

Alarms and Audit Trail

Alarms are elements of fundamental importance in a SCADA system. Alarms signal when an event occurs, allowing the system or the human monitor to take appropriate action. Events may occur, for example, when a device encounters an abnormal condition or when a parameter being monitored falls outside a normal range. In some cases the system may be able to automatically provide the desired response, such as sending a corrective signal to a device. In other cases human monitors need to be informed, by a visual display, an audio signal, or a remote message.

In FactoryStudio, alarms are associated with tags. An alarm may be triggered when a tag reaches a certain value, or when it falls outside a certain range. Alarms may be linked to standard responses such as visual displays or sounds. Alternately, custom code may be provided to create almost any desired response.

Alarms are assigned priorities, so that the most urgent can be handled if there are multiple alarms at the same time. Also, for convenience, alarms are organized in groups, and each group shares some common properties. In particular, an alarm's group determines if and how the alarm must be acknowledged, and if and for how long it should be recorded in the system log.

To minimize spurious alarms, a time interval may be established before each alarm may be triggered after system startup, and a minimum time may be set before the same alarm can be triggered again.

On this page:

- [Configuring Alarms and Events](#)
- [The Audit Trail](#)
- [The Alarm Database](#)
- [Alarm Visualization](#)
- [Acknowledgments and Subscriptions](#)
- [The Alarm Namespace](#)

Configuring Alarms and Events

Alarm Groups

Every alarm and system event belongs to an alarm group. Alarm groups allow you to configure common settings for a collection of alarms. The group settings determine such things as whether an acknowledgment is required, whether a sound plays, what is logged, and how alarms display. FactoryStudio comes with several predefined alarm groups that you can use, and you can create your own groups as well. The predefined groups are:

- Warning (Warning messages that do not require acknowledgment)
- Critical (Critical Messages that require acknowledgment)
- AuditTrail (System Events log)

To configure alarm groups:

- Go to **Edit > Alarms > Groups**.
- Enter or select information, as needed.

Column	Description
Name	Enter a name for the alarm group. The system lets you know if the name is not valid.
AckRequired	If required, the alarm stays in the alarm list until someone acknowledges the alarm by double-clicking it in the application.
ActiveTimeDeadband and	Enter a time dead band that will create a delay from when the alarm happens plus the dead band configured, and will only trigger the alarm after this
AckTimeout	Defines a timeout to ack the alarm. If the alarm is not acknowledged after the specified time, the alarm becomes active again
AutoAckTime	If the alarm is not acknowledged after the specified time, the system acknowledges the alarm.

Sound	Select the sound that will play when the alarm occurs.
Show	Select List for the alarm to display in the alarm window in the application
LogEvents	Select when you want the alarm to be logged to the alarm historian: • None—Alarms are not logged. • Active—Log when alarm is active. • ActiveAck—Log when the alarm is active and acknowledged. • ActiveNorm—Log when the alarm returns to normal. • All—Log all of the above conditions.]
Colors	Select the colors you want to use for each state: • Active—Configured alarm is in the alarm state. • Normalized—Configured alarm was in the alarm state, but is no longer in the alarm state and still needs to be acknowledged. • Acknowledged—Configured alarm has been acknowledged, but still is in the alarm state.]
NotificationMethod	Calls a Script > Class method that trigger a code when the alarm happens
Description	Enter a description of this alarm group.
[Other columns]	For definitions of other columns that are available in many tables, see "Common Column Definitions" .

Alarms and Events

To configure an individual alarm, the first step is to identify the tag on which the alarm will be based. It is okay to have multiple alarms associated with the same tag value.

Next, you must define the condition that will trigger the alarm. Possible conditions are based on comparing the value of the tag to two specific values that you define called the *Limit* and the *Setpoint*. The available conditions are described in the table below.

Additional parameters to be configured for each alarm include the group for the alarm, the priority, and the message that will be displayed in the alarm list.

To configure Alarms:

- Go to **Edit > Alarms > Items**.

Enter or select information, as needed.

Column	Description
TagName	Enter a tag name or click ... to select a tag.
Condition	Select the condition you want to use for this alarm event (see the table below).
Limit	Enter a value for the alarm limit to be used for the condition you selected.
Deadband	Set the deadband, which represents the time that must pass after an alarm before a second alarm can be generated
Setpoint	Set a value or a tag to compare for conditions that are based on a deviation.
SetpointDeadband	Set the deadband for the setpoint when compared with the deviation
Group	Select the alarm group that should control what happens when this alarm occurs.
Area	Once created under Alarms > Areas, the areas will be displayed here and will be able to be chosen.

Priority	Enter a priority value that controls where the alarm displays in the list. The higher the number the higher the priority. You can use the same priority for more than one alarm event. Enter 0 (zero) for alarms to be at the end of the list.
Message	Enter the text that displays in the alarm list.
[Other columns]	For definitions of other columns that are available in many tables, see "Common Column Definitions" .

The conditions that can be set are described in the following table:

Hi	Tag >= limit
HiHi	Tag >= limit (when acknowledged, a Hi alarm is automatically acknowledged to the same Tag)
Lo	Tag <= limit
LoLo	Tag <= limit (when acknowledged, a Lo alarm is automatically acknowledged to the same Tag)
RateOfChange	Tag rate of change >= limit
DeviationMinor	Absolute value (tag - Setpoint) > limit
DeviationMajor	Absolute value (tag - Setpoint) > limit
Equal	Tag = limit
GreaterThan	Tag > limit
GreaterEqual	Tag >= limit
LessThan	Tag < limit
LessEqual	Tag <= limit
Changed	Tag value changed
ChangedUp	Tag value increased
ChangedDown	Tag value decreased
NotEqual	Tag different from limit



The limit is a specific value on which the condition is based. The setpoint, if used, may be either a value or another tag.

General Alarm Options

Two general options should be set that apply to all alarms. These determine how the system handles alarms during startup and how long alarms and events remain in the alarm historian.

To configure general alarm options:

- Go to **Edit > Alarms > Groups**.
- In the Initial Disable Time field, enter how long the system should wait during startup before considering an alarm state to be an alarm. This provides a short time for the system to stabilize before alarms may be triggered.
- In the Life Time field, enter how long the alarm historian retains the alarm data, in days.

The Audit Trail

The audit trail provides a log of selected system events for a specified length of time.

To configure the Audit Trail:

- Go to **Edit > Alarms > Groups**.
- Click **Settings**.
- The AuditTrail window displays.
- To enable event logging, select the Enable option.
- After selecting Enable, select the events you want to log.
- Click **OK**.

The Alarm Database

The database used to store the Alarms is defined in Edit-Datasets-DBs by the database connection object with the name AlarmHistorian. By default, when a new project is created, the AlarmHistorian is defined to use the Tatsoft built-in embedded SQL database TatsoftDB.

The TatsoftDB database should be used for databases up to 10GB. If it is expected to create more than 10GB, you should define another SQL for Tag Historian Database. In order to define another Database to store the Alarm Historian database, you need to create a new database connection, as explained on "Configuring Database Connections" on page 105, and name it AlarmHistorian.

The system does not allow duplicate names, so to create a DB connection with the name AlarmHistorian you must rename or delete the existing row using that name.

The Alarm Module automatically creates the required tables in the database. An example of table schema is available opening any file with extension .TAlarm, created when running application using the default TatsoftDB as the alarm logging database.

Alarm Visualization

To visualize alarms and events, you can use the built-in object on the displays or reports. See section "[Configuring an Alarm Window](#)" for more information.

Acknowledgments and Subscriptions

Acknowledging Alarms

There are several methods for acknowledging alarms:

- Using Alarm visualization object in the displays.
- Tag properties.
- Alarm group or Alarm item runtime properties.

Acknowledge All Alarms

You can use the <Alarm.AckAll> property that acknowledges all active Alarms .

Acknowledge Single Alarm or Highest Priority Alarm

The <Alarm.PriorityItem.UnAck> property allows acknowledgment of the highest priority Alarm configured in <Edit.Alarms.Items> the "Priority" column if there is a pending ack alarm.

Acknowledge a Specific Alarm

To acknowledge a specific alarm use the <Alarm.Items.IDxx.Unack> property. To verify the contents of the IDxx Alarm columns, go to Alarm item and add the ID column (right-click in the table and select "ID").

Subscriptions to Notifications

In order to apply custom actions using .NET scripts, you can subscribe to notifications of alarms and events.

These notifications are typically used to send SMS or email warnings, execute custom calculations, add custom notification messages or audio alarms, text-to-speech audio alarms, and to perform any kind of action defined by custom programming.

To subscribe to alarm events, create a method in any Script Class of type Server, with the following prototype:

```
void Notification(AlarmEventInfo[] info)
```

Finally, select that method on Edit > Alarms >Groups at the NotificationMethod Column.

The name of the method can vary. What is important is the expected parameters. The AlarmEventInfo structure is defined in the namespace Alarm on the runtime classes reference, <http://www.tatsoft.com/help/fs-2014/runtime/index.html>.

The Alarm Namespace

The Alarm namespace has the properties of the Alarm server.

The Alarm.Group object has the list of all defined groups and their properties.

The Alarm.Item object has all alarm items and their properties.

The following tag properties are connected with the Alarm module:

tag.tagName.Hi: configuration and runtime status of the HI alarm, similar names for the other alarm types.

See "[Namespaces](#)" for the complete programming reference on runtime objects.