Guide to the ArchestrA™ Alarm Control

Invensys Systems, Inc.

Contents

Welcome

This guide describes configuring and using the ArchestrA Alarm control. This control is delivered as part of the ArchestrA Symbol Editor and can be used in ArchestrA symbols to show current and historical alarms and events in a grid.

You can view this document online or you can print it, in part or whole, by using the print feature in Adobe Acrobat Reader.

This guide assumes you know how to use Microsoft Windows, including navigating menus, moving from application to application, and moving objects on the screen. If you need help with these tasks, see the Microsoft online help.

This guide also assumes you know how to use Microsoft SQL Server. For help with SQL Server, see the Microsoft online help.

In some areas of the Application Server, you can also right-click to open a menu. The items listed on this menu change, depending on where you are in the product. All items listed on this menu are available as items on the main menus.

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
Bold	Menus, commands, dialog box names, and dialog box options.
Monospace	Code samples and display text.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using.

- Details of how to recreate the problem.

- The exact wording of the error messages you saw.

- Any relevant output listing from the Log Viewer or any other diagnostic applications.

- Details of what you did to try to solve the problem(s) and your results.

- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

About the ArchestrA Alarm Control

The ArchestrA Alarm Control is a graphical element you can use in your ArchestrA symbols to show current and historical alarms and events.

The ArchestrA Alarm Control replaces the Alarm Viewer control and Alarm DB View control in the InTouch HMI and extends alarm visualization to the ArchestrA Graphics environment.

You can place the ArchestrA Alarm Control directly from the Tools panel in the ArchestrA Symbol Editor onto the canvas. You can customize it to your needs by adding further graphics, interactions, and scripts.

You can deploy a managed InTouch application containing ArchestrA Alarm Controls to a remote node and visualize and interact with alarms at run time with InTouch WindowViewer.

For this documentation, the ArchestrA Alarm Control is simply referred to as "Alarm Control."

We recommend you have a basic understanding of the InTouch Alarm system before continuing. For more information, see the *InTouch HMI Alarms and Events Guide*.

Client Modes

The Alarm Control supports five different client modes, which can be grouped depending on their data source.

Alarm Manager

The Alarm Manager manages currently active alarms (summary alarms) and recent alarms and events (historical alarms and events). These types of alarms and events are held in the InTouch internal alarm memory.

Current Alarms

When the Alarm Control is showing alarms in "Current Alarms" mode, it is showing currently active alarms directly from the Alarm Manager.

Recent Alarms and Events

When the Alarm Control is showing alarms in "Recent Alarms and Events" mode, it is showing historical alarms and events stored in Alarm Manager.

Unlike the "Current Alarms" mode, the "Recent Alarms and Events mode" shows time point data, such as alarm transitions and events, instead of continuous conditions.

Alarm Database

The Alarm Database stores alarms and events from the Alarm Manager to a SQL Server database. You can use the Alarm DB Logger utility to continuously log alarms and events to the Alarm Database.

Historical Alarms

When the Alarm Control is configured in "Historical Alarms" mode, only alarms stored in the Alarm Database are shown.

Historical Events

When the Alarm Control is configured in "Historical Events" mode, only events stored in the Alarm Database are shown.

Historical Alarms and Events

When the Alarm Control is configured in "Historical Alarms and Events" mode, both alarms and events stored in the Alarm Database are shown.

Switching Between Client Modes

The client mode and many other features are controlled by properties and methods.

By default, the Alarm Control is set to show current alarms. You can change the client mode also at run time by using the Alarm Control properties.

Using the Alarm Control in ArchestrA Symbols

You can use the ArchestrA Alarm control as a faceplate so that when the operator clicks an icon, an ArchestrA Alarm control showing a specific alarm area opens.
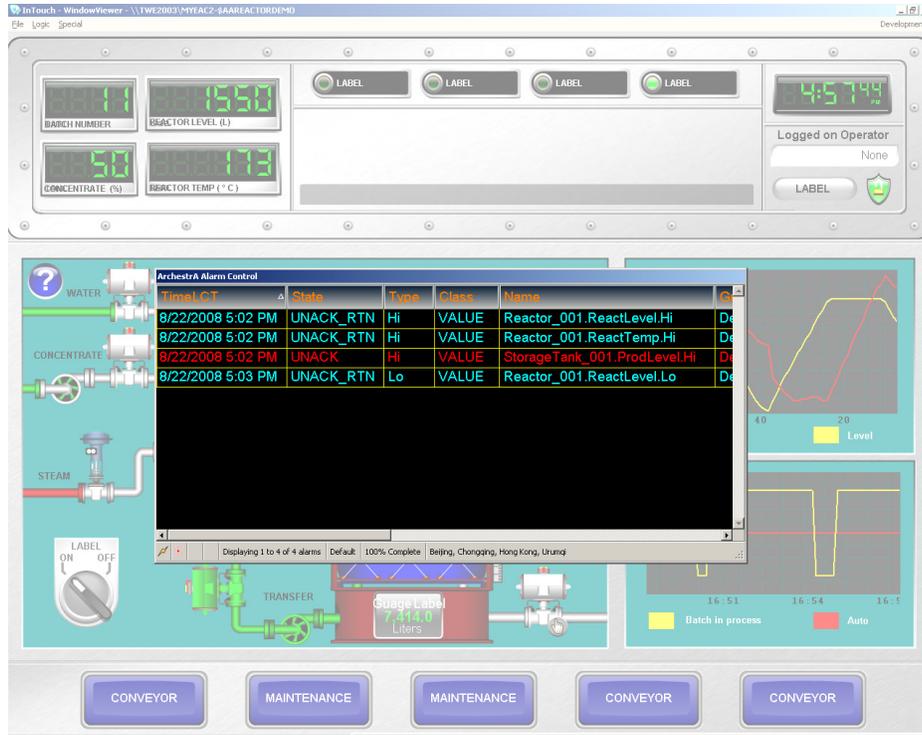
You can also configure the ArchestrA Alarm control to interact with the Galaxy namespace and other ArchestrA symbols by mapping its properties to ArchestrA attributes and symbol elements.

The Alarm Control can be placed into ArchestrA Symbols hosted by Automation Object templates and instances. You can configure them to retrieve alarms from their hosting Area object or their hosting Automation object.

Alarm Acknowledgement

You can configure the Alarm Control to require an alarm to be acknowledged even if the condition causing the alarm has passed. This ensures that an operator is aware of events that caused a temporary alarm state but have returned to normal.

You acknowledge alarms at run time using a shortcut menu or through script methods.

Current Value and Quality Display

The Alarm Control in one of the current client modes shows continuously the current value and quality of a tag or attribute in alarm state.

You can see the current value and quality of tags or attributes in alarm from:

● InTouch running on the local computer.

● Galaxy namespace.

Note You cannot see current value and quality data from InTouch tags running on a remote computer.

Alarm Queries

The Alarm Control supports the standard InTouch and Galaxy alarm query formats, such as:

```
\galaxy!Area_001
```
```
\intouch!Group_A
```
The alarm query syntax changes when you use the run-time alarm comment language switching feature. For more information, see Alarm Query Syntax when Register Using Galaxy_<GalaxyName> is Enabled on page 19.

The Alarm Control also supports relative references for Galaxy alarms in alarm queries. For all alarm modes, relative references are resolved at run time at the point of query to the Alarm Manager or Alarm Database.

You must put the reference part of the alarm query between less-than (<) and greater-than (>) characters.

The following tables shows examples of alarm queries.

Alarm Query	Description
\provider!group	Shows all alarms from the given provider and group. For example: `\intouch!Group_A`
\provider!group!tagname	Shows all alarms from the given provider, group and tag. For example: \galaxy!Mixing_Area!RotorCtrl

Alarm Query	Description
\\node\provider!group	Shows all alarms from the given provider and group from a given node. For example: `\\remote\intouch!Group_B`
\\node\provider!group!tagname	Shows all alarms from the given provider, group and tag from a given node. For example: `\\grnode\galaxy!Packaging_Area!Wrapper1`
HotBackupName	Shows all alarms from primary or backup alarm provider as configured in the Hot Backup Manager.
\galaxy!<me.Area>!<me.tagname>.*	Shows all alarms from the Automation Object. Alarms from other Automation Objects in the same area are ignored.
\galaxy!<myArea.tagname> or \galaxy!<me.Area>	Shows all alarms from the Area object hosting the Automation Object
\galaxy!<myPlatform.tagname>	Shows all alarms from the Winplatform object hosting the Automation Object.
\galaxy!<myContainer.tagname>	Shows all alarms from the container Automation Object. At run-time the Alarm Control resolves the Container attribute to detect the container.
\galaxy!<myEngine.tagname>	Shows all alarms from the AppEngine object hosting the Automation Object. At run-time the Alarm Control resolves the MyEngine attribute to detect the host.
\\Node:IP Address\InTouch!$System	On Windows Vista and Windows Server 2008 operating systems, if Window Viewer is started from a remote client session use a query of this form to access the alarms from the Alarm Manager running in the remote client session.

Note On Windows Vista and later operating systems, only one alarm provider is supported per node.

Alarm Query Syntax when Register Using Galaxy_<GalaxyName> is Enabled

The run-time alarm comment language switching feature requires slightly different alarm query syntax. In the WinPlatform object, when you enable InTouch alarm provider, you can enable **Register using Galaxy_<GalaxyName> instead of Galaxy**.

This option will register the platform to the alarm subsytem using the Galaxy name preferred by "Galaxy_" intead of just the word "Galaxy". This allows an InTouch application to monitor alarms from multiple Galaxies and avoid name conflicts.

Syntax changes slightly when Galaxy_GalaxyName is enabled:

- Use \\ for machine name.
- Use \ for Galaxy or Galaxy_<GalaxyName>.
- Use ! for Area.

For example: \\Galaxy\MyGalaxy!Area001.

If Galaxy_GalaxyName is not enabled in WinPlatform, then the default behavior described in Alarm Queries on page 17 applies.

You can determine if Galaxy_<GalaxyName> has been enabled by monitoring the run-time attribute of the platform ITAlarmProvider.ProviderNameAsGalaxyNameEnabled.

Alarm Filtering

The Alarm Control unites the Query Favorites concept of the InTouch Alarm Viewer control and the Filter Favorites concept of the InTouch Alarm DB View control.

The Query Favorites of InTouch Alarm Viewer control define a set of alarm provider, alarm group, an optional node name, and a priority range under one name. The alarm provider, alarm group, and the node name are used for subscribing to a specific alarm group. The priority range on the other hand is used to filter the alarms from the given alarm group.

The Filter Favorites of InTouch Alarm DB View control define a set of any number of criteria you want to filter from the Alarm Database under one name.

In summary, Filter Favorites fulfill a purely filtering function whereas Query Favorites fulfill a subscription and a filtering function at the same time.

The Alarm Control filtering feature unites both these concepts by exclusively using filter conditions and subscribing to the necessary alarm providers on demand.

The filter conditions can be re-used between different client modes. For example, if you define node name, provider name, alarm group, and a priority range for the current alarms, you can also use this filter to retrieve the historized alarm data of the same source from the Alarm Database instead.

Alarm Queries to Query Filters Translation

As with InTouch alarm controls, you can define queries for current alarms in the \\node\provider!group format, but they are translated by the Alarm Control to a filter after you save.

For example, the query string \\GRNode\galaxy!MixingArea is translated to the following filter string:

Node = 'GRNode' AND Provider='galaxy' AND Group='MixingArea'

You can modify the filter in a tree to query only alarms in the priority range 1 to 250, such as:

```
AND
    Node = 'GRNode'
    Provider = 'Galaxy'
    Group = 'MixingArea'
    Priority >= '1'
    Priority <= '250'
```

Alarm Hiding

The "hiding" and "unhiding" of alarm records is known in the corresponding InTouch alarm controls as "suppressing" and "unsuppressing".

When the Alarm Control is hiding alarms, it ignores certain alarms. If an alarm matches the exclusion criteria, it is not visible.

The actual alarm generation is completely unaffected by hiding. Alarm records are still logged into the alarm history.

As in the InTouch HMI, you can unhide specific alarms and also use properties and methods to interact with the alarm hiding feature at run time.

Alarm Control Grid Freezing

You can freeze the Alarm Control to prevent the Alarm control tree from being updated with any further changes.

For example, if new alarms occur while the Alarm Control is frozen, the new alarms are only shown after you unfreeze the Alarm Control.

You can configure a time period after which the Alarm Control automatically unfreezes to avoid the Alarm Control being unknowingly frozen. For example, the operator leaves the workstation and returns without realizing that the Alarm Control is still frozen.

The Alarm Control unfreezes automatically if one of the following changes:

- Alarm Mode

- Alarm Query

- Query Filter

Alarm Sorting

Like InTouch alarm controls, you can sort the alarms in ascending or descending direction for selected columns.

The Alarm Control supports alarm sorting for up to three columns at design time and run time.

At run time, the operator can configure sorting of even more columns by clicking on the column headers of the Alarm Control.
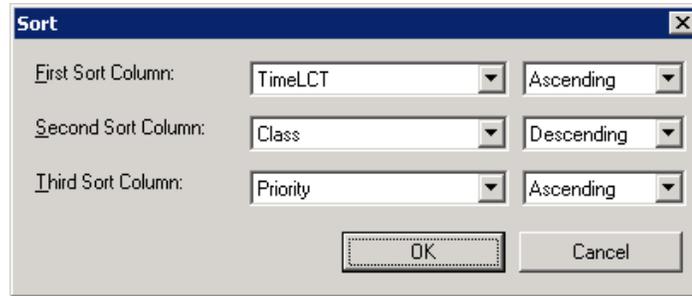
Status Bar

The status bar of the Alarm Control resembles the status bars of the InTouch alarm controls, with the following differences:

- Alarm Control shows also the alarm client time zone.

- Alarm Control querying the Alarm Database has a Requery button to more easily retrieve data from the Alarm Database.

- Alarm Control shows the current client mode as an icon.

Chapter 2

Configuring the Alarm Control

This section shows you how to place an Alarm Control onto the canvas and configure it. You can configure it either with the **Edit Animations** dialog box, or by changing individual properties in the Properties Editor.

After placing the Alarm Control onto the canvas, you can configure the:

- Client Mode to show current alarms, recent alarms and events, or historical alarms and/or events.

- Colors for the Alarm Control grid, window, heading, and alarm records.

- Order and width of the grid columns and their headers.

- Sorting order of alarm records.

- Filtering for alarm records and save the filters as favorites for re-use.

- Time format and zone for the alarm record time stamps.

- Run-time behavior for the Alarm Control, such as:

  - If the operator can resize columns or select multiple records at run time.
  - Access to specified options of the shortcut menu at run time.

Placing the Alarm Control into an ArchestrA Symbol

You can easily place the ArchestrA Alarm Control into an ArchestrA Symbol by placing it onto the canvas.

To place the Alarm Control into an ArchestrA Symbol

1 Open the ArchestrA Symbol in the ArchestrA Symbol Editor.

2 On the **Tools** panel, click the ArchestrA Alarm Control icon. The cursor appears in insert mode.

3 Click on the canvas where you want to place the Alarm Control.
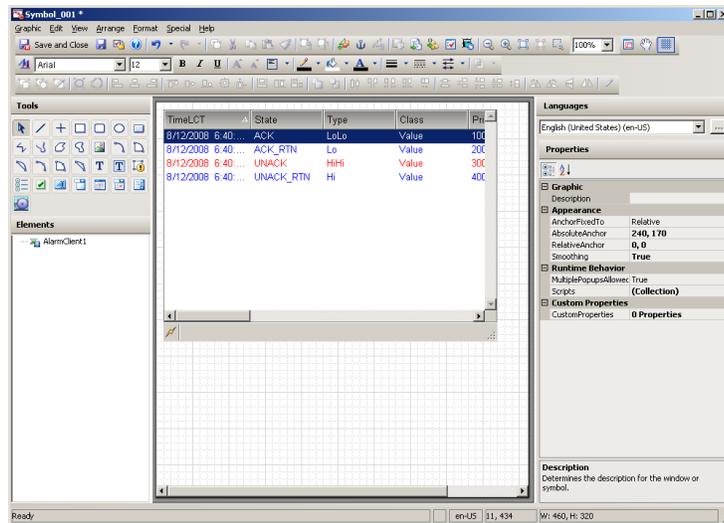
Setting the Alarm Control Properties

Like all other graphical objects in the ArchestrA Symbol Editor, you can set some of the properties of the selected Alarm Control directly in the Properties Editor.
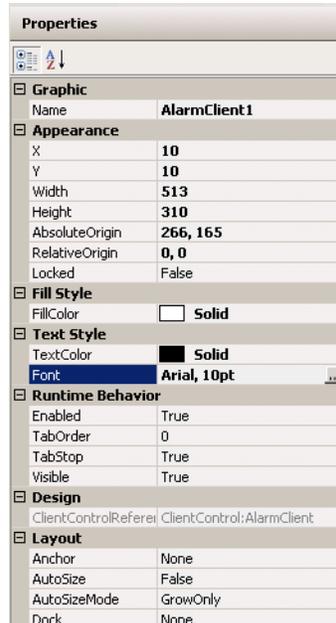
We recommend you configure the Alarm Control with the **Edit Animations** dialog box and only use the Properties Editor to edit the configuration afterward.

Showing Current Alarms or Recent Alarms and Events

You can set the Alarm Control to either show:

* Current alarms

* Recent alarms and events

You use the **ClientMode Property** integer property in scripting to switch the Alarm Control to show current alarm or recent alarms and events at run time.

You can also configure a comment to use when alarms are acknowledged at run time. Use the **AckComment.UseDefault Property** Boolean property and **AckComment.DefaultValue Property** string property in scripting to use a default acknowledgement comment at run time.

To show current alarms

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.
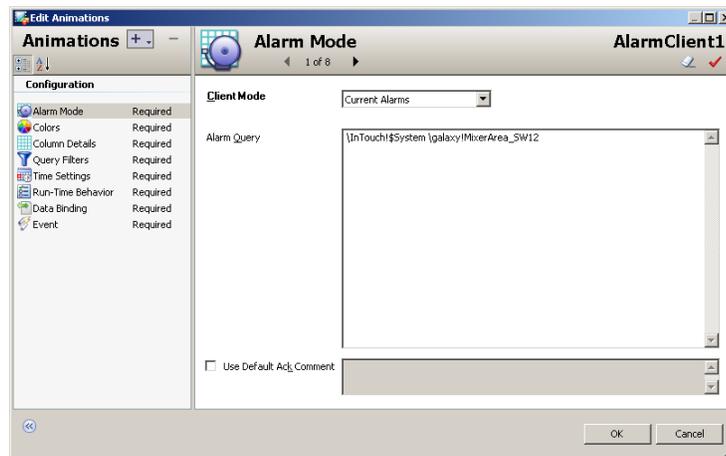
2   If necessary, click **Alarm Mode**. The **Alarm Mode** page appears.
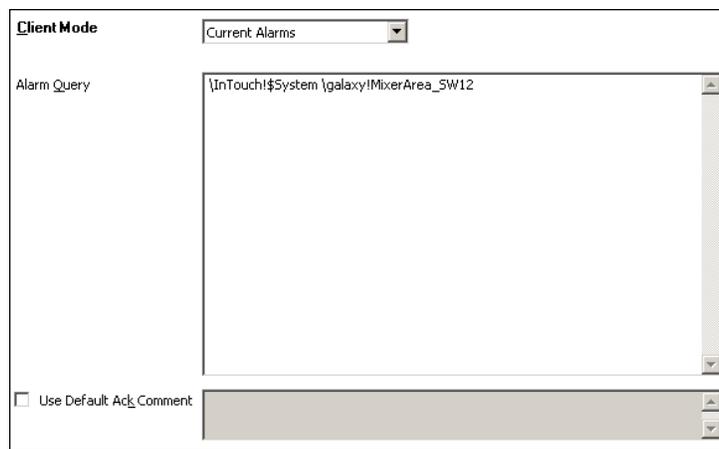
3   In the **Client Mode** list, click **Current Alarms**.

4   In the **Alarm Query** box, type the alarm query. To create a new line in the Alarm Query box, press **Ctrl + Enter**. For more information on the valid syntax, see Alarm Queries on page 17.

5   If you want to want to use a default acknowledgement comment, select the **Use Default Ack Comment** check box and type a comment in the text box.

6   Click **OK**.

To show recent alarms and events

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Alarm Mode**. The **Alarm Mode** page appears.

3   In the **Client Mode** list, click **Recent Alarms and Events**.
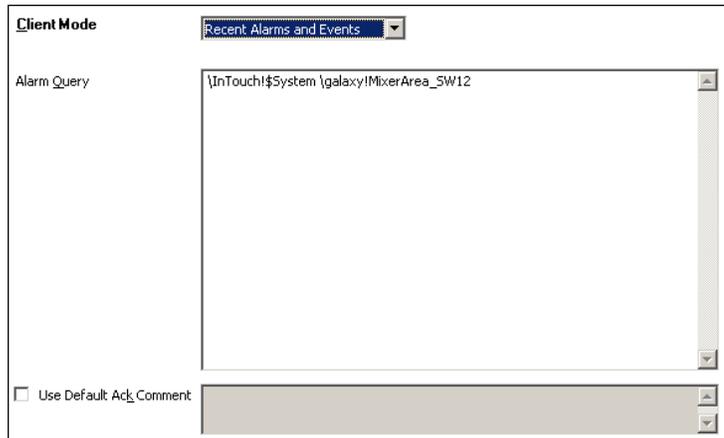
4   In the **Alarm Query** box, type the alarm query. To create a new line in the Alarm Query box, press **Ctrl + Enter.**

The alarm query must follow one of the following syntax:

- \\node\provider!group

- \provider!group

- HotBackupName

For example:

```
\intouch!$system
\galaxy!Area_001
```

For Alarm Controls hosted by Automation Object templates or instances, you can specify one of the following alarm queries:

- \galaxy!<myArea.Tagname> to retrieve alarms and events from the Area object hosting the Automation Object template or instance.

- \galaxy!<me.Area>!<me.Tagname>.* to retrieve alarms and events from the Automation Object template or instance.

For more information on alarm queries, see Alarm Queries on page 17

5   If you want to want to use a default acknowledgement comment, select the **Use Default Ack Comment** check box and type a comment in the text box.

6   Click **OK**.

Showing Historical Alarms and/or Events

You can set the Alarm Control to show one of the following:

- Historical alarms from the Alarm Database

- Historical events from the Alarm Database

- Historical alarms and events from the Alarm Database

When you configure the Alarm Control to show historical alarms and/or events, you also configure the following:

- Server name hosting the Alarm Database

- Authentication information to connect to the Alarm Database

- Maximum number of records to retrieve from the Alarm Database

- Time range or duration to show in the Alarm Control.

- If the Alarm Control should update to the current client time

For more information on creating an alarm database and logging alarms, see *Recording Alarms into an Alarm Database* in the *InTouch HMI Alarms and Events Guide.*

Use the following properties in scripting to switch the client mode and configure the database connection, such as:

- ClientMode Property on page 94

- Database.Authentication Property on page 102

- Database.Name Property on page 103

- Database.Password Property on page 103

- Database.ServerName Property on page 103

- Database.UserID Property on page 104

- Domain Property on page 104

To show historical alarms and/or events

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Alarm Mode**. The **Alarm Mode** page appears.

3 In the **Client Mode** list, click:

- **Historical Alarms** to only show alarms from the Alarm Database. No events are shown.

- **Historical Events** to only show events from the Alarm Database. No alarms are shown.

- **Historical Alarms and Events** to show both alarms and events from the Alarm Database.

4 In the **Authentication Mode** list, click one of the following:

- **Windows Integrated** to use the authentication of the currently logged-on Windows user.

- **Windows Account** to use a given Windows user authentication.

- **SQL Server** to use SQL Server authentication mode.

5 In the **Server Name** list, either select or type the name of the server hosting the Alarm Database.

6 In the **Database Name** box, type the name of the Alarm Database. By default, this is WWALMDB.

7 If you are using **Windows Account** authentication mode, type the domain, user name, and password in the **Domain**, **User Name** and **Password** boxes.

8 If you are using **SQL Server** authentication mode, type user name and password in the **User Name** and **Password** boxes.

9 Click **Test Connection**. The connection to the Alarm Database is tested and a result message appears. If necessary, check your authentication information.

10 Click **OK**.

To set maximum records and time range

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Alarm Mode**. The **Alarm Mode** page appears.

3   Make sure the Client Mode is set to **Historical Alarms**, **Historical Events**, or **Historical Alarms and Events**.

4   In the **Maximum Records** box, type the number of records to view from the control at one instance. The valid range of maximum records is from 1 to 32766.

    You can also use the **MaxDatabaseRecords Property** property in scripting to set the maximum records at run time.

5   To use a pre-defined time interval, select an interval from the middle list of the **Time Range** pickers.
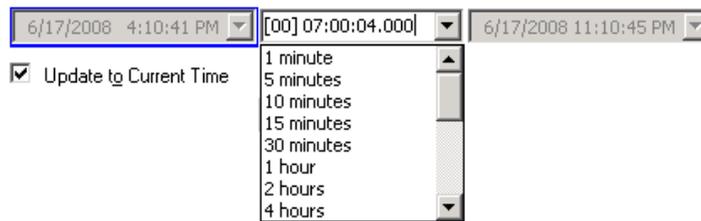
6   To use a specific start time and end time, clear **Update to Current Time**, and select the start time from the list at the left and the end time from the list at the right of the **Time Range** pickers.
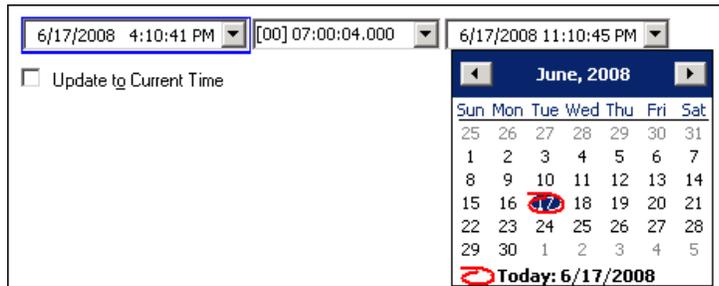
    You can also use the **TimeSelector.*** methods and properties in scripting to set the start date, end date, or duration at run time. For more information, see the Scripting the Alarm Control on page 79.

7   Click **OK**.

Setting Alarm Control Colors

You can show different types of alarm records in different colors so the operator can more easily identify certain types of alarms.

You can configure the Alarm Control with priority breakpoints to show alarm records within the resulting priority ranges in different colors.

You can also configure the control background color, the grid color, and the heading colors.

Setting Event Record Colors

You can set text color and background color for event alarm records. Use the **EventColor.ForeGround Property** and **EventColor.BackGround Property** properties in scripting to set the event alarm record text color and background color at run time.

To set text and background colors for event records

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Colors**. The **Colors** page appears.
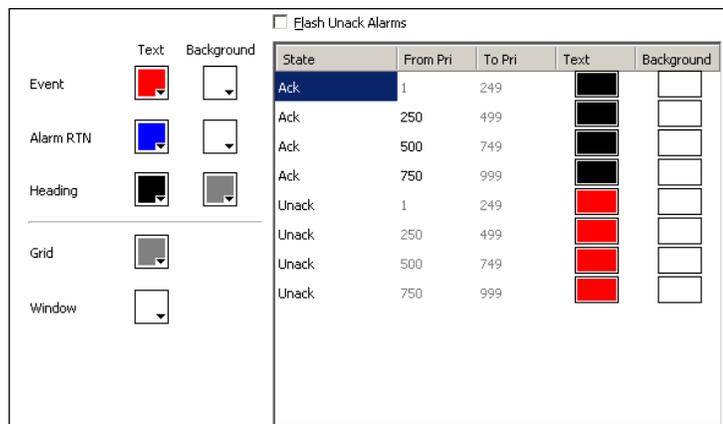
3 Configure the event record text color. Do the following:

   a Click the color field next to **Event** and under **Text**. The color picker appears.

   b Select a color and click **OK**.

4 Configure the event record background color. Do the following:

   a Click the color field next to **Event** and under **Background**. The color picker appears.

   b Select a color and click **OK**.

5 Click **OK**.

Setting Return To Normal Record Colors

You can set text color and background color for "return to normal" alarm records. Use the **AlarmColor.RTN.ForeGround Property** and **AlarmColor.RTN.BackGround Property** properties in scripting to set the "return to normal" alarm record text color and background color at run time.

To set text and background colors for "return to normal" records

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Colors**. The **Colors** page appears.

3   Configure the "return to normal" record text color. Do the following:

   a   Click the color field next to **Alarm RTN** and under **Text**. The color picker appears.

   b   Select a color and click **OK**.

4   Configure the "return to normal" record background color. Do the following:

   a   Click the color field next to **Alarm RTN** and under **Background**. The color picker appears.

   b   Select a color and click **OK**.

5   Click **OK**.

Setting Heading, Grid, and Window Color

You can set text color and background color for the heading, the grid color, and the Alarm Control window color. Use the corresponding **HeadingColor.ForeGround Property**, **HeadingColor.BackGround Property**, **GridColor Property**, and **WindowColor Property** properties in scripting to set the colors for heading, grid, and window.

To set heading, grid, and window color for the Alarm Control

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Colors**. The **Colors** page appears.

3   Do one of the following:

a   Configure the heading text color by clicking the color box next to **Heading** and under **Text**. If the color box does not open, you need to select the **Show Heading** option on the **Run-Time Behavior** page first.

b   Configure the heading background color by clicking the color box next to **Heading** and under **Background**. If the color box does not open, you need to select the **Show Heading** option on the **Run-Time Behavior** page first.

c   Configure the grid color by clicking the color box next to **Grid**. If the color box does not open, you need to select the **Show Grid** option on the **Run-Time Behavior** page first.

d   Configure the window color by clicking the color box next to **Window**.

Setting Priority Ranges for Alarm Records

You can use alarm priority ranges to filter alarms. The Alarm Control can show alarms within a given range with a different text and background color. Use the **AlarmColor.Range Property** property group in scripting to set the breakpoints at run time.

The Alarm Control supports four alarm ranges defined by three breakpoints:

1 < breakpoint 1 < breakpoint 2 < breakpoint 3 < 999

To set priority ranges for alarm records

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Colors**. The **Colors** page appears.

3   In the **From Pri** column in the list at the right, locate the break point you want to change. These are values except 1 or 999.

4   Click on the value and type a new value in the range between the previous breakpoint and the next breakpoint.

☐ Flash Unack Alarms

State	From Pri	To Pri	Text	Background
Ack	1	249		
Ack	250	499		
Ack	500	749		
Ack	750	999		
Unack	1	249		
Unack	250	499		
Unack	500	749		
Unack	750	999		

5   Press **Enter**. All priority values in the list are updated.

6   Click **OK**.

Example

If you use the color configuration in the procedure above, the Alarm Control at run time could have following appearance:

Setting Colors for Acknowledged Alarms

You can set the text and background colors for records of acknowledged alarms. For each of the priority ranges, you can set a text color and a background color. Use the **AlarmColor.Ack.ForeGround Property** and **AlarmColor.Ack.BackGround Property** property groups in scripting to set the text color and background color for acknowledged alarms in each priority range at run time.

To set colors for acknowledged alarm records

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Colors**. The **Colors** page appears.

3 In the list at the right, locate the **Ack** record and priority range for which you want to change the text or background color.

4 Click the color box in the **Text** or **Background** column of the line. The color picker appears.

5 Select a color and click **OK**.

6 Click **OK**.

Setting Colors for Unacknowledged Alarms

You can set the text and background colors for records of unacknowledged alarms. For each of the priority ranges, you can set a text color and a background color. Use the **AlarmColor.UnAck.ForeGround Property** and **AlarmColor.UnAck.BackGround Property** property groups in scripting to set the text color and background color for unacknowledged alarms in each priority range at run time.

To set colors for unacknowledged alarm records

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Colors**. The **Colors** page appears.

3   In the list at the right, locate the **Unack** record and priority range for which you want to change the text or background color.

4   Click the color box in the **Text** or **Background** column of the line. The color picker appears.

5   Select a color and click **OK**.

6   Click **OK**.

Setting Unacknowledged Alarms to Flash

Instead of showing unacknowledged alarm records in predefined constant text and background color, you can configure the Alarm Control to flash unacknowledged alarms in another text and background colors.

The unacknowledged alarm records flash between the colors of the Unack alarms and the colors of the Flash Unack alarms. Use the **FlashUnAckAlarms Property** Boolean property in scripting to set unacknowledged alarm records to flash at run time. Use the **AlarmColor.UnAck.Flash.ForeGround Property** and **AlarmColor.UnAck.Flash.BackGround Property** property groups in scripting to set the text color and background color for flashing unacknowledged alarms in each priority range at run time.

To set flashing and colors for unacknowledged alarm records

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Colors**. The **Colors** page appears.

3   Select the **Flash Unack Alarms** check box.

Note You cannot select the **Flash UnAck Alarms** check box if the client mode is set to one of the historical modes.

4 In the list on the right, locate the **Unack** record and priority range for which you want to change the text or background color. Do the following:

  a Click the color box in the **Text** or **Background** column of the line. The color picker appears.

  b Select a color and click **OK**.

5 Locate the **Flash Unack** record and priority range for which you want to change the text or background color. Do the following:

  a Click the color box in the **Text** or **Background** column of the line. The color picker appears.

  b Select a color and click **OK**.

6 Click **OK**.

Renaming, Resizing, and Reordering Column Headers

You can rename, resize, and change the order of column headers in the Alarm Control.
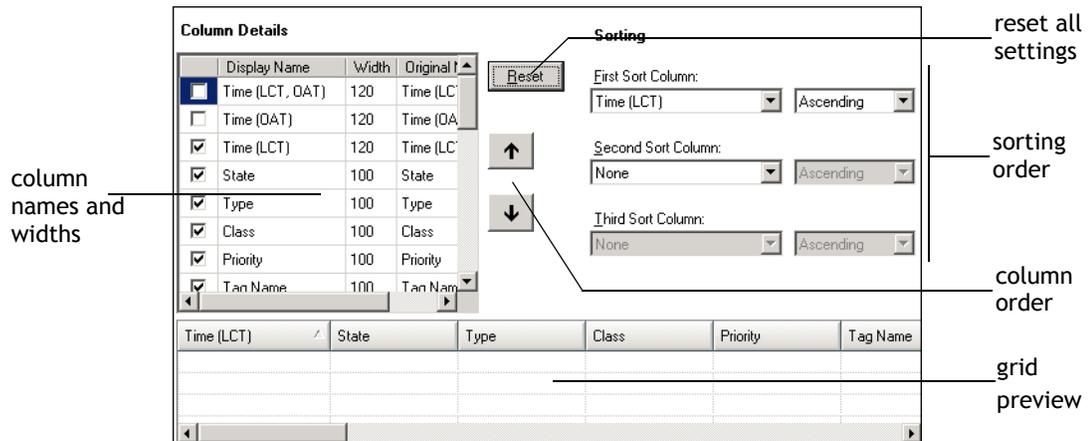
All changes you make in the Column Details list are shown in the grid preview.

You can also use the grid preview to resize columns or change their order with the pointer.

Column headers can be localized along with other symbol text when you export, translate, and reimport languge files. The translated language files must be imported to the InTouch HMI for run-time language switching. For further information, see Chapter 11 Working with Languages in the *Application Server User's Guide*.

> **Important** If you rename or reorder column headers, you must repeat the symbol text translation procedures. If you do not, your changes will not be available for run-time language switching.

Renaming Column Headers

You can rename the column headers in the Alarm Control.

To rename column headers

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Column Details**. The **Column Details** page appears.

3 In the **Column Details** list, locate the column header you want to rename and click on it.

4 Type a new name and press **Enter**. The **Column Details** list and the grid preview are updated.
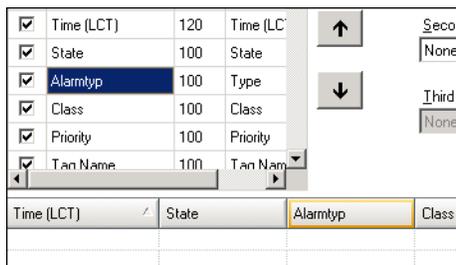
5 Click **OK**.

Resizing Columns

You can resize the column headers in the Alarm Control either by:

● Typing in a numeric value.

● Dragging the column header boundary width with the pointer in the grid preview.

To resize the column numerically

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Column Details**. The **Column Details** page appears.

3 In the **Column Details** list, locate the name of the column you want to resize and click on the **Width** value in the row.

4 Type a new width in pixels and press **Enter**. The **Column Details** list and the grid preview are updated.

5 Click **OK**.

To resize the column graphically

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Column Details**. The **Column Details** page appears.

3 In the grid preview, locate the column you want to resize and drag the column boundary to resize the column. The width value of the **Column Details** list is updated.

4 Click **OK**.

Changing the Order of Columns

You can change the order of the columns in the Alarm Control by:

• Moving column names up and down in the **Column Details** list using buttons.

• Dragging the column header with the pointer in the grid preview.

You also can reset the column widths and order to their default values. Resetting the column widths and order also resets the names to their default values.

To change the column order

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Column Details**. The **Column Details** page appears.

   3   Do one of the following:

   - Click arrow up and arrow down to reposition the columns.

   - In the grid preview, drag the name of the column you want to reposition and drop it to the left of another column to reposition it.
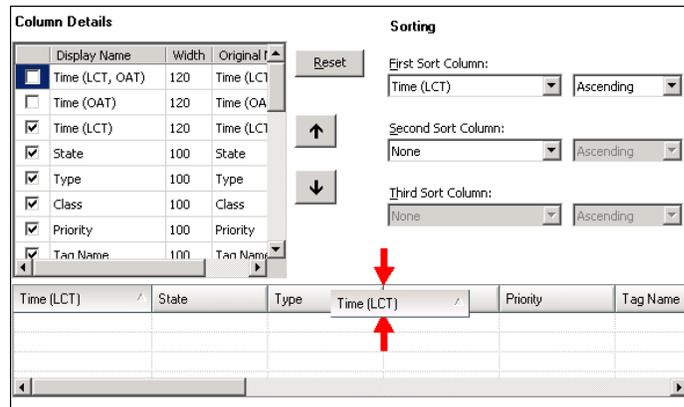
   The grid preview and the **Column Details** list shows the new column order.

   4   Click **OK**.

   To reset column widths and order

   1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

   2   Click **Column Details**. The **Column Details** page appears.

   3   Click **Reset**. The column widths, names, and order are reset to their default values.

   4   Click **OK**.

Sorting Alarms

You can configure how the Alarm Control sorts alarm records at run time. By default, the Alarm Control lists alarm records by time in ascending order.

You can sort alarm records in ascending or descending order based on a primary column, an optional secondary sort column, and an optional tertiary sort column.

You can configure the sorting columns and directions either in lists or with the grid preview. Use the **SortColumn.First Property**, **SortColumn.Second Property**, and **SortColumn.Third Property** properties in scripting to set the columns to be sorted at run time. Use the **SortOrder.First Property**, **SortOrder.Second Property**, and **SortOrder.Third Property** properties in scripting to set the sort direction for each at run time.

To set sorting columns and directions with lists

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Column Details**. The **Column Details** page appears.

3   In the **Sorting** area, do the following:

   a   Select the primary sort column in the **First Sort Column** list and a sorting direction in the list to its right.

   b   Optionally, select the secondary sort column in the **Second Sort Column** list and a sorting direction in the list to its right.

   c   If you set the **Second Sort Column**, optionally select the tertiary sort column in the **Third Sort Column** list and a sorting direction in the list to its right.

The grid preview is updated and shows arrows for the sorted columns and their sort directions.

**To set sorting columns and directions with the grid preview**

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Column Details**. The **Column Details** page appears.

3 In the grid preview, click on a column to select it for sorting. An arrow appears on the column header and the change is also shown in the **Sorting** area lists.

4 To change the sorting direction, click on the column header again. The arrow changes on the column header and the change is also shown in the **Sorting** area lists.

Note If you click on a column header after releasing the **Shift** key, all sorting information is lost and the selected column is the new primary sorting criteria.

5 To set secondary and tertiary sorting, hold the **Shift** key and repeat from step 3.

6 Release the **Shift** key.

7 Click **OK**.

Filtering Alarms

You can filter current and historical alarms by using query filters. A query filter is a collection of filter criteria in a logical construct.

For example, you can filter alarms by defining a query filter that only shows alarms with priorities larger than 500 and smaller than 750.

You can re-use the filter queries you define for historical alarms for current alarms and vice versa. You can also re-use filter queries you define at design-time at run time and vice versa.

Important Query filters for current alarms and recent alarms and events require at least **Provider** and **Group** as filter criteria. These must use the equals sign.

When you use TimeLCT, TimeOAT, or TimeLCTOAT as filter criteria for historical alarm modes, you need make sure that the TimeSelector.StartDate and TimeSelector.EndDate properties do not limit the query. Otherwise the Alarm Control can possibly not return all alarm and event records.

Set the TimeSelector.StartDate property earlier than any time filtering requirement, and the TimeSelector.EndDate later than any time filtering requirement.

Using Wildcards in Queries

In current alarm queries, you can use wildcards only in the Tagname part of the query and not in the Provider, Group, or Node part of the query. A valid example is:

```
\galaxy!Mixing!RotorBlade*
```
In query filters that are used for current queries, the same restrictions apply.

In query filters that are used for historical queries, you must convert the operator and wildcard to SQL syntax according to the following table:

	Current Query	Historical Query
Operator	=	Like
Wildcard	*	%

For example:

```
Provider = 'galaxy' AND Group = 'Mixing'
  AND Name Like 'RotorBlade%'
```
If you want to use a query filter containing a wildcard for a current query and a historical query, create two separate query filters.

Using an Existing Query Filter

You can use an existing query filter to filter the alarms shown in the ArchestrA Alarm Control. You can also use the **Favorite Property** string property in scripting to switch to an existing query filter at run-time.

To use an existing query filter

1　Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2　Click **Query Filters**. The **Query Filters** page appears.

3　In the **Query Filter Favorites** list, select a query filter.

4　Click **OK**.

Adding a New Query Filter

You can define a new query filter to filter the alarms shown in the ArchestrA Alarm Control. The new query filter is saved as a favorite in the **Query Filter Favorites** list.

To add a new query filter

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Query Filters**. The **Query Filters** page appears.
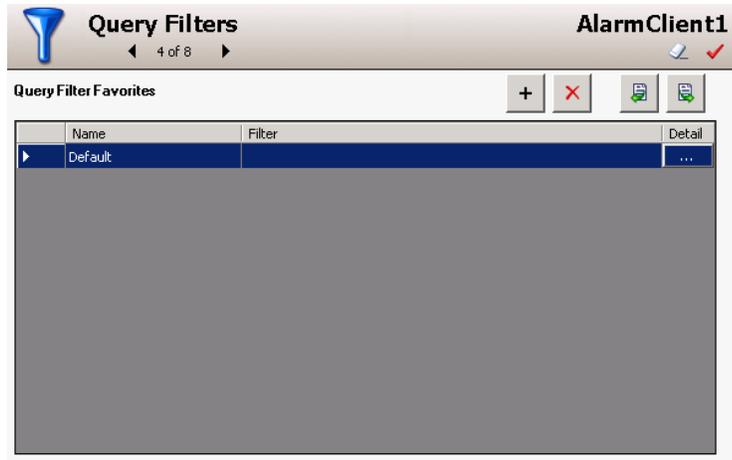
3 Click **Add**. The **Add Filter** dialog box appears.
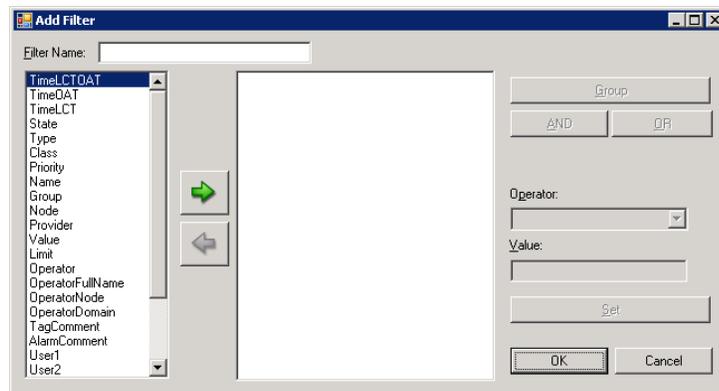
For more information, see Constructing Filters on page 45.

Constructing Filters

You use the **Add Filter** or **Modify Filter** dialog box to create or edit a filter graphically.

Filter criteria   Construction area
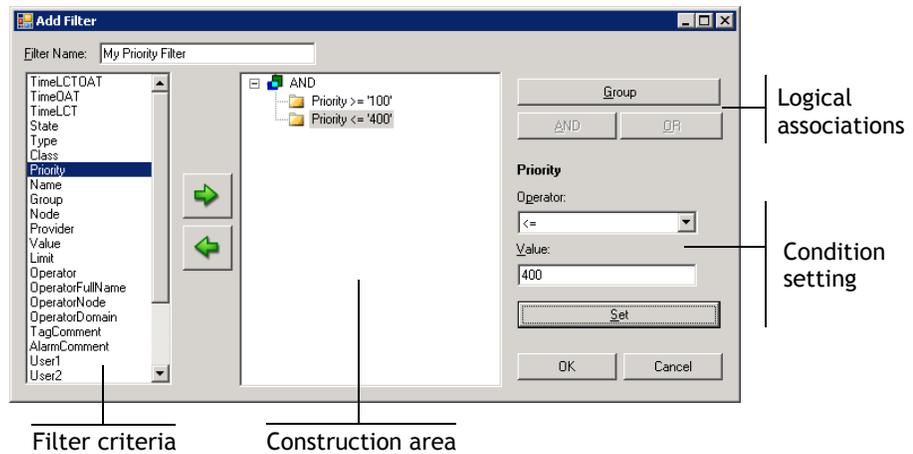
To construct a filter

1 If you want to change the filter name, type a new unique name in the **Filter Name** box.

2 Add filter criteria to the construction area by selecting a column name on the left and clicking the right arrow button. When you add filter criteria to the construction area, they are automatically logically connected by AND.

3 If necessary, remove filter criteria by selecting them in the filter construction area and clicking the left arrow button.

4 To change the logical operator, select it in the filter construction area, and then either:

  • Click **AND** or **OR**.

  • Right-click and select **AND** or **OR** from the shortcut menu.

5 To group filter criteria logically, either:

  • Drag a filter criteria in the construction area over another filter criteria.

- Select one filter criteria, click **Group**, and then click the other filter criteria.
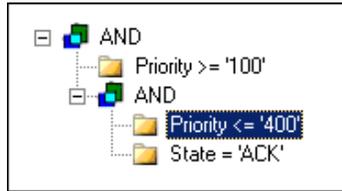
By default, the filter criteria are logically grouped with AND. If necessary, you can select the **AND** item in the tree and click **OR** to change it to an OR grouping.

6 Assign values to filter criteria.

Note If you are using the Value column as a filter criteria, you may get unexpected results at run time. The items in the Value column are sorted alphabetically, not numerically. This is because the Value column can contain strings.

Do the following:

a Select a filter criteria in the construction area.

b Select an operator from the **Operator** list.

c Type or select a value in the **Value** box.

d Click **Set**. The filter criteria is updated in the construction area.

7 To cut, copy, or paste individual filter criteria or filter criteria branches, right-click on the filter criteria and select the appropriate option from the shortcut menu.

8 When you are done, click **OK**.

Modifying an Existing Query Filter

You can modify an existing query filter using the **Modify Filter** dialog box.

To modify an existing query filter

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Query Filters**. The **Query Filters** page appears.

3   Select an existing query filter in the **Query Filter Favorites** list.

4   Click the ellipsis button. The **Modify Filter** dialog box appears. For more information, see Constructing Filters on page 45.

5   Click **OK**.

Deleting a Query Filter Favorite

You can delete any non-default query filter favorites.

To delete a query filter favorite

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Query Filters**. The **Query Filters** page appears.

3   Select an existing query filter in the **Query Filter Favorites** list.

4   Click the **Delete** button.

5   When a message appears, click **Yes**.

Exporting Query Filter Favorites

You can export the query filter favorites list to an XML file. The XML file containing the query filter favorites can be imported to other Alarm Control in design time or run time. Do not edit this file directly. The default query filter favorite is not exported to the XML file.

To export the query filter favorites list

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Query Filters**. The **Query Filters** page appears.

3   Click the **Export** button. The **Export Query Filter Favorites** dialog box appears.

4   Select a location and a name for the XML file and click **Save**.

Importing Query Filter Favorites

You can import the query filter favorites list from an XML file.

To import the query filter favorites list

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Query Filters**. The **Query Filters** page appears.

3 Click **Import**. The **Import Query Filter Favorites** dialog box appears.
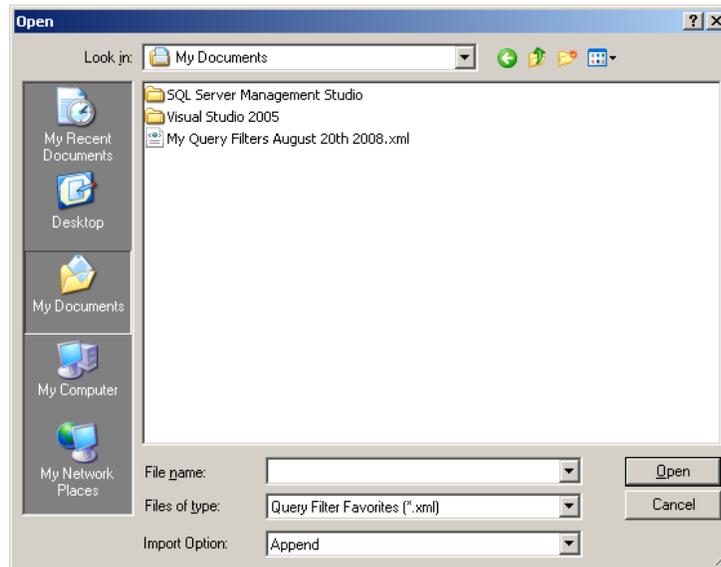
4 In the **Import Option** list, click either:

- **Append** to append the imported query filters to the existing query filters. If query filter names in the imported XML conflict with existing query filters, you are prompted to confirm the import for each filter.

- **Overwrite** to replace all existing query filters with the imported query filters.

5 Browse to the XML file and click **Open**.

Setting Time Zone and Format

You can set the time zone in which the client shows the alarm and event records. By default, the time zone is set to the client computer's current time zone at design time. Use the **TimeZone.TimeZone Property**, **Time.Type Property**, and **Time.Format Property** properties in scripting to set the time zone, time type, and time format at run time.

You can also set the time format of the alarm and event records. You can select between two different time format sets:

- Wonderware Time Format: same as the InTouch Alarm Viewer control and InTouch Alarm DB View control of InTouch version 10.0 and later.

- .NET Time Format: defined by Microsoft .NET Framework time format conventions.

Setting the Time Zone

You can set the time zone in which the Alarm Control shows the alarm and event records.

You can either set the time display to a predefined time zone, or to the client time zone. The client time zone is the time zone of the computer on which the Alarm Control is running.

The **Client Time Zone** setting is useful if you are deploying an application using the Alarm Control to a different time zone.

For example, if you develop your application in the "Pacific Time" zone and deploy it to two computers in the time zones "Central Time" and "Eastern Time", you can ensure the Alarm Control shows the local time for each deployment by setting the time zone to **Client Time Zone**.

To set the time zone

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Time Settings**. The **Time Settings** page appears.

3 In the **Time Zone** list, select a time zone.

4 Click **OK**.

Setting the Wonderware Time Format

You can set the Wonderware time format in which the Alarm Control shows the alarm and event records. You can either use a predefined datetime format, or compose one.

To set the Wonderware time format

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Time Settings**. The **Time Settings** page appears.

3 In the **Time Format** area, do the following:

   a Make sure **Use .NET Time Format** is cleared.

   b Click a date format from the list at the left. The equivalent date format code appears in the box above.

   c Click a time format from the list at the right. The equivalent time format code is appended to the format string in the box above.

4 If you want to customize the datetime format, modify the codes in the box as follows:

Code	Purpose	Example
%m	Two-digit month	03
%b	Three-letter month	Mar
%B	Full month name	March
%d	Two-digit day	17
%Y	Four-digit year	2008
%y	Two-digit year	08
%#x	Full day and date	Tuesday, March 11, 2008
%H	Hours in 24 hour format	14
%I	Hours in 12 hour format	2
%M	Minutes	55
%S	Seconds	34
%s	Milliseconds	223
%p	AM or PM	PM

5 Click **OK**.

Setting the .NET Datetime Format

You can set the .NET datetime format in which the Alarm Control shows the alarm and event records. You can either use a predefined datetime format, or compose one. The predefined date format is based on the short date format setting of the operating system and may vary from computer to computer.

To set the .NET datetime format

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Time Settings**. The **Time Settings** page appears.

3 In the **Time Format** area, do the following:

   a Select the **Use .NET Time Format** check box.

   b Click a date format from the list at the left. The equivalent date format code appears in the box above.

   c Click a time format from the list at the right. The equivalent time format code is appended to the format string in the box above.

4 If you want to customize the datetime format, modify the codes in the box as in the table below. For more information, see the Microsoft Knowledge database on .NET datetime formats.

Code	Purpose	Example
M	Single-digit month	9
MM	Two-digit month	09
MMM	Three-letter month	Sep
MMMM	Full month name	September
d	Single-digit day	8
dd	Two-digit day	08
ddd	Abbreviated day of the week	Mon.
dddd	Day of the week	Monday
yyyy	Four-digit year	2008
yy	Two-digit year	08
HH	Hours in 24 hour format	14
hh	Hours in 12 hour format	2
mm	Minutes	55

Code	Purpose	Example
ss	Seconds	34
fff	Milliseconds	223
tt	AM or PM	PM

5 Click **OK**.

Configuring Run-Time Behavior

You can configure the behavior and appearance of the Alarm Control at run time, for example:

- Showing and Hiding parts of the Alarm Control.

- Specifying if the Alarm Control queries the alarm database when it starts up.

- Scrolling to new alarms.

- Hiding warnings, errors, and messages.

- Restricting operator access to parts of the Alarm Control.

- Specifying Alarm Control freeze behavior.

- Customizing the "no records" message.

- Customizing the run-time shortcut menu.

Showing Heading, Grid, or Status Bar

You can show and hide parts of the Alarm Control at run time, such as the heading, grid, or status bar. Use the **ShowHeading Property**, **ShowGrid Property**, and **ShowStatusBar Property** properties in scripting to show or hide the heading, grid, and status bar at run time.

Caution If you hide the status bar, you will not be able to see important indicators, such as the New Alarms, Hidden Alarms, and Frozen Grid indicators.

To show the heading, grid, or status bar at run time

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3 Show or hide the part. Do any of the following:
   - Select the **Show Heading** check box to show the heading at run time, or clear it to hide the heading at run time.
   - Select the **Show Grid** check box to show the grid at run time, or clear it to hide the grid at run time.
   - Select the **Show Status Bar** check box to show the status bar at run time, or clear it to hide the status bar at run time.

4 Click **OK**.

Automatically Querying for Alarms on Start Up

You can configure the Alarm Control to automatically query the Alarm Manager or Alarm Database when the control starts up at run time. Use the **QueryStartup Property** property in scripting to control the start up behavior at run time.

By default, current alarms and recent alarms and events are automatically queried when the Alarm Control starts at run time. You can disable the automatic query if the Alarm Control is:

- Configured to mainly use query filters.

- Driven mainly by scripts.

To query the Alarm Manager or Alarm Database automatically on start up

1 Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2 Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3 Select the **Query on Startup** check box.

4 Click **OK**.

Scrolling Automatically to New Alarms

If the operator is viewing multiple pages of alarms, new alarms may go unnoticed. You can configure the Alarm Control to scroll automatically to new alarms. Use the **AutoScroll Property** Boolean property in scripting to scroll automatically to new alarms.

However, if the Alarm Control scrolls automatically to new alarms, it may be hard for the operator to view and analyze older alarms if new alarms occur. If the Alarm Control is frozen, it will not scroll automatically to new alarms.

To scroll automatically to new alarms

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Select the **Auto Scroll to New Alarms** check box.

4   Click **OK**.

Hiding Errors, Warnings, and Status Messages

You can prevent a message dialog box from opening when errors, warnings, or status messages occur in the Alarm Control. Even if you hide errors, warnings, and status messages, the messages are sent to the ArchestrA Logger. Use the **HideErrors Property** property in scripting to hide error, warning, and status messages at run time.

To hide error and warning messages

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Select the **Hide Errors and Warnings** check box.

4   Click **OK**.

Restricting User Access to Rows and Columns

You can prevent the operator from:

- Resizing columns.

- Selecting rows.

- Selecting multiple rows.

Use this feature for interfaces where it is easy to accidentally resize columns or select rows. For example, if the Alarm Control is running on a small display, use the **AllowColumnResize Property** and **RowSelection Property** properties in scripting to control the ability to resize columns and select rows at run time.

To prevent the operator from resizing columns

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Clear the **Allow Column Resizing** check box.

4   Click **OK**.

To prevent the operator from selecting rows

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   In the **Row Selection** list, click:
- **No** to prevent operator from selecting rows.

- **Single** to allow operator to only select one row.

- **Multiple** to allow operator select multiple rows.

4   Click **OK**.

Retain Hiding when Changing Alarm Query Filter

You can configure the Alarm Control to hide alarms even if the alarm query filter changes. Use the **RetainHidden Property** property in scripting to retain the hiding of alarms at run time.

To retain hiding when change the alarm query filter

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Select the **Retain Hidden** check box.

4   Click **OK**.

Overriding the Frozen Grid

You can configure the Alarm Control to unfreeze the grid after a given time in seconds. Use this option to make sure that new alarms appear on the grid after a specified time. Use the **AutoResumeDuration Property** property in scripting to unfreeze the Alarm Control after a certain duration at run time.

The Alarm Control also unfreezes if you change one of the following:

•   Alarm Mode

•   Alarm Query

•   Query Filter

To override the frozen grid

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Select the **Auto Resume after** check box and type the number of seconds after which the grid unfreezes.

4   Click **OK**.

Customizing the "No Records" Message

You can customize the message that appears when there are no records to show in the grid. Use the **NoRecordsMessage.Enabled Property** and **NoRecordsMessage.Message Property** properties in scripting to customize the "no records" message at run time.

To customize the "no records" message

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Select the **Show Custom 'No Records' Message** check box and type a message you want to show in the Alarm Control when there are no alarm records.

4   Click **OK**.

Changing the Language of the "No Records" Message

You can change the language of the message that appears when there are no records to show in the grid.

To change the language of the "No Records" Message

1   Right-click on the canvas and click **Scripts**. The **Edit Scripts** dialog box appears.

2   Click the Add icon and give the script a name, for example ChangeLanguage.

3   In the **Expression** box, type:
```
intouch:$Language
```

4   In the **Trigger** list, click **DataChange**.

5   In the script area, type the following:
```
If intouch:$Language == 1033 then ' Switch to
English
AlarmClient1.NoRecordsMessage.Message = "No
Records";
    else If intouch:$Language == 1031 then ' Switch
    to German
    AlarmClient1.NoRecordsMessage.Message = "Keine
    Einträge";
        else if intouch:$Language == 1036 then '
        Switch to French
        AlarmClient1.NoRecordsMessage.Message =
        "Aucun enregistrement";
        endif;
    endif;
endif;
```

6   Click **OK**.

Configuring the Run-Time Shortcut Menu

You can configure the run-time shortcut menu to show only selected options or to be hidden at run time. The shortcut menus of the Alarm Control showing historical alarms (or events) and the Alarm Control showing current alarms (or recent alarms and events) are different.

Ack All
Ack Visible
Ack Selected Groups
Ack Selected Tags
Ack Selected Priorities

Ack Selected	
Ack Others	▶
Hide Selected	
Hide Others	▶
Hidden...	
Sort...	
Query Filters...	
Freeze	
Statistics...	
Requery	
Reset	

Hide All
Hide Visible
Hide Selected Groups
Hide Selected Tags
Hide Selected Priorities
Unhide All

Sort...
Query Filters...
Requery
Reset

Current alarms mode   Historical alarms mode

For the current alarms shortcut menu, you can also show or hide entire shortcut submenus. Use the **ContextMenu.*** and the **ShowContextMenu Property** properties in scripting to control if shortcut menu items appear or not at run time. For more information, see the ContextMenu.AckAll Property on page 95 and following.

To hide the shortcut menu

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Clear the **Show Context Menu** check box.

4   Click **OK**.

To show or hide shortcut menu options

1   Double-click the Alarm Control on the canvas. The **Edit Animations** dialog box appears.

2   Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.

3   Make sure the **Show Context Menu** check box is selected.

☑ Show Context Menu

☑ Ack Selected	☑ Ack All
☑ Ack Others ▶	☑ Ack Visible
☑ Hide Selected	☑ Ack Selected Groups
☑ Hide Others ▶	☑ Ack Selected Tags
☑ Hidden...	☑ Ack Selected Priorities
☑ Sort...	
☑ Query Filters...	☑ Hide All
☑ Freeze	☑ Hide Visible
☑ Statistics...	☑ Hide Selected Groups
☑ Requery	☑ Hide Selected Tags
☑ Reset	☑ Hide Selected Priorities
	☑ Unhide All

4   In the shortcut menu lists, do the following:

a   Select the options you want to appear on the run-time shortcut menu (if applicable for the selected client mode).

b   Clear the options you want to hide from the operator on the run-time shortcut menu.

5   Click **OK**.

Chapter 3

Using the Alarm Control at Run Time

This section shows how you can interact with the Alarm Control at run time, such as:

- Refreshing the Alarm Control grid to show the most current alarms.

- Using the status bar to view various information about the alarm records.

- Acknowledging, hiding, filtering, or sorting alarms.

- Freezing the Alarm Control grid.

- Switching between client modes.

- Switching between languages.

Refreshing the Alarm Control Grid

You can refresh the Alarm Control grid at run time. Depending on which client mode is selected, the Alarm Control:

- Requeries the Alarm Manager for latest current alarm records from all configured providers.

- Retrieves alarm record data from the Alarm Database based on the time range settings.

To refresh the alarm control grid

1   Right-click the Alarm Control grid at run time. The shortcut menu appears.

2   Click **Requery**.

Using Status Bar Information

The status bar shows you information about the current Alarm Control grid. Depending on the client mode, the status bar information shows different information.

Using Status Bar Information of Current Modes

If the Alarm Control is showing current alarms or recent alarms and events, the status bar shows the following:

Element	Icon(s)	Description
Client Mode		Indicates the Alarm Control is showing current alarms (or recent alarms and events).
New Alarms		Appears if new alarms have occurred. If you move the pointer over the indicator, the tooltip shows you how many alarms are unacknowledged.
Hidden Alarms		Appears if any alarms are currently hidden. If you move the pointer over the indicator, the tooltip shows you how many alarms are hidden.
Frozen Grid		Appears if the Alarm Control is currently frozen.

Element	Icon(s)	Description
Alarm Records		Displaying 1 to 13 of 28 alarms
		Shows the total number of alarm records and which alarms are currently shown.
Query Filter		Default
		Shows the name of the current query filter favorite.
Retrieval		100% Complete
		Shows the percentage of alarms retrieved from all alarm providers.
		If this percentage is less than 100%, not all alarm providers are providing alarm data. Use the **Alarm Statistics** dialog box to detect which alarm providers are not providing alarm data.
Time Zone		Beijing, Chongqing, Hong Kong, Urumqi
		Shows the current time zone of the Alarm Control. Move the pointer over the time zone to show the full information in a tool tip.

Using Status Bar Information of Historical Modes

If the Alarm Control is showing historical alarms or events, the status bar shows the following:

Element	Description
Client Mode	Indicates the Alarm Control is showing historical alarms and/or events.
Alarm Records	Displaying 1 to 13 of 28 alarms Shows the total number of alarm records and which alarms are currently shown.
Alarm Database	localhost - WWAlmDb Shows the name of the server hosting the Alarm Database and the Alarm Database name.
Connection Status	Connected Shows the connection status to the Alarm Database.
Time Zone	Beijing, Chongqing, Hong Kong, Urumqi Shows the current time zone of the Alarm Control. Move the pointer over the time zone to show the full information in a tool tip.
Requery	Click this button to retrieve latest alarm records from the Alarm Database.

Acknowledging Alarms

You can acknowledge alarm records in alarm state directly from the Alarm Control. You can acknowledge:

- One or more selected alarms.

- All alarms, including alarms not visible due to the limited space of the Alarm Control.

- All visible alarms.

- All alarms with common values, such as provider names, group names, priority ranges, and tag names. You can simplify alarm acknowledgement for the operator by using methods in scripting. For more information, see Ack.All() Method on page 125.

To acknowledge a selected alarms using the Alarm Control grid

1 Select one or more alarms in alarm state.

2 Right-click the Alarm Control grid and click **Ack Selected**.

3 If no default acknowledgement statement is configured for the Alarm Control, the **Ack Comment** dialog box appears.

4 Type an alarm acknowledgement comment and click **OK**.

To acknowledge other alarms using the Alarm Control grid

1 Select one or more alarms in alarm state.

2 Right-click the Alarm Control grid, point to **Ack Others**, and click one of the following:
   - **Ack All** to acknowledge all alarms in alarm state.
   - **Ack Visible** to acknowledge all visible alarms.
   - **Ack Selected Group** to acknowledge alarms with the same provider names and group names of one or more selected alarms in alarm state.
   - **Ack Selected Tag** to acknowledge alarms with the same provider names, group names, and tag names within the priority ranges of one or more selected alarms in alarm state.
   - **Ack Selected Priority** to acknowledge alarms with the same provider names, group names, and within the priority ranges of one or more selected alarms in alarm state.

3 If no default acknowledgement statement is configured for the Alarm Control, the **Ack Comment** dialog box appears.

4 Type an alarm acknowledgement comment and click **OK**.

Sorting Alarms at Run Time

You can sort alarms at run time in similar way as design time. Any changes you make to the sorting at run time are lost when you switch back to design time. You can simplify alarm sorting for the operator by using methods in scripting. For more information, see SetSort() Method on page 141 and Show.Sort() Method on page 142.

Note If you are sorting by the Value column, the items in the column are sorted alphabetically, not numerically. This is because the Value column can contain strings.

To set sorting columns and directions with lists at run time

1   Right-click the Alarm Control grid and click **Sort**. The **Sort** dialog box appears.
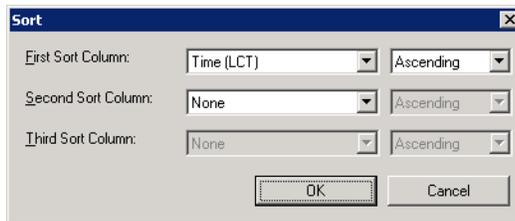
2   In the **First Sort Column** list, select the first sort column and a sorting direction in the list to its right.

3   Optionally, select the second sort column in the **Second Sort Column** list and a sorting direction in the list to its right.

4   If you set the **Second Sort Column**, optionally select the third sort column in the **Third Sort Column** list and a sorting direction in the list to its right.

5   Click **OK**.

To set sorting columns and directions in the grid at run time

1   In the Alarm Control grid, click on a column header to set sorting for the column. An arrow appears on the column header.

2   To change the sorting direction, click on the column header again. The arrow changes direction on the header.

Note If you click on a column header after releasing the **Shift** key, all sorting information is lost and the selected column is the new primary sorting criteria.

3   To set sorting for second and third columns, repeat step 3 while pressing the **Shift** key.

4   Release the **Shift** key.

Filtering Alarms at Run Time

You can filter alarms at run time by using the filters you defined at design time.

If you did not define a filter according to your needs at design time, you can still create new filters at run time, or modify existing filters.

If you saved filters to an XML file, you can load them from a file at run-time.

Filters you define at run-time are not saved for use at design-time. To re-use filters you create or modify at run-time, export the filter list to an XML file, and import the XML file at design-time.

If you are showing historical alarms or events, you can use the filtering mechanism provided by the grid technology instead of using filter favorites.

Using an Existing Query Filter

At run time, you can use any filter you defined at design time, regardless if you defined it for the current modes or historical modes. You can also use scripting to switch to an existing query filter. For more information, see Favorite Property on page 106.

To use an existing query filter

1   Right-click the Alarm Control grid and click **Query Filters**. The **Query Filters** dialog box appears.

2   Select the filter from the list and click **OK**. The alarm records are filtered and the current filter name appears in the status bar.
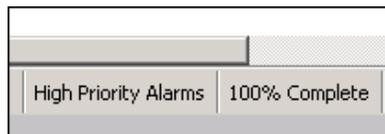
Adding a New Query Filter

At run time, you can create new query filters to limit the number of alarm records.

The new query filters are not stored for future use and are only valid for the current session. If you want to store them for future use, you must also export the query filters to an XML file. For more information, see Exporting Query Filter Favorites on page 69.

To add a new query filter

1   Right-click the Alarm Control grid and click **Query Filters**. The **Query Filters** dialog box appears.

2   The configuration is the same as in design time. For more information, see Adding a New Query Filter on page 44.

Modifying an Existing Query Filter

At run time, you can modify a query filter.

The modification of query filters is not saved for future use and is only valid for the current session. If you want to save the modifications, you must also export the query filters to an XML file. For more information, see Exporting Query Filter Favorites on page 69.

To modify an existing query filter

1   Right-click the Alarm Control grid and click **Query Filters**. The **Query Filters** dialog box appears.

2   The configuration is the same as in design time. For more information, see Modifying an Existing Query Filter on page 47.

Deleting a Query Filter

At run time, you can delete a query filter.

After you delete a query filter at run time, it is only deleted for the current session. If you want to save the list of query filters without the deleted query filter, you must export the query filters to an XML file. For more information, see Exporting Query Filter Favorites on page 69.

To delete an existing query filter

1   Right-click the Alarm Control grid and click **Query Filters**. The **Query Filters** dialog box appears.

2   The configuration is the same as in design time. For more information, see Deleting a Query Filter Favorite on page 47.

Importing Query Filter Favorites

At run time, you can import the list of query filters from an XML file.

To import Query Filter Favorites

1   Right-click the Alarm Control grid and click **Query Filters**. The **Query Filters** dialog box appears.

2   The configuration is the same as in design time. For more information, see Importing Query Filter Favorites on page 48.

Exporting Query Filter Favorites

At run time, you can export the list of query filters to an XML file for future use. After exporting, you can import the query filter from the XML into design time.

Note The default query filter favorite is not exported to the XML file.

To export Query Filter Favorites

1   Right-click the Alarm Control grid and click **Query Filters**. The **Query Filters** dialog box appears.

2   The configuration is the same as in design time. For more information, see Exporting Query Filter Favorites on page 47.

Filtering Alarms with Client-Based Filtering

The grid technology used in the Alarm Control lets you filter the grid contents after the data has been retrieved from the data source.

You can filter historical alarms and/or events in the following ways for any selected column:

Filter	Description
(All)	No filtering, all records are shown for the selected column.
(Custom)	Lets you configure a more complex filter for the selected column, for example a filter that can compare values of different columns.
(Blanks)	Filters by showing blank values only.
(NonBlanks)	Filters by showing non blank values only.
Values	Filters by the selected value.

If a filter is applied to any column in the Alarm Control, the filter icon in the column header appears in blue .

To filter alarms with client-based filtering

1 Click the filter icon on the column you want to filter by. A menu appears.

2 Select one of the following:
- **(All)** to switch off filtering.
- **(Custom)** to define a more complex filter.
- **(Blanks)** to filter by blank values.
- **(NonBlanks)** to filter by non blank values.
- A value to filter by the value.

If you selected **(Custom)**, a dialog box appears.
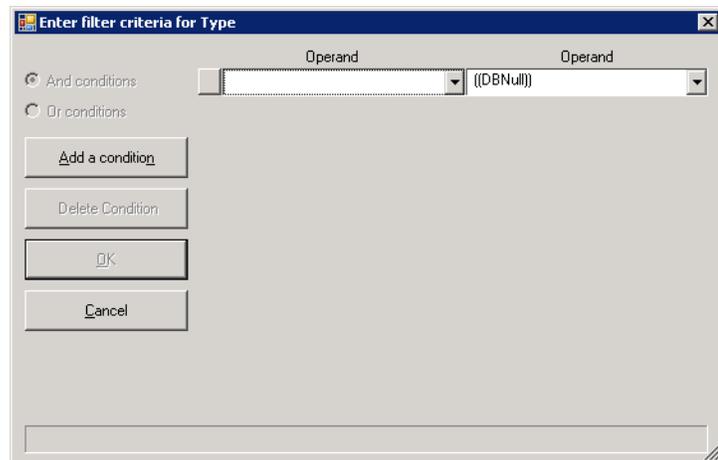
3 Do one of the following:

- Select a different operator for the current condition.
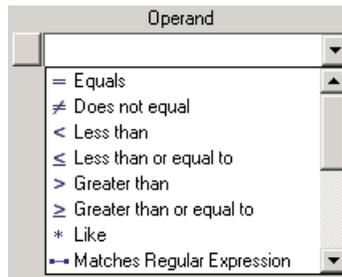
- Type or select a different operand for the current condition. The operand can be a value, or the value of a different column in the same row.

- Click **Add a condition** to add more conditions to the filter.
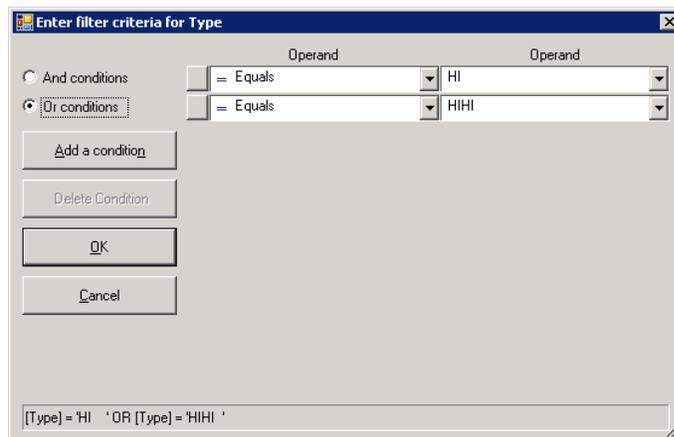
- Click **Delete Condition** to delete one or more selected conditions. (You can mark the condition by clicking on the button to the left of each condition.)

4 Click **OK**.

Resetting the Grid

You can reset the column widths, column order, and names to their last design-time values. When you reset the grid, the query filter is also reset to its default. You can also reset the grid by using a method in scripting. For more information, see Reset() Method on page 138.

To reset the grid

◆ Right-click the Alarm Control grid and click **Reset**.

Hiding Alarms

You can temporarily remove specified alarms from the Alarm Control by hiding them. You can hide:

• All alarms, including alarms not visible due to the limited space of the Alarm Control.

• All visible alarms.

• One or more selected alarms.

• All alarms with the same provider names and group names of one or more selected alarms.

• All alarms with the same provider names, group names, and within the priority ranges of one or more selected alarms.

• All alarms with the same provider names, group names, and tag names within the priority ranges of one or more selected alarms.

You can also view which alarms are hidden and unhide them. You can simplify alarm hiding and unhiding for the operator by using methods in scripting. For more information, see Hide.All() Method on page 133.

To hide all alarms

◆ Right-click the Alarm Control grid, point to **Hide Others**, and click **Hide All**.

To hide all visible alarms

◆ Right-click the Alarm Control grid, point to **Hide Others**, and click **Hide Visible**.

To hide selected alarms

1   Select one or more alarms in alarm state.

2   Right-click the Alarm Control grid and click **Hide Selected**.

To hide alarms with common parameters

1   Select one or more alarms.

2   Right-click the Alarm Control grid, point to **Hide Others**, and click one of the following:

    • **Hide Selected Group** to hide alarms with the same provider names and group names of one or more selected alarms.

    • **Hide Selected Tag** to hide alarms with the same provider names, group names, and tag names within the priority ranges of one or more selected alarms.

    • **Hide Selected Priority** to hide alarms with the same provider names, group names, and within the priority ranges of one or more selected alarms.

To unhide alarms

1   Right-click the Alarm Control grid and click **Hidden**. The **Hidden Alarms** dialog box appears.

2   Select the alarms you want to unhide and click **Unhide**.

3   Click **Close**.

Showing Alarm Statistics

You can view alarm statistics at run time to see which alarm providers are providing the alarm data. You can also use scripting to show alarm statistics at run time. For more information, see Show.Statistics() Method on page 142.

To show alarm statistics

1   Right-click the Alarm Control grid and click **Statistics**. The **Alarm Statistics** dialog box appears.
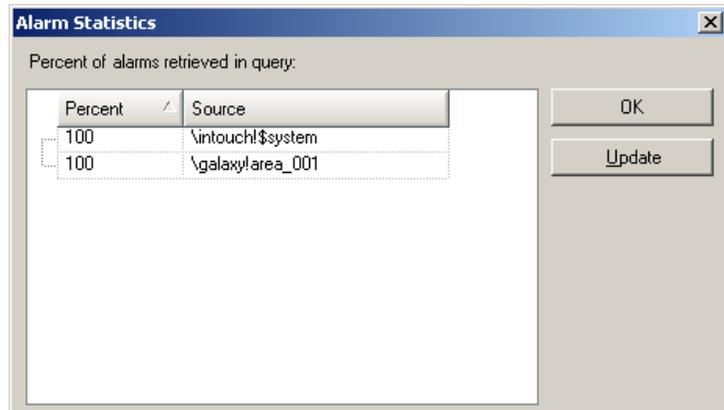
Note If you use an Alarm Hotbackup name as alarm query, you can expand the Hotbackup name in the **Alarm Statistics** dialog box to show the individual percentages of retrieval for the configured primary and backup alarm provider.

2   Click **Update** to update the statistics.

3   Click **Close**.

Freezing and Unfreezing the Alarm Control Grid

You can freeze the Alarm Control grid at run time so that no more updates are shown.

After you unfreeze the Alarm Control, the grid updates with the new alarm records and any other updates while the grid was frozen.You can also use scripting to freeze and unfreeze the Alarm Control grid at run time. For more information, see FreezeDisplay() Method on page 131.

To freeze or unfreeze the Alarm Control grid

1   Right-click the Alarm Control grid. The shortcut menu appears.
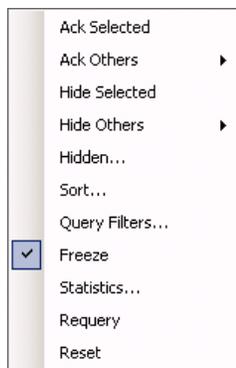
A check mark next to the **Freeze** option indicates if the grid is currently frozen.

2   Click **Freeze**. The Alarm Control grid is either frozen or unfrozen.

Switching between Client Modes

You can switch between client modes at run time by changing the Alarm Control ClientMode property. The easiest way to do this, is to configure an ArchestrA script to interact with the Alarm Control ClientMode property at design time.

To switch between client modes

1   Place the Alarm Control on the ArchestrA symbol.

2   Paste a button on the canvas and change its caption to **Current Alarms**.

3   Double-click the button and configure it with the following action script:

```
AlarmControlGrid1.ClientMode = 1;
```

4 Click **OK**.

5 Repeat steps 2 to 4 for the following buttons:

Button Caption	Action script
Recent Alarms and Events	`AlarmControlGrid1.ClientMode = 2;`
Historical Alarms	`AlarmControlGrid1.ClientMode = 3;`
Historical Events	`AlarmControlGrid1.ClientMode = 4;`
Historical Alarms and Events	`AlarmControlGrid1.ClientMode = 5;`

6 Save and close the ArchestrA symbol.

7 Create a new managed InTouch application and open it in WindowMaker.

8 Place the ArchestrA symbol on a new InTouch window.

9 Switch to WindowViewer to test your application.

10 Click **Historical Alarms** to show historical alarms instead of current alarms.

By default, the Alarm Control tries to connect to the alarm database called WWALMDB on the local computer using the currently logged on user. If you are using a different configuration, you can use value input links or action script to set the following properties:

- **Database.ServerName Property**

- **Database.UserID Property**

- **Database.Password Property**

- **Domain Property**

- **Database.Name Property**

- **Database.Authentication Property**

Switching Run-Time Languages

You can switch the language of the Alarm Control in the same way as other parts of your InTouch application. When you switch language, the alarm state, alarm class, alarm type, the various alarm comment fields, and the column headers are switched to the selected language.

To switch the language

◆   Do one of the following:

   • In WindowViewer on the **Special** menu, point to **Languages**, and then click the language you want to switch to.

   • In WindowMaker, use the InTouch QuickScript **SwitchDisplayLanguage** in a button action script to switch the language. At run time, click the button to switch the language.

   • In WindowMaker, use the system tag $Language in a button action script and assign it to the language code you want to switch to. At run time, click the button to switch the language.

For more information about run-time language switching, see Chapter 11, Working with Languages, in the *Application Server User's Guide.*

Important If you rename or reorder column headers, you must repeat the symbol text translation procedures. If you do not, your changes will not be available for run-time language switching.

Chapter 4

Scripting the Alarm Control

This section shows you the properties, methods, and events for the Alarm Control.

Alarm Control Properties

This section describes all the properties available for scripting in the Alarm Control.

AckComment.DefaultValue Property

The AckComment.DefaultValue property is a read-write string property that gets or sets the default acknowledgement comment when the AckComment.UseDefault property is TRUE.

Syntax

result = AlarmClient.AckComment.DefaultValue;

AlarmClient.AckComment.DefaultValue = *ackComment*;

Example

AlarmClient1.AckComment.UseDefault = 1;

AlarmClient1.AckComment.DefaultValue = "This alarm is
  acknowledged by John Smith";

Remarks

For more information, see Showing Current Alarms or Recent Alarms and Events on page 25.

AckComment.UseDefault Property

The AckComment.UseDefault property is a read-write Boolean property that gets or sets the usage of the default acknowledgement comment.

Syntax

result = *AlarmClient*.AckComment.UseDefault;

AlarmClient.AckComment.UseDefault = *useComment*;

Example

```
AlarmClient1.AckComment.UseDefault = 1;

AlarmClient1.AckComment.DefaultValue = "This alarm is
  acknowledged by John Smith";
```

Remarks

For more information, see Showing Current Alarms or Recent Alarms and Events on page 25.

AlarmColor.Ack.BackGround Property

The AlarmColor.Ack.BackGround property is an array of read-write integer properties that get or set the background colors of all acknowledged alarm records.

Index	Purpose
0	Sets the background color of all acknowledged alarm records in all priority ranges.
1	Gets or sets the background color of acknowledged alarm records in the priority range 1 to AlarmColor.Range[1].
2	Gets or sets the background color of acknowledged alarm records in the priority range AlarmColor.Range[1] to AlarmColor.Range[2].
3	Gets or sets the background color of acknowledged alarm records in the priority range AlarmColor.Range[2] to AlarmColor.Range[3].
4	Gets or sets the background color of acknowledged alarm records in the priority range AlarmColor.Range[3] to 999.

Syntax

```
Color = AlarmClient.AlarmColor.Ack.BackGround[n];

AlarmClient.AlarmColor.Ack.BackGround[n] = Color;
```

Parameters

n
  Index from 0 to 4.

Color
  Color of background.

Examples

```
AlarmClient1.AlarmColor.Ack.BackGround[0] = Color.Red;

AlarmClient1.AlarmColor.Ack.BackGround[1] =
  Color.FromARGB(0,128,0);

AlarmClient1.AlarmColor.Ack.BackGround[2] = Color.Grey;

AlarmClient1.AlarmColor.Ack.BackGround[3] =
  Color.Yellow;

AlarmClient1.AlarmColor.Ack.BackGround[4] =
  Color.Black;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.Ack.ForeGround Property

The AlarmColor.Ack.ForeGround property is an array of read-write integer properties that get or set the text colors of all acknowledged alarm records.

Index	Purpose
0	Sets the text color of all acknowledged alarm records in all priority ranges.
1	Gets or sets the text color of acknowledged alarm records in the priority range 1 to AlarmColor.Range[1].
2	Gets or sets the text color of acknowledged alarm records in the priority range AlarmColor.Range[1] to AlarmColor.Range[2].
3	Gets or sets the text color of acknowledged alarm records in the priority range AlarmColor.Range[2] to AlarmColor.Range[3].
4	Gets or sets the text color of acknowledged alarm records in the priority range AlarmColor.Range[3] to 999.

Syntax

```
Color = AlarmClient.AlarmColor.Ack.ForeGround[n];

AlarmClient.AlarmColor.Ack.ForeGround[n] = Color;
```

Parameters

n
  Index from 0 to 4.

Color
  Color of text.

Examples

```
AlarmClient1.AlarmColor.Ack.ForeGround[0] =
  Color.Black;

AlarmClient1.AlarmColor.Ack.ForeGround[1] = Color.Blue;

AlarmClient1.AlarmColor.Ack.ForeGround[2] =
  Color.Green;

AlarmClient1.AlarmColor.Ack.ForeGround[3] =
  Color.Yellow;

AlarmClient1.AlarmColor.Ack.ForeGround[4] =
  Color.FromARGB(0,128,0);
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.Ack.RTN.BackGround Property

The AlarmColor.Ack.RTN.BackGround property is a read-write color property that gets or sets the background color of acknowledged alarm records that "return to normal" (ACK_RTN).

Syntax

```
Color = AlarmClient.AlarmColor.Ack.RTN.BackGround;

AlarmClient.AlarmColor.Ack.RTN.BackGround = Color;
```

Parameters

Color
  Color of background.

Return Value

Returns the background color of acknowledged alarms that "return to normal".

Example

```
AlarmClient1.AlarmColor.Ack.RTN.BackGround =
  Color.Blue;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.Ack.RTN.ForeGround Property

The AlarmColor.Ack.RTN.ForeGround property is a read-write color property that gets or sets the text color of acknowledged alarm records that "return to normal" (ACK_RTN).

Syntax

```
Color = AlarmClient.AlarmColor.Ack.RTN.ForeGround;

AlarmClient.AlarmColor.Ack.RTN.ForeGround = Color;
```

Parameters

Color
  Color of text.

Example

```
AlarmClient1.AlarmColor.Ack.RTN.ForeGround =
  Color.Black;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.Range Property

The AlarmColor.Range property is an array of read-write integer properties that get or set the boundaries of the priority ranges.

You can use priority ranges to classify, group, and emphasize alarms and events belonging to a certain priority range.

The boundaries must fulfill the following condition:

1 < Range[1] < Range[2] < Range[3] < 999

By default, the boundaries are set as follows:

• AlarmColor.Range[1]	250
• AlarmColor.Range[2]	500
• AlarmColor.Range[3]	750

Syntax

```
RangeN = AlarmClient.AlarmColor.Range[N];

AlarmClient.AlarmColor.Range[1] = RangeN;
```

Parameters

N

　Range index 1, 2, or 3.

Example

The following example defines four priority ranges (1 to 50, 51 to 600, 601 to 800, and 801 to 999):

```
AlarmClient1.AlarmColor.Range[1] = 50;

AlarmClient1.AlarmColor.Range[2] = 600;

AlarmClient1.AlarmColor.Range[3] = 800;
```

Remarks

For more information, see Setting Priority Ranges for Alarm Records on page 34.

AlarmColor.RTN.BackGround Property

The AlarmColor.RTN.BackGround property is a read-write color property that gets or sets the background color of alarm records that "return to normal" (ACK_RTN and UNACK_RTN).

Syntax

```
Color = AlarmClient.AlarmColor.RTN.BackGround;

AlarmClient.AlarmColor.RTN.BackGround = Color;
```

Parameters

Color

　　Color of background.

Example

```
AlarmClient1.AlarmColor.RTN.BackGround = Color.Blue;
```

Remarks

For more information, see Setting Return To Normal Record Colors on page 32.

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.RTN.ForeGround Property

The AlarmColor.RTN.ForeGround property is a read-write color property that gets or sets the text color of alarm records that "return to normal" (ACK_RTN and UNACK_RTN).

Syntax

```
Color = AlarmClient.AlarmColor.RTN.ForeGround;

AlarmClient.AlarmColor.RTN.ForeGround = Color;
```

Parameters

Color
  Color of text.

Example

```
AlarmClient1.AlarmColor.RTN.ForeGround = Color.Yellow;
```

Remarks

For more information, see Setting Return To Normal Record Colors on page 32.

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.UnAck.BackGround Property

The AlarmColor.UnAck.BackGround property is an array of read-write integer properties that get or set the background colors of all unacknowledged alarm records.

Index	Purpose
0	Sets the background color of all unacknowledged alarm records in all priority ranges.
1	Gets or sets the background color of unacknowledged alarm records in the priority range 1 to AlarmColor.Range[1].
2	Gets or sets the background color of unacknowledged alarm records in the priority range AlarmColor.Range[1] to AlarmColor.Range[2].

Index	Purpose
3	Gets or sets the background color of unacknowledged alarm records in the priority range AlarmColor.Range[2] to AlarmColor.Range[3].
4	Gets or sets the background color of unacknowledged alarm records in the priority range AlarmColor.Range[3] to 999.

Syntax

```
Color = AlarmClient.AlarmColor.UnAck.BackGround[n];
```

```
AlarmClient.AlarmColor.UnAck.BackGround[n] = Color;
```

Parameters

n
  Index from 0 to 4.

Color
  Color of background.

Example

```
AlarmClient1.AlarmColor.UnAck.BackGround[0] =
  Color.Blue;
```

```
AlarmClient1.AlarmColor.UnAck.BackGround[1] =
  Color.ARGB(223,113,76);
```

```
AlarmClient1.AlarmColor.UnAck.BackGround[2] =
  Color.Yellow;
```

```
AlarmClient1.AlarmColor.UnAck.BackGround[3] =
  Color.Green;
```

```
AlarmClient1.AlarmColor.UnAck.BackGround[4] =
  Color.White;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.UnAck.Flash.BackGround Property

The AlarmColor.UnAck.Flash.BackGround property is an array of read-write color properties that get or set the background colors of all flashing unacknowledged alarm records.

Index	Purpose
0	Sets the background color of all flashing unacknowledged alarm records in all priority ranges.
1	Gets or sets the background color of flashing unacknowledged alarm records in the priority range 1 to AlarmColor.Range[1].
2	Gets or sets the background color of flashing unacknowledged alarm records in the priority range AlarmColor.Range[1] to AlarmColor.Range[2].
3	Gets or sets the background color of flashing unacknowledged alarm records in the priority range AlarmColor.Range[2] to AlarmColor.Range[3].
4	Gets or sets the background color of flashing unacknowledged alarm records in the priority range AlarmColor.Range[3] to 999.

Syntax

```
Color =
  AlarmClient.AlarmColor.UnAck.Flash.BackGround[n];
```

```
AlarmClient.AlarmColor.UnAck.Flash.BackGround[n] =
  Color;
```

Parameters

n
  Index from 1 to 4.

Color
  Color of background.

Example

```
AlarmClient1.AlarmColor.UnAck.Flash.BackGround[1] =
  Color.ARGB(223,113,76);
```

```
AlarmClient1.AlarmColor.UnAck.Flash.BackGround[2] =
  Color.Yellow;
```

```
AlarmClient1.AlarmColor.UnAck.Flash.BackGround[3] =
  Color.Green;
```

```
AlarmClient1.AlarmColor.UnAck.Flash.BackGround[4] =
  Color.White;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.UnAck.Flash.ForeGround Property

The AlarmColor.UnAck.Flash.ForeGround property is an array of read-write color properties that get or set the text colors of all flashing unacknowledged alarm records.

Index	Purpose
0	Sets the text color of all flashing unacknowledged alarm records in all priority ranges.
1	Gets or sets the text color of flashing unacknowledged alarm records in the priority range 1 to AlarmColor.Range[1].
2	Gets or sets the text color of flashing unacknowledged alarm records in the priority range AlarmColor.Range[1] to AlarmColor.Range[2].
3	Gets or sets the text color of flashing unacknowledged alarm records in the priority range AlarmColor.Range[2] to AlarmColor.Range[3].
4	Gets or sets the text color of flashing unacknowledged alarm records in the priority range AlarmColor.Range[3] to 999.

Syntax

```
Color =
  AlarmClient.AlarmColor.UnAck.Flash.ForeGround[n];
```

AlarmClient.AlarmColor.UnAck.Flash.ForeGround[*n*] =
  Color;

Parameters

n
  Index from 1 to 4.

Color
  Color of text.

Examples

```
AlarmClient1.AlarmColor.UnAck.Flash.ForeGround[1] =
   Color.ARGB(223,113,76);
```

```
AlarmClient1.AlarmColor.UnAck.Flash.ForeGround[2] =
   Color.Yellow;
```

```
AlarmClient1.AlarmColor.UnAck.Flash.ForeGround[3] =
   Color.Green;
```

```
AlarmClient1.AlarmColor.UnAck.Flash.ForeGround[4] =
   Color.White;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.UnAck.ForeGround Property

The AlarmColor.UnAck.ForeGround property is an array of read-write integer properties that get or set the text colors of all unacknowledged alarm records.

Index	Purpose
0	Sets the text color of all unacknowledged alarm records in all priority ranges.
1	Gets or sets the text color of unacknowledged alarm records in the priority range 1 to AlarmColor.Range[1].
2	Gets or sets the text color of unacknowledged alarm records in the priority range AlarmColor.Range[1] to AlarmColor.Range[2].
3	Gets or sets the text color of unacknowledged alarm records in the priority range AlarmColor.Range[2] to AlarmColor.Range[3].
4	Gets or sets the text color of unacknowledged alarm records in the priority range AlarmColor.Range[3] to 999.

Syntax

```
Color = AlarmClient.AlarmColor.UnAck.ForeGround[n];
```

```
AlarmClient.AlarmColor.UnAck.ForeGround[n] = Color;
```

Parameters

n
  Index from 0 to 4.

Color
  Color of text.

Example

```
AlarmClient1.AlarmColor.UnAck.ForeGround[0] =
  Color.Blue;
```

```
AlarmClient1.AlarmColor.UnAck.ForeGround[1] =
  Color.ARGB(223,113,76);
```

```
AlarmClient1.AlarmColor.UnAck.ForeGround[2] =
  Color.Yellow;
```

```
AlarmClient1.AlarmColor.UnAck.ForeGround[3] =
  Color.Green;
```

```
AlarmClient1.AlarmColor.UnAck.ForeGround[4] =
  Color.White;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.UnAck.RTN.BackGround Property

The AlarmColor.UnAck.RTN.BackGround property is a read-write color property that gets or sets the background color of unacknowledged alarm records that "return to normal" (UNACK_RTN).

Syntax

```
Color = AlarmClient.AlarmColor.UnAck.RTN.BackGround;
```

```
AlarmClient.AlarmColor.UnAck.RTN.BackGround = Color;
```

Parameters

Color
  Color of background.

Example

```
AlarmClient1.AlarmColor.UnAck.RTN.BackGround =
  Color.Blue;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmColor.UnAck.RTN.ForeGround Property

The AlarmColor.UnAck.RTN.ForeGround property is a read-write color property that gets or sets the text color of unacknowledged alarm records that "return to normal" (UNACK_RTN).

Syntax

```
Color = AlarmClient.AlarmColor.UnAck.RTN.ForeGround;

AlarmClient.AlarmColor.UnAck.RTN.ForeGround = Color;
```

Parameters

Color
  Color of text.

Example

```
AlarmClient1.AlarmColor.UnAck.RTN.ForeGround =
  Color.FromARGB(0,0,0);
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

AlarmQuery Property

The AlarmQuery property is a read-write string property that gets or sets the alarm query.

Syntax

```
result = AlarmClient.AlarmQuery;

AlarmClient.AlarmQuery = AlmQry;
```

Parameters

AlmQry
   Alarm query string in format \\node\provider!group where node is optional.

Example

```
AlarmClient.AlarmQuery = "\intouch!GroupA";
```

Remarks

After you write a new value to the AlarmQuery property, the Alarm Control is updated. If you are using the default query filter, the query is updated with the new node, provider, and group name.

AllowColumnResize Property

The AllowColumnResize property is a read-write Boolean property that gets or sets the ability to resize the columns at run time.

Syntax

```
result = AlarmClient.AllowColumnResize;

AlarmClient.AllowColumnResize = allowColResizing;
```

AutoResumeDuration Property

The AutoResumeDuration property is a read-write integer property that gets or sets the time in seconds after which the grid becomes unfrozen and resumes showing alarms.

Set this value to 0 to disable auto resume.

Syntax

```
result = AlarmClient.AutoResumeDuration;

AlarmClient.AllowColumnResize = timeout;
```

AutoScroll Property

The AutoScroll property is a read-write Boolean property that gets or sets automatic scrolling to new alarms.

Syntax

result = AlarmClient.AutoScroll;

AlarmClient.AutoScroll = *allowAutoscroll*;

ClientMode Property

The ClientMode property is a read-write integer property that gets or sets the client mode for the Alarm Control. Use one of the following values:

Value	Client Mode
1	Current Alarms
2	Recent Alarms and Events
3	Historical Alarms
4	Historical Events
5	Historical Alarms and Events

Syntax

result = AlarmClient.ClientMode;

AlarmClient.ClientMode = *clientMode*;

Example

AlarmClient1.ClientMode = 2;

LogMessage("Alarm client set to Recent Alarms and
  Events");

Remarks

For more information, see Showing Current Alarms or Recent Alarms and Events on page 25.

ConnectStatus Property

The ConnectStatus property is a read-only string property that gets the status of the connection to the Alarm Database.

Syntax

```
result = AlarmClient.ConnectStatus;
```

Return Value

Returns the status of the connection to the alarm database. Can be "Connected," "Not connected," or "In progress."

Example

```
alive = AlarmClient1.ConnectStatus;

if alive == "Connected" then
    LogMessage("The Alarm Control is currently connected
    to the Alarm Database");

else
    LogMessage("The Alarm Control is either currently
    connecting to the Alarm Database or not
    connected.");

endif;
```

ContextMenu.AckAll Property

The ContextMenu.AckAll property is a read-write Boolean property that gets or sets the appearance of the **Ack All** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckAll;

AlarmClient.ContextMenu.AckAll = AckAllVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.AckOthers Property

The ContextMenu.AckOthers property is a read-write Boolean property that gets or sets the appearance of the **Ack Others** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckOthers;

AlarmClient.ContextMenu.AckOthers = AckOthersVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.AckSelected Property

The ContextMenu.AckSelected property is a read-write Boolean property that gets or sets the appearance of the **Ack Selected** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckSelected;
```

```
AlarmClient.ContextMenu.AckSelected = AckSelectedVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.AckSelectedGroups Property

The ContextMenu.AckSelectedGroups property is a read-write Boolean property that gets or sets the appearance of the **Ack Selected Groups** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckSelectedGroups;
```

```
AlarmClient.ContextMenu.AckSelectedGroups =
  AckSelGrpsVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.AckSelectedPriorities Property

The ContextMenu.AckSelectedPriorities property is a read-write Boolean property that gets or sets the appearance of the **Ack Selected Priorities** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckSelectedPriorities;
```

```
AlarmClient.ContextMenu.AckSelectedPriorities =
  AckSelPriVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.AckSelectedTags Property

The ContextMenu.AckSelectedTags property is a read-write Boolean property that gets or sets the appearance of the **Ack Selected Tags** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckSelectedTags;

AlarmClient.ContextMenu.AckSelectedTags =
  AckSelTagsVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.AckVisible Property

The ContextMenu.AckVisible property is a read-write Boolean property that gets or sets the appearance of the **Ack Visible** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.AckVisible;

AlarmClient.ContextMenu.AckVisible = AckVisVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Favorites Property

The ContextMenu.Favorites property is a read-write Boolean property that gets or sets the appearance of the **Query Filters** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Favorites;

AlarmClient.ContextMenu.Favorites = FavsVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Freeze Property

The ContextMenu.Freeze property is a read-write Boolean property that gets or sets the appearance of the **Freeze** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Freeze;

AlarmClient.ContextMenu.Freeze = FreezeVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Hidden Property

The ContextMenu.Hidden property is a read-write Boolean property that gets or sets the appearance of the **Hidden** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Hidden;

AlarmClient.ContextMenu.Hidden = HiddenVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideAll Property

The ContextMenu.HideAll property is a read-write Boolean property that gets or sets the appearance of the **Hide All** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.HideAll;

AlarmClient.ContextMenu.HideAll = HideAllVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideOthers Property

The ContextMenu.HideOthers property is a read-write Boolean property that gets or sets the appearance of the **Hide Others** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.HideOthers;

AlarmClient.ContextMenu.HideOthers = HideOthersVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideSelected Property

The ContextMenu.HideSelected property is a read-write Boolean property that gets or sets the appearance of the **Hide Selected** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.HideSelected;

AlarmClient.ContextMenu.HideSelected = HideSelVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideSelectedGroups Property

The ContextMenu.HideSelectedGroups property is a read-write Boolean property that gets or sets the appearance of the **Hide Selected Groups** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.HideSelectedGroups;

AlarmClient.ContextMenu.HideSelectedGroups =
  HideSelGrpsVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideSelectedPriorities Property

The ContextMenu.HideSelectedPriorities property is a read-write Boolean property that gets or sets the appearance of the **Hide Selected Priorities** option on the shortcut menu.

Syntax

```
result =
  AlarmClient.ContextMenu.HideSelectedPriorities;

AlarmClient.ContextMenu.HideSelectedPriorities =
  HideSelPrisVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideSelectedTags Property

The ContextMenu.HideSelectedTags property is a read-write Boolean property that gets or sets the appearance of the **Hide Selected Tags** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.HideSelectedTags;

AlarmClient.ContextMenu.HideSelectedTags =
  HideSelTagsVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.HideVisible Property

The ContextMenu.HideVisible property is a read-write Boolean property that gets or sets the appearance of the **Hide Visible** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.HideVisible;

AlarmClient.ContextMenu.HideVisible = HideVisVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Requery Property

The ContextMenu.Requery property is a read-write Boolean property that gets or sets the appearance of the **Requery** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Requery;

AlarmClient.ContextMenu.Requery = RequeryVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Reset Property

The ContextMenu.Reset property is a read-write Boolean property that gets or sets the appearance of the **Reset** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Reset;

AlarmClient.ContextMenu.Reset = ResetVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Sort Property

The ContextMenu.Sort property is a read-write Boolean property that gets or sets the appearance of the **Sort** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Sort;

AlarmClient.ContextMenu.Sort = SortVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.Statistics Property

The ContextMenu.Statistics property is a read-write Boolean property that gets or sets the appearance of the **Statistics** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.Statistics;

AlarmClient.ContextMenu.Statistics = StatsVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

ContextMenu.UnhideAll Property

The ContextMenu.UnhideAll property is a read-write Boolean property that gets or sets the appearance of the **Unhide All** option on the shortcut menu.

Syntax

```
result = AlarmClient.ContextMenu.UnhideAll;

AlarmClient.ContextMenu.UnhideAll = UnhideAllVis;
```

Remarks

For more information, see Configuring the Run-Time Shortcut Menu on page 58.

Database.Authentication Property

The Database.Authentication property is a read-write string property that gets or sets the authentication mode to connect to the Alarm Database. Possible values are:

• Windows Integrated

• Windows Account

• SQL Server

The default value is "Windows Integrated".

Syntax

```
result = AlarmClient.Database.Authentication;

AlarmClient.Database.Authentication = AuthMode;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Database.Name Property

The Database.Name property is a read-write string property that gets or sets the name of the Alarm Database. The default value is "WWALMDB".

If you change the Database.Name property at run time, you need to call the Connect method to connect to the new alarm database.

Syntax

```
result = AlarmClient.Database.Name;

AlarmClient.Database.Name = AlmDBName;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Database.Password Property

The Database.Password property is a read-write string property that gets or sets the password associated with the user name to connect to the Alarm Database.

Syntax

```
result = AlarmClient.Database.Password;

AlarmClient.Database.Password = Psswrd;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Database.ServerName Property

The Database.ServerName property is a read-write string property that gets or sets the name of the server that hosts the Alarm Database.

Syntax

```
result = AlarmClient.Database.ServerName;

AlarmClient.Database.ServerName = SrvName;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Database.UserID Property

The Database.UserID property is a read-write string property that gets or sets the name of user authorized to access the Alarm Database.

Syntax

```
result = AlarmClient.Database.UserID;

AlarmClient.Database.UserID = UserName;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Domain Property

The Domain property is a read-write string property that gets or sets the domain name of the user to connect to the Alarm Database.

Syntax

```
result = AlarmClient.Domain;

AlarmClient.Domain = DomName;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Enabled Property

The Enabled property is a read-write Boolean property that gets or sets the enablement of Alarm Control. When the Alarm Control is disabled, alarm records are still updated, but the operator cannot interact with the control.

The operator can still use scripting to interact with the control.

Syntax

```
result = AlarmClient.Enabled;

AlarmClient.Enabled = EnableFlag;
```

EventColor.BackGround Property

The EventColor.BackGround property is a read-write color property that gets or sets the background color of event records.

Syntax

```
Color = AlarmClient.EventColor.BackGround;
AlarmClient.EventColor.BackGround = Color;
```

Parameters

Color
   Color of background.

Example

```
AlarmClient1.EventColor.BackGround = Color.Blue;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

EventColor.ForeGround Property

The EventColor.ForeGround property is a read-write color property that gets or sets the text color of event records.

Syntax

```
Color = AlarmClient.EventColor.ForeGround;
AlarmClient.EventColor.ForeGround = Color;
```

Parameters

Color
   Color of text.

Example

```
AlarmClient1.EventColor.ForeGround = Color.Blue;
```

Remarks

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

Favorite Property

The Favorite property is a read-write string property that gets or sets the name of the current query filter favorite.

Syntax

```
QueryFilterName = AlarmClient.Favorite;

AlarmClient.Favorite = QueryFilterName;
```

Parameters

QueryFilterName
  The name of a query filter favorite.

Example

The following example sets the current Alarm Control grid to the Query Filter Favorite with the name "All Hi Priority Alarms".

```
AlarmClient1.Favorite = "All Hi Priority Alarms";
```

Remarks

You can also use this property to reset the currently used query filter to its default with the following script:

```
AlarmClient.Favorite = "Default";
```

FlashUnAckAlarms Property

The FlashUnAckAlarms property is a read-write Boolean property that gets or sets the flashing of unacknowledged alarm records.

Syntax

```
result = AlarmClient.FlashUnAckAlarms;

AlarmClient.FlashUnAckAlarms = FlashUnAckRecs;
```

Remarks

For more information, see Setting Unacknowledged Alarms to Flash on page 36.

GridColor Property

The GridColor property is a read-write color property that gets or sets the color of the grid lines.

Syntax

```
Color = AlarmClient.GridColor;

AlarmClient.GridColor = Color;
```

Parameters

Color
  Color of the grid lines.

Example

```
AlarmClient1.GridColor = Color.Black;
```

Remarks

For more information, see Setting Heading, Grid, and Window Color on page 32.

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

HeadingColor.BackGround Property

The HeadingColor.BackGround property is a read-write color property that gets or sets the background color of the heading.

Syntax

```
Color = AlarmClient.HeadingColor.BackGround;
```
```
AlarmClient.HeadingColor.BackGround = Color;
```

Parameters

Color
  Color of background.

Example

```
AlarmClient1.HeadingColor.BackGround = Color.Blue;
```

Remarks

For more information, see Setting Heading, Grid, and Window Color on page 32.

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

HeadingColor.ForeGround Property

The HeadingColor.ForeGround property is a read-write color property that gets or sets the text color of the heading.

Syntax

```
Color = AlarmClient.HeadingColor.ForeGround;

AlarmClient.HeadingColor.ForeGround = Color;
```

Parameters

Color
  Color of text.

Example

```
AlarmClient1.HeadingColor.ForeGround = Color.Blue;
```

Remarks

For more information, see Setting Heading, Grid, and Window Color on page 32.

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

Height Property

The Height property is a read-write integer property that gets or sets the height of the Alarm Control in pixels.

Syntax

```
result = AlarmClient.Height;

AlarmClient.Height = Hght;
```

HiddenAlarms Property

The HiddenAlarms property is a read-only integer property that gets the number of hidden alarms.

Syntax

```
Result = AlarmClient.HiddenAlarms;
```

Example

```
LogMessage("There are " +
  Text(AlarmClient1.HiddenAlarms,"#")+" hidden
  alarms.");
```

HideErrors Property

The HideErrors property is a read-write Boolean property that gets or sets the Hide Errors option.

- TRUE - Run-time errors, warnings, and status messages are written to the ArchestrA Logger. No pop-ups appear.

- FALSE - Run-time errors, warnings, and status messages pop-up and are also written to the ArchestrA Logger.

Syntax

```
result = AlarmClient.HideErrors;
```

```
AlarmClient.HideErrors = SilentMode;
```

Remarks

For more information, see Hiding Errors, Warnings, and Status Messages on page 54.

MaxDatabaseRecords Property

The MaxDatabaseRecords property is a read-write integer property that gets or sets the maximum database records to retrieve. The valid range is 1 to 32766.

Syntax

```
result = AlarmClient.MaxDatabaseRecords;
```

```
AlarmClient.MaxDatabaseRecords = MaxRecs;
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

NewAlarmEventMode Property

The NewAlarmEventMode property is an read-write integer property that gets or sets the trigger behavior of the New Alarm event.

Syntax

EMode = AlarmClient.NewAlarmEventMode;

AlarmClient.NewAlarmEventMode = *EMode;*

Parameters

EMode

Event mode with following possible values:

Value	Description
0	The NewAlarm event cannot be triggered. (default).
1	The NewAlarm event is triggered only one time the first time a new alarm occurs.
2	The NewAlarm event is triggered every time a new alarm occurs.

NoRecordsMessage.Enabled Property

The NoRecordsMessage.Enabled property is a read-write Boolean property that gets or sets the visibility of a custom message when no alarm records are available.

Syntax

result = AlarmClient.NoRecordsMessage.Enabled;

AlarmClient.NoRecordsMessage.Enabled = *showMessage;*

Example

AlarmClient1.NoRecordsMessage.Enabled = 1;

AlarmClient1.NoRecordsMessage.Message = "There are no
  alarm records available";

Remarks

Use this property in combination with the NoRecordsMessage.Message property.

NoRecordsMessage.Message Property

The NoRecordsMessage.Message property is a read-write string property that gets or sets the custom message text when no alarm records are available and the **NoRecordsMessage.Enabled** property value is TRUE.

Syntax

result = AlarmClient.NoRecordsMessage.Message;

AlarmClient.NoRecordsMessage.Message = *myCustomMessage;*

Example

AlarmClient1.NoRecordsMessage.Enabled = 1;

AlarmClient1.NoRecordsMessage.Message = "There are no
  alarm records available";

Remarks

Use this property in combination with the NoRecordsMessage.Enabled property.

QueryStartup Property

The QueryStartup property is a read-write Boolean property that gets or sets or sets the automatic update of the Alarm Control on startup.

Syntax

result = AlarmClient.QueryStartup;

AlarmClient.QueryStartup = *AutoQry;*

Remarks

For more information, see Automatically Querying for Alarms on Start Up on page 53.

RetainHidden Property

The RetainHidden property is a read-write Boolean property that gets or sets the retention of hidden alarms or events when the alarm query or query filter to retrieve records changes at run time.

Syntax

result = AlarmClient.RetainHidden;

AlarmClient.RetainHidden = *RetainHddn;*

Remarks

For more information, see Retain Hiding when Changing Alarm Query Filter on page 56.

RowCount Property

The RowCount property is a read-only integer property that gets the number of records shown in the Alarm Control grid.

For current alarms (and recent alarms and events), the **RowCount** property value is always the same as the **TotalRowCount** property value.

For historical alarms, if the Alarm Control retrieves more alarm records than specified by the **MaxDatabaseRecords** property value, it splits these into multiple pages.

The **RowCount** property shows how many alarm records are currently shown on the current page. The RowCount property value is the same as the **MaxDatabaseRecords** property value, with exception of the last page.

Syntax

```
Result = AlarmClient.RowCount;
```

Example

```
NRows = AlarmClient1.RowCount;

LogMessage("There are " + Text(NRows, "#") + " alarm
  records on the retrieved page.");
```

RowSelection Property

The RowSelection property is a read-write string property that determines if row selection is allowed at run time. The following values are possible:

Value	Description
No	Operator cannot select rows.
Single	Operator can only select one row at a time.
Multiple	Operator can select one or more rows.

The default value is "Multiple".

Syntax

```
Result = AlarmClient.RowSelection;

AlarmClient.RowSelection = RwSel;
```

Example

```
AlarmClient1.RowSelection = "Multiple";
```

Remarks

For more information, see Restricting User Access to Rows and Columns on page 55.

SelectedCount Property

The SelectedCount property is a read-only integer property that gets the total number of selected alarm records.

Syntax

Result = AlarmClient.SelectedCount;

Return Value

Returns the number of selected alarm records.

Example

```
NSelRows = AlarmClient1.SelectedCount;

If NSelRows > 5 Then
    LogMessage("There are more than 5 rows selected.");

Endif;
```

ShowContextMenu Property

The ShowContextMenu property is a read-write Boolean property that gets or sets the ability to open the shortcut menu at run time.

Syntax

result = AlarmClient.ShowContextMenu;

AlarmClient.ShowContextMenu = *ContxtMnuAvail;*

ShowGrid Property

The ShowGrid property is a read-write Boolean property that gets or sets the appearance of grid lines.

Syntax

result = AlarmClient.ShowGrid;

AlarmClient.ShowGrid = *showGrid;*

ShowGroupByHeader Property

The ShowGroupByHeader property is a read-write Boolean property to show or hide the column grouping label at the top of the run-time Alarm Control in the historical mode. Set the ShowGroupByHeader property to true to show the label "Drag a column header here to group by that column".

Syntax

result = AlarmClient.ShowGroupByHeader;

AlarmClient.ShowGroupByHeader = *ShowGroupByHeader;*

ShowHeading Property

The ShowHeading property is a read-write Boolean property that gets or sets the visibility of the grid heading at run time.

Syntax

```
result = AlarmClient.ShowHeading;

AlarmClient.ShowHeading = showHeading;
```

ShowStatusBar Property

The ShowStatusBar property is a read-write Boolean property that gets or sets the visibility of the status bar at run time.

Syntax

```
result = AlarmClient.ShowStatusBar;

AlarmClient.ShowStatusBar = showStatusBar;
```

SortColumn.First Property

The SortColumn.First property is a read-write string property that gets or sets the first sort column.

The default value is "Time (LCT)".

Syntax

```
result = AlarmClient.SortColumn.First;

AlarmClient.SortColumn.First = sortByFirst;
```

Example

```
AlarmClient1.SortColumn.First = "Class";
```

Remarks

Use this property in connection with the SortOrder.First to determine the sorting direction.

SortColumn.Second Property

The SortColumn.Second property is a read-write string property that gets or sets the second sort column.

The default value is blank.

Syntax

```
result = AlarmClient.SortColumn.Second;

AlarmClient.SortColumn.Second = sortBySecond;
```

Example

```
AlarmClient1.SortColumn.Second = "Type";
```

Remarks

Use this property in connection with the SortOrder.Second to determine the sorting direction.

SortColumn.Third Property

The SortColumn.Third property is a read-write string property that gets or sets the third sort column.

The default value is blank.

Syntax

result = AlarmClient.SortColumn.Third;

AlarmClient.SortColumn.Third = *sortByThird*;

Example

AlarmClient1.SortColumn.Third = "State";

Remarks

Use this property in connection with the SortOrder.Third to determine the sorting direction.

SortOrder.First Property

The SortOrder.First property is a read-write Boolean property that gets or sets the sorting direction of the first sort column. The following values are possible:

Value	Description
FALSE	Ascending sorting direction
TRUE	Descending sorting direction

The default value is FALSE (Ascending).

Syntax

result = AlarmClient.SortOrder.First;

AlarmClient.SortOrder.First = *sortDirFirst*;

Remarks

Use this property in connection with the SortColumn.First to determine which column is sorted.

SortOrder.Second Property

The SortOrder.Second property is a read-write Boolean property that gets or sets the sorting direction of the second sort column. The following values are possible:

Value	Description
FALSE	Ascending sorting direction
TRUE	Descending sorting direction

The default value is FALSE (Ascending).

Syntax

result = AlarmClient.SortOrder.Second;

AlarmClient.SortOrder.Second = *sortDirSecond*;

Remarks

Use this property in connection with the SortColumn.Second to determine which column is sorted.

SortOrder.Third Property

The SortOrder.Third property is a read-write Boolean property that gets or sets the sorting direction of the third sort column. The following values are possible:

Value	Description
FALSE	Ascending sorting direction
TRUE	Descending sorting direction

The default value is FALSE (Ascending).

Syntax

result = AlarmClient.SortOrder.Third;

AlarmClient.SortOrder.Third = *sortDirThird*;

Remarks

Use this property in connection with the SortColumn.Third to determine which column is sorted.

Time.Format Property

The Time.Format property is a read-write string property that gets or sets the date and time formats of the alarm records in the Alarm Control.

You can either use the .NET time format or the Wonderware time format. Set the Time.Type property to determine which time format type to use.

Syntax

result = AlarmClient.Time.Format;

AlarmClient.Time.Format = *TmFormat*;

Example

This example shows the time format in French format (day/month/year) using the .NET datetime type.

```
AlarmClient1.Time.Type = 1;
AlarmClient1.Time.Format = "dd/MM/yyyy";
```

Remarks

For more information about the .NET time format, see Setting the .NET Datetime Format on page 51.

For more information about the Wonderware time format, see Setting the Wonderware Time Format on page 50.

Time.Type Property

The Time.Type property is a read-write Boolean property that gets or sets the time format type of the alarm records. The following values are possible:

Value	Description
FALSE	Wonderware time format
TRUE	.NET time format (default)

Syntax

result = AlarmClient.Time.Type;

AlarmClient.Time.Type = *TmType*;

Example

This example shows the time format in German format (day.month.year) using the Wonderware datetime type.

```
AlarmClient1.Time.Type = 0;
AlarmClient1.Time.Format = "%d.%m.%Y %H:%M:%S";
```

Remarks

For more information about the .NET time format, see Setting the .NET Datetime Format on page 51.

For more information about the Wonderware time format, see Setting the Wonderware Time Format on page 50.

TimeSelector Property

The TimeSelector property gets the Time Range Picker object used in the Alarm Control. You can use it in scripting to shorten the code using its properties and methods.

For the individual properties and methods, see the following properties, or the methods starting at TimeSelector.GetStartAndEndTimes() Method on page 142.

Example 1

```
dim TRP as object;

TRP = AlarmClient1.TimeSelector;

Timeselect = TRP;

StartDate = TRP.StartDate;

EndDate = TRP.EndDate;

duration = TRP.TimeDuration;
```

Example 2

```
dim TRP as object;

TRP = AlarmClient1.TimeSelector;

TRP.SetStartAndEndTimes(StartDate, EndDate, Duration);
```

TimeSelector.DurationMS Property

The TimeSelector.DurationMS property is a read-write integer property that gets the time duration measured in milliseconds.

The start time of the Alarm control (TimeSelector.StartDate) is calculated as the end time (TimeSelector.EndDate) minus the new time duration (TimeSelector.DurationMS).

When you set the value of the TimeSelector.DurationMS property, the TimeSelector.TimeDuration property is set to 0.

The default value is 3600000.

Syntax

```
result = AlarmClient.TimeSelector.DurationMS;

AlarmClient.TimeSelector.DurationMS = Value;
```

Example

```
AlarmClient1.TimeSelector.DurationMS = 1800000;

// The Alarm Control now retrieves alarms from the last
  30 minutes.
```

TimeSelector.EndDate Property

The TimeSelector.EndDate property is a read-only string property that gets the end date and time of the Alarm Control.

The default value is the time the Alarm Control is placed on the canvas. If the **Update to Current Time** option is enabled, the TimeSelector.EndDate property is updated with the current time.

Note To set the end date and time of the Alarm Control, use the TimeSelector.SetStartAndEndTimes() Method method.

Syntax

```
result = AlarmClient.TimeSelector.EndDate;
```

Example

```
LogMessage(AlarmClient1.TimeSelector.EndDate);
```

TimeSelector.StartDate Property

The TimeSelector.StartDate property is a read-only string property that gets the start date and time of the Alarm Control.

The default value is the time the Alarm Control is placed on the canvas. If the **Update to Current Time** option is enabled, the TimeSelector.StartDate property is updated as current time minus duration.

Note To set the start date and time of the Alarm Control, use the TimeSelector.SetStartAndEndTimes() Method method.

Syntax

```
result = AlarmClient.TimeSelector.StartDate;
```

Example

```
LogMessage(AlarmClient1.TimeSelector.StartDate);
```

TimeSelector.TimeDuration Property

The TimeSelector.TimeDuration property is a read-write integer property that gets or sets the time duration. The start time of the Alarm control (TimeSelector.StartDate) is calculated as the end time (TimeSelector.EndDate) minus the new time duration.

The TimeSelector.TimeDuration can have one of the following values:

Value	Description
0	Custom
1	The last minute.
2	The last five minutes.
3	The last ten minutes.
4	The last 15 minutes.
5	The last 30 minutes.
6	The last hour.
7	The last two hours.
8	The last four hours.
9	The last eight hours.
10	The last 12 hours.
11	The last 24 hours.
12	The last two days.
13	The last week.
14	The last two weeks.
15	The last month.
16	The last three months.
17	One minute.
18	Five minutes.
19	Ten minutes.
20	15 minutes.
21	30 minutes.
22	One hour.
23	Two hours.

Value	Description
24	Four hours.
25	Eight hours.
26	12 hours.
27	24 hours.
28	Two days.
29	One week.
30	Two weeks.
31	One month.
32	Three months.
33	Yesterday: 0:00:00 of the previous day to 0:00:00 of the current day.
34	Current day: 0:00:00 of the current day to the current time.
35	Previous hour: The start of the previous hour to the start of the current hour.
36	Current hour: The start of the current hour to the current time.

The default value is 6 (Last Hour).

Syntax

result = *AlarmClient*.TimeSelector.TimeDuration;

AlarmClient.TimeSelector.TimeDuration = *Value*;

Example

AlarmClient1.TimeSelector.TimeDuration = 5;

// The Alarm Control now retrieves alarms from the last
  30 minutes.

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

TimeZone.TimeZone Property

The TimeZone.TimeZone property is a read-write string property that gets or sets the time zone of the Alarm Control.

The default value depends on the current setting of the operating system.

If you want to show time stamps using the local time of the computer, set the TimeZone.TimeZone property to an empty string.

Syntax

```
result = AlarmClient.TimeZone.TimeZone;

AlarmClient.TimeZone.TimeZone = TimeZone;
```

Example

```
AlarmClient1.TimeZone.TimeZone = "(GMT-09:00) Alaska";
```

Remarks

For more information, see Setting Time Zone and Format on page 48.

TotalRowCount Property

The TotalRowCount property is a read-only integer property that gets the total number of alarm records in the Alarm Control.

For current alarms (and recent alarms and events), the **RowCount** property value is always the same as the **TotalRowCount** property value.

For historical alarms, if the Alarm Control retrieves more alarm records than specified by the **MaxDatabaseRecords** property value, it splits these into multiple pages.

The **RowCount** property value shows how many alarm records are currently shown on the current page, whereas the **TotalRowCount** property value shows how many alarm records are retrieved from the alarm database.

Syntax

```
Result = AlarmClient.TotalRowCount;
```

Return Value

Returns the end date and time of the Alarm Control in historical mode.

Example

```
NTRows = AlarmClient1.TotalRowCount;

If (NTRows > 1000) then
    LogMessage("More than 1000 records are currently in
    the Alarm Control");

Endif;
```

UnAckAlarms Property

The UnAckAlarms property is a read-only integer property that gets the number of unacknowledged alarm records in the Alarm Control.

Syntax

Result = AlarmClient.UnackAlarms;

Return Value

Returns the number of unacknowledged alarm records in the Alarm Control.

Example

```
NUnack = AlarmClient1.UnAckAlarms;

If NUnack > 10 Then
    LogMessage("There are more than 10 unacknowledged
    alarms in the grid!");

Endif;
```

UpdateToCurrentTime Property

The UpdateToCurrentTime property is a read-write Boolean property that gets or sets the **Update to Current Time** option.

If you set this property to TRUE, the Alarm Control end time is set to the current time and the start time is calculated as the difference of end time and duration. Whenever you refresh the Alarm Control, the end time is set as current time.

If you set this property to FALSE, the Alarm Control uses the end time, duration, and start time as defined by the Time Range Picker control.

The default value is TRUE.

Syntax

result = AlarmClient.UpdateToCurrentTime;

AlarmClient.UpdateToCurrentTime = *UpdToCurrTime*;

Example

```
AlarmClient1.UpdateToCurrentTime = 1;

AlarmClient1.Requery();
```

Remarks

For more information, see Showing Historical Alarms and/or Events on page 28.

Visible Property

The Visible property is a read-write Boolean property that gets or sets the visibility of the Alarm Control.

Syntax

```
result = AlarmClient.Visible;
AlarmClient.Visible = Boolean;
```

Width Property

The Width property is a read-write integer property that gets or sets the width of the Alarm Control in pixels.

Syntax

```
result = AlarmClient.Width;
AlarmClient.Width = Wdth;
```

WindowColor Property

The WindowColor property is a read-write color property that gets or sets the color of the Alarm Control background.

Syntax

```
Color = AlarmClient.WindowColor;
AlarmClient.WindowColor = Color;
```

Parameters

Color
  Color of background.

Example

```
AlarmClient1.WindowColor = Color.FromARGB(240,200,198);
```

Remarks

For more information, see Setting Heading, Grid, and Window Color on page 32.

Color is a .NET Framework data type. You can use various Color methods to set the color, such as a predefined color name, FromARGB(), FromKnownColor(), and FromName().

For a list of the .NET color names and the hexadecimal codes, see .NET Colors on page 147.

For more information on the color methods, see the online Microsoft documentation for .NET Framework Development.

X Property

The X property is a read-write integer property that gets or sets the horizontal position of the Alarm Control in relation to the left edge of the InTouch window in which it appears.

Syntax

```
result = AlarmClient.X;

AlarmClient.X = LeftPos;
```

Y Property

The Y property is a read-write integer property that gets or sets the vertical position of the Alarm Control in relation to the top edge of the InTouch window in which it appears.

Syntax

```
result = AlarmClient.Y;

AlarmClient.Y = TopPos;
```

Alarm Control Methods

This section describes the methods available for scripting in the Alarm Control.

AboutBox() Method

The AboutBox method shows the **About** dialog box of the Alarm Control.

Syntax

```
AlarmClient.AboutBox();
```

Ack.All() Method

The Ack.All method acknowledges all alarms in the Alarm Control, including those not shown.

Syntax

```
AlarmClient.Ack.All(AckComment);
```

Parameters

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.All("Alarm is acknowledged");
```

Ack.Group() Method

The Ack.Group method acknowledges all alarms for a given alarm source and group.

The alarm source and group names are case-insensitive.

Syntax

```
AlarmClient.Ack.Group(AlarmSource, Group, AckComment);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.Group("\\machine1\galaxy", "Area_001",
  "All alarms in Area_001 acknowledged");
```

Ack.Priority() Method

The Ack.Priority method acknowledges all alarms for a given alarm source, group, and priority range.

The alarm source and group names are case-insensitive.

Syntax

```
AlarmClient.Ack.Priority(AlarmSource, Group,
  FromPriority, ToPriority, AckComment);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

FromPriority
  Starting priority of alarms. For example, 100.

ToPriority
  End priority of alarms. For example, 900.

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
GrpName = "ValveGroup";

AlarmClient1.Ack.Priority("\intouch", GrpName, 250,
    500, "All local InTouch alarms in the ValveGroup
    alarm group with priorities from 250 to 500 are now
    acknowledged.");
```

Ack.Selected() Method

The Ack.Selected method acknowledges all selected alarms.

Syntax

```
AlarmClient.Ack.Selected(AckComment);
```

Parameters

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.Selected("This selected alarm is
    acknowledged");
```

Ack.SelectedGroup() Method

The Ack.SelectedGroup method acknowledges all alarms that have the same alarm sources and groups as one or more selected alarms.

Syntax

```
AlarmClient.Ack.SelectedGroup(AckComment);
```

Parameters

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.SelectedGroup("Alarm acknowledged");
```

Ack.SelectedPriority () Method

The Ack.SelectedPriority method acknowledges all alarms that have the same alarm sources, groups, and within the priority ranges as one or more selected alarms.

Syntax

```
AlarmClient.Ack.SelectedPriority(AckComment);
```

Parameters

AckComment
   A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.SelectedPriority("Alarm
   acknowledged");
```

Ack.SelectedTag() Method

The Ack.SelectedTag method acknowledges all alarms that have the same alarm sources, groups, tags, and within the priority ranges as one or more selected alarms.

Syntax

```
AlarmClient.Ack.SelectedTag(AckComment);
```

Parameters

AckComment
   A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.SelectedTag("Alarm acknowledged");
```

Ack.Tag() Method

The Ack.Tag method acknowledges all alarms for a given alarm source, group, tag name, and priority range.

The alarm source, group names, and tag names are case-insensitive.

Syntax

```
AlarmClient.Ack.Tag(AlarmSource, Group, Tag,
   FromPriority, ToPriority, AckComment);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

Tag
  The name of the alarm tag. For example, ValveTag1.

FromPriority
  Starting priority of alarms. For example, 100.

ToPriority
  End priority of alarms. For example, 900.

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
AckComment = "All ArchestrA alarm records of the
  attribute Valve17 in the group (area) Vessel_25B of
  the galaxy on machine25 with priorities from 1 to 99
  are now acknowledged.";

AlarmClient1.Ack.Tag("\\machine25\galaxy",
  "Vessel_25B", "Valve17", 1, 99, AckComment);
```

Ack.Visible() Method

The Ack.Visible method acknowledges all alarms currently visible in the Alarm Control.

Syntax

```
AlarmClient.Ack.Visible(AckComment);
```

Parameters

AckComment
  A string indicating the alarm acknowledgement comment.

Example

```
AlarmClient1.Ack.Visible("Alarm acknowledged");
```

Connect() Method

The Connect method connects the Alarm Control to the Alarm Database.

Syntax

```
AlarmClient.Connect();
```

Disconnect() Method

The Disconnect method disconnects the Alarm Control from the Alarm Database.

Syntax

```
AlarmClient.Disconnect();
```

Favorites.Export() Method

The Favorites.Emport method exports the list of query filter favorites list to an XML file.

Syntax

```
AlarmClient.Favorites.Export(FilePath, FileName);
```

Parameters

FilePath
  Name of the path to export the query filter favorites file.

FileName
  Name of the query filter favorites file to export.

Example

```
AlarmClient1.Favorites.Export("c:\",
  "MyFavorites.xml");
```

Favorites.Import() Method

The Favorites.Import method imports the list of query filter favorites list from an XML file. You can either overwrite the exisiting query filter favorites with the new favorites, or append them.

Syntax

```
AlarmClient.Favorites.Import(FilePath, FileName,
  OverwriteAppend);
```

Parameters

FilePath
Name of the path to the query filter favorites file to import.

FileName
Name of the query filter favorites file to import.

OverwriteAppend
String determining if the import of the query filter favorites overwrites existing favorites, or appends to existing favorites. Set to one of the following:

- Overwrite to overwrite existing query filter favorites.

- Append to append to existing query filter favorites.If a query filter with the same name already exists, it is not overwritten by the query filter in the file.

Example

```
AlarmClient1.Favorites.Import("c:\MyFavs\", "Favs.xml",
  "Overwrite");
```

FreezeDisplay() Method

The FreezeDisplay method freezes or unfreezes the Alarm Control. The following values are possible:

Value	Description
TRUE	Freezes the Alarm Control.
FALSE	Unfreezes the Alarm Control.

Syntax

```
AlarmClient.FreezeDisplay(FreezeFlag);
```

Parameters

FreezeFlag
Boolean value or expression (TRUE = freeze control, FALSE = unfreeze control)

Example

```
AlarmClient1.FreezeDisplay($hour > 17 OR $hour<9 );
```

```
LogMessage("The Alarm Control is frozen between 6 PM
  and 8 AM.");
```

GetItem() Method

The GetItem method returns the data at the given row and column. The row is given as a zero-based index. You need to specify 0 to retrieve data from the 1st row. The column name can either be the original column name, or the displayed column name.

Syntax

Result = AlarmClient.GetItem(*RowNumber, ColumnName*);

Parameters

RowNumber
An integer row number for the alarm record containing the value you want to fetch.

ColumnName
Name of the column.

Return Value

Returns the data at the given row and column as a string value.

Example

```
Data1 = AlarmClient1.GetItem(5, "Current Value");

LogMessage("The current value of the 6th alarm record
  is " + Data1);
```

Remarks

To get alarm record data from the currently selected row in a given column name, use the **GetSelectedItem** method.

GetLastError() Method

The GetLastError method returns the last error message. This is useful if the Hide Errors option is selected.

Syntax

ErrMsg = AlarmClient.GetLastError();

Return Value

Returns the last error message.

Example

```
ErrMsg = AlarmClient1.GetLastError();

ComboBox1.AddItem(ErrMsg);
```

GetSelectedItem() Method

The GetSelectedItem method returns the data at the currently selected row and specified column. The column name can either be the original column name, or the displayed column name.

Syntax

Result = AlarmClient.GetSelectedItem(*ColumnName*);

Parameters

ColumnName
  Name of the column.

Return Value

Returns the data in the currently selected row and specified column as a string value.

Example

```
Data2 = AlarmClient1.GetSelectedItem ("State");

LogMessage("The current state of the selected alarm
  record is " + Data2);
```

Remarks

To get alarm record data from a given column name and row index, use the **GetItem** method.

Hide.All() Method

The Hide.All method hides all current alarms in the Alarm Control, including future alarms.

Syntax

AlarmClient.Hide.All();

Hide.Group() Method

The Hide.Group method hides all alarms for a given alarm source and group.

The alarm source and group names are case-insensitive.

Syntax

```
AlarmClient.Hide.Group(AlarmSource, Group);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

Example

```
AlarmClient1.Hide.Group("\\machine1\galaxy",
  "Area_001");

LogMessage("All alarms in Area_001 hidden.");
```

Hide.Priority() Method

The Hide.Priority method hides all alarms for a given alarm source, group, and priority range.

The alarm source and group names are case-insensitive.

Syntax

```
AlarmClient.Hide.Priority(AlarmSource, Group,
  FromPriority, ToPriority);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

FromPriority
  Starting priority of alarms. For example, 100.

ToPriority
  End priority of alarms. For example, 900.

Example

```
GrpName = "ValveGroup";

AlarmClient1.Hide.Priority("\intouch", GrpName, 250,
    500);

LogMessage("All local InTouch alarms in the ValveGroup
    alarm group with priorities from 250 to 500 are now
    hidden.");
```

Hide.Selected() Method

The Hide.Selected method hides all selected alarms.

Syntax

```
AlarmClient.Hide.Selected();
```

Hide.SelectedGroup() Method

The Hide.SelectedGroup method hides all alarms that have the same alarm sources and groups as one or more selected alarms.

Syntax

```
AlarmClient.Hide.SelectedGroup();
```

Hide.SelectedPriority() Method

The Hide.SelectedPriority method hides all alarms that have the same alarm sources, groups, and within the priority ranges as one or more selected alarms.

Syntax

```
AlarmClient.Hide.SelectedPriority();
```

Hide.SelectedTag() Method

The Hide.SelectedTag method hides all alarms that have the same alarm sources, groups, tag names, and within the priority ranges as one or more selected alarms.

Syntax

```
AlarmClient.Hide.SelectedTag();
```

Remarks
None

Hide.Tag() Method

The Hide.Tag method hides all alarms for a given alarm source, group, tag name, and priority range.

The alarm source, group name, and tag names are case-insensitive.

Syntax

```
AlarmClient.Hide.Tag(AlarmSource, Group, Tag,
  FromPriority, ToPriority);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

Tag
  The name of the alarm tag. For example, ValveTag1.

FromPriority
  Starting priority of alarms. For example, 100.

ToPriority
  End priority of alarms. For example, 900.

Example

```
AlarmClient1.Hide.Tag("\\machine25\galaxy",
  "Vessel_25B", "Valve17", 1, 99);

LogMessage("All ArchestrA alarm records of the
  attribute Valve17 in the group (area) Vessel_25B of
  the galaxy on machine25 with priorities from 1 to 99
  are now hidden.");
```

Hide.Visible() Method

The Hide.Visible method hides all alarms currently visible in the Alarm Control.

Syntax

```
AlarmClient.Hide.Visible();
```

MoveWindow() Method

The MoveWindow method scrolls the alarm records in the control in a given direction.

Syntax

AlarmClient.MoveWindow(*ScrollDir*, *Repeat*);

Parameters

ScrollDir

String indicating the direction to scroll. This parameter is case-insensitive. See the following table.

ScrollDir	Description
LineDn	Line down. The Repeat parameter controls the number of lines to be scrolled.
LineUp	Line up. The Repeat parameter controls the number of lines to be scrolled.
PageDn	Page down. The Repeat parameter controls the number of pages to be scrolled.
PageUp	Page up. The Repeat parameter controls the number of pages to be scrolled.
Top	To the top of the control
Bottom	To the bottom of the control.
PageRt	Page to the right. The Repeat parameter controls the number of pages to be scrolled.
PageLf	Page to the left. The Repeat parameter controls the number of pages to be scrolled.
Right	Scrolls right. The Repeat parameter controls the number of columns to be scrolled.
Left	Scrolls left. The Repeat parameter controls the number of columns to be scrolled.
Home	Scrolls to the top row and left most column of the control.

Repeat

Number of times to repeat the scroll action.

Example

```
AlarmClient1.MoveWindow ("Bottom", 0);
```

Requery() Method

The Requery method refreshes the alarm records in the Alarm Control.

For current alarms and recent alarms and events, the control requeries the Alarm Manager. For historical alarms or events, the control retrieves alarm records from the Alarm Database.

Syntax

```
AlarmClient.Requery();
```

Reset() Method

The Reset method resets column widths and the column order to their last known design-time settings. The Reset method also resets the current query filter to the default query.

Syntax

```
AlarmClient.Reset();
```

Select.All() Method

The Select.All method selects all alarms in the Alarm Control.

Syntax

```
AlarmClient.Select.All();
```

Select.Group() Method

The Select.Group method selects all alarms for a given provider and group.

Syntax

```
AlarmClient.Select.Group(AlarmSource, Group);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

Example

```
AlarmClient1.Select.Group("\\machine1\galaxy",
  "Area_001");
```

```
LogMessage("All galaxy alarms of group Area_001 from
  machine1 are now selected.");
```

Select.Item() Method

The Select.Item method selects an alarm record at a given zero-based row number.

Syntax

AlarmClient.Select.Item(*RowNumber*);

Parameters

RowNumber
An integer row number for the alarm record to select. The first row in the control is 0.

Example

```
AlarmClient1.Select.Item(5);
```

```
LogMessage("The alarm record in the 6th row (index 5)
  is now selected.");
```

Select.Priority() Method

The Select.Priority method selects all alarms for a given alarm source, group, and priority range.

Syntax

AlarmClient.Select.Priority(*AlarmSource*, *Group*,
  FromPriority, *ToPriority*);

Parameters

AlarmSource
The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
The name of the alarm group. For example, $system.

FromPriority
Starting priority of alarms. For example, 100.

ToPriority
End priority of alarms. For example, 900.

Example

```
GrpName = "ValveGroup";

AlarmClient1.Select.Priority("\intouch", GrpName, 250,
   500);

LogMessage("All local InTouch alarms in the ValveGroup
   alarm group with priorities from 250 to 500 are now
   selected.");
```

Select.Tag() Method

The Select.Tag method selects all alarms for a given alarm source, group, tag name, and priority range.

Syntax

```
AlarmClient.Select.Tag(AlarmSource, Group, Tag,
   FromPriority, ToPriority);
```

Parameters

AlarmSource
  The name of the provider and optionally node providing alarms including backslash. For example:

```
\\node1\galaxy
\intouch
```

Group
  The name of the alarm group. For example, $system.

Tag
  The name of the alarm tag. For example, ValveTag1.

FromPriority
  Starting priority of alarms. For example, 100.

ToPriority
  End priority of alarms. For example, 900.

Example

```
AlarmClient1.Select.Tag("\\machine25\galaxy",
   "Vessel_25B", "Valve17", 1, 99);

LogMessage("All ArchestrA alarm records of the
   attribute Valve17 in the group (area) Vessel_25B of
   the galaxy on machine25 with priorities from 1 to 99
   are now selected.");
```

SetSort() Method

The SetSort method sets the level of sorting according to the defined sort columns and sort orders.

Syntax

```
AlarmClient.SetSort(Level);
```

Parameters

Level

The level of sorting:

Value	Description
1	Only use the primary sort column.
2	Use primary and secondary sort columns.
3	Use primary, secondary, and tertiary sort columns.

Example

```
AlarmClient1.SetSort(2);
```

Remarks

Use the **Show.Sort** method to open the **Sort** dialog box instead.

Show.Context() Method

The Show.Context method opens the shortcut menu at run time. This method ignores the ShowContextMenu property setting and always shows the context menu.

Syntax

```
AlarmClient.Show.Context();
```

Show.Favorite() Method

The Show.Favorite method opens the **Query Filters** dialog box.

Syntax

```
AlarmClient.Show.Favorite();
```

Show.Hidden() Method

The Show.Hidden method opens the **Hidden Alarms** dialog box.

Syntax

```
AlarmClient.Show.Hidden();
```

Show.Sort() Method

The Show.Sort method opens the **Sort** dialog box.

Syntax

```
AlarmClient.Show.Sort();
```

Show.Statistics() Method

The Show.Statistics method opens the **Alarm Statistics** dialog box.

Syntax

```
AlarmClient.Show.Statistics();
```

TimeSelector.GetStartAndEndTimes() Method

The TimeSelector.GetStartAndEndTimes method gets the start and end times for the query.

Syntax

```
AlarmClient.GetStartAndEndTimes(StartTime, EndTime);
```

Parameters

StartTime
  String attribute, custom property, or element property to retrieve the start time.

EndTime
  String attribute, custom property, or element property to retrieve the end time.

Example

```
dim SDate as string;

dim EDate as string;

AlarmClient1.TimeSelector.GetStartAndEndTimes(SDate,
  EDate);

StartDate = SDate;

EndDate = EDate;
```

TimeSelector.RefreshTimes() Method

The TimeSelector.RefreshTimes method sets the time period for the query by updating the end time to current time and recalculates the start time based on the new end time and duration.

If you set the Boolean parameter to TRUE, the OnChange event is triggered if the time is updated.

Only use this method, if the **Update to Current Time** option is cleared or the **UpdateToCurrentTime** property is FALSE.

Note This method does not work if the **UpdatetoCurrentTime** property value is TRUE.

Syntax

AlarmClient.TimeSelector.RefreshTimes(*TriggerEvent*);

Example

```
dtag = 1;
AlarmClient.TimeSelector.RefreshTimes(dtag);
```

TimeSelector.SetStartAndEndTimes() Method

The TimeSelector.SetStartAndEndTimes method sets the start and end times for the query.

You must specify one of the following parameter combinations:

- Start time and end time. Set the Duration parameter to 0.

- Start time and duration. Set the EndTime parameter to "".

- End time and duration. Set the StartTime parameter to "".

- Start time, duration, and end time. The Alarm Control shows an error message if start time plus duration is not equal to end time.

Syntax

AlarmClient.SetStartAndEndTimes(*StartTime*, *EndTime*, *Duration*);

Parameters

StartTime
  String value or expression indicating the start time.

EndTime
  String value or expression indicating the end time.

Duration
  Duration enum. For more information on possible values, see TimeSelector.TimeDuration Property on page 120.

Example

```
AlarmClient1.TimeSelector.SetStartAndEndTimes("08/31/2
  008 15:33:43","09/01/2009 15:33:43",0);
```

Toggle.All() Method

The Toggle.All method reverses the selection of all alarm records. Selected alarms are cleared, and unselected alarms are selected.

Syntax

```
AlarmClient.Toggle.All();
```

Toggle.Item() Method

The Toggle.Item method reverses the selection of a given alarm record. If the given alarm record is selected, the selection is cleared; otherwise, it is selected.

Syntax

```
AlarmClient.Toggle.Item(RowNumber);
```

Parameters

RowNumber
  An integer row number for the alarm record to reverse the selection. The first row in the control is 0.

Example

```
AlarmClient1.Toggle.Item(5);
```

```
LogMessage("The selection of the alarm record in the
  6th row (index 5) is now reversed.");
```

UnhideAll() Method

The UnhideAll method unhides all hidden alarms.

Syntax

AlarmClient.UnhideAll();

UnSelectAll() Method

The UnSelectAll method unselects all alarm records.

Syntax

AlarmClient.UnSelectAll();

Configuring Events

You can execute an action script when the Alarm Control triggers an event. Examples of basic events are:

- Click: The user clicks the Alarm Control.

- DoubleClick: The user double-clicks the Alarm Control.

- Startup: The Alarm Control opens at run time.

- Shutdown: The Alarm Control closes at run time.

The Click, DoubleClick, Startup, and Shutdown events are standard for all .NET client controls. For more information, see the *Creating and Managing ArchestrA Graphics Users Guide*.

The Alarm Control has one event of its own that is triggered when a new alarm occurs, the NewAlarm event.

Configuring the NewAlarm Event

You can configure the NewAlarm event to execute an ArchestrA symbol script whenever a new alarm occurs.

You can control the trigger behavior with the NewAlarmEventMode property. For more information, see NewAlarmEventMode Property on page 110.

To configure the NewAlarm event

1   Double-click the Alarm Control. The **Edit Animations** dialog box appears.

2   Click **Event**. The **Event** page appears.

3 In the **Event** list, click **NewAlarm**.
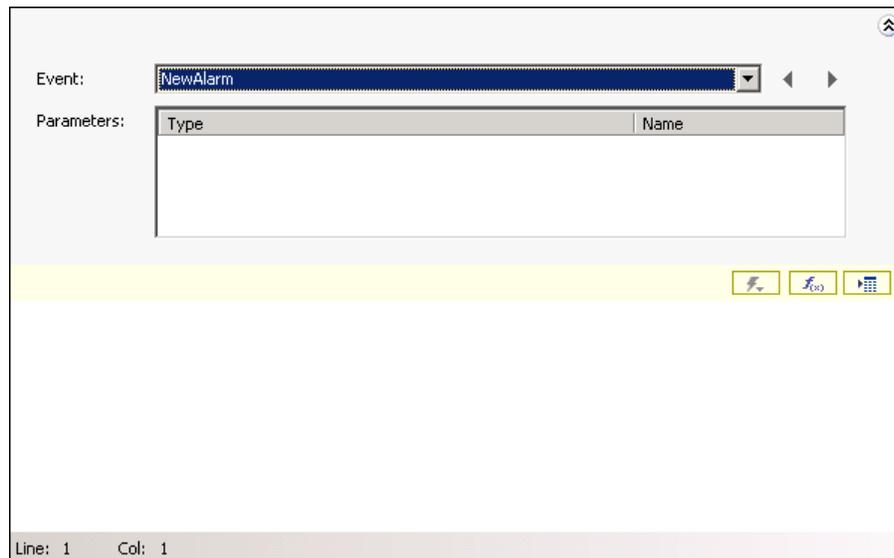
4 In the script area, type the script you want to execute when a new alarm occurs, for example:

```
AlertIcon.Visible = true;
```

5 You must also set the NewAlarmEventMode property to 1 or 2 to enable the NewAlarm event trigger. Do the following:

a On the **Special** menu, click **Scripts**. The **Edit Scripts** dialog box appears.

b Make sure **Trigger type** is set to **On Show**.

c In the script area, type the following:

```
AlarmClient1.NewAlarmEventMode = 1;
```

d If you want the script to be executed every time a new alarm occurs, set the NewAlarmEventMode property to 2 instead.

e Click **OK**.

.NET Colors

The following table is an overview of the color .NET color names with hexadecimal code.

Color with Hex Code	Color with Hex Code	Color with Hex Code
AliceBlue #F0F8FF	AntiqueWhite #FAEBD7	Aqua #00FFFF
Aquamarine #7FFFD4	Azure #F0FFFF	Beige #F5F5DC
Bisque #FFE4C4	Black #000000	BlanchedAlmond #FFEBCD
Blue #0000FF	BlueViolet #8A2BE2	Brown #A52A2A
BurlyWood #DEB887	CadetBlue #5F9EA0	Chartreuse #7FFF00
Chocolate #D2691E	Coral #FF7F50	CornflowerBlue #6495ED
Cornsilk #FFF8DC	Crimson #DC143C	Cyan #00FFFF
DarkBlue #00008B	DarkCyan #008B8B	DarkGoldenrod #B8860B
DarkGray #A9A9A9	DarkGreen #006400	DarkKhaki #BDB76B
DarkMagenta #8B008B	DarkOliveGreen #556B2F	DarkOrange #FF8C00
DarkOrchid #9932CC	DarkRed #8B0000	DarkSalmon #E9967A
DarkSeaGreen #8FBC8B	DarkSlateBlue #483D8B	DarkSlateGray #2F4F4F
DarkTurquoise #00CED1	DarkViolet #9400D3	DeepPink #FF1493
DeepSkyBlue #00BFFF	DimGray #696969	DodgerBlue #1E90FF
Firebrick #B22222	FloralWhite #FFFAF0	ForestGreen #228B22
Fuchsia #FF00FF	Gainsboro #DCDCDC	GhostWhite #F8F8FF
Gold #FFD700	Goldenrod #DAA520	Gray #808080
Green #008000	GreenYellow #ADFF2F	Honeydew #F0FFF0
HotPink #FF69B4	IndianRed #CD5C5C	Indigo #4B0082
Ivory #FFFFF0	Khaki #F0E68C	Lavender #E6E6FA
LavenderBlush #FFF0F5	LawnGreen #7CFC00	LemonChiffon #FFFACD
LightBlue #ADD8E6	LightCoral #F08080	LightCyan #E0FFFF
LightGoldenrodYellow #FAFAD2	LightGray #D3D3D3	LightGreen #90EE90
LightPink #FFB6C1	LightSalmon #FFA07A	LightSeaGreen #20B2AA
LightSkyBlue #87CEFA	LightSlateGray #778899	LightSteelBlue #B0C4DE
LightYellow #FFFFE0	Lime #00FF00	LimeGreen #32CD32

Color with Hex Code	Color with Hex Code	Color with Hex Code
Linen #FAF0E6	Magenta #FF00FF	Maroon #800000
MediumAquamarine #66CDAA	MediumBlue #0000CD	MediumOrchid #BA55D3
MediumPurple #9370DB	MediumSeaGreen #3CB371	MediumSlateBlue #7B68EE
MediumSpringGreen #00FA9A	MediumTurquoise #48D1CC	MediumVioletRed #C71585
MidnightBlue #191970	MintCream #F5FFFA	MistyRose #FFE4E1
Moccasin #FFE4B5	NavajoWhite #FFDEAD	Navy #000080
OldLace #FDF5E6	Olive #808000	OliveDrab #6B8E23
Orange #FFA500	OrangeRed #FF4500	Orchid #DA70D6
PaleGoldenrod #EEE8AA	PaleGreen #98FB98	PaleTurquoise #AFEEEE
PaleVioletRed #DB7093	PapayaWhip #FFEFD5	PeachPuff #FFDAB9
Peru #CD853F	Pink #FFC0CB	Plum #DDA0DD
PowderBlue #B0E0E6	Purple #800080	Red #FF0000
RosyBrown #BC8F8F	RoyalBlue #4169E1	SaddleBrown #8B4513
Salmon #FA8072	SandyBrown #F4A460	SeaGreen #2E8B57
SeaShell #FFF5EE	Sienna #A0522D	Silver #C0C0C0
SkyBlue #87CEEB	SlateBlue #6A5ACD	SlateGray #708090
Snow #FFFAFA	SpringGreen #00FF7F	SteelBlue #4682B4
Tan #D2B48C	Teal #008080	Thistle #D8BFD8
Tomato #FF6347	Transparent #FFFFFF	Turquoise #40E0D0
Violet #EE82EE	Wheat #F5DEB3	White #FFFFFF
WhiteSmoke #F5F5F5	Yellow #FFFF00	YellowGreen #9ACD32

Chapter 5

Transferring Alarm Configuration from InTouch

You can transfer the configuration of the InTouch Alarm Viewer control and the InTouch Alarm DB View control to the configuration of the ArchestrA Alarm Control.

You can also map the InTouch alarm control properties and methods to the properties and methods of the ArchestrA Alarm Control.

Transferring the InTouch Alarm Viewer Control Configuration

You can transfer the configuration of the InTouch Alarm Viewer control tabs options to the ArchestrA Alarm Control.

Transferring Configuration of the Control Name Tab

You can transfer the configuration of the **Control Name** tab options of the InTouch Alarm Viewer control to the ArchestrA Alarm Control.
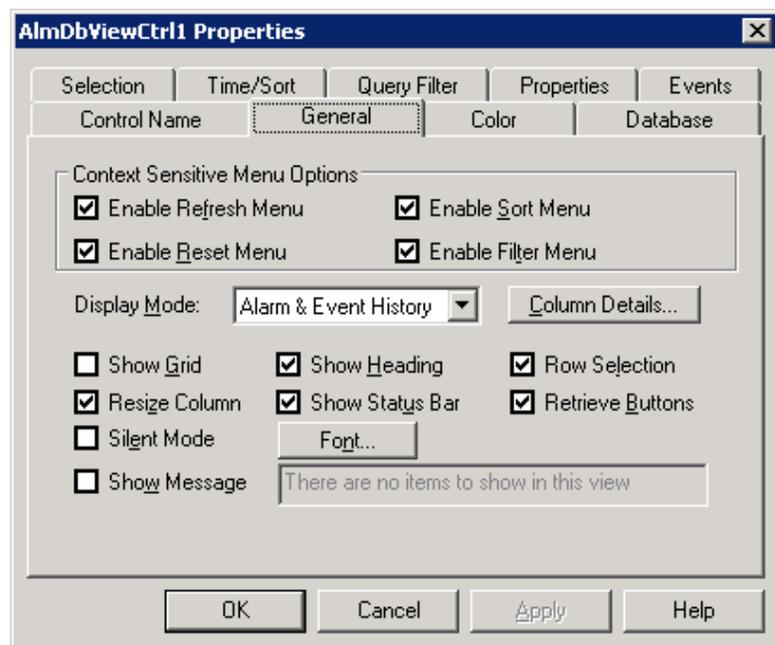

InTouch option	Alarm Control option
ControlName	You can rename the ArchestrA Alarm Control the same way as any other elements on the canvas. For more information, see the *Creating and Managing ArchestrA Graphics User's Guide*.
Left, Top, Width, and Height	You can directly edit the positioning options in the same way as any other element on the canvas. Edit the following properties in the Properties Editor: **X**, **Y**, **Width**, and **Height**.
Visible	You can directly edit the visibility option in the same way as any other element on the canvas. In the Properties Editor, edit the **Visible** property.
GUID	This option has no meaning in the ArchestrA Alarm Control.

Transferring Configuration of the General Tab

You can transfer the configuration of the **General** tab options of the InTouch Alarm Viewer control to the ArchestrA Alarm Control.
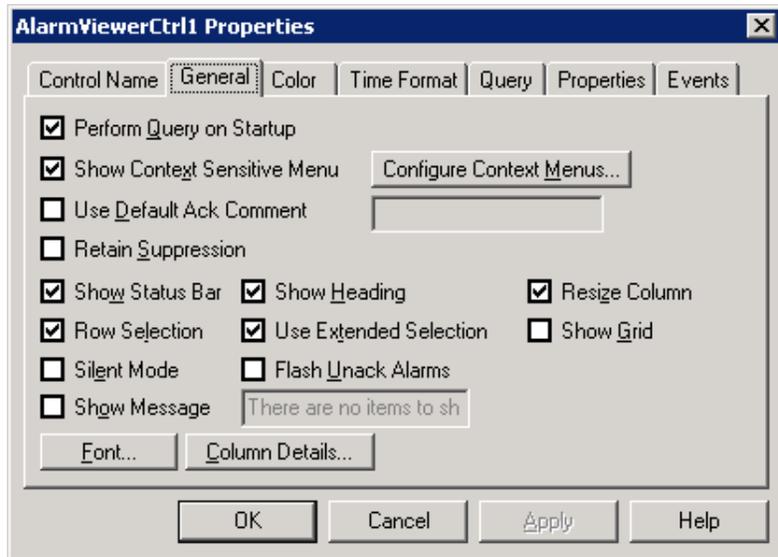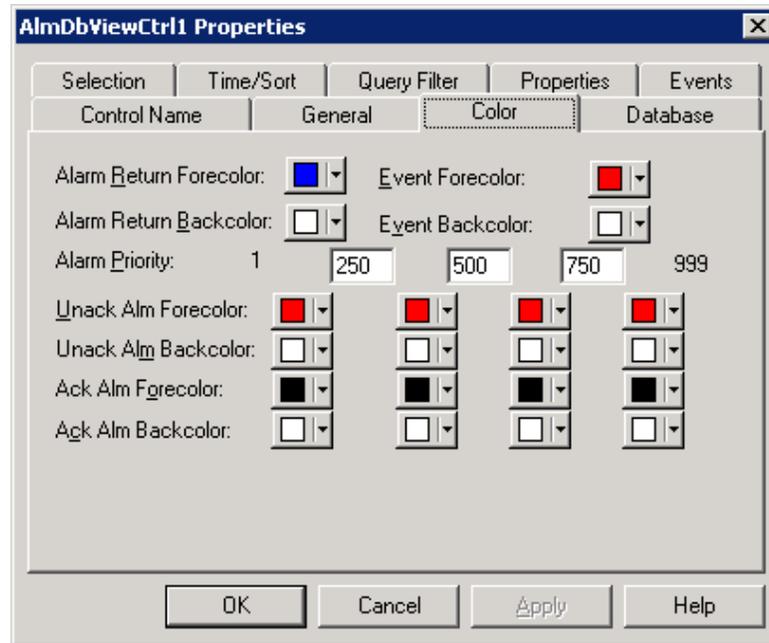

InTouch option	Alarm Control option
Perform Query on Startup	In the ArchestrA Alarm Control, this option is called **Query on Startup**. You can configure this option on the **Run-Time Behavior** page.
Show Context Sensitive Menu	In the ArchestrA Alarm Control, this option is called **Show Context Menu**. You can configure this option on the **Run-Time Behavior** page.
Configure Context Menus	In the ArchestrA Alarm Control, you can configure the availability of individual shortcut menu options at run-time directly on the **Run-Time Behavior** page.
Use Default Ack Comment	In the ArchestrA Alarm Control, you can configure the **Use Default Ack Comment** option on the **Alarm Mode** page, when either Current Alarms or Recent Alarms and Events is selected as client type.
Retain Suppression	In the ArchestrA Alarm Control, this option is called **Retain Hidden**. You can configure it on the **Run-Time Behavior** page.

InTouch option	Alarm Control option
Show Status Bar	In the ArchestrA Alarm Control, you can configure the **Show Status Bar** option on the **Run-Time Behavior** page.
Show Heading	In the ArchestrA Alarm Control, you can configure the **Show Heading** option on the **Run-Time Behavior** page.
Resize Column	In the ArchestrA Alarm Control, this option is called **Allow Column Resizing**. You can configure it on the **Run-Time Behavior** page.
Row Selection	In the ArchestrA Alarm Control, this option is called **Row Selection**. You can configure it on the **Run-Time Behavior** page.
Use Extended Selection	In the ArchestrA Alarm Control, this option is called **Row Selection**. You can configure it on the **Run-Time Behavior** page.
Show Grid	In the ArchestrA Alarm Control, you can configure the **Show Grid** option on the **Run-Time Behavior** page.
Silent Mode	In the ArchestrA Alarm Control, this option is called **Hide Errors and Warnings**. You can configure it on the **Run-Time Behavior** page.
Flash Unack Alarms	In the ArchestrA Alarm Control, you can configure the **Flash Unack Alarms** option on the **Colors** page.
Show Message	In the ArchestrA Alarm Control, this option is called **Show Custom 'No Records' Message**. You can configure it on the **Run-Time Behavior** page.
Font	You can configure this option from the ArchestrA Symbol Editor page. Select the ArchestrA Alarm Control on the canvas and select an appropriate font type, size, and style on the menu bars.
Column Details	In the ArchestrA Alarm Control, you can configure the column details directly on the **Column Details** page.

Transferring Configuration of the Color Tab

You can transfer the configuration of the **Color** tab options of the InTouch Alarm Viewer control to the ArchestrA Alarm Control.
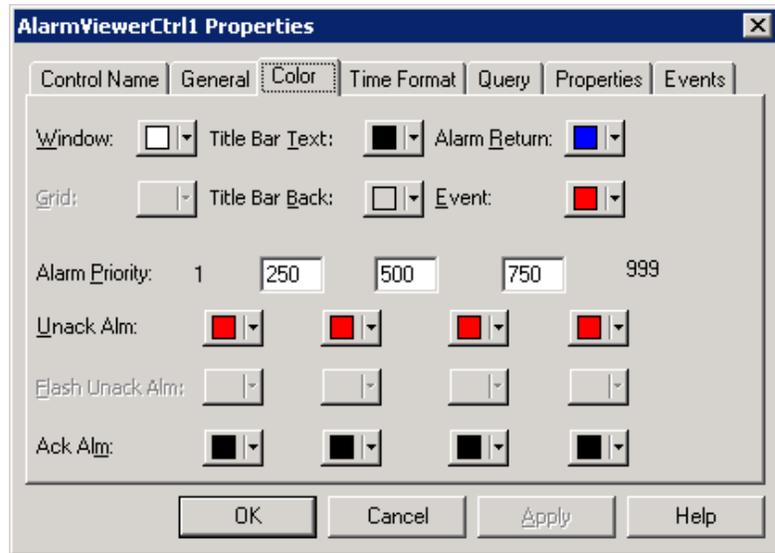
All the options of the **Color** tab in the InTouch Alarm Viewer control can be set on the **Colors** page of the ArchestrA Alarm Control.

The following table shows you some minor differences in wording:

InTouch Alarm Viewer control	ArchestrA Alarm Control
Title Bar Text	Heading Text
Title Bar Back	Heading Background
Alarm Return	Alarm RTN

You can also set the background color in addition to the text color for most of the alarm records.

You can set the alarm priority range breakpoints directly in the table in the **From Pri** column.

Transferring Configuration of the Time Format Tab

You can transfer the configuration of the **Time Format** tab options of the InTouch Alarm Viewer control to the ArchestrA Alarm Control.

InTouch option	Alarm Control option
Time Format	In the ArchestrA Alarm Control, you can configure the **Time Format** option on the **Time Settings** page.
Displayed Time	This option has no meaning in the ArchestrA Alarm Control. All alarm records are shown with the following time stamps in the Alarm Control grid: • **Time (OAT)**: Original Alarm Time • **Time (LCT)**: Last Changed Time • **Time (LCT, OAT)**: Last Changed Time, but Original Alarm Time if the alarm record is unacknowledged.
Displayed Time Zone	In the ArchestrA Alarm Control, this option is called **Time Zone**. You can configure it on the **Time Settings** page. You need to explicitly configure the time zone for the correct time stamp.

Transferring Configuration of the Query Tab

You can transfer the configuration of the **Query** tab options of the InTouch Alarm Viewer control to the ArchestrA Alarm Control.

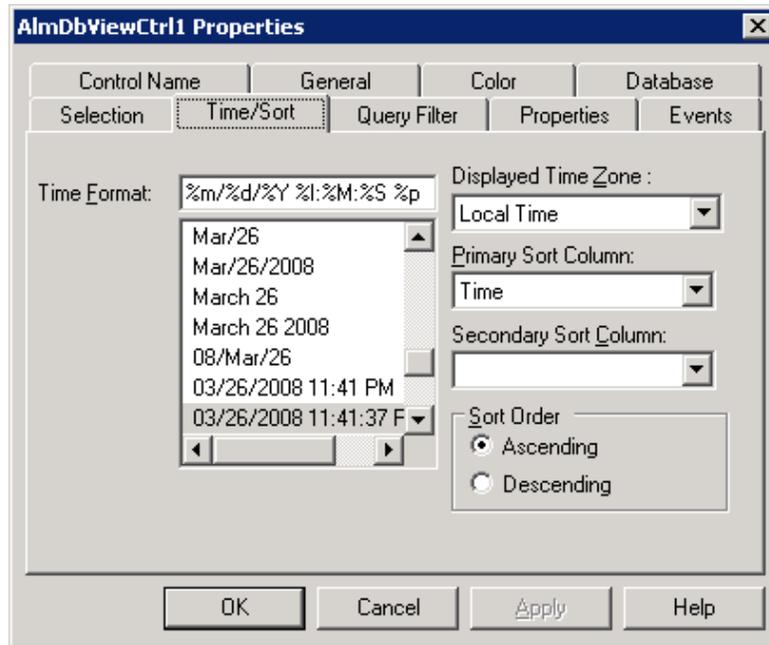| InTouch option	Alarm Control option
From Priority, To Priority	In the ArchestrA Alarm Control, you can only set the priority limits as part of a query filter on the **Query Filters** page. For more information, see Filtering Alarms on page 42.
Alarm State	In the ArchestrA Alarm Control, you can only set the alarm state limitation as part of a query filter on the **Query Filters** page. For more information, see Filtering Alarms on page 37.
Query Type	In the ArchestrA Alarm Control, you can set the **Client Mode** option on the **Alarm Mode** page as follows: • For query type "Summary", set the client mode to **Current Alarms**. • For query type "Historical", set the client mode to **Recent Alarms and Events**.
Alarm Query	In the ArchestrA Alarm Control, you can configure the **Alarm Query** option on the **Alarm Mode** page.

InTouch option	Alarm Control option
Query Favorites File, Edit Query Favorites	In the ArchestrA Alarm Control, all query favorites and filter favorites are managed on one page and are interchangeable between different client modes. To access the Query Filter Favorites, open the **Query Filters** page.
Sort Column	In the ArchestrA Alarm Control, you can configure the sorting of alarm records on the **Column Details** page.
Auto Scroll to New Alarms	In the ArchestrA Alarm Control, you can configure the **Auto Scroll to New Alarms** on the **Run-Time Behavior** page.
Secondary Sort Column, Sort Direction	In the ArchestrA Alarm Control, you can configure the sorting of alarm records on the **Column Details** page.

Transferring Configuration of the Properties Tab

You can set the properties of the ArchestrA Alarm Control in the **Properties Editor** when the Alarm Control is selected on the canvas.
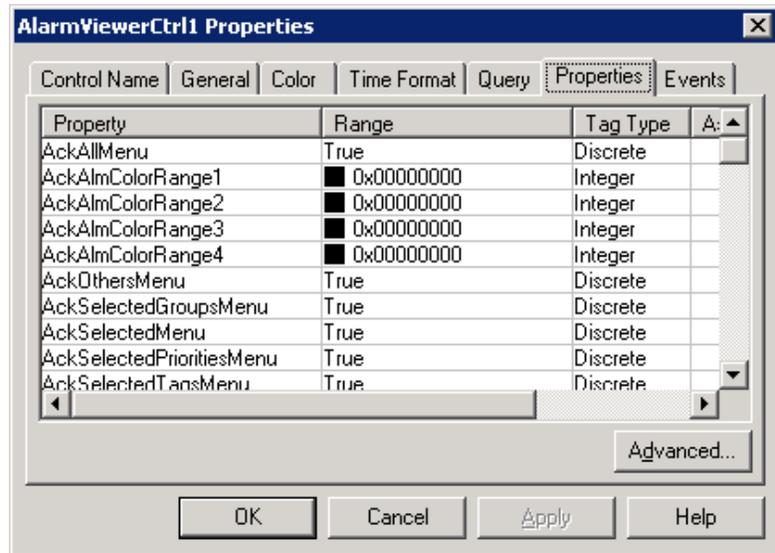
For more information on the exact mapping between the InTouch Alarm Viewer control properties and ArchestrA Alarm Control properties, see Mapping Properties and Methods on page 169.

The advanced property filtering feature does not exist in the ArchestrA Alarm Control. However, when you browse for properties of the ArchestrA Alarm Control from other elements with the **Galaxy Browser**, you can filter the properties. Also, the properties of the ArchestrA Alarm Control are logically grouped in the Properties Editor.

Transferring Script Configuration on the Events Tab

You can configure scripts for events of the ArchestrA Alarm Control on the **Event** animation page. The events are the same as the events for the InTouch Alarm Viewer control:

- Click
- DoubleClick
- New Alarm

- Shutdown
- StartUp

Transferring the InTouch Alarm DB View Control Configuration

You can transfer the configuration of the InTouch Alarm DB View control tabs options to the ArchestrA Alarm Control.

Transferring Configuration of the Control Name Tab

You can transfer the configuration of the **Control Name** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.

InTouch option	Alarm Control option
ControlName	You can rename the ArchestrA Alarm Control the same way as any other elements on the canvas. For more information, see the *Creating and Managing ArchestrA Graphics User's Guide*.
Left, Top, Width, and Height	You can directly edit the positioning options in the same way as any other element on the canvas. Edit the following properties in the Properties Editor: **X**, **Y**, **Width**, and **Height**.

InTouch option	Alarm Control option
Visible	You can directly edit the visibility option in the same way as any other element on the canvas. In the Properties Editor, edit the **Visible** property.
GUID	This option has no meaning in the ArchestrA Alarm Control.

Transferring Configuration of the General Tab

You can transfer the configuration of the **General** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.

InTouch option	Alarm Control option
Enable Refresh Menu	In the ArchestrA Alarm Control, you can configure the availability of the **Requery** shortcut menu option on the **Run-Time Behavior** page.
Enable Sort Menu	In the ArchestrA Alarm Control, you can configure the availability of the **Sort** shortcut menu option on the **Run-Time Behavior** page.

InTouch option	Alarm Control option
Enabled Reset Menu	In the ArchestrA Alarm Control, you can configure the availability of the **Reset** shortcut menu option on the **Run-Time Behavior** page.
Enabled Filter Menu	In the ArchestrA Alarm Control, you can configure the availability of the **Query Filters** shortcut menu option on the **Run-Time Behavior** page.
Display Mode	In the ArchestrA Alarm Control, set the **Client Mode** on the **Alarm Mode** page to the same setting as the **Display Mode** setting in the InTouch Alarm DB View control.
Column Details	In the ArchestrA Alarm Control, you can configure the column details directly on the **Column Details** page.
Show Grid	In the ArchestrA Alarm Control, you can configure the **Show Grid** option on the **Run-Time Behavior** page.
Show Heading	In the ArchestrA Alarm Control, you can configure the **Show Heading** option on the **Run-Time Behavior** page.
Row Selection	In the ArchestrA Alarm Control, this option is called **Row Selection**. You can configure it on the **Run-Time Behavior** page.
Resize Column	In the ArchestrA Alarm Control, this option is called **Allow Column Resizing**. You can configure it on the **Run-Time Behavior** page.
Show Status Bar	In the ArchestrA Alarm Control, you can configure the **Show Status Bar** option on the **Run-Time Behavior** page.
Retrieve Buttons	In the ArchestrA Alarm Control, the retrieve buttons are not available. The underlying grid technology handles the alarm retrieval from the alarm database.

InTouch option	Alarm Control option
Silent Mode	In the ArchestrA Alarm Control, this option is called **Hide Errors and Warnings**. You can configure it on the **Run-Time Behavior** page.
Font	You can configure this option from the ArchestrA Symbol Editor page. Select the ArchestrA Alarm Control on the canvas and select an appropriate font type, size, and style on the menu bars.
Show Message	In the ArchestrA Alarm Control, this option is called **Show Custom 'No Records' Message**. You can configure it on the **Run-Time Behavior** page.

Transferring Configuration of the Color Tab

You can transfer the configuration of the **Color** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.

All the options of the **Color** tab in the InTouch Alarm DB View control can be set on the **Colors** page of the ArchestrA Alarm Control.

The following table shows you some minor differences in wording:

InTouch Alarm DB View control	ArchestrA Alarm Control
Forecolor	Text
Backcolor	Background
Alm	n/a
Return	RTN

You can set the alarm priority range breakpoints directly in the table in the **From Pri** column.

Transferring Configuration of the Database Tab

You can transfer the configuration of the **Database** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.
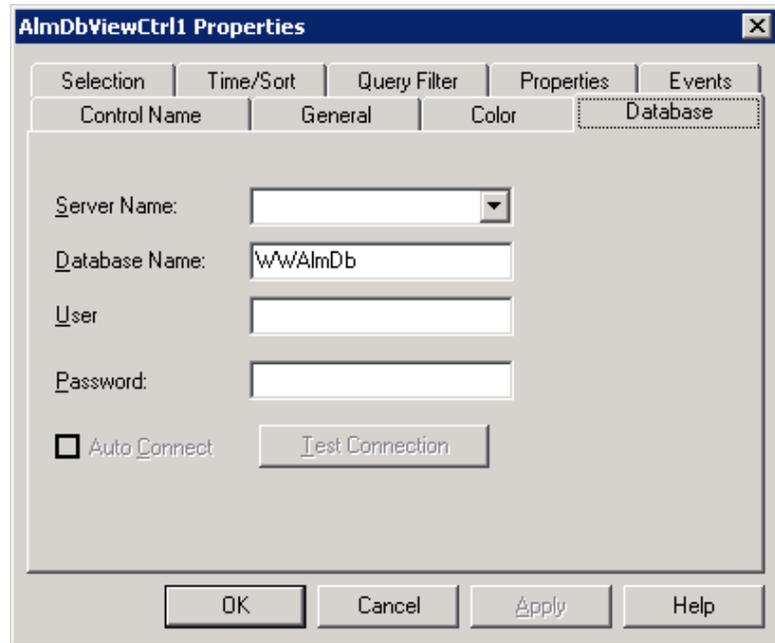
In the ArchestrA Alarm Control, you can configure the following options on the **Alarm Mode** page:

- **Server Name**
- **User**
- **Test Connection**
- **Database Name**
- **Password**

In the ArchestrA Alarm Control, the **Auto Connect** option is called **Query on Startup**. You can configure it on the **Run-Time Behavior** page.

The configuration for the Alarm Database only appears if the **Client Mode** is set to **Historical Alarms**, **Historical Events**, or **Historical Alarms and Events**.

Transferring Configuration of the Selection Tab

You can transfer the configuration of the **Selection** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.
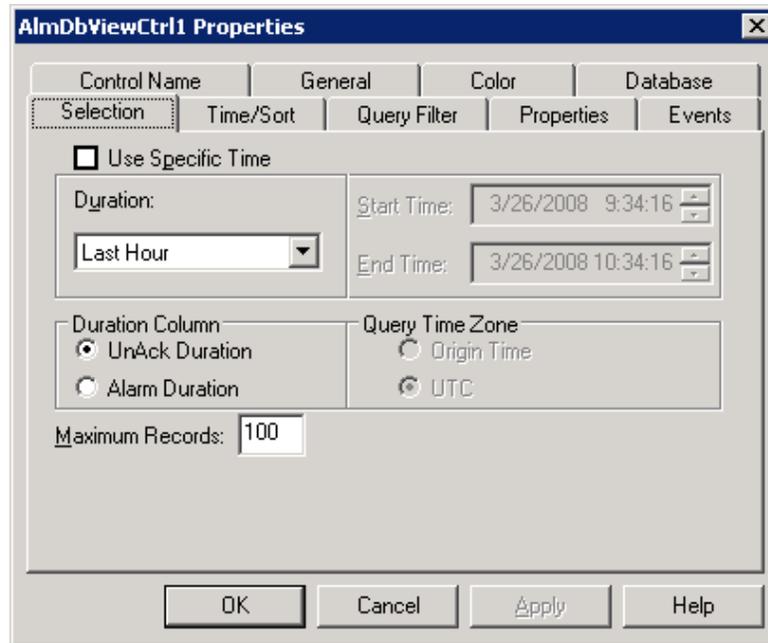
InTouch option	Alarm Control option
Use Specific Time, Start Time, End Time	In the ArchestrA Alarm Control, you can set these options directly in the Time Range Picker control on the **Alarm Mode** page.
	When you select a time from either the start time or end time part of the Time Range Picker control, the Alarm Control is automatically set to use a specific time.
	To keep the specific start and end time, you must also clear **Update to Current Time**. When you refresh the Alarm Control grid at run time, the time range stays fixed to the given start and end time.

InTouch option	Alarm Control option
Duration	In the ArchestrA Alarm Control, you can set this option directly in the Time Range Picker control on the **Alarm Mode** page. When you select a duration from the center part of the Time Range Picker control, the Alarm Control is automatically set to use a time offset. To keep the duration, you must also select the **Update to Current Time** check box. When you refresh the Alarm Control grid at run time, the end time is set to the current time and the Alarm Control shows the alarms within the set duration.
UnAck Duration, Alarm Duration	In the ArchestrA Alarm Control, you cannot configure the Unack Duration and Alarm Duration settings. The Alarm Control grid shows both UnAck Duration and Alarm Duration in separate columns.
Query Time Zone	In the ArchestrA Alarm Control, you can configure the **Time Zone** setting on the **Time Settings** page.
Maximum Records	In the ArchestrA Alarm Control, you can configure the **Maximum Records** setting on the **Alarm Mode** page.

Transferring Configuration of the Time/Sort Tab

You can transfer the configuration of the **Time/Sort** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.

InTouch option	Alarm Control option
Time Format	In the ArchestrA Alarm Control, you can configure the **Time Format** setting on the **Time Settings** page.
Displayed Time Zone	In the ArchestrA Alarm Control, you can configure the **Time Zone** setting on the **Time Settings** page.
Primary Sort Column, Secondary Sort Column, Sort Order	In the ArchestrA Alarm Control, you can configure the sorting options on the **Column Details** page.

Transferring Configuration of the Query Filter Tab

You can transfer the configuration of the **Query Filter** tab options of the InTouch Alarm DB View control to the ArchestrA Alarm Control.
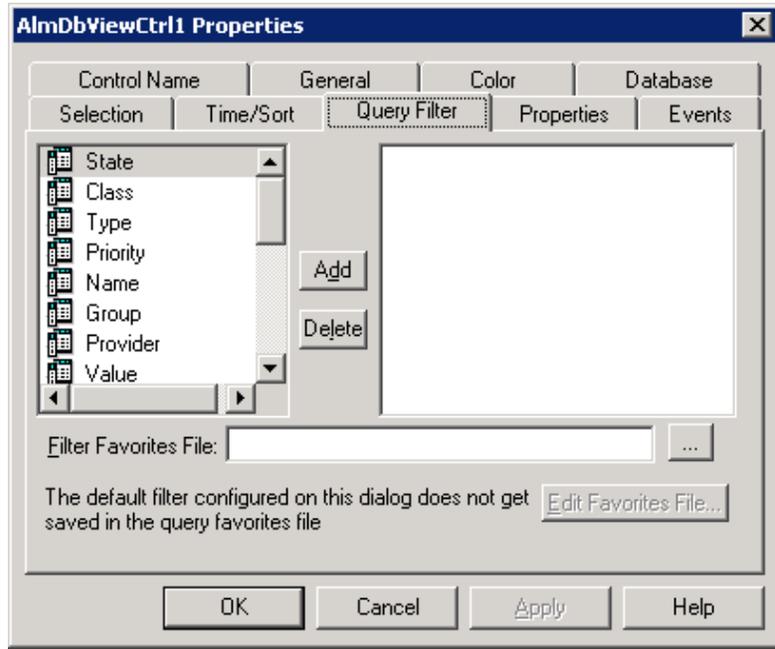
In the ArchestrA Alarm Control, all query favorites and filter favorites are managed on one page and are interchangeable between different client modes. To access the Query Filter Favorites, open the **Query Filters** page.

Transferring Configuration of the Properties Tab

You can set the properties of the ArchestrA Alarm Control in the **Properties Editor** when the Alarm Control is selected on the canvas.
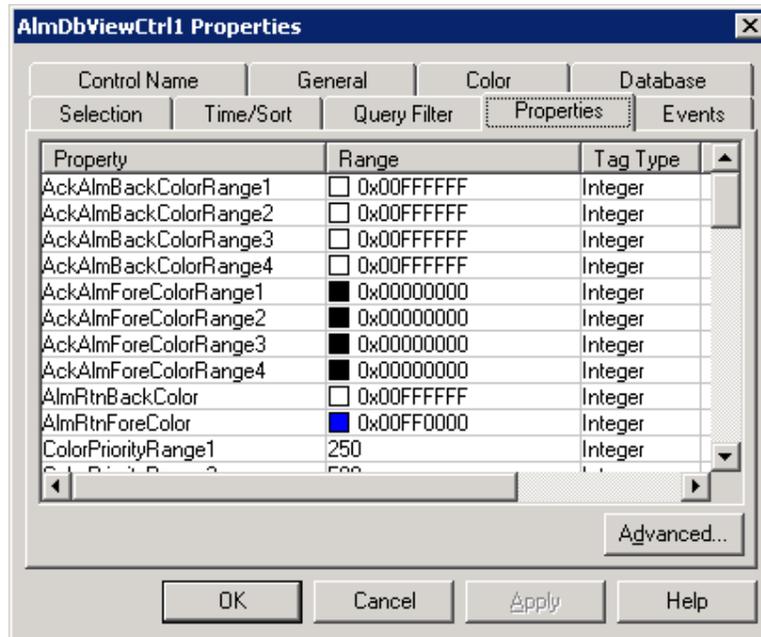
For more information on the exact mapping between the InTouch Alarm DB View control properties and ArchestrA Alarm Control properties, see Mapping Properties and Methods on page 169.

The advanced property filtering feature does not exist in the ArchestrA Alarm Control. However, when you browse for properties of the ArchestrA Alarm Control from other elements with the **Galaxy Browser**, you can filter the properties. Also, the properties of the ArchestrA Alarm Control are logically grouped in the Properties Editor.

Transferring Scripts Configuration on the Events Tab

You can configure scripts for events of the ArchestrA Alarm Control on the **Event** animation page. The events are the same as the events for the InTouch Alarm DB View control:

- Click
- DoubleClick
- NewAlarm

- Shutdown
- StartUp
-

For more information, see Configuring Events on page 145.

Transferring Query Favorites Configuration

You can only transfer query favorites configuration from InTouch to the ArchestrA Alarm Control by recreating the filters on the Query Filters page.

If you intend to use a the query filter in one of the current client modes, make sure you also include **Provider** and **Group** as filter criteria.

Mapping Properties and Methods

The following table shows all properties and methods of the InTouch Alarm Viewer control and InTouch Alarm DB View controls and their corresponding properties and methods of the ArchestrA Alarm Control.

InTouch alarm control property or method	ArchestrA Alarm Control property or method
AboutBox()	AboutBox() Method on page 125
AckAll()	Ack.All() Method on page 125
AckAllMenu	ContextMenu.AckAll Property on page 95
AckAlmBackColor	AlarmColor.Ack.BackGround Property on page 80
AckAlmBackColorRange1	AlarmColor.Ack.BackGround Property on page 80
AckAlmBackColorRange2	AlarmColor.Ack.BackGround Property on page 80
AckAlmBackColorRange3	AlarmColor.Ack.BackGround Property on page 80
AckAlmBackColorRange4	AlarmColor.Ack.BackGround Property on page 80
AckAlmColorRange1	AlarmColor.Ack.ForeGround Property on page 82
AckAlmColorRange2	AlarmColor.Ack.ForeGround Property on page 82
AckAlmColorRange3	AlarmColor.Ack.ForeGround Property on page 82
AckAlmColorRange4	AlarmColor.Ack.ForeGround Property on page 82

InTouch alarm control property or method	ArchestrA Alarm Control property or method
AckAlmForeColor	AlarmColor.Ack.ForeGround Property on page 82
AckAlmForeColorRange1	AlarmColor.Ack.ForeGround Property on page 82
AckAlmForeColorRange2	AlarmColor.Ack.ForeGround Property on page 82
AckAlmForeColorRange3	AlarmColor.Ack.ForeGround Property on page 82
AckAlmForeColorRange4	AlarmColor.Ack.ForeGround Property on page 82
AckGroup()	Ack.Group() Method on page 126
AckOthersMenu	ContextMenu.AckOthers Property on page 95
AckPriority()	Ack.Priority() Method on page 126
AckRtnBackColor	AlarmColor.RTN.BackGround Property on page 85
AckRtnForeColor	AlarmColor.RTN.ForeGround Property on page 86
AckSelected()	Ack.Selected() Method on page 127
AckSelectedGroup()	Ack.SelectedGroup() Method on page 127
AckSelectedGroupsMenu	ContextMenu.AckSelectedGroups Property on page 96
AckSelectedMenu	ContextMenu.AckSelected Property on page 96
AckSelectedPrioritiesMenu	ContextMenu.AckSelectedPriorities Property on page 96
AckSelectedPriority()	Ack.SelectedPriority () Method on page 128
AckSelectedTag()	Ack.SelectedTag() Method on page 128
AckSelectedTagsMenu	ContextMenu.AckSelectedTags Property on page 97
AckTag()	Ack.Tag() Method on page 128
AckVisible()	Ack.Visible() Method on page 129

InTouch alarm control property or method	ArchestrA Alarm Control property or method
AckVisibleMenu	ContextMenu.AckVisible Property on page 97
AlarmQuery	AlarmQuery Property on page 93
AlarmState	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.
AlmRtnBackColor	AlarmColor.RTN.BackGround Property on page 85
AlmRtnColor	AlarmColor.RTN.ForeGround Property on page 86
AlmRtnForeColor	AlarmColor.RTN.ForeGround Property on page 86
ApplyDefaultQuery()	Favorite Property on page 106
ApplyQuery()	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.
AutoConnect	QueryStartup Property on page 111
AutoScroll	AutoScroll Property on page 94
ColorPriorityRange1	AlarmColor.Range Property on page 84
ColorPriorityRange2	AlarmColor.Range Property on page 84
ColorPriorityRange3	AlarmColor.Range Property on page 84
ColumnResize	AllowColumnResize Property on page 93
Connect()	Connect() Method on page 130
ConnectStatus	ConnectStatus Property on page 95
CustomMessage	NoRecordsMessage.Message Property on page 111
DefaultAckComment	AckComment.DefaultValue Property on page 79

InTouch alarm control property or method	ArchestrA Alarm Control property or method
DisplayedTime	This option has no meaning in the ArchestrA Alarm Control. All three times are shown in the Alarm Control: • Original Alarm Time • Last Changed Time • Last Changed Time, but Original Alarm Time for unacknowledged alarms
DisplayedTimeZone	TimeZone.TimeZone Property on page 122
DisplayMode	ClientMode Property on page 94
Duration	TimeSelector.TimeDuration Property on page 120
EndTime	TimeSelector.EndDate Property on page 119
EventBackColor	EventColor.BackGround Property on page 105
EventColor	EventColor.ForeGround Property on page 105
EventForeColor	EventColor.ForeGround Property on page 105
ExtendedSelection	RowSelection Property on page 112
FilterFavoritesFile	No corresponding property. The file name is used as a parameter for the Favorites.Export() Method and Favorites.Import() Method methods.
FilterMenu	ContextMenu.Favorites Property on page 97
FilterName	Favorite Property on page 106
FlashUnackAlarms	FlashUnAckAlarms Property on page 106
FlashUnAckAlmColorRange1	AlarmColor.UnAck.Flash.ForeGround Property on page 89
FlashUnAckAlmColorRange2	AlarmColor.UnAck.Flash.ForeGround Property on page 89

InTouch alarm control property or method	ArchestrA Alarm Control property or method
FlashUnAckAlmColorRange3	AlarmColor.UnAck.Flash.ForeGround Property on page 89
FlashUnAckAlmColorRange4	AlarmColor.UnAck.Flash.ForeGround Property on page 89
Font	You can only set the font at design time, not at run time.
FreezeDisplay()	FreezeDisplay() Method on page 131
FreezeMenu	ContextMenu.Freeze Property on page 98
FromPriority	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.
GetItem()	GetItem() Method on page 132
GetLastError()	GetLastError() Method on page 132
GetNext()	No corresponding property. Alarm records are retrieved one by one from the Alarm Database after the initial set of alarm records is retrieved. The initial set is defined by the **Maximum Records** setting.
GetPrevious()	No corresponding property. Alarm records are retrieved one by one from the Alarm Database after the initial set of alarm records is retrieved. The initial set is defined by the **Maximum Records** setting.
GetSelectedItem()	GetSelectedItem() Method on page 133
GridColor	GridColor Property on page 106
GroupExactMatch	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.

InTouch alarm control property or method	ArchestrA Alarm Control property or method
GroupName	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.
MaxRecords	MaxDatabaseRecords Property on page 109
MoveWindow()	MoveWindow() Method on page 137
NewAlarmEventMode	NewAlarmEventMode Property on page 110
Password	Database.Authentication Property on page 102
PrimarySort	SortOrder.First Property on page 115
ProviderExactMatch	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.
ProviderName	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite property. For more information, see Favorite Property on page 106.
QueryFavoritesFile	No corresponding property. The file name is used as a parameter for the Favorites.Export() Method and Favorites.Import() Method methods.
QueryFavoritesMenu	ContextMenu.Favorites Property on page 97
QueryName	Favorite Property on page 106
QueryStartup	QueryStartup Property on page 111
QueryTimeZone	TimeZone.TimeZone Property on page 122
QueryType	ClientMode Property on page 94
Refresh()	Requery() Method on page 138
RefreshMenu	ContextMenu.Requery Property on page 101

InTouch alarm control property or method	ArchestrA Alarm Control property or method
Requery()	Requery() Method on page 138
RequeryMenu	ContextMenu.Requery Property on page 101
Reset()	Reset() Method on page 138
ResetMenu	ContextMenu.Reset Property on page 101
RetainSuppression	RetainHidden Property on page 111
RowCount	RowCount Property on page 112
RowSelection	RowSelection Property on page 112
SecondarySort	SortColumn.Second Property on page 114
SecondarySortColumn	SortColumn.Second Property on page 114
SelectAll()	To select all records, see Select.All() Method on page 138. To reverse the selection of all records, see Toggle.All() Method on page 144.
SelectedCount	SelectedCount Property on page 113
SelectGroup()	Select.Group() Method on page 138
SelectItem()	To select a given alarm record, see Select.Item() Method on page 139. To reverse the selection of a given alarm record, see Toggle.Item() Method on page 144.
SelectPriority()	Select.Priority() Method on page 139
SelectQuery()	Favorite Property on page 106
SelectTag()	Select.Tag() Method on page 140
ServerName	Database.ServerName Property on page 103
SetQueryByName	Favorite Property on page 106
SetSort()	SetSort() Method on page 141
ShowContext()	Show.Context() Method on page 141
ShowContextMenu	ShowContextMenu Property on page 113

InTouch alarm control property or method	ArchestrA Alarm Control property or method
ShowDate	There is no equivalent functionality in the ArchestrA Alarm Control.
ShowFetch	No corresponding property. The buttons for retrieving sets of alarm records from the Alarm Database do not exist in the ArchestrA Alarm Control.
ShowFilter()	Show.Favorite() Method on page 141
ShowGrid	ShowGrid Property on page 113
ShowHeading	ShowHeading Property on page 114
ShowMessage	NoRecordsMessage.Enabled Property on page 110
ShowQueryFavorites()	Show.Favorite() Method on page 141
ShowSort()	Show.Sort() Method on page 142
ShowStatistics()	Show.Statistics() Method on page 142
ShowStatusBar	ShowStatusBar Property on page 114
ShowSuppression()	Show.Hidden() Method on page 142
SilentMode	HideErrors Property on page 109
SortColumn	You can set three sort columns in the ArchestrA Alarm Control. To set the first column, see SortColumn.First Property on page 114.
SortMenu	ContextMenu.Sort Property on page 101
SortOnCol()	To set the first sort column, see SortColumn.First Property on page 114. To set the sort order of the first sort column, see SortOrder.First Property on page 115.
SortOrder	SortOrder.First Property on page 115
SpecificTime	UpdateToCurrentTime Property on page 123
StartTime	TimeSelector.StartDate Property on page 119
StatsMenu	ContextMenu.Statistics Property on page 102
SuppressAll()	Hide.All() Method on page 133

InTouch alarm control property or method	ArchestrA Alarm Control property or method
SuppressAllMenu	ContextMenu.HideAll Property on page 98
SuppressedAlarms	HiddenAlarms Property on page 109
SuppressGroup()	Hide.Group() Method on page 134
SuppressionMenu	ContextMenu.Hidden Property on page 98
SuppressOthersMenu	ContextMenu.HideOthers Property on page 99
SuppressPriority()	Hide.Priority() Method on page 134
SuppressSelected()	Hide.Selected() Method on page 135
SuppressSelectedGroup()	Hide.SelectedGroup() Method on page 135
SuppressSelectedGroupsMenu	ContextMenu.HideSelectedGroups Property on page 99
SuppressSelectedMenu	ContextMenu.HideSelected Property on page 99
SuppressSelectedPrioritiesMenu	ContextMenu.HideSelectedPriorities Property on page 100
SuppressSelectedPriority()	Hide.SelectedPriority() Method on page 135
SuppressSelectedTagsMenu	ContextMenu.HideSelectedTags Property on page 100
SuppressSelectedTag()	Hide.SelectedTag() Method on page 135
SuppressTag()	Hide.Tag() Method on page 136
SuppressVisible()	Hide.Visible() Method on page 136
SuppressVisibleMenu	ContextMenu.HideVisible Property on page 100
Time	Time.Type Property on page 117 and Time.Format Property on page 116
TimeFormat	Time.Format Property on page 116 and Time.Type Property on page 117
TitleBackColor	HeadingColor.BackGround Property on page 107

InTouch alarm control property or method	ArchestrA Alarm Control property or method
TitleForeColor	HeadingColor.ForeGround Property on page 108
ToPriority	No corresponding property. Configure a Query Filter favorite at design time instead and use the Favorite Property. For more information, see Favorite Property on page 106.
TotalAlarms	TotalRowCount Property on page 122
TotalRowCount	TotalRowCount Property on page 122
UnAckAlarms	UnAckAlarms Property on page 123
UnAckAlmBackColor	AlarmColor.UnAck.BackGround Property on page 86
UnAckAlmBackColorRange1	AlarmColor.UnAck.BackGround Property on page 86
UnAckAlmBackColorRange2	AlarmColor.UnAck.BackGround Property on page 86
UnAckAlmBackColorRange3	AlarmColor.UnAck.BackGround Property on page 86
UnAckAlmBackColorRange4	AlarmColor.UnAck.BackGround Property on page 86
UnAckAlmColorRange1	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmColorRange2	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmColorRange3	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmColorRange4	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmForeColor	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmForeColorRange1	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmForeColorRange2	AlarmColor.UnAck.ForeGround Property on page 90
UnAckAlmForeColorRange3	AlarmColor.UnAck.ForeGround Property on page 90

InTouch alarm control property or method	ArchestrA Alarm Control property or method
UnAckAlmForeColorRange4	AlarmColor.UnAck.ForeGround Property on page 90
UnAckOrAlarmDuration	No corresponding property. UnAck Duration and Alarm Duration are shown in the Alarm Control grid.
UnSelectAll()	UnSelectAll() Method on page 145
UnSuppressAll()	UnhideAll() Method on page 145
UnsuppressAllMenu	ContextMenu.UnhideAll Property on page 102
UseDefaultAckComment	AckComment.UseDefault Property on page 80
UserID	Database.UserID Property on page 104
Visible	Visible Property on page 124
WindowColor	WindowColor Property on page 124

Index

Symbols

A