Alarms Groups Reference

Organize alarms into groups.

Alarm Groups (Reference) define shared characteristics and behaviors for collections of related alarm items in your FrameworX solution. Alarm Groups enable:

- Collective configuration of alarm behaviors
- Shared acknowledgment requirements
- Common notification methods
- Unified visual and audio alerts
- Centralized logging preferences
- Consistent timeout settings

The platform provides pre-defined groups (Critical, Warning, AuditTrail) for common use cases, plus support for custom groups.

Configuration Properties
Pre-Defined Groups
Creating Alarm Groups
Acknowledgment Settings
Notification Configuration
Deadband and Timing
Runtime Access
Common Configurations
Best Practices Checklist
Troubleshooting

Configuration Properties

Property	Description	Required
Name	Group identifier	Yes
AckRequired	Requires operator acknowledgment	Yes
ActiveTimeDeadband	Time period to ignore active alarms	No
AckTimeout	Timeout period for acknowledgment	No
AutoAckTime	Auto-acknowledge after this time	No
Sound	Audio file for alarm notification	No
Show	Display alarms from this group	Yes
LogEvents	Log events to historian	Yes
Colors	Visual identification colors	No
NotificationMethod	Script method for notifications. From FrameworX 10.1.6, also a webhook or MQTT broker URI (`http://`, `https://`, `mqtt://`, `mqtts://`). See Alarms Notifications Reference.	No
Category	Category designation	No
Description	Group purpose and details	No

Pre-Defined Groups

Group	Purpose	Default Settings
Critical	High-priority alarms requiring immediate action	AckRequired=True, LogEvents=True, High Priority
Warning	Medium/low-priority informational alarms	AckRequired=False, LogEvents=True, Medium Priority
AuditTrail	Hidden alarms for audit logging only	Show=False, LogEvents=True, No Display

Creating Alarm Groups

1. Navigate to **Alarms** **Groups**
2. Click **Plus icon**
3. Enter **Name** and **Description**
4. Click **OK**
5. Edit properties by double-clicking
6. Press **Enter** to save changes

Before deleting: Ensure no AlarmItems are linked to the group. Check via **Alarms** **Items** and use **Track Changes** **Cross Reference**.

Acknowledgment Settings

Property	Behavior
AckRequired	Alarm stays active until acknowledged, even if condition clears
AckTimeout	Maximum time allowed for acknowledgment
AutoAckTime	Automatically acknowledge after specified time

Example flow:

```
Alarm Triggered  Wait for Ack  AutoAckTime expires  Auto-acknowledged

         Manual Ack  Acknowledged
```

Notification Configuration

Sound Alerts

- Configure via **Sound** property
- Plays continuously until acknowledged
- Different sounds for different severities
- See [Alarm Groups Sounds] for audio file setup

Visual Indicators

- **Colors** property defines alarm states
- Applied in alarm displays
- Consistent across all items in group

Custom Notifications

NotificationMethod calls Script Class methods:

```
// In Script Class

public void SendCriticalAlert(AlarmEventArgs args)
{
    // Send email
    EmailService.Send(args.Message, args.Priority);

    // Send SMS
    SMSService.Alert(args.TagName, args.Value);

    // Log to external system
    ExternalLogger.LogAlarm(args);
}
```

Configure: `NotificationMethod = "Script.Class.ServerMain.SendCriticalAlert"`

Webhook and MQTT Notifications

From FrameworX 10.1.6, NotificationMethod also accepts a webhook or MQTT broker URI, for example `https://alerts.example.com/fx/alarms` or `mqtts://broker.example.com/plant1/alarms`. The runtime then sends the alarm events of the group to the target with no script, and keeps them in a store-and-forward buffer while the target is unreachable. For the URI formats, the message content and the delivery rules, see Alarms Notifications Reference.

Deadband and Timing

ActiveTimeDeadband

Prevents alarm flooding during startup or known conditions:

- Ignores alarms for specified seconds
- Useful during system initialization
- Prevents nuisance alarms

Time-Based Acknowledgment

```
AckTimeout = 300  // Must acknowledge within 5 minutes
AutoAckTime = 600 // Auto-acknowledge after 10 minutes if not handled
```

Runtime Access

Access group properties at runtime:

```
// Check if acknowledgment required
bool needsAck = @Alarm.Group.Critical.AckRequired;

// Get active alarm count for group
int criticalCount = @Alarm.Group.Critical.TotalCount;

// Disable group (set Disable to 0 to re-enable)
@Alarm.Group.Warning.Disable = 1;
```

Acknowledge All Alarms in a Group

`AckAll` on a group is a change-triggered command property, not a method. Each change of value acknowledges every unacknowledged alarm item in the group. Writing the same value again does nothing, so a button that always writes 1 acknowledges only the first time. To acknowledge again, change the value once: use the Toggle action (ActionType ToggleValue), or write the current value + 1. Setting the value to 0 and back to 1 in the same script does not reliably acknowledge. This behavior applies to FrameworX 10.1.5 and 10.1.6.

```
// Acknowledge every unacknowledged alarm in the Critical group
@Alarm.Group.Critical.AckAll = @Alarm.Group.Critical.AckAll + 1;
```

Common Configurations

Critical Process Alarms

```
Name: ProcessCritical
AckRequired: True
Sound: Critical.wav
LogEvents: True
AutoAckTime: 0 (never)
Colors: Red
NotificationMethod: Script.Class.Notifications.SendCritical
```

Informational Logging

```
Name: InfoLog
AckRequired: False
Show: False
LogEvents: True
Category: Audit
```

Maintenance Alarms

```
Name: Maintenance
AckRequired: True
AutoAckTime: 3600 (1 hour)
LogEvents: True
NotificationMethod: Script.Class.Maintenance.ScheduleWork
```

Best Practices Checklist

- **Group by severity** - Critical, Warning, Information
- **Consistent sounds** - Same severity = same sound
- **Clear naming** - Group name indicates purpose
- **Document settings** - Use Description field
- **Test notifications** - Verify all alert methods work
- **Set appropriate timeouts** - Balance responsiveness vs alarm fatigue
- **Use categories** - Organize groups logically

Troubleshooting

Alarms not showing:

- Check **Show** property is True
- Verify display filter includes group
- Confirm AlarmItems assigned to group

No sound playing:

- Verify sound file path correct
- Check audio system enabled
- Confirm AckRequired if sound should repeat

Notifications not sent:

- Verify NotificationMethod syntax
- Check Script Class method exists
- Review method error logs
- For a webhook or MQTT target (from FrameworX 10.1.6), read `Alarm.LastErrorMessage`. It says when events were buffered and why.

Auto-acknowledge not working:

- Confirm AutoAckTime > 0
- Check AckRequired is True
- Verify no manual ack occurred

In this section...
