Alarms Notifications Reference

Send alarm events to a webhook or an MQTT broker.

Alarm Notifications (Reference) describes how an alarm group sends its alarm events to an external system. *Available from FrameworX 10.1.6.*

From FrameworX 10.1.6, the **NotificationMethod** column of an alarm group accepts one of two kinds of target:

- A Script Class method, as in earlier versions.
- A webhook or MQTT broker address, written as a URI starting with `http://`, `https://`, `mqtt://` or `mqtts://`.

With a URI target, the runtime sends the events of the group directly, with no script. Events wait in a store-and-forward buffer while the target is unreachable and go out automatically once it responds again, also after a runtime restart.

In 10.1.5, NotificationMethod accepts a Script Class method only.

Configuring a Notification Target
Store-and-Forward
Monitoring Delivery
Troubleshooting

Configuring a Notification Target

1. Open **Alarms** **Groups**.
2. In the **NotificationMethod** column of the group, enter the target URI.
3. Run the solution. Every alarm item of the group sends its events to the target.

Each group has one NotificationMethod value. Leave it empty to send no notifications. To send the events of different alarms to different targets, place the alarm items in different groups.

Target	Format	Example
Script Class method	`Class.Method`	`Notifications.SendCritical`
Webhook	`http://host[:port]/path` or `https://host[:port]/path`	`https://alerts.example.com/fx/alarms`
MQTT broker	`mqtt://[user:password@]host[:port]/topic`	`mqtt://broker.local:1883/plant1/alarms`
MQTT broker over TLS	`mqtts://[user:password@]host[:port]/topic`	`mqtts://broker.example.com/plant1/alarms`

Events Sent

An alarm item of the group sends an event when it becomes active, when it returns to normal and when it is acknowledged. The runtime collects the events per target and sends them in batches, about once per second.

Message Format

Webhook and MQTT targets receive the same JSON message:

```
{
  "idempotencyKey": "<key>",
  "events": [ { ... }, { ... } ]
}
```

- **events** holds one entry per alarm event, with the alarm fields such as the tag name, message, group, area, priority, state and timestamps.
- **idempotencyKey** identifies the batch. A batch sent again after an outage carries the same key, so the receiver uses the key to discard a batch it already processed.

Webhook Targets

- The runtime sends an HTTP POST with content type `application/json`.
- The request also carries the key in the `Idempotency-Key` header.
- Any 2xx response counts as delivered.
- An HTTP 5xx, 408 or 429 response, a connection error or a timeout puts the batch in the store-and-forward buffer for a later resend.
- Any other 4xx response means the request itself is wrong, so the runtime does not resend the batch. The Alarm module's last error message shows the HTTP code. Correct the target address or the receiver configuration.

MQTT Targets

- The URI path is the MQTT topic, and it is required. For example, `mqtt://broker.local/plant1/alarms` publishes to the topic `plant1/alarms`.
- Default ports are 1883 for `mqtt://` and 8883 for `mqtts://`.
- Messages are published with QoS 1 (at least once), without the retain flag.
- For a broker requiring credentials, place them before the host: `mqtt://user:password@host/topic`. A user name with no password, for example an access token, is written as `mqtt://token@host/topic`. Write special characters of the user name or password as URL escapes, for example `%40` for `@`.

Certificate Trust for mqtts

For an `mqtts://` target, the broker certificate must be trusted on the computer running the runtime. When the broker presents a certificate the computer does not trust, the runtime refuses the connection. The Alarm module's last error message names the broker, the failed policy check, and the certificate subject and issuer, so you add the missing trust. The events stay in the buffer and go out once the certificate chain is trusted.

Windows shares one TLS session per host name. The runtime does not distinguish a trusted and an untrusted TLS port on the same host name.

Store-and-Forward

When a webhook or MQTT target is unreachable, the runtime buffers the events and sends them again automatically:

- Buffered events are kept on disk, in a file with the extension `.dbAlarmEgressStFw`, in the same folder the runtime uses for the alarm database store-and-forward files.
- The runtime retries the target on its own, starting 5 seconds after the failure and doubling the wait up to 5 minutes between attempts.
- Events go out in the order they happened. While a backlog exists, new events queue behind it.
- After a runtime restart, buffered events resume sending on their own as soon as the target is reachable. No new alarm event is needed.
- A resent batch carries the same idempotency key as the first attempt.
- The buffer holds up to 100,000 batches. When it is full, the oldest batch is dropped to make room for the new one.

Monitoring Delivery

The property `Alarm.LastErrorMessage` shows the most recent error of the Alarm module. While notifications are not delivered live, it says the events were buffered and why, for example:

```
Alarm events to 'https://alerts.example.com/fx/alarms' were buffered for automatic resend: Webhook endpoint
returned HTTP 503 (Service Unavailable) at <local time>
```

The runtime adds the local time of the error at the end of the message. Credentials in the target address are masked in this message.

Troubleshooting

Symptom	Check
No message arrives	Confirm the alarm item belongs to the group with the NotificationMethod target, and read `Alarm.LastErrorMessage`.
Message says the events were buffered	The target is unreachable or answered with a temporary error. Fix the connection. The buffered events go out on their own.
mqtts connection refused	Install the broker certificate chain as trusted on the runtime computer.
Webhook returns HTTP 4xx	Correct the URI path or the receiver configuration. These batches are not resent.
MQTT target rejected without a topic	Add the topic as the URI path, for example `mqtt://broker:1883/alarms/events`.

For Script Class methods as NotificationMethod, see Alarms Groups Reference and AlarmGroup Notifications Code.

In this section...
