Alarm Namespace Reference

Access alarm data via scripts.

The Alarm Namespace in .NET allows for Alarm management through properties and methods, enabling access to specific alarm attributes and settings. Use **`@Alarm`** to reference the Alarm Namespace, followed by attribute accessors like `.Item`, `.Group`, `.Area`, or `.GlobalSettings` for detailed control.

Methods such as `.GetItemList` or `.GetChildrenAreas` facilitate list retrieval and area categorization, while properties like `.TagName` or `.Sound` adjust individual alarm settings, enhancing real-time monitoring and condition evaluation capabilities.

Alarm Properties
Alarm Namespace
Acknowledge-All Commands
Alarm Priority Order

Alarm Properties

Access Alarms Namespace to see the properties of the alarm server. You can also find two more objects and its properties, they are as follow:

- The Alarm Group Class has the list of all defined Alarm Groups and their properties.
- The Alarm Item Class has all Alarm Items and their properties.

The tag properties are connected with the Alarms module. See the example below:

Example	
`Tag.Tagname.Hi.Limit`	Configuration and runtime status of the HI Limit condition for the tag. The naming of all tag properties follow this same method.

To learn about the basic concepts of namespaces and objects, you can refer to Unified Namespace.

Alarm Namespace

The Alarm Namespace in .NET offers a structured approach to alarm management, leveraging dot notation for intuitive access to alarm properties and methods. Starting with **`@Alarm`** to specify the Alarm namespace context, dot notation enables navigation through alarm attributes such as `.Item` for individual alarms, `.Group` for categorized alarms, `.Area` for location-based alarms, and `.GlobalSettings` for module-wide configurations. This notation facilitates the retrieval of specific alarm details, manipulation of settings, and execution of methods like `.GetItemList` for monitoring and managing alarms in real-time applications. It exposes properties and methods from the .NET objects used by the Alarms runtime. You can use these properties and methods on your Displays or to create Scripts.

Alarm Item
`@Alarm.Item.ID0.TagName`

`@Alarm` refers to the Alarm Namespace.

`.Item` accesses the specific item (or alarm point) attributes.

`.ID0` specifies the first item or a unique identifier for a specific alarm item in a list or collection of alarms.

`.TagName` is the property that holds the name of the tag associated with the alarm item, such as a sensor name, a machine part, or any identifier used within the system.

In code, this retrieves the tag name of the first alarm item, which is a string.

Alarm Group
`@Alarm.Group.Warning.Sound`

`@Alarm` refers to the Alarm Namespace.

`.Group` denotes a specific group of alarms or a categorization within the alarm system.

`.Warning` is the selected group of alarms classified as warnings.

`.Sound` refers to the sound property for Warning alarms, indicating what sound to play when a warning alarm is triggered.

This line of code is used to retrieve or set the sound that should be played when a warning-level alarm occurs.

Alarm Area

```
@Alarm.Area.Brewery.GetChildrenAreas
```

`@Alarm` refers to the Alarm Namespace.

`.Area` represents a logical or physical area within the environment being monitored.

`.Brewery` specifies a specific area named 'Brewery', in a facility that produces beer.

`.GetChildrenAreas` is a method that returns a list of areas within the Brewery area.

When executed, this code provides a collection of all the sub-areas or zones within the 'Brewery' area.

Audit Trail

```
@Alarm.GlobalSettings.AuditTrail.TagChanges
```

`@Alarm` refers to the Alarm Namespace.

`.GlobalSettings` accesses the global settings of the alarm module.

`.AuditTrail` refers to a logging or history-tracking feature within the alarm system.

`.TagChanges` indicates a specific setting or log that records all changes made to tag properties within the alarm system.

This is used to review or activate the tracking of all changes made to alarm tags across the entire system for auditing purposes.

Alarm GetItemList

```
@Alarm.GetItemList
```

`@Alarm` refers to the Alarm Namespace.

`.GetItemList` is a method that returns a list of all the items (alarms, sensors, etc.) that are being monitored by the alarm module.

Acknowledge-All Commands

The acknowledge-all commands are change-triggered properties, not methods. Each change of value acknowledges. 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 All

```
// Every alarm item in every group
@Alarm.AckAll = @Alarm.AckAll + 1;

// Every alarm item in one group
@Alarm.Group.Critical.AckAll = @Alarm.Group.Critical.AckAll + 1;

// An area and all its child areas
@Alarm.Area.Production.AckAll = @Alarm.Area.Production.AckAll + 1;

// Only the items assigned directly to the area
@Alarm.Area.Production.AckAllLocal = @Alarm.Area.Production.AckAllLocal + 1;
```

Alarm Priority Order

Each alarm item has a numeric Priority. `@Alarm.GlobalSettings.PriorityOrder` sets which direction ranks as more important. The Alarm module uses it to pick the top alarm: `@Alarm.PriorityItem`, and the `PriorityItem` of each group and area.

Value	Meaning
0 (default)	A higher Priority number is more important. Priority 10 outranks priority 1.
1	A lower Priority number is more important. Priority 1 outranks priority 10.

The setting is not on the Alarms > Global Settings page and is not stored in the solution. It starts at 0 on every run. To use 1, set it from a Script Task that runs at server startup. Changing it while the solution runs updates the top alarm right away.

Make priority 1 the most critical

```
@Alarm.GlobalSettings.PriorityOrder = 1;
```

In this section...
