

# Alarms and Audit Trail

Alarms are fundamental elements of a SCADA system because they signal when events occur, allowing the system or the human monitor to take appropriate action. Events may occur, for example, when devices encounter abnormal conditions or when parameters fall outside normal ranges. 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. When tags reach certain values or fall outside certain ranges, alarms maybe be triggered. 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, which allows the most urgent alarms to be handled if there are multiple ones at the same time. Also, for convenience, alarms are organized into groups, and each group shares common properties. In particular, an alarm's group determines if and how the alarm must be acknowledged as well as if and for how long it should be recorded in the system log.

For when a system is set up, a time interval can be established that will give the system a time buffer for any possible spurious alarms, and a minimum time can be set before the same alarm can be triggered again.

---

## 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. A group's settings determine such things as whether or not an acknowledgment is required, whether or not a sound plays, what information is logged, and/or how alarms display.

FactoryStudio comes with several predefined alarm groups that can be used, and/or you can create your own groups. 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	Enter a time deadband that will create a delay after an alarm occurs. The alarm will trigger again only after the deadband.
AckTimeout	Defines a timeout for acknowledging 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 the 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: <ul style="list-style-type: none"> <li>• None—Alarms are not logged.</li> <li>• Active—Log when alarm is active.</li> <li>• ActiveAck—Log when the alarm is active and acknowledged.</li> <li>• ActiveNorm—Log when the alarm returns to normal.</li> <li>• All—Log all of the above conditions.</li> </ul>
Colors	Select the colors you want to use for each state: <ul style="list-style-type: none"> <li>• Active—Configured alarm is in the alarm state.</li> <li>• Normalized—Configured alarm was in the alarm state, but is no longer in the alarm state and still needs to be acknowledged.</li> <li>• Acknowledged—Configured alarm has been acknowledged, but it is still in the alarm state.</li> </ul>
NotificationMethod	Calls a <b>Script &gt; Class</b> method that triggers a code when the alarm happens.
Description	Enter a description of for the alarm group.
[Other columns]	For definitions of other columns that are available in many tables, see <a href="#">Common Column Definitions</a> .

## Alarms and Events

The first step in configuring an individual alarm 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 the *Limit* and the *Set point*, which are two specific values that you define. The available conditions are described in the table below.

Additional parameters that can 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	Sets the deadband value. E.g.: for a Hi alarm item with limit as 100 and deadband as 10, the limit value is now 110; for a Lo alarm item, setting the deadband as 10 means the limit value is 90.
Setpoint	Sets a value or a tag that you want to compare other values against to determine if there is a deviation.
SetpointDeadband	Sets 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 <b>Alarms &gt; Areas</b> , 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 <a href="#">Common Column Definitions</a> .



### PriorityOrder Property

It is also possible to set the priority using the PriorityOrder property in CodeBehind or in PropertyWatch. In this case, the LOWER the number, the HIGHER the priority. Default is 0 (highest priority).

E.g.: `@Alarm.PriorityOrder = 1`

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 for the same Tag)
Lo	Tag <= limit
LoLo	Tag <= limit (when acknowledged, a Lo alarm is automatically acknowledged for 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

There are two general options that apply to all alarms and should be set for 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 a time for how long the system should wait during startup before considering an alarm state to be an alarm. This gives the system time to stabilize before any alarms can be triggered.
- In the Life Time field, enter the number of days for how long the alarm historian retains the alarm data.

## The Audit Trail

The audit trail provides a log of any 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 and is defined by the AlarmHistorian, which is a database connection object. By default, when a new project is created, the AlarmHistorian is defined to use the built-in embedded SQL database, TatsoftDB.

The TatsoftDB database should be used for databases up to 10GB. If you expect to create more than 10GB of data, you should define another SQL database for another Tag Historian. In order to define another database that will store the Alarm Historian, you need to create a new database connection, as explained in "Configuring Database Connections", and name it AlarmHistorian.

The system does not allow duplicate names, so you will need to rename or delete the existing row using the name.

When you use the default TatsoftDB as the alarm logging database, the Alarm Module will automatically create the required tables in the database. To see the table schema for these tables, you can look at any of the database table files, which have the extension TAlarm.

## Alarm Visualization

To visualize alarms and events, you can use the built-in object in your 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 the alarm visualization object in the displays.
- Tag properties.
- Alarm group or the alarm item's runtime properties.

#### *Acknowledge All Alarms*

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

#### *Acknowledge a Single Alarm or the Highest Priority Alarm*

The <Alarm.PriorityItem.UnAck> property allows for the 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 the alarm item and add the ID column (right-click in the table and select "ID").

### Subscriptions to Notifications

In order to apply a custom action using .NET scripts, 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 the method in the NotificationMethod column on Edit > Alarms > Groups.

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 tag properties are connected with the alarm module: For example: tag.tagname.Hi: configuration and runtime status of the HI alarm. The naming of all tag properties follow this same method.

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