Alarms Items Reference

Define alarm conditions and settings.

Alarm Items (Reference) define individual tags or expressions monitored for alarm conditions within your FrameworX solution. Alarm Items are the fundamental alarm objects that:

- Monitor tag values against configured conditions
- Trigger alarm events when conditions are met
- Associate with AlarmGroups for shared behaviors
- Organize within AlarmAreas for hierarchical structure
- Support multiple limit values for shift-based operations

Each Alarm Item represents a single monitored condition that generates alarms based on your defined thresholds and settings.

Two Alarm Items on the same tag and condition are not supported, because the tag keeps one link per condition: both rows raise alarms, but `Tag.<name>.<Condition>` and the saved limits follow only one of them.

Configuration Properties
Alarm Conditions
Creating Alarm Items
Shift-Based Limits
Deadband Configuration
Runtime Properties
Best Practices Checklist
Troubleshooting

Configuration Properties

Property	Description	Required
TagName	The tag being monitored for alarm events	Yes
Condition	Trigger condition (Hi, Lo, RateOfChange, etc.)	Yes
Limit	Primary threshold value	Yes
Limit1	Secondary threshold (for shift-based limits)	No
Limit2	Tertiary threshold (for shift-based limits)	No
Group	Associated AlarmGroup for shared properties	Yes
Area	AlarmArea for hierarchical organization	No
Priority	Alarm priority level	No
Message	Text displayed when alarm triggers	No
Disable	Enable/disable the alarm	No
Deadband	Range where changes are ignored	No
Setpoint	Target value for deviation alarms	No
SetpointDeadband	Range around setpoint to ignore	No
AuxValue	Additional information field	No
AuxValue2	Secondary auxiliary value	No
AuxValue3	Tertiary auxiliary value	No
Comment	Notes about the alarm item	No

Alarm Conditions

Condition	Description	Formula
Hi	High alarm	Tag Limit
HiHi	High-high alarm (auto-acks Hi)	Tag Limit
Lo	Low alarm	Tag Limit

LoLo	Low-low alarm (auto-acks Lo)	Tag Limit
RateOfChange	Rate exceeds limit	Tag/t Limit (s)
DeviationMinor	Minor deviation from setpoint	|Tag - Setpoint| > Limit
DeviationMajor	Major deviation from setpoint	|Tag - Setpoint| > Limit
Equal	Exact match	Tag = Limit
GreaterThan	Greater than	Tag > Limit
GreaterEqual	Greater or equal	Tag Limit
LessThan	Less than	Tag < Limit
LessEqual	Less or equal	Tag Limit
Changed	Any change	Tag value changed
ChangedUp	Value increased	Tag value increased
ChangedDown	Value decreased	Tag value decreased
NotEqual	Not equal	Tag Limit



Alarms will be generated even when the quality differs from 192 (good).

Creating Alarm Items

1. Navigate to **Alarms Items**
2. Enter **TagName** or select from tag list
3. Select **Condition** type
4. Set **Limit** threshold value
5. Choose **AlarmGroup** for behavior
6. Press **Enter** to save

Example configuration:

```
TagName: Tank1.Temperature
Condition: HiHi
Limit: 95
Group: Critical
Message: "Tank 1 Critical Temperature"
Priority: 100
```

Shift-Based Limits

Use multiple limit values for different operational periods:

Property	Usage
Limit	Default/primary threshold
Limit1	Secondary threshold (e.g., day shift)
Limit2	Tertiary threshold (e.g., night shift)

Enable via **Alarms Global Settings Enable LimitsByShift**

Deadband Configuration

Prevents alarm chattering:

Value Deadband:

- Alarm triggers at Limit
- Must drop below (Limit - Deadband) to clear
- Must exceed (Limit + Deadband) to re-trigger

Setpoint Deadband:

- Applied around setpoint for deviation alarms
- Prevents minor fluctuations from triggering

Runtime Properties

Access alarm item state at runtime via `@Alarm.Item.[ItemName]`:

```
// Check alarm state
bool active = @Alarm.Item.HighTemp.Alarm == 1;
DateTime activeTime = @Alarm.Item.HighTemp.ActiveTime;

// Get current values
string currentValue = @Alarm.Item.HighTemp.LastValue;
double limitValue = @Alarm.Item.HighTemp.Limit;
```

Best Practices Checklist

- **Use meaningful names** - TagName should clearly identify the monitored point
- **Set appropriate deadbands** - Prevent alarm flooding from oscillating values
- **Configure priority levels** - Critical alarms get higher priority
- **Write descriptive messages** - Include context and severity in alarm text
- **Group related alarms** - Use AlarmGroups for common behaviors
- **Document with comments** - Use Comment field for maintenance notes
- **Test conditions** - Verify alarms trigger at expected values

---

Troubleshooting

Alarm not triggering:

- Verify TagName is valid
- Check Condition logic matches requirement
- Ensure not Disabled
- Confirm AlarmGroup is active

Excessive alarms:

- Increase Deadband value
- Review trigger Condition
- Check for tag oscillation
- Consider time deadband in AlarmGroup

Wrong threshold:

- Verify Limit value
- Check shift-based limits if enabled
- Confirm Setpoint for deviation alarms

In this section...