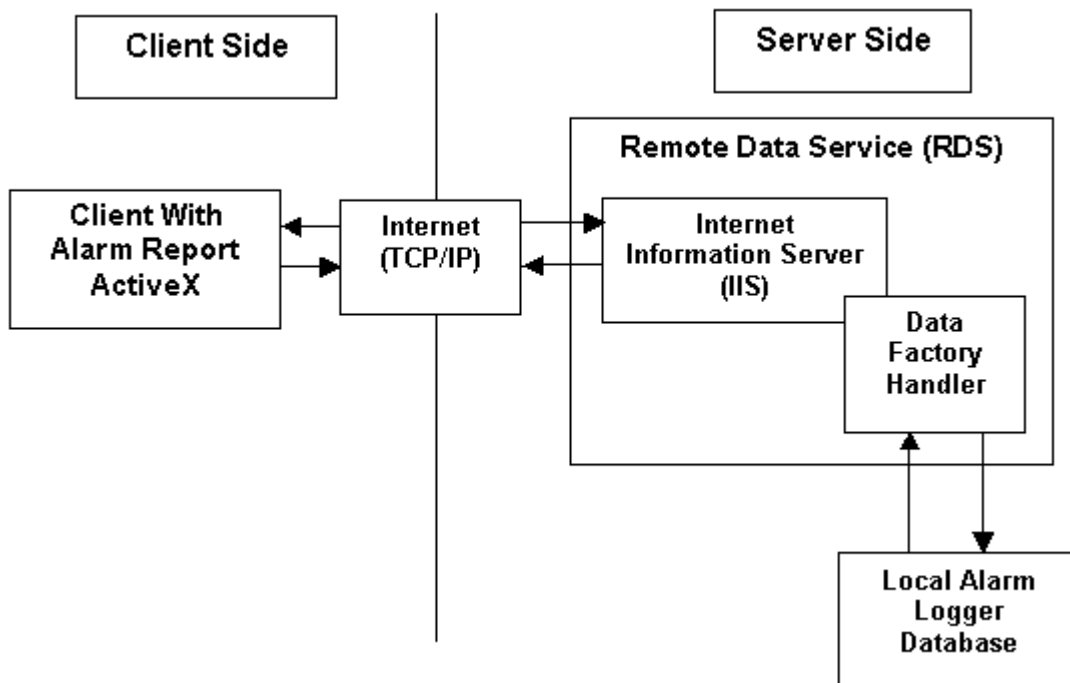
OK Cancel Apply Help

Alarm Report ActiveX Web Access Tab

When Internet Access to the logger database is enabled, the client makes a request over the Internet to the IIS Remote Data Service. The RDS uses a special component, called the Data Factory Handler, to relay the request to the local logger database on the server, as shown in the figure below. The RDS retrieves the relevant information for the alarm report from the logger database so that the client can download the data from the server over the Internet. The client machine then displays the data in the Report ActiveX per the client's report configuration settings.

In most typical scenarios, the Report ActiveX configuration is done on the server side. Those steps include:

- Connection to the database is established by clicking the **Connection Parameters** button on the **General** tab of the **Report ActiveX Properties** dialog and using the **Database Connection** dialog.
- Internet access settings are enabled on the **Web Access** tab.
- Configuration data saved as an *.awr file or a *.gdf display containing the Report ActiveX are published to an HTML document on a WebHMI server.



Downloading Alarm Report Data From a Remote Logger Database

Tips for Remote Data Access

A good understanding of Remote Data Access principles is a key element to building a successful solution for a Web-enabled alarm reporting system. Since the volume of alarm databases can reach hundreds of megabytes, it is very important to keep in mind the following aspects.

Maximum Number of Downloadable Database Records and Adequate Record Filtering

The setting for the maximum number of records to download (Nmax) indicates that any database query operations would return no more than Nmax alarm records (lines to the grid). If more than Nmax records from the current table match the current filtering criteria, then only the portion of the record set containing top Nmax records is returned. To avoid such record set cut-off, strong restrictive record filters should be used all the time. Since the record filters are processed on the database side, the database server uses all existing records to build the record set.

Proper Choice of the Back-end Database

It should be noted that the use of Microsoft Access databases (*.mdb) is not recommended for Web Access. Poor memory management in Microsoft Jet Engine creates a serious scalability problem for any type of Web database

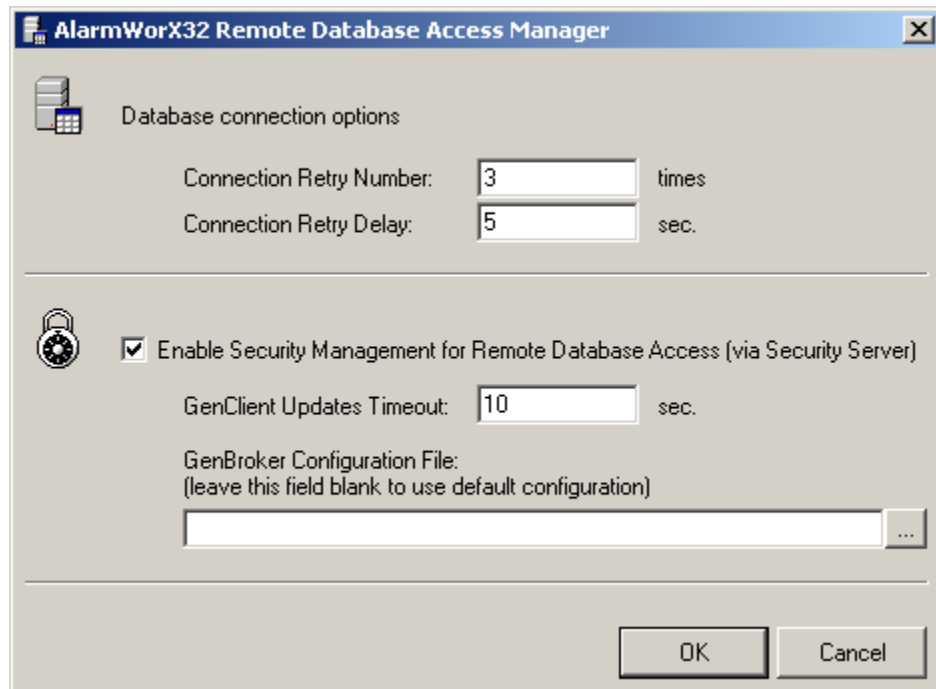
application. **The recommended back-end database for Web alarming is MSDE (Microsoft Desktop Engine) or Microsoft SQL Server.**

Table Management in Alarm Logger Configuration

To provide further optimization for the Alarm Logger database operations, the table management should be enabled in the logger configuration. See the Alarm Logger Configurator documentation for more information.

Using the Remote Database Access Manager

The **AlarmWorX32 Remote Database Access Manager**, shown in the figure below, is a separate utility (the **AWXRep32RDSSM.exe** file) located in the GENESIS32 Bin directory. This utility configures the communication parameters for the Data Factory Handler (described above) so that the AlarmWorX32 Report ActiveX can connect to a remote Alarm Logger database and retrieve data from the database. It also enables security for remote database access via the GENESIS32 Security Server.



AlarmWorX32 Remote Database Access Manager

Database Connection Options

The **Database Connection Options** section of the AlarmWorX32 Remote Database Access Manager sets the following connection parameters for the Data Factory Handler:

- **Connection Retry Number:** Specifies the maximum number of times the Data Factory Handler will try to reconnect to the Alarm Logger database in case the connection fails.
- **Connection Retry Delay:** Sets the amount of time (in seconds) the Data Factory Handler waits between subsequent connection retries to the Alarm Logger database.

Security Management

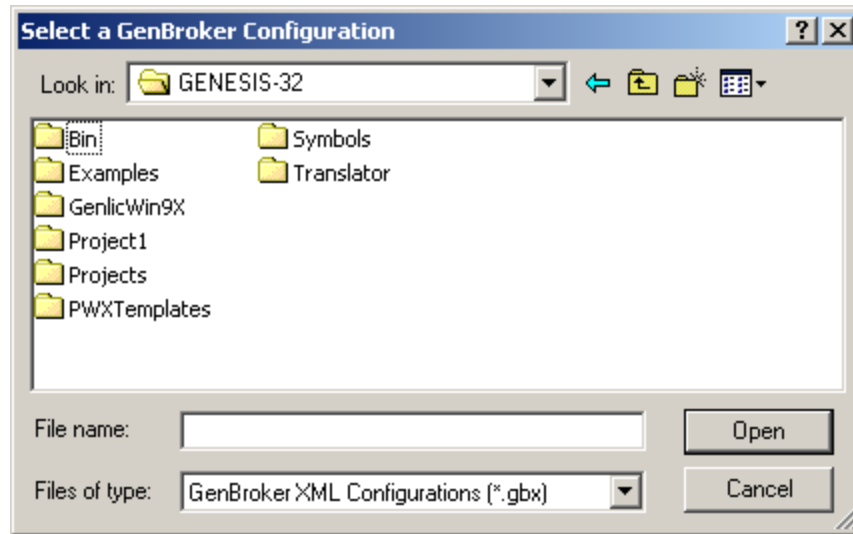
The AlarmWorX32 Remote Database Access Manager allows you to enable security for remote database access via the GENESIS32 Security Server. Check the **Enable Security Management for Remote Database Access (via Security Server)** check box. When security is enabled, each time a user tries to access the Alarm Logger Database, the Security Server will check to verify whether that user is allowed to have remote access to the database.

Note: For information about the Security Server configuration and login, please see the Security Configurator and Security Login Help documentation.

Because the database is accessed over the Internet via TCP/IP communications, GenBroker communications over OPC over TCP/IP must also be enabled. Thus, you must specify a **GenBroker Configuration File (.gbx)**. Clicking the ... button to the right of this field allows you to browse for a GenBroker configuration file, as shown in the figure below. If you do not specify a GenBroker configuration file, or if no Security Server is specified in the GenBroker configuration, then the default system GenBroker configuration file will be used.

Note: If you are using WebHMI, it is recommended that the Security Server be located on the same machine as the WebHMI Server. For information about how to configure GenBroker communications, please see the GenBroker Help documentation.

The **GenClient Updates Timeout** sets the maximum amount of time (in seconds) the Data Factory Handler will wait for Security Server validation of a data request from the client (i.e. the client with the Alarm Report ActiveX installed) before returning an "Access Denied" error.



Selecting a GenBroker Configuration File

Security

To access the GENESIS32 Security Server, either select **Security > Configuration** from the AlarmWorX32 Container **Tools** menu, or click the Security button on the toolbar. This opens the Security Server Login dialog box, as shown below.

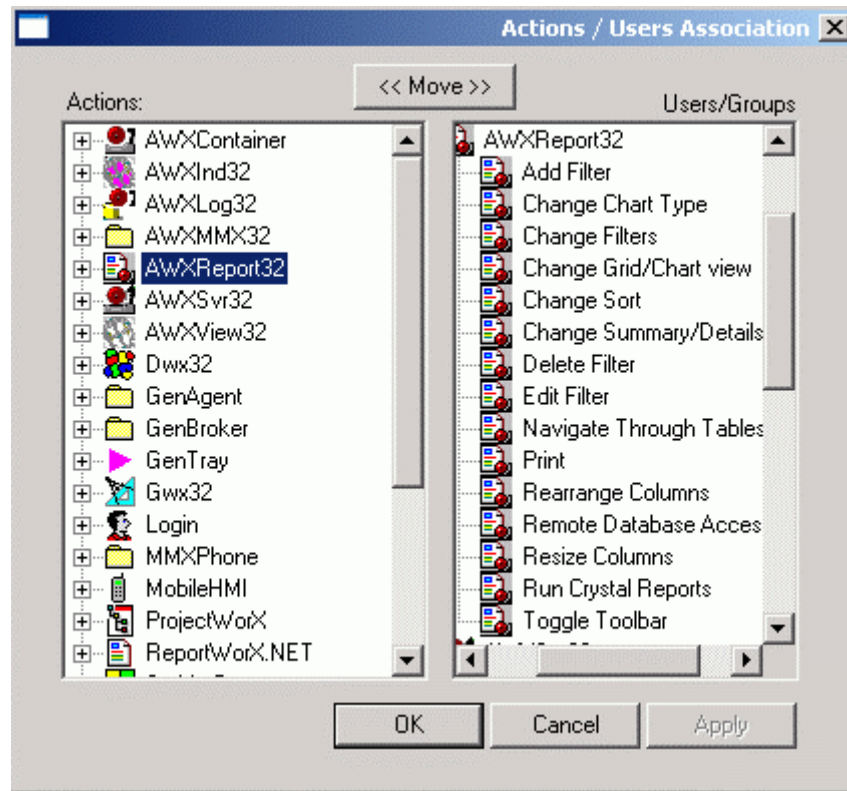


Security Server Login

Enter your **User Name** and **Password**, and then click **OK**. This opens the Security Server Configurator, where you can access the Security Server. Depending on your configured level of security, you may or may not be able to edit components of the Security Server.

Note: For more information about security, please see the Security Configurator help documentation.

Once a user has gained access, the Security Server dialog box will appear. From the **Edit** menu, choose **Application Actions**. The **Actions/Users Association** dialog box appears as shown below.



Editing Security Actions and User Associations

The Security Server can be set up to allow access to several different actions from this dialog. To set the options for the AlarmWorX32 Report ActiveX, select **AWXReport32** from the list in the right pane. This brings up the list in the left pane, which includes the following options:

- Add Filter
- Change Chart Type
- Change Filters
- Change Grid/Chart View
- Change Sort
- Change Summary/Details View
- Delete Filter
- Edit Filter
- Navigate Through Tables
- Print
- Rearrange Columns
- Resize Columns
- Remote Database Access
- Run Crystal Reports
- Toggle Toolbar

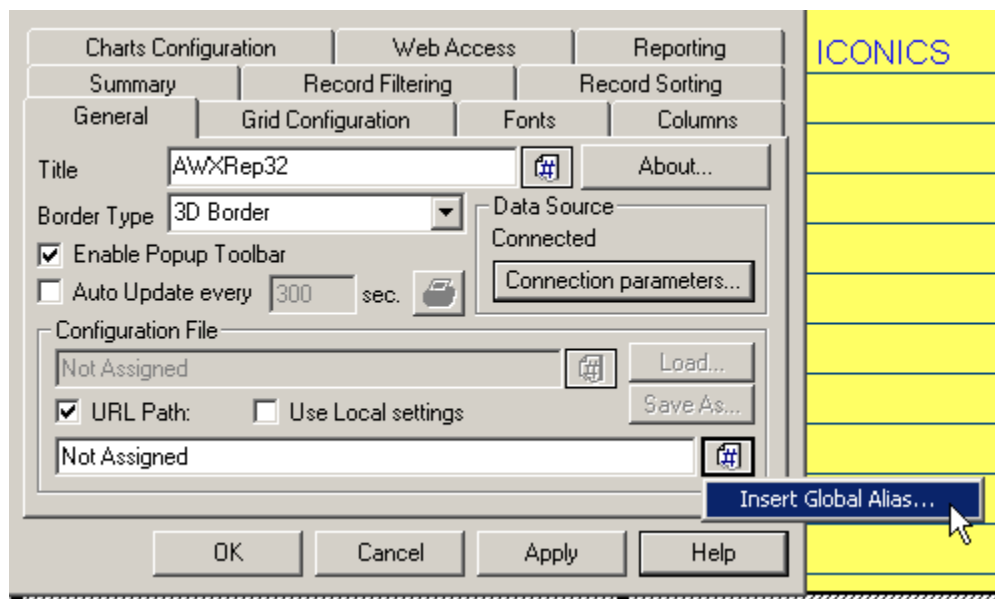
Global-Aliasing Support

The AlarmWorX32 Report ActiveX supports Global Aliasing by interfacing to the ICONICS Global Aliasing System Engine Server.

Currently the AlarmWorX32 Report ActiveX supports machine-level and process-level aliasing. All fields that support aliasing have been interfaced to the Global Alias Browser dialog, which is evoked by clicking on the # button in the configuration.

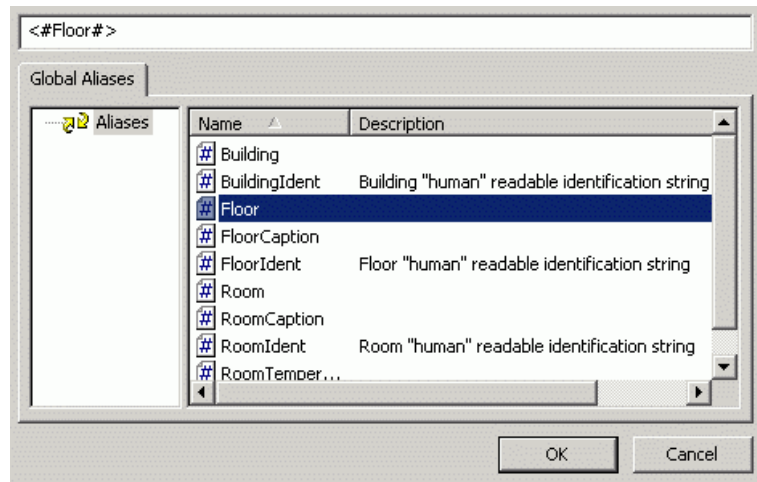
When specifying a global alias in the **Alarm Report ActiveX Properties** dialog box, you can also select a global alias from the **Global Alias Browser**, which includes all global aliases in the Global Aliasing System database. This eliminates the need to manually type in the alias name. Clicking the # button and selecting **Insert Global Alias** from the pop-up menu opens the Global Alias Browser, as shown in the figure below. The Global Alias Browser is available for the following:

- **Title, Configuration File, and URL Path** sections of the **Alarm Report ActiveX Properties** dialog box **General** tab
- Records filtering



Opening the Global Alias Browser in the Report Properties

All global aliases that are configured in the Global Alias Engine Configurator are conveniently available to choose from inside the Unified Data 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.

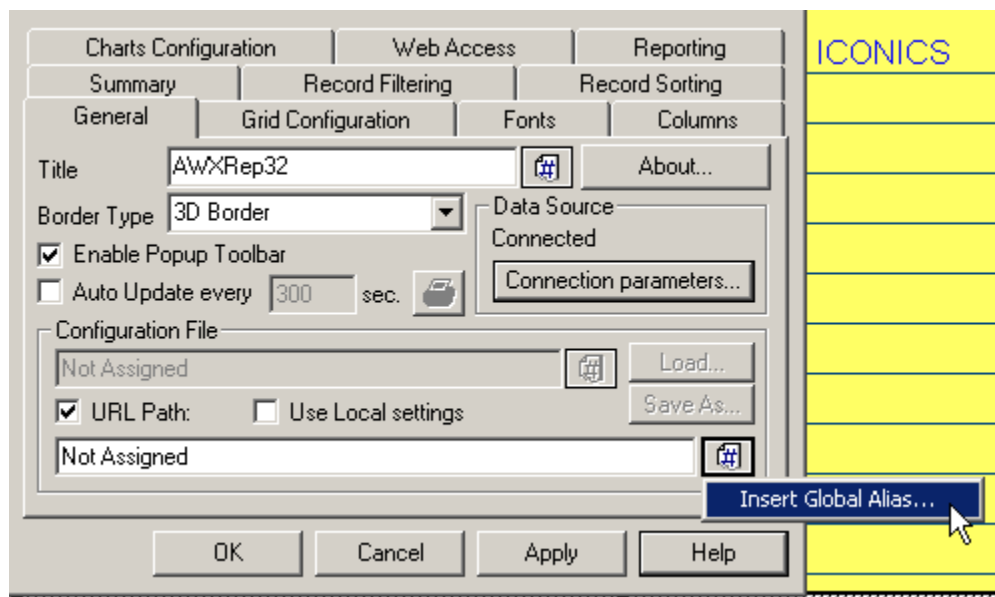


Selecting an Alias From the Global Alias Browser

Asynchronous Downloading features

The AlarmWorX32 Report ActiveX supports asynchronous downloading of configuration data when used inside ICONICS ActiveX document servers, such as GraphWorX32, to produce documents, which can be viewed using Microsoft Internet Explorer.

To enable this feature, select **URL Path** from the **General** tab of the **Alarm Report ActiveX Properties** dialog box, shown below, and supply a URL path. In the case of viewing ActiveX documents within an intranet setup, you can supply a server name and directory specifying the location of the AlarmWorX32 Report configuration data (must be a shared directory), by using the convention "file:\\Server Name\\Directory\\File Name.awr."



Alarm Report ActiveX Properties: General Tab

Runtime Mode

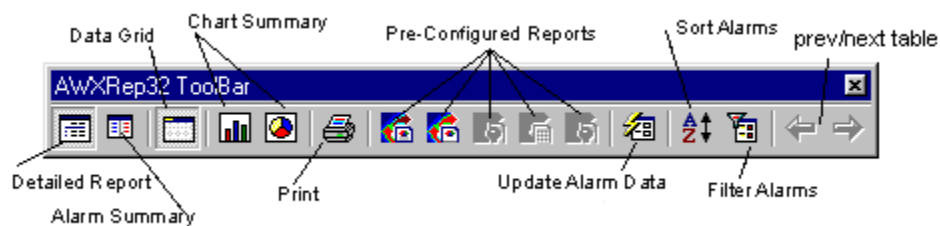
To enter runtime, either click the **Enter Runtime** button on the toolbar or select **Runtime Mode** from the **Actions** menu. Once in runtime, the following data grid appears.

AWXRep32 - [EventLog]			
	EventTime	Tag	Message
	12/3/99 1:09:13 PM	Scale	Reading on the sca
	12/3/99 1:09:13 PM	Tank1	The level in tank1 is
	12/3/99 1:09:13 PM	Ash Content	Ash Content of Tan
	12/3/99 1:09:13 PM	Box Line	Belt1 on the Box Li
	12/3/99 1:09:13 PM	Coolant Level	Warp core breach
	12/3/99 1:09:13 PM	Pump1	Pressure in pump1
	12/3/99 1:09:13 PM	Alkaline Level	Alkaline level in Tan
	12/3/99 1:09:25 PM	Coolant Level	Coolant leak detect
	12/3/99 1:09:41 PM	Humidity	Core humidity is hig
	12/3/99 1:09:45 PM	Tank PSI	The PSI in Tank1 is
	9/19/00 12:23:19 PM	AWXSvr32	ICONICS Alarm/Vor
	12/3/99 1:09:45 PM	Humidity	Core humidity is too
	12/3/99 1:09:45 PM	Belt Speed	Belt Speed of Pump

Runtime Mode

Note: If runtime mode does not show any data, make sure you have established a connection with an Alarm Logger database. You can connect to a database by clicking **Connection Parameters** on the **General** tab of the **Alarm Report ActiveX Properties** dialog box while in configuration mode. Also, if you deleted one of the three default columns (EventTime, Tag, or Message), the Report ActiveX will be disabled in runtime mode.

Double-clicking anywhere on the Alarm Report ActiveX during runtime mode opens the **Alarm Report** toolbar, as shown in the figure below. Using the toolbar, you can execute certain operations and change how the data are viewed.



Alarm Report Toolbar

There are two different views to choose from: Data Grid or Chart Summary. When viewing the Data Grid, you can further choose from a detailed report or a summary report. These views are described in the sections below along with the other tool button options.

Data Grid
Chart Summary
Printing Current Alarms
Choosing a Crystal Report Template
Updating Alarm Data
Adjust Columns width
Sorting and Filtering using AWXRep32 ToolBar

Data Grid

By default, the Alarm Report ActiveX runtime screen displays the **Data Grid > Details** view, as shown below. The Data Grid displays all data for the alarms in the report. The reported elements are determined by the columns selected during configuration. Additionally, you may be able to change the view slightly by changing the width of the columns (if this option was enabled during configuration).

AWXRep32 - [EventLog]		
EventTime	Tag	Message
12/5/2000 9:37:31 AM	Alkaline Level	Alkaline level in Tank1 is too low. Contents are basic
12/5/2000 9:37:31 AM	Ash Content	Ash Content of Tank1 is too low. Contents are acidic
12/5/2000 9:38:11 AM	Belt Speed	Belt Speed is normal
12/5/2000 9:37:31 AM	Box Line	Belt1 on the Box Line is slipping.
12/5/2000 9:38:11 AM	Compressor	Compressor gauge is reading normal
12/5/2000 9:38:11 AM	Coolant Level	Coolant leak detected.
12/5/2000 9:37:55 AM	Critical Mass	Critical Mass is ok.
12/5/2000 9:38:11 AM	Humidity	Humidity is normal.
12/5/2000 9:38:11 AM	Level Gauge	Level gauge is normal.
12/5/2000 9:37:55 AM	Pressure	Pressure in valve1 is normal
12/5/2000 9:37:31 AM	Pump1	Pressure in pump1 is too low
12/5/2000 9:37:31 AM	Scale	Reading on the scale is low.
12/5/2000 9:38:11 AM	Tank PSI	PSI in Tank1 is normal.
12/5/2000 9:37:31 AM	Tank1	The level in tank1 is too low.
12/5/2000 9:38:11 AM	""Pressure"	""Awx\Deviation\Pres\HiHi"

Data Grid Details View in Runtime Mode

To view a summary of the data, double-click the grid and then click the **Alarm Summary** button on the **Alarm Report** toolbar. The view changes to display only the Tag name and the Alarm Summary, as shown in the figure below. The data are listed, from highest to lowest, by the number of alarms.

AWXRep32 - [EventLog (Summary)]	
Tag	Summary
``Pressure"	155
``Temperature"	161
Alkaline Level	1
Ash Content	1
Belt Speed	41
Box Line	1
Compressor	35
Coolant Level	21
Critical Mass	22
Humidity	42
Level Gauge	41
Pressure	25
Pump1	1
Scale	1
Tank PSI	41
Tank1	1

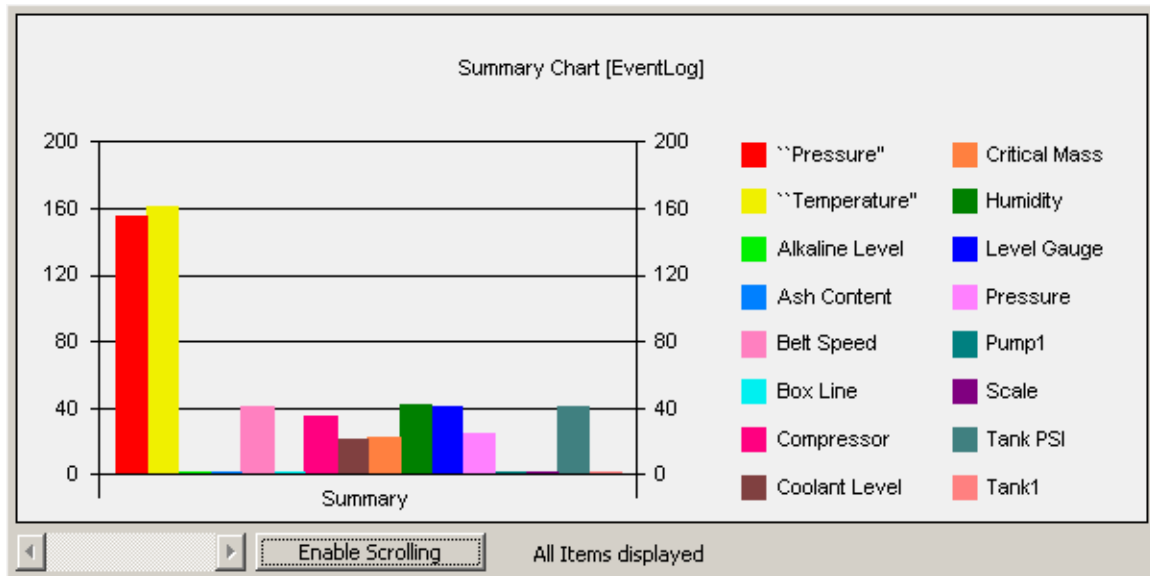
Data Grid Summary View in Runtime Mode

Chart Summary

To view a bar chart or pie chart summary of the Alarm Report data during runtime mode, double-click the Alarm Report ActiveX and then click the **Chart Summary** buttons on the **Alarm Report** toolbar. The number of items charted depends on the options selected on the **Record Filtering** tab of the **Alarm Report ActiveX Properties** dialog box. If no filters are enabled, all alarm types are charted.

Bar Charts

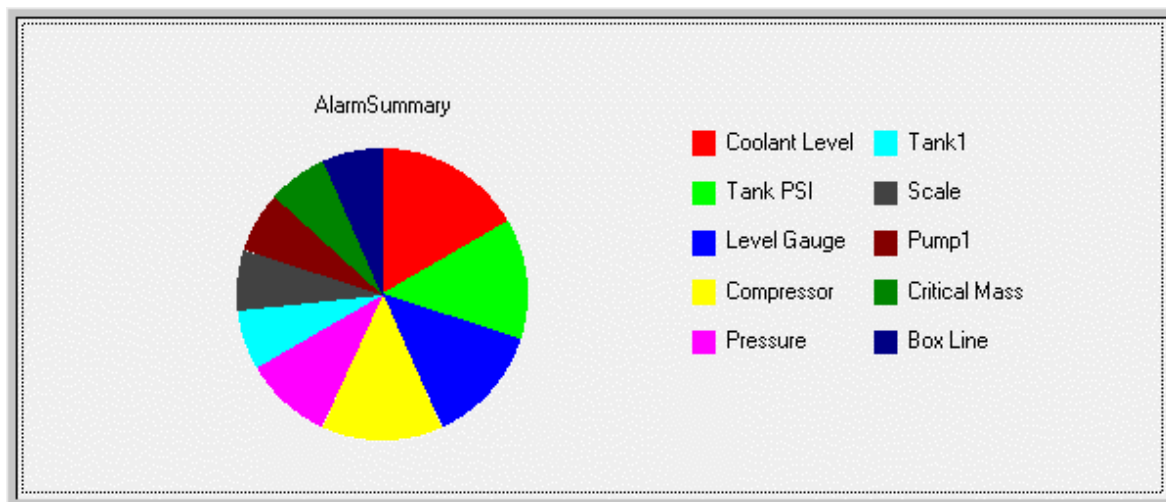
A **Bar Summary Chart**, shown in the figure below, shows how many alarms of each type are present in the database that is being reported. If scrolling was enabled in the **Charts Configuration** tab of the **Alarm Report ActiveX Properties** dialog box, an **Enable Scrolling** button appears at the base of the bar chart summary, as shown below. Click this button to scroll left or right between bars on the chart using the arrows.



Bar Chart Summary

Pie Charts

A **Pie Summary Chart**, shown in the figure below, shows what percentage of each alarm type exists in the report.



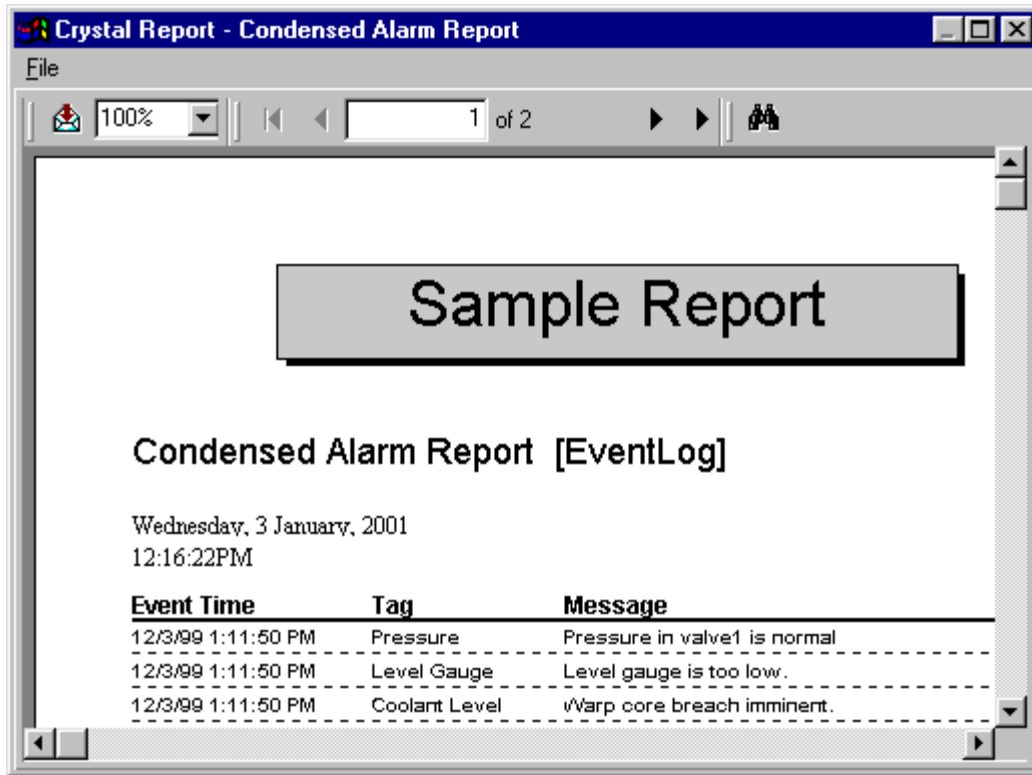
Pie Chart Summary

Printing Current Alarms

The **Print Current Alarms** button on the **Alarm Report** toolbar opens the standard **Print** dialog box, where you can choose your printing settings. The current view will be printed with the date, time, description, and page number at the top of the page.

Choosing a Crystal Report Template

If you have configured any reporting templates on the **Reporting'** tab in configuration mode, you will see one to five reports on the **Alarm Report** toolbar from which to choose. Clicking on a report button launches a Crystal Reports viewer, showing the selected report. From there, all of the features of Crystal Reports are at your disposal for printing, formatting, and saving your report. A sample Crystal Report is shown below.



Sample Report

Updating Alarm Data

Clicking the **Update Alarm Data** button on the **Alarm Report** toolbar refreshes the data shown in the Alarm Report ActiveX during runtime with the most current alarm data from the connected Alarm Logger database.

Adjusting Column Width

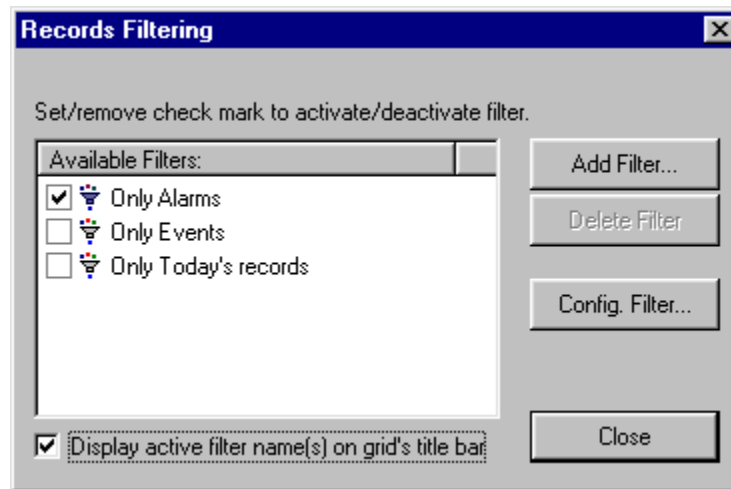
During runtime mode, you can adjust each column in the Alarm Report ActiveX data grid independently by moving the cursor over the vertical line between each column (a double arrow will appear) and moving the line while holding down the mouse button.

Sorting and Filtering Data Using the Alarm Report Toolbar

The last two buttons before the left and right arrows on the **Alarm Report** toolbar are used to sort and filter the alarm data.

Clicking the **Sort Alarms** button on the toolbar opens the **Sorting Fields** dialog box accessed through the **Records Sorting** tab of the **Alarm Report ActiveX Properties** dialog box.

Clicking the **Filter Alarms** button on the toolbar opens the **Records Filtering** dialog box, shown below. For information about adding, removing, and configuring filters, please see the **Record Filtering** section.



Alarm Filtering



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

Automation Interfaces

Available Control Properties

Available Control Methods

Introduction

This reference describes the OLE Automation features available in the AlarmWorX32 Report.

Automation Interfaces

The AlarmWorX32 Report provides a COM interface that allows automation interfaces run from within the ActiveX container to manipulate the Report ActiveX control 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 Report must be made available by choosing **Project > Components** from the main menu in the VB or VBA development environment and selecting **ICONICS AWXRep32 ActiveX** in the list of available components.

The description of every property or method in this reference is followed by the code example for typical use of that property or method. The example code is written in Visual Basic for Applications (VBA) language, built in the AlarmWorX32 container application (AwX32.exe). It assumes that AWXRep32 ActiveX is embedded within **AlarmWorX** display and its name property is set to "AWXRep321."

Available Control Properties

AWXRep32Title

Type: String

Description

Sets/gets title string on General property page.

Example

```
Dim strNewTitle as String  
ThisDocument.AWXRep321.AWXRep32Title=strNewTitle
```

AppearanceMode

Type: Long

Description

Determines the appearance of data grid. This property can be one of the following:

0 - flat look
1 - 3D look

Example

Sets 3D look for the grid.

```
ThisDocument.AWXRep321.AppearanceMode=1
```

BackColor

Type: OLE_COLOR

Description

Sets/gets grid background color.

Example

Sets grid background color to cyan.

```
ThisDocument.AWXRep321.BackColor=RGB(0,255,255)
```

BorderType

Type: String

Description

Sets/gets control's border appearance. Can be one of the following:

"0" - none
"1" - flat border
"2" - 3D border

Example

Sets 3D look for the control's border.

```
ThisDocument.AWXRep321.BorderType="2"
```

ChartBackColor

Type: OLE_COLOR

Description

Sets/gets summary chart background color.

Example

Sets summary chart background color to white.

```
ThisDocument.AWXRep321.ChartBackColor=RGB(255,255,255)
```

ChartLegendHeight

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartLegendLeft

ChartLegendTop

Type: Short

Description

These properties determine the location of the corresponding chart element within the chart area. The coordinates are given in percents of the total width or height of the chart area, assuming that the coordinate origin (point {0, 0}) is located at the left top corner of the chart area.

ChartLegendVisible

Type: Boolean

Description

These properties set/get the visibility of chart legend and/or title. Setting either property to FALSE will hide the corresponding chart element.

Default value: TRUE

ChartLegendWidth

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartPlotHeight

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartPlotLeft

ChartPlotTop

Type: Short

Description

These properties determine the location of the corresponding chart element within the chart area. The coordinates are given in percents of the total width or height of the chart area, assuming that the coordinate origin (point {0, 0}) is located at the left top corner of the chart area.

ChartPlotWidth

Type: Short

Description

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartTitle

Type: String

Description

Sets/gets a string value that appears in the summary chart title area.

Example

```
ThisDocument.AWXRep321.ChartTitle="Alarm Summary By Tags"
```

ChartTitleHeight**Type: Short****Description**

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartTitleLeft**ChartTitleTop****Type: Short****Description**

These properties determine the location of the corresponding chart element within the chart area. The coordinates are given in percents of the total width or height of the chart area, assuming that the coordinate origin (point {0, 0}) is located at the left top corner of the chart area.

ChartTitleUpdateOnDataSelect**Type: Boolean****Description**

Sets/gets the value of a flag that determines whether the chart title is updated when user clicks on a data series (either bar or pie segment) inside the plot. If set to TRUE (default), the name of selected data series is attached to the chart title.

ChartTitleVisible**Type: Boolean****Description**

These properties set/get the visibility of chart legend and/or title. Setting either property to FALSE will hide the corresponding chart element.

Default value: TRUE

ChartTitleWidth**Type: Short****Description**

These properties determine the dimensions of the corresponding chart element. The width and height of a particular chart element are given in percents of the total width or height of the chart area.

ChartType

Type: Integer

Description

Sets/gets the type of alarm summary chart. Can be one of the following:

0 - bar type

1 - pie type

Example

Sets summary chart to pie type.

```
ThisDocument.AWXRep321.ChartType=1
```

Columns

Type: String

Description

Sets/gets comma-separated list of database columns to be shown on grid.

Example

This line configures control to show only EventTime, Tag and Message columns.

```
ThisDocument.AWXRep321.Columns="EventTime, Tag, Message"
```

Remarks

If database column name contains one or more space characters, the column name should be enclosed in square brackets, as in this example:

```
ThisDocument.AWXRep321.Columns="[Column 1], [Column 2]"
```

ConnectionString

Type: String

Description

Sets/gets ADO connection string used by AWXReport32 control to access data.

Example

```
Dim strConnString as String
strConnString="Provider=Microsoft.Jet.OLEDB.4.0; _
Data Source=c:\alarms\AWXLog32.mdb; _
User ID=Admin; Password=SYSTEM"
ThisDocument.AWXRep321. ConnectionString = strConnString
```

DisplayActiveFilterName

Type: Boolean

Description

Sets/gets Boolean value that controls whether the active filter(s) name(s) is(are) shown on the grid title bar.

Example

```
ThisDocument.AWXRep321. DisplayActiveFilterName = TRUE
```

DisplayGridTitle

Type: Boolean

Description

Sets/gets Boolean value that controls the visibility of the grid title bar.

Example

```
ThisDocument.AWXRep321. DisplayGridTitle = TRUE
```

EnableToolbar

Type: Boolean

Description

Setting this property to FALSE prohibits the floating toolbar to pop up when user

double-clicks inside the control.

Default value - TRUE.

Example

To disable tools

```
ThisDocument.AWXRep321.EnableTools=FALSE
```

FilteringString

Type: String

Description

Sets/gets SQL WHERE statement that determines filtering of alarm data.

Example

These statements configure control to show only Humidity alarms or Pressure alarms.

```
Dim strFiltrStmt as String
strFiltrStmt="WHERE (Tag='Humidity' OR Tag='Pressure') "
ThisDocument.AWXRep321. FilteringString = strFiltrStmt
```

Font

Type: StdFont

Description

Returns a reference to the control's default font.

Example

To set control's default font to Arial, 14 pt., use the following statements:

```
With ThisDocument.AWXRep321.Font
.Name="Arial"
.Size=14
End With
```

ForeColor

Type: OLE_COLOR

Description

Sets/gets Forecolor property of data grid.

Example

```
ThisDocument.AWXRep321. ForeColor=RGB(0,0,0)
```

GridBackColor

Type: OLE_COLOR

Description

Sets/gets color value used to paint the gaps between rightmost column and right edge of the grid, last row and bottom edge of the grid, and row headers column (the 'back area' of the grid).

Example

```
ThisDocument.AWXRep321. GridBackColor =RGB(0,127,0)
```

GridBorderStyle

Type: Long

Description

Sets/gets the style of data grid border. Can be one of the following:

0 - no border

1 - fixed single border style

Example

```
ThisDocument.AWXRep321.GridBorderStyle=1
```

GridDefRowHeight

Type: Integer

Description

Sets/gets the default row height for data grid.

Example

```
ThisDocument.AWXRep321. GridDefRowHeight=16
```

GridHorizLinesColor and GridVertLinesColor

Type: OLE_COLOR

Description

Sets/gets color for horizontal or vertical grid lines.

Example

```
ThisDocument.AWXRep321. GridHorizLinesColor  
=RGB(127,127,127)
```

```
ThisDocument.AWXRep321. GridVertLinesColor =  
RGB(127,127,127)
```

GridHorizLineStyle and GridVertLineStyle

Type: Integer

Description

Sets/gets style for horizontal or vertical grid lines. Can be one of the following:

0 - no line

1 - solid line

2 - dot line

3 - dashdot line

4 - dashdotdot line

Example

```
ThisDocument.AWXRep321. GridHorizLineStyle =0  
ThisDocument.AWXRep321. GridVertLineStyle = 2
```

GridHorizLinesWidth and GridVertLinesWidth

Type: Integer

Description

Sets/gets the widths for the horizontal or vertical grid lines when corresponding grid line style (property **GridHorizLineStyle** or **GridVertLineStyle**) is set to 1 - solid line. For all other line styles, these properties must be set to 1.

Example

Sets both horizontal and vertical grid lines to solid line style, 2 pixels wide.

```
ThisDocument.AWXRep321. GridHorizLineStyle =1  
ThisDocument.AWXRep321. GridVertLineStyle = 1  
ThisDocument.AWXRep321. GridHorizLinesWidth =2  
ThisDocument.AWXRep321. GridVertLinesWidth = 2
```

GridTitleBackColor

Type: OLE_COLOR

Description

Sets/gets the background color for the grid's title bar.

Example

Sets title bar background color to dark blue.

```
ThisDocument.AWXRep321. GridTitleBackColor =GRB(0,0,127)
```

GridTitleTextColor

Type: OLE_COLOR

Description

Sets/gets the text color for the grid's title bar.

Example

Sets grid's title bar text color to yellow.

```
ThisDocument.AWXRep321. GridTitleBackColor  
=GRB(255,255,0)
```

HeaderColor

Type: OLE_COLOR

Description

Sets/gets the background color for grid's column headers.

Example

Sets color of grid's column header to dark green.

```
ThisDocument.AWXRep321. GridTitleBackColor =GRB(0,127,0)
```

IsConnected

Type: Boolean

Description

This is read-only property, indicating that control is connected to the database.

Example

```
If ThisDocument.AWXRep321.IsConnected Then  
  `` do something here ``  
End If
```

NoSorting

Type: Boolean

Description

Sets/gets **Boolean** value that indicates whether records sorting has been enabled or not. By default, this property is set to TRUE (sorting disabled), providing best data retrieval performance.

Example

To enable sorting:

```
ThisDocument.AWXRep321.NoSorting=FALSE
```

QueryText

Type: String

Description

Sets/gets the SQL statement used to populate data grid in detailed report mode.

Example

```
Dim strQry as String  
strQry=ThisDocument.AWXRep321.QueryText  
MsgBox strQry
```

Remarks

Though this property is read/write, it is not recommended to set **QueryText** directly. Instead, you can set the **Columns**, **FilteringString**, and **SortingString** properties, then the value of **QueryText** will be automatically updated.

QueryTextSummary

Type: String

Description

Sets/gets the SQL statement used to populate data grid in alarm summary mode.

Example

```
Dim strQrySmr as String  
strQrySmr=ThisDocument.AWXRep321.QueryTextSummary  
MsgBox strQrySmr
```

Remarks

Though this property is read/write, it is not recommended to set **QueryTextSummary** directly. Instead, you can set the **SummaryColumn**, **FilteringString**, **SortingString**, or **SummaryFieldSortDir** properties. The value of **QueryTextSummary** will then be automatically updated.

ReportFileAlarms

FileAlarmsCondensed

FileAlarmsEvents

ReportFileEvents

ReportFileEventsCondensed

These properties should be considered obsolete in version 6.1. They are supported only for backwards compatibility.

RowDividerStyle

Type: String

Description

Sets/gets the appearance of row divider on data grid (horizontal grid lines).
Can be one of the following:

- 0 - No dividers
- 1 - Black line
- 2 - Dark gray line
- 3 - Raised
- 4 - Inset
- 5 - Use ForeColor
- 6 - Light gray line

This property can be used to set the appearance of both vertical and horizontal grid lines in one statement. The same can be done using the **GridHorizLineColor** and **GridVertLinesColor** properties independently.

Example

To set Black row divider:

```
ThisDocument.AWXRep321.RowDividerStyle=1
```

ShowSummaryChart

Type: Boolean

Description

Sets/gets the **Boolean** value that controls how alarm records are reported.
Use this property to toggle between **Chart** and **Grid Display** modes.

Examples

This statement brings control into **Chart Display** mode:

```
ThisDocument.AWXRep321. ShowSummaryChart=TRUE
```

This statement brings control into **Grid Display** mode:

```
ThisDocument.AWXRep321. ShowSummaryChart=FALSE
```

ShowSummaryGrid

Type: Boolean

Description

Sets/gets the **Boolean** value that controls how alarm records are reported in **Grid Display** mode. Use this property to toggle between **Detailed** and **Summary** views in **Grid Display** mode.

Remarks

If control is in the **Chart** mode, setting **ShowSummaryGrid** property will not have immediate effect. Only after **ShowSummaryChart** property is switched to **FALSE** (**Grid** mode), will the desired grid view be displayed.

Examples

These statements bring the control into **Detailed Grid Display** mode:

```
ThisDocument.AWXRep321. ShowSummaryChart =FALSE
```

```
ThisDocument.AWXRep321. ShowSummaryGrid =FALSE
```

This statement brings the control into **Summary Grid Display** mode:

```
ThisDocument.AWXRep321. ShowSummaryGrid =TRUE
```

SortByDbClick

Type: Boolean

Description

Sets/gets **Boolean** value indicating that records will be sorted by values in the particular grid's column when user double-clicks on that column's header. Subsequent double clicks on the same column header will reverse sorting direction.

Remarks

The sorting action will occur only if the **NoSorting** property is set to **FALSE**.

Example

```
ThisDocument.AWXRep321. NoSorting =FALSE
```

```
ThisDocument.AWXRep321. SortByDbClick=TRUE
```

SortingString

Type: String

Description

Sets/gets SQL ORDER BY statement that determines sorting of alarm data.

Example

Configures control to sort alarm data by EventTime in ascending order.

```
Dim strSortStmt as String  
strSortStmt="ORDER BY EventTime ASC "  
ThisDocument.AWXRep321. SortingString = strSortStmt
```

SummaryColumn

Type: String

Description

Sets/gets database column name, by which the alarm records are grouped for **Summary Grid** view and **Summary Chart**. By default, this is "[Tag]" column.

Example

This statement changes the database column to summarize records by "Area":

```
ThisDocument.AWXRep321.SummaryColumn="[Area]"
```

SummaryFieldSortDir

Type: Integer

Description

Sets/gets sort direction for the **Summary** field on the **Summary Grid** view or **Summary Chart**. Can be one of the following:

- 0 - ascending
- 1 - descending

Example

This statement changes the **Summary** field sort direction to descending:

```
ThisDocument.AWXRep321. SummaryFieldSortDir=1
```

TableName**Type: String****Description**

Sets/gets name of the current database table.

Example

```
ThisDocument.AWXRep321.TableName="EventLog"
```

ToolsVisible**Type: Boolean****Description**

Sets/gets visibility of toolbar.

Example

To show toolbar:

```
ThisDocument.AWXRep321.ToolsVisible=TRUE
```

ToolXPos**Type: Integer****Description**

Sets/gets the X coordinate of toolbar's left top corner (in device units).

Example

```
ThisDocument.AWXRep321.ToolXPos=255
```

ToolYPos**Type: Integer****Description**

Sets/gets the Y coordinate of toolbar's left top corner (in device units).

Example

```
ThisDocument.AWXRep321.ToolYPos=100
```

AutoUpdateInterval**Type: Short**

Description

Sets/gets the number of seconds between automatic updates.

Example

ThisDocument.AWXRep321.AutoUpdateInterval=60

BlinkOnUpdate

Type: Boolean

Description

Sets/gets whether or not the Report ActiveX will blink on update.

Example

ThisDocument.AWXRep321.BlinkOnUpdate=True

ChartGlobalScaling

Type: Boolean

Description

Enables/disables global scaling of the chart.

Example

ThisDocument.AWXRep321.ChartGlobalScaling=True

ChartScrolling

Type: Boolean

Description

Enables/disables chart scrolling.

Example

ThisDocument.AWXRep321.ChartScrolling=True

ChartScrollSize

Type: Short

Description

Indicates how many items you can scroll through on the chart at one time.

Example

ThisDocument.AWXRep321.ChartScrollSize=10

ChartTruncateNumber

Type: Short

Description

Sets/gets the size at which long strings are truncated. Must have ChartTruncateStrings property set to “True” in order for this to take effect.

Example

ThisDocument.AWXRep321.ChartTruncateNumber=8

ChartTruncateStrings

Type: Boolean

Description

Enables/disable the truncation of strings longer than the ChartTruncateNumber.

Example

ThisDocument.AWXRep321.ChartTruncateStrings=True

EnableAutoUpdate

Type: Boolean

Description

Enables/disables automatic updates.

Example

ThisDocument.AWXRep321.EnableAutoUpdate=True

EnableWebAccess;

Type: Boolean

Description

Enables/disables Report ActiveX web access.

Example

ThisDocument.AWXRep321.EnableWebAccess=True

FileName

Type: BSTR

Description

Sets/gets the name of the Report ActiveX configuration file.

Example

ThisDocument.AWXRep321.FileName=”C:\AwxRepConfig.awr”

MaxRecords

Type: Long

Description

Sets/gets the maximum number of records that the Report ActiveX will fit.

Example

ThisDocument.AWXRep321.MaxRecords=1024

RemoteDataServer

Type: BSTR

Description

This is the Internet Information Server for remote data access.

Example

ThisDocument.AWXRep321.RemoteDataServer = "http://localhost"

URLPathName

Type: BSTR

Description

Sets/gets the URL path for a configuration file over the web. Must set URLPathUsed to "True" for this to take effect.

Example

ThisDocument.AWXRep321.URLPathName=
"http://www.iconics.com/awxrepcfg.awr"

URLPathUsed

Type: Boolean

Description

Enables/disables the use of a URL path.

Example

ThisDocument.AWXRep321.URLPathUsed=True

UseLocalSettings;

Type: Boolean

Description

Tells a URL-enabled Report ActiveX whether or not to use local settings.

Example

ThisDocument.AWXRep321.UseLocalSettings=True

Available Control Methods

AboutBox

Shows **About** dialog box.

Example

```
ThisDocument.AWXRep321.AboutBox
```

Connect As Boolean

Forces the control to establish connection to database, using the information set in **ConnectionString** and **TableName**. If control is already connected, the current connection will be closed and a new connection will be established. This method can be used in VBA scripts automatically to switch the report ActiveX between different databases.

Return Value

On success, returns TRUE; if for any reason connection cannot be established, returns FALSE.

Example

```
Dim strConnString as String
strConnString="Provider=Microsoft.Jet.OLEDB.4.0; _
    Data Source=c:\alarms\SomeOtherDatabase.mdb; _
    User ID=Admin; Password=SYSTEM"
ThisDocument.AWXRep321.ConnectionString = strConnString
ThisDocument.AWXRep321.TableName="SomeOtherTable"
ThisDocument.AWXRep321.Connect
    ThisDocument.AWXRep321.Refresh
```

GetColumnHeaderText(iColIndex As Integer) As String

Returns the header text for the grid column specified by iColIndex argument. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property

Example

```
Dim strHeader1
    strHeader1=ThisDocument.AWXRep321.GetColumnHeaderT
ext(1)
```

GetColumnWidth(iColIndex As Integer) As Integer

Returns the width of the column specified by iColIndex argument. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property. The width is given in logical units, depending on the selected font.

Example

Retrieves the current width on the leftmost column:

```
Dim cx1 As Integer
cx1=ThisDocument.AWXRep321.GetColumnWidth(1)
```

LaunchReport(iTemplateIndex As Integer) As Integer

Creates Crystal Report based on the template specified by iTemplateIndex parameter. The value of argument iTemplateIndex can be 0 through 4 (there might be up to 5 different Crystal Report templates).

Return value

On success, returns 0; if a template for certain index does not exist or is not configured (in which case the **Crystal Report** button on the toolbar at the position corresponding to that index, counting from left, is grayed), return value is (-1).

Example

Creates Crystal Report based on the third template from the left, as seen on the **Reporter's** toolbar:

```
If ThisDocument.AWXRep321.LaunchReport(2)=-1 Then
    MsgBox " It looks like the template
specified is not configured"
End If
```

Print

Description

Prints detailed alarm data, alarm summary, data or alarm summary chart depending on the current state of AWXReport32 control.

Example

```
ThisDocument.AWXRep321.Print
```

Refresh

Description

Updates alarm data by requering the database.

Example

```
ThisDocument.AWXRep321.Refresh
```

SetColumnHeaderText(iColIndex As Integer, strNewText As String)

Sets the header text for the grid column specified by iColIndex argument. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property.

Example

```
ThisDocument.AWXRep321.SetColumnHeaderText(1, "Column1")
```

SetColumnWidth(iColIndex As Integer, iNewWidth As Integer)

Sets the width of the column specified by iColIndex argument to iNewWidth. The iColIndex argument is 1 - based index of the column, counting from the left of the grid. In **Summary Grid** view, iColIndex can be either 1 or 2; in **Detailed Grid** view, it can be from 1 to the maximum number of entries in the **Columns** property. The width is given in logical units, depending on the selected font.

Example

Doubles the current width on the leftmost column:

```
Dim cx1 As Integer
cx1=ThisDocument.AWXRep321.GetColumnWidth(1)
ThisDocument.AWXRep321.SetColumnWidth(1, 2*cx1)
```

LoadConfigFile(BSTR newVal) as Boolean

Loads the configuration file specified in newVal.

Example

```
Dim ret As Boolean
ret = AWXRep321.LoadConfigFile("C:\My Configs\cfg.awr")
```

SaveConfigFile(BSTR newVal) as Boolean

Saves your current Report ActiveX configuration to the file specified in newVal.

Example

```
Dim ret As Boolean
ret = AWXRep321.SaveConfigFile("C:\My Configs\cfg.awr")
```

LoadConfigURL(BSTR newVal) as Boolean

Loads the configuration file specified in the URL string newVal.

Example

```
Dim ret As Boolean
ret =
AWXRep321.LoadConfigURL("http://localhost/cfg.awr")
```

SetDBType(short NewDBType)

Sets the database type.

Example

```
AWXRep321.SetDBType(1)
```

GetColumnsNumber() as Short

Returns the number of columns in your current report configuration.

Example

```
MsgBox AWXRep321.GetColumnsNumber
```

GetFiltersNumber() as Short

Returns the number of filters currently available to your report.

Example

```
MsgBox AWXRep321.GetFiltersNumber
```

GetSortEntriesNumber() as Short

Returns the number of columns on which your report is currently sorting.

Example

```
MsgBox AWXRep321.GetSortEntriesNumber
```

GetColumn(short iIndex) as OGridColumnWrapper

Gets the Column object corresponding to iIndex. Returns "Nothing" if the index is invalid. Column index numbers start at 1.

Example

```
Dim col As OGridColumnWrapper
Set col = AWXRep321.GetColumn(1)
```

AddColumn(BSTR newCol) as OGridColumnWrapper

Adds the specified column to your report. Returns "Nothing" if argument is not a valid column name. You can then modify the column through the OGridColumnWrapper automation properties and methods.

Example

```
Dim col As OGridColumnWrapper  
Set col = AWXRep321.AddColumn("Severity")
```

SwitchColumns(short iIndex1, short iIndex2)

Switches the display order of the two columns corresponding to the specified indices. Returns "Nothing" if the column index is invalid.

Example

```
AWXRep321.SwitchColumns(1, 2)
```

DeleteColumn(short iIndex)

Deletes the column corresponding to the specified index. Returns "Nothing" if the column index is invalid.

Example

```
AWXRep321.DeleteColumn(2)
```

GetSummaryColumn(short iIndex) as OGridSummaryColumnWrapper

Gets the OGridSummaryColumnWrapper object corresponding to the specified column index. The summary column is the header associated with each column. Returns "Nothing" if the column index is invalid.

Example

```
Dim sumcol As OGridSummaryColumnWrapper  
Set sumcol = AWXRep321.GetSummaryColumn(1)
```

UpdateColumnsInfo()

Forces a refresh of the column information.

Example

```
AWXRep321.UpdateColumnsInfo
```

GetFilter(short iIndex) as OFilterWrapper

Gets the filter corresponding to the specified index.

Example

```
Dim filter As OFilterWrapper  
Set filter = AWXRep321.GetFilter(1)
```

AddFilter(BSTR newCol) as OFilterWrapper

Adds a filter on the specified column. You can then modify the filter through the OFilterWrapper automation properties and methods.

Example

```
Dim filter As OFilterWrapper  
Set filter = AWXRep321.AddFilter("Severity")
```

DeleteFilter(short iIndex)

Deletes the specified filter.

Example

```
AWXRep321.DeleteFilter(1)
```

UpdateFiltersInfo()

Forces an update of the filter information.

Example

```
AWXRep321.UpdateFiltersInfo
```

GetSortEntry(short iIndex) as OSortEntryWrapper

Returns an OSortEntryWrapper object corresponding to the specified index. Returns "Nothing" if the index is invalid.

Example

```
Dim sort As OSortEntryWrapper  
Set sort = AWXRep321.GetSortEntry(1)
```

AddSortEntry(BSTR newCol) as OSortEntryWrapper

Adds a sort for the specified column. You can then modify the sort entry through the OSortEntryWrapper automation properties and methods.

Example

```
Dim sort As OSortEntryWrapper  
Set sort = AWXRep321.AddSortEntry("Tag")
```

DeleteSortEntry(short iIndex)

Deletes the sort entry corresponding to the specified index.

Example

```
AWXRep321.DeleteSortEntry(1)
```

UpdateSortInfo()

Forces an update of the sort entry information.

Example

```
AWXRep321.UpdateSortInfo
```

SwitchSortEntries(short iIndex1, short iIndex2)**GetCellText(integer iColumnIndex, long iRowIndex) as VARIANT**

Returns the contents of the cell specified by the grid coordinates.

Example

```
Dim cell As Variant  
cell = AWXRep.GetCellText(1,1)  
MsgBox cell
```

GetGridRowCount() as VARIANT

Returns the number of rows in the grid. Returns -1 if there are no records or the database is not yet connected.

Example

```
Msgbox AWXRep321.GetGridRowCount
```

GetSelectedRowIndex() as VARIANT

Returns the index of the row that is currently selected. Returns -1 if no cell is selected.

Example

```
MsgBox AWXRep321.GetSelectedRowIndex
```

GetSelectedColIndex() as VARIANT

Returns the index of the column that is currently selected. Returns -1 if no cell is selected.

Example

```
MsgBox AWXRep321.GetSelectedColIndex
```

PrintToDefault(numOfCopies as Integer, orientation as Integer) as Long

Sends the current report to the default printer. Use 0 for portrait orientation and 1 for landscape.

Example

```
AWXRep321.PrintToDefault(1,0)
```

PrintToDevice(BSTR strDriverName, BSTR strDeviceName,BSTR strPortName, numberOfCopies as Integer, orientation as Integer) as Long

Prints to the specified device.

CanNavigateNextTable() as Boolean

Returns whether or not the report can navigate to the next table. Returns false if there are no other tables.

Example

```
MsgBox AWXRep321.CanNavigateNextTable
```

CanNavigatePrevTable() as Boolean

Returns whether or not the report can navigate to the previous table. Returns false if there are no other tables.

Example

```
MsgBox AWXRep321.CanNavigatePrevTable
```

NavigateNextTable()

Navigates to the next available table. Does nothing if no other tables exist, or this is the last table already.

Example

```
MsgBox AWXRep321.NavigateNextTable
```

NavigatePrevTable()

Navigates to the previous table. Does nothing if you are already on the first table.

Example

```
MsgBox AWXRep321.NavigatePrevTable
```

PrintReport(short iReportIndex, short nCopies, long IStartPage, long IEndPage) as Short

Prints the report to the default printer from page X to page Y.

Example

```
ret = AWXRep321.PrintReport(1, 1, 1, 5)
```

PrintReportToDevice(short iReportIndex, BSTR strDriverName, BSTR strDeviceName, BSTR strPortName, short nCopies, long IStartPage, long IEndPage) as Short

Prints the report to the specified device from page X to page Y.

GetHighlightedRange() as VARIANT

Returns a pair of coordinates corresponding to the upper left selected grid and the lower right selected grid.

Example

```
Dim sel As Variant
sel = AWXRep321.GetHighlightedRange
For Each coordinate In sel
    MsgBox coordinate
Next coordinate
```

ScrollUp()**ScrollDwn()****ScrollPgUp()****ScrollPgDwn()****ScrollLeft()****ScrollRight()****ScrollPgLeft()****ScrollPgRight()****SelectGridColumn(short iIndex)**

Selects the specified column in the grid.

Example

```
AWXRep321.SelectGridColumn(2)
```

GetSelectedGridColumns() as VARIANT

Returns a list of the currently selected grid columns. If no columns are selected, returns "Nothing".

Example

```
Dim n As Variant
n = AWXRep321.GetSelectedGridColumns
```

GetAllColumns() as VARIANT

Returns a list of all columns available to the report.

Example

```
Dim cols As Variant
cols = AWXRep321.GetAllColumns
For Each col In cols
    MsgBox col
```

Next col

GetAutoPrintSupport() as IOAutoPrintSupport

Returns an OAutoPrintSupport object to be used for configuring auto printing. See the OAutoPrintSupport documentation below for its properties.

Example

```
Dim aps As OAutoPrintSupport
Set aps = AWXRep321.GetAutoPrintSupport
```

GetCRTemplate(short iReportIndex) as IOCRTemplate

Returns a Crystal Reports template object. See the OCRTemplate documentation below for more information on its properties and methods.

Example

```
Dim cr As OCRTemplate
Set cr = AWXRep321.GetCRTemplate(1)
```

ReplaceHost(BSTR OldHostName, BSTR NewHostName) as Long

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
```

ReplaceFilePath(BSTR OldSubstring, BSTR NewSubstring) as Long

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

```

ReplaceHostEx(BSTR OldHostNameSubstring, BSTR NewHostNameSubstring, BOOL MatchCase, BOOL MatchWholeWord) as Long

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

```

Available Control Events

OnUpdate()

OnChartSeriesSelect(BSTR strSelectedItem)

OnGridSelectionChanged(boolean IsDragging)

OnGridLButtonDbcClk(short iColIndex, long iRowIndex)

OnGridLButtonClicked(short iColIndex, long iRowIndex)

OnGridRButtonDbcClk(short iColIndex, long iRowIndex)

OnGridRButtonClicked(short iColIndex, long iRowIndex)

OnGridMButtonDbcClk(short iColIndex, long iRowIndex)

OnGridMButtonClicked(short iColIndex, long iRowIndex)

OnChartDbcClick()

OnChartMouseUp(short Button, short Shift)

OnChartMouseDown(short Button, short Shift)

OnChartClicked()

Available Automation Objects

OGridColumnWrapper,
OGridSummaryColumnWrapper object properties:

HeaderTitle

Type: BSTR

Description

Sets/gets the value of the column header field.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.HeaderTitle = "New Column Header"
ThisDisplay.AWXRep321.UpdateColumnsInfo
```

The above example will change the header of column 1 in the Alarm Report ActiveX.

ColumnName

Type: BSTR

Description

Sets/gets the value of the column name field.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
Msgbox column.ColumnName
```

The above example will display the column name for column 1.

Width

Type: Short

Description

Sets/gets the width of the column

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.Width = 100
```

The above example will set the width of column 1 equal to 100.

DataType

Type: Short

Description

Sets/gets the data type of the column

Example

```
Dim column As OGridColumnWrapper  
Set column = ThisDisplay.AWXRep321.GetColumn(1)  
Msgbox column.DataType
```

The above example will display the data type of column 1.

ModifyFont

Type: Boolean

Description

Enables/Disables the modified cell font.

Example

```
Dim column As OGridColumnWrapper  
Set column = ThisDisplay.AWXRep321.GetColumn(1)  
column.ModifyFont = False
```

The above example will disable the modified font for column 1.

ModifyHeadFont

Type: Boolean

Description

Enables/Disables the modified header font.

Example

```
Dim column As OGridColumnWrapper  
Set column = ThisDisplay.AWXRep321.GetColumn(1)  
column.ModifyHeadFont = False
```

The above example will disable modified the header font for column 1.

ModifyTextColor

Type: Boolean

Description

Enables/Disables the modified text color.

Example

```
Dim column As OGridColumnWrapper  
Set column = ThisDisplay.AWXRep321.GetColumn(1)  
column.ModifyTextColor = True  
column.TextColor = RGB(0, 255, 0)
```

The above example will change the text color for column 1 to green.

TextColor

Type: OLE_COLOR

Description

Sets/gets the color of the column text.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.ModifyTextColor = True
column.TextColor = RGB(0, 255, 0)
```

The above example will change the text color for column 1 to green.

ModifyBackColor

Type: Boolean

Description

Enables/Disables the modified background color.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.ModifyBackColor = True
column.BackColor = RGB(0, 255, 0)
```

The above example will change the background color for column 1 to green.

BackColor

Type: OLE_COLOR

Description

Sets/gets the background color of the column.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.ModifyBackColor = True
column.BackColor = RGB(0, 255, 0)
```

The above example will change the background color for column 1 to green.

ModifyCell

Type: Boolean

Description

Enables/Disables changes to the cell justification and wrap text settings.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.ModifyCell = True
column.CellJustify = 0
```

The above example will change the column to left justified.

CellWrapText

Type: Boolean

Description

Enables/Disables column text wrapping.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.ModifyCell = True
column.CellWrap = True
```

The above example will enable cell wrapping.

CellJustify

Type: Short

Description

Sets/Gets the column justification

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.ModifyCell = True
column.CellJustify = 0
```

The above example will change the column text to left justified.

HeadJustify

Type: Short

Description

Sets/Gets the column justification

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
column.HeadJustify = 0
```

The above example will change the column header to left justified.

CellFont

Type: IfontDisp*

Description

Sets/Gets the cell font

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
Dim fnt As IfontDisp
Set fnt = column.CellFont
fnt.Bold = True
fnt.Italic = True
Set column.CellFont = fnt
```

The above example will turn on bold and italic for column 1.

HeadFont

Type: IfontDisp*

Description

Sets/Gets the header font

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.AWXRep321.GetColumn(1)
Dim fnt As IfontDisp
Set fnt = column.HeadFont
fnt.Bold = True
fnt.Italic = True
Set column.HeadFont = fnt
```

The above example will turn on bold and italic for column 1.

OFilterWrapper properties:

Active

Type: Boolean

Description

Activates/Deactivates the indicated filter

Example

```
Dim filter As ofilterwrapper
```

```
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
filter.Active = True
ThisDisplay.AWXRep321.UpdateFiltersInfo
```

The above example will activate filter number 1.

FilterName

Type: BSTR

Description

Sets/Gets the filter name

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
filter.FilterName = "New Filter"
ThisDisplay.AWXRep321.UpdateFiltersInfo
```

The above example will rename filter number 1 to the name "New Filter".

ColumnName

Type: BSTR

Description

Sets/Gets the name of the column to filter

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
filter.ColumnName = "[Tag]"
ThisDisplay.AWXRep321.UpdateFiltersInfo
```

The above example will change the filter to filter on column "[Tag]".

DataType

Type: Short

Description

Gets the data type of the column selected in the filter.

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
Msgbox filter.DataType
```

The above example will display the data type of the column selected in the filter.

FilterType

Type: Short

Description

Sets/Gets the type of the filter. Can be one of the following:

- 0 – Inclusive Filter
- 1 – Exclusive Filter
- 2 – Custom Filter

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
filter.FilterType = 0
ThisDisplay.AWXRep321.UpdateFiltersInfo
```

The above example will set the filter type as an Inclusive Filter.

IncludedValues

Type: VARIANT

Description

Sets/Gets the column values included by the filter.

Example

```
Dim myfilter As ofilterwrapper
Set myfilter = ThisDisplay.AWXRep321.GetFilter(1)
Dim included As String
Dim varstring
Dim myvar As Variant
included = ""
myvar = myfilter.IncludedValues
For Each varstring In myvar
    included = included & CStr(varstring) & "; "
Next varstring
Msgbox included
```

The above example will display a list of the included values separated by semicolons.

ExcludedValues

Type: VARIANT

Description

Sets/Gets the column values excluded by the filter.

Example

```
Dim myfilter As ofilterwrapper
Set myfilter = ThisDisplay.AWXRep321.GetFilter(1)
Dim excluded As String
Dim varstring
Dim myvar As Variant
```

```

excluded = ""
myvar = myfilter.ExcludedValues
For Each varstring In myvar
    excluded = excluded & CStr(varstring) & "; "
Next varstring
Msgbox excluded

```

The above example will display a list of the included values separated by semicolons.

TimeMode

Type: Short

Description

Sets/Gets the date/time type for inclusive or exclusive filters. Can be one of the following:

- 0 – Today
- 1 – Single Day
- 2 – Date Range
- 3 – Last

Example

```

Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
Filter.TimeMode = 2
ThisDisplay.AWXRep321.UpdateFiltersInfo

```

The above example will set date/time to “single day”.

Days

Type: Short

Description

Sets/Gets the last x Days value for Date/Time filters

Example

```

Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
Filter.Days = 1

```

The above example will set the filter to show the last 1 day.

Hours

Type: Short

Description

Sets/Gets the last x Hours value for Date/Time filters

Example

```
Dim filter As ofilterwrapper  
Set filter = ThisDisplay.AWXRep321.GetFilter(1)  
Filter.Hours = 1
```

The above example will set the filter to show the last 1 hour.

Minutes

Type: Short

Description

Sets/Gets the last x Minutes value for Date/Time filters

Example

```
Dim filter As ofilterwrapper  
Set filter = ThisDisplay.AWXRep321.GetFilter(1)  
Filter.Minutes = 1
```

The above example will set the filter to show the last 1 minute.

DateTimeFrom

Type: Variant

Description

Sets/Gets the Time From setting for filters that are set for Date Range. This setting is also used to determine which day is selected for Single Day filters.

Example

```
Dim filter As ofilterwrapper  
Set filter = ThisDisplay.AWXRep321.GetFilter(1)  
Filter.DateTimeFrom = "7/27/2003 12:00:00 AM"  
Filter.DateTimeTo = "7/28/2003 12:00:00 AM"
```

The above example will set the filter to show any message occurring on 7/27/2003.

DateTimeTo

Type: Variant

Description

Sets/Gets the Time To setting for filters that are set for Date Range.

Example

```
Dim filter As ofilterwrapper  
Set filter = ThisDisplay.AWXRep321.GetFilter(1)  
Filter.DateTimeFrom = "7/27/2003 12:00:00 AM"  
Filter.DateTimeTo = "7/28/2003 12:00:00 AM"
```

The above example will set the filter to show any message occurring on 7/27/2003.

CustomSQL

Type: BSTR

Description

Sets/Gets the custom SQL string that is used when the Filter Type is a Custom Filter.

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
Filter.CustomSQL = "[Tag]
IN('Humidity', 'Pump1', 'Tank1')"
```

The above example will set the custom SQL to select messages with where the Tag field is 'Humidity', 'Pump1', or 'Tank1'.

OFilterWrapper methods:

GetSQL() as BSTR

Type: BSTR

Description

Returns the custom SQL string that is used when the Filter Type is a Custom Filter. The returned string is enclosed in parenthesis.

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
MsgBox filter.CustomSQL
```

The above example will return the Custom SQL string.

IsValid() as Boolean

Type: BSTR

Description

Returns whether or not the filter is valid.

Example

```
Dim filter As ofilterwrapper
Set filter = ThisDisplay.AWXRep321.GetFilter(1)
If filter.IsValid Then
    MsgBox "This filter is valid."
Else
    MsgBox "This filter is invalid."
End If
```

The above example will say let you know if the filter is valid.

OSortEntryWrapper properties:

ColumnName

Type: BSTR

Description

Sets/Gets the column name to sort on.

Example

```
Dim sort As OSortEntryWrapper
Set sort = ThisDisplay.AWXRep321.GetSortEntry(1)
sort.ColumnName = "[Tag]"
ThisDisplay.AWXRep321.UpdateSortInfo
```

The above example sets the sort column as [Tag].

SortDir

Type: Short

Description

Sets/Gets sort direction. Possible values are:

- 0 – Do not sort on this field
- 1 – Sort Ascending
- 2 – Sort Descending

Example

```
Dim sort As OSortEntryWrapper
Set sort = ThisDisplay.AWXRep321.GetSortEntry(1)
Sort.sortdir = 1
ThisDisplay.AWXRep321.UpdateSortInfo
```

The above example sets the sort direction as ascending.

ResortMode

Type: Short

Description

Sets/Gets the re-sort settings. Possible values are:

- 0 – Do not re-sort after translating
- 1 – Re-sort after translating

Example

```
Dim sort As OSortEntryWrapper
Set sort = ThisDisplay.AWXRep321.GetSortEntry(1)
Sort.ResortMode = 1
ThisDisplay.AWXRep321.UpdateSortInfo
```

The above example sets the sort entry to re-sort after translating

OAutoPrintSupport properties:

EnableAutoPrint

Type: Boolean

Description

Enables/Disables auto printing of Alarm Report

Example

```
Dim autoprint As OAutoPrintSupport
Set autoprint =
ThisDisplay.AWXRep321.GetAutoPrintSupport()
autoprint.EnableAutoPrint = True
```

The above example enables auto print support.

AutoPrintMode

Type: Short

Description

Gets/Sets the printing mode. Possible values are:

- 0 – Default printer
- 1 – Selected Devices

Example

```
Dim autoprint As OAutoPrintSupport
Set autoprint =
ThisDisplay.AWXRep321.GetAutoPrintSupport()
autoprint.AutoPrintMode = 0
```

The above example selects default printer as the printing mode.

Orientation

Type: Short

Description

Gets/Sets the printer orientation. Possible values are:

- 0 – Portrait
- 1 – Landscape

Example

```
Dim autoprint As OAutoPrintSupport
Set autoprint =
ThisDisplay.AWXRep321.GetAutoPrintSupport()
autoprint.Orientation = 0
```

The above example selects portrait as the orientation.

NumberOfCopies

Type: Short

Description

Gets/Sets the number of copies to be printed.

Example

```
Dim autoprint As OAutoPrintSupport
Set autoprint =
ThisDisplay.AWXRep321.GetAutoPrintSupport()
Autoprint.NumberofCopies = 5
```

The above example sets the number of copies to 5.

PrintContent

Type: Short

Description

Gets/Sets content of the pages to be printed. Possible values are:

- 0 - None
- 1 – Crystal Report – Design #5
- 2 – Crystal Report – Design #4
- 4 – Crystal Report – Design #3
- 8 – Detailed Alarm Report
- 16 – Condensed Alarm Report
- 32 – Current Grid / Chart View

* To include multiple choices in auto-printing just add the values together. For example, 63 = all report views.

Example

```
Dim autoprint As OAutoPrintSupport
Set autoprint =
ThisDisplay.AWXRep321.GetAutoPrintSupport()
autoprint.PrintContent = 63
```

The above example sets all pages to be printed.

SpecialPrinter1

Type: IOSpecialPrinter*

Description

Gets/Sets settings for an additional printer.

Example

```

Dim autoprint As OAutoPrintSupport
Set autoprint =
ThisDisplay.AWXRep321.GetAutoPrintSupport()
Dim specprint As OSpecialPrinter
Set specprint = autoprint.SpecialPrinter1()
specprint.Orientation = 0

```

The above example sets the orientation of the additional printer as portrait.

IOSpecialPrinter* SpecialPrinter2

Type: IOSpecialPrinter*

Description

Gets/Sets settings for an additional printer.

Example

See Example for SpecialPrinter1

IOSpecialPrinter* SpecialPrinter3

Type: IOSpecialPrinter*

Description

Gets/Sets settings for an additional printer.

Example

See Example for SpecialPrinter1

IOSpecialPrinter* SpecialPrinter4

Type: IOSpecialPrinter*

Description

Gets/Sets settings for an additional printer.

Example

See Example for SpecialPrinter1

IOSpecialPrinter* SpecialPrinter5

Type: IOSpecialPrinter*

Description

Gets/Sets settings for an additional printer.

Example

See Example for SpecialPrinter1

OSpecialPrinter properties:

DriverName

Type: BSTR

Description

Gets/Sets the driver name for the printer.

Example

```
Dim specprint As OSpecialPrinter
Msgbox specprint.DriverName
```

This example displays the name of the driver.

DeviceName

Type: BSTR

Description

Gets/Sets the device name for the printer.

Example

```
Dim specprint As OSpecialPrinter
Msgbox specprint.DeviceName
```

This example displays the name of the device.

PortName

Type: BSTR

Description

Gets/Sets the port name.

Example

```
Dim specprint As OSpecialPrinter
Msgbox Specprint.PortName
```

This example displays the name of the port.

NumberOfCopies

Type: short

Description

Gets/Sets the number of copies to print.

Example

```
Dim specprint As OSpecialPrinter  
specprint.NumberOfCopies = 5
```

Orientation

Type: short

Description

Gets/Sets the orientation for printing

Example

```
Dim specprint As OSpecialPrinter  
specprint.Orientation = 0  
This example sets the orientation as portrait.
```

OCRLine properties:

long Style

long PenWidth

OLE_COLOR Color

OCRText properties:

BSTR Text

IFontDisp* Font;

short FontPointSize

OLE_COLOR ForeColor

OLE_COLOR BackColor

OLE_COLOR BorderColor

short Justify

short Left

short Width

short Top

short Height

bool DropShadow

long LeftBorder

long TopBorder

long RightBorder

long BottomBorder

OCRField properties:

BSTR Field

BSTR HeaderText

short Width

OCRTemplate properties:

OLE_COLOR DataColor

IFontDisp* HeaderFont

short HeaderFontSize

IFontDisp* DataFont

short DataFontSize

OLE_COLOR HeaderColor

IOCRText* MainTitle

IOCRText* SubTitle

IOCRLine* HeaderDivider

IOCRLine* DataRowDivider

short XtraHeaderSpace

short XtraDataSpace

BSTR TemplateName

float BottomMargin

float TopMargin

float LeftMargin

float RightMargin

short PaperType

short ReportingMode

boolean HasUpperLimit

long UpperLimit

boolean AttachCurrTableName

short Orientation

short SortingMode

OCRTemplate methods:

GetFieldsCount() as Short

AddField(BSTR NewFieldName, BSTR NewFieldHeaderText, short NewFieldWidth) as IOCRField

DeleteField(short FieldIndex)

SwitchFields(short FieldIndex1, short FieldIndex2)

GetFieldObject(short FieldIndex) as IOCRField

GetFiltersNumber() as Short

GetSortEntriesNumber() as Short

GetFilter(short ilIndex) as IOFilterWrapper

AddFilter(BSTR newCol) as IOFilterWrapper

DeleteFilter(short ilIndex)

GetSortEntry(short ilIndex) as IOSortEntryWrapper

AddSortEntry(BSTR newCol) as IOSortEntryWrapper

DeleteSortEntry(short ilIndex)

SwitchSortEntries(short ilIndex1, short ilIndex2)