

Shifts and notifications

27 September 2026

Two separate things that work together:

- A **shift** says what patrolling is expected at a site: when, how often, how long a round takes. The system compares the guards' clockings with it and writes what happened — patrol started, completed, missed points, failed — into the site's **occurrence book (eOB)**, where the control room sees it. It can also make the site's units show and sound alerts.
- A **notification** says who is told about what, by email, Telegram or a post to the customer's own system (a webhook). Shift events only reach a person through a notification (*Shift Notifications*, below). Other notifications don't need a shift at all: a unit that stops reporting, a point that is clocked, a report every morning.

Written for whoever sets these up for a customer, and for whoever has to work out why an alert did or didn't arrive. Code references are for when you need to look deeper; `docs/NOTIFICATIONS.md` is the technical reference for the notification engine.

Part 1 — Shifts

What a shift is

A shift belongs to one **site**. It patrols **every group directly inside that site** — normally the site's points. Adding or removing a point changes what the shift patrols straight away; nothing is stored per shift.

Keep only points directly inside a site that has a shift. Anything else there (an "Unknown ..." group, a guard) also counts as a point to patrol, and since it is never clocked the patrol can never be complete.

A site can have more than one shift, for example a day shift and a night shift.

Setting one up

Quick setup → **Shifts** (`/quicksetupshifts` , permission `menu_shifts_group`) lists every shift on the sites you can reach, with an **Add** form. Click a shift to edit it (`/editshift/<id>`).

Times are entered in your own timezone. Every time is in **minutes** unless it says otherwise.

Field	What it does
Shift name	Shown in the eOB and on reports.

Field	What it does
Site	The site whose points are patrolled.
Start time / End time	When the shift runs. Leave both empty for a shift that never ends (24 hours a day). A shift may run over midnight (e.g. 18:00–06:00).
Round time	How long a patrol round may take. 0 means no rounds — the shift only watches <i>Max time between points</i> (or just the ack timer).
Rest time	Time between rounds, after a round ends. Empty for no rest: the next round starts straight away.
Fastest round time	A round completed quicker than this is logged as <i>Patrol too fast</i> .
Max time between points	If this long passes with no point clocked, <i>Too long between points</i> is logged.
Max time before first point	If no point is clocked this long after a round starts, <i>Patrol start late</i> is logged.
% Leeway	The share of the site's points that counts as a complete patrol. 100 means every point.
Days of week	All, weekdays, weekends or one day.
Day of month	Only run on that day of the month (0 for every day). See <i>Things to know</i> .
Fixed timing	On (the default): rounds stay on a fixed timetable, e.g. every hour on the hour; finishing early doesn't start the next round early. Off: the next round (or rest) starts as soon as a round is complete.
Nominal ack time / Random variance	The ack timer — see below. 0 turns it off.
Guard acknowledges alerts	The site's units ask the guard to acknowledge each alert.
Signal types	Which shift events the site's units show and sound (see <i>Alerts on the site's units</i>).

A **new** shift waits for its next start time before it begins. **Saving** an existing shift restarts it from now if it is inside its hours.

What happens during a shift

A shift moves between four states: **offline** (outside its hours), **round time** (a patrol should be happening), **rest time**, and **in shift** (on duty with no rounds — round time 0). Each change, and each problem, is logged as an eOB entry for the site:

Event	When	Priority
Shift start	the start time is reached	–
Patrol start	a round begins	–
Patrol start late	no point clocked within <i>Max time before first point</i>	2 (red)

Event	When	Priority
Too long between points	no point clocked within <i>Max time between points</i>	2 (red)
Patrol complete	enough points clocked (the leeway), not too fast	–
Patrol too fast	complete, but quicker than <i>Fastest round time</i>	1 (yellow)
Patrol end	the round's time is up (or it was completed, with fixed timing off)	–
Incomplete patrol	the round ended below the leeway, and there is rest time	1 (yellow)
Missed point	the round ended below the leeway (but at least half of it)	1 (yellow)
Failed to patrol	the round ended below half the leeway	3 (red)
Rest time start / end	around the rest period	–
Ack timeout	the guard didn't acknowledge the ack timer in time	1 (red)
Shift end	the end time is reached	–

Priority events (1–3) are left **unacknowledged** in the eOB until someone acknowledges them; the others are logged as already seen. The same priority event is not logged twice within 5 minutes.

Progress is the percentage of the site's points clocked since the round started. It is shown in the control-room screens and on the GuardSense page after a scan.

The ack timer

For a site with a **Stationary Guard** or **Staff Management Unit**: the unit asks the guard to press acknowledge within a time. *Nominal ack time* $\pm$ half the *Random variance* (in minutes) picks a new random time each time, so the guard can't predict it. It is sent to the site's units at shift start, at every patrol start, and after each clocking. If the guard doesn't respond, the unit reports it, *Ack timeout* is logged, and a new timer is sent.

Alerts on the site's units

The **Signal types** ticked on the shift decide which events the site's controllable units (Stationary Guard, SMU) sound. Every event is shown on the unit's screen in its colour with the guard text (e.g. *Patrol has failed*) and what's next (e.g. *Round end 14:00*); only the ticked ones make a noise.

The MissQue — how a shift keeps time

A shift has to act at times when nothing else happens: a round ends, a guard is late, the shift starts. The **MissQue** (`MissQue` table) holds, for each shift, **the next minute it needs looking at**. Every minute the server's `/check` tick looks for MissQue rows due **that exact minute** and runs the shift's `Shift.next()`, which updates the state, logs the events, and writes the next minute back. A clocking on the site also runs `next()` straight away and replaces the row.

The MissQue also drives the *Missed Clockings*, *Time Between* and *Missed Signals* notifications (Part 2): each has a row saying when to check next.

When a minute is missed. If `/check` doesn't run for a while (a restart, the server down), rows fall due without being looked at. The next tick picks up **everything that is due or overdue**, oldest first:

- an overdue **notification** check (more than 5 minutes late) is **rescheduled without sending** — it missed its moment, and clockings sent while the server was down may still be arriving. It alerts again only if the next check is missed too;
- an overdue **shift** is run the way saving it does, so a shift whose start minute was missed still starts if it is inside its hours, and one in the middle of a round catches up;
- a row that fails with an error is logged and tried again an hour later, so it can't hold up the others.

(Before 25 Sep 2026 only rows due in that exact minute were looked at, so an overdue row was never picked up again and its shift or notification went quiet for good.)

To look at the queue: the site's shift shows its next check time (or *Missing from MissQue*) on the control-room screens and the GuardSense page; **Admin** → **Debug Info** (`/debuginfo`) lists every MissQue row; **View shift debug** on the shift's edit page (`/report/debugshift/<id>`) shows each run of `next()` — the state it was in, what it decided, the next time it set. Saving a shift still rebuilds its row.

Things to know

- A shift starts in its **start minute**, when it is saved inside its hours, or when its missed start is caught up (above). *Shift end* is only logged in the exact end minute.
 - **Days of week** and **day of month** both decide the next start; a day of month that never comes (the 31st of a short month) is given up on after 62 days.
 - A shift with no points on its site is taken offline.
 - Shifts are web-only: the API has no shift endpoints.
-

Part 2 — Notifications

How a notification is put together

Every notification is three things (`Notification` in `models.py`):

1. **A trigger** — *when*: a time, a clocking, a signal from a unit, something that *didn't* happen, an eOB entry.
2. **A message or report** — *what*: a message with `{{placeholders}}` , or a report (detailed clocking, overview, health check, guard evaluation, ...) sent as a PDF.
3. **A contact** — *who*: an email address, a Telegram chat, or a post contact (a URL the customer's own system receives JSON on; see `docs/POST-NOTIFICATIONS.md`). A cell number can't receive anything.

It can be limited to a number of sends (it switches itself off after the last), and switched on and off.

When a notification is created for a **Telegram** contact from the advanced page and most of the simple pages, that person gets a message saying they've been subscribed, with a link to unsubscribe. They can also see and stop them from the bot's ... menu (*List Subscriptions, Unsubscribe from all*).

Delivery. A notification is queued (`SendQue`) and sent by the background worker, usually within a minute. A contact that is sent many messages close together is held back: for email more than about 10 with under 5 minutes between each, for Telegram 30. On a dev machine only Super Users receive anything.

Every send is logged: Notifications → View sent (`/listnotificationssent`) shows what was sent to whom and the answer from the mail server or Telegram. From there you can **preview** a message or report, **resend** one, or resend everything that was held back.

Which page for what

The pages below create the common kinds with a simple form. **Notifications → List all** (`/listnotifications`) shows every notification you can reach, whichever page made it; from there you can edit, switch off, duplicate to another contact, or delete, and open the **advanced** page for anything else.

I want to be told...	Page	Trigger
when a unit is tampered with, its battery is low, or it stops reporting	Quick setup → Device Notifications	signal / missed signal
about a site's patrol events (failed patrol, late start...)	Quick setup → Shift Notifications	eOB entry
every time a point, guard or site is clocked	Notifications → Group Clockings	clocking
when a unit hasn't reported for a while	Notifications → Missed Signals	missed signal
when too few of a site's points are clocked in a patrol interval	Notifications → Missed Clockings	missed clocking
when a site goes too long without any clocking	Notifications → Time Between Clockings	time between
both of the last two, set up together	Notifications → Failed To Patrol	missed clocking + time between
a message at a set time (once or repeating)	Notifications → Reminders	fixed / repeating time
a report by email every day, week...	the report's own page → Automate	repeating time
anything else (any trigger with any report)	List all → New notification (advanced)	any

Device Notifications — tamper, battery low, not communicating

Quick setup → Device Notifications (`/quicksetupnotification`). Choose a contact; you get a list of units with three tick boxes each:

- **Tamper** — the unit reports a tamper signal.
- **Battery low** — the unit reports battery low.

- **Not communicating** — the unit hasn't been heard from for 3 hours (checked every 3 hours from 08:00). When it reports again, a one-off *"has spoken to the server again"* message follows.

Unticking a box switches that notification off; ticking it again switches it back on. Tamper and battery low depend on the unit reporting those signals; batons do.

Both Device and Shift Notifications need the `menu_quick_notifications` permission, and only offer contacts and units you manage.

Shift Notifications — a site's patrol events

Quick setup → **Shift Notifications** (`/shiftnotification`). Choose a contact; each site with a shift has two tick boxes: **All signals** (every eOB entry for the site) or **Priority only** (the ones needing attention — late, missed, failed, too fast, ack timeout). The message is the event's text with the site name. Each contact has their own setting per site; saving the page for one contact never touches anyone else's.

The advanced page can do the same for a group *above* the sites (e.g. a whole profile, "include everything below") and can limit it to one kind of event.

Group Clockings — every time something is clocked

Pick the group (a point, a guard, a site...) and a contact; each clocking sends a message. Placeholders:

`{{groupclocked}}`, `{{groupabove}}`, `{{lastuser}}`, `{{lastlocation}}`, `{{triggertime}}`.

Missed Signals — a unit has gone quiet

Pick the unit, a contact and after how many hours (1–24) to warn. Checked from the next whole hour, then every that-many hours. Tick **Notify me when next signal comes through** to also get a message when it reports again.

Missed Clockings — a patrol interval without enough points

Pick the site (or any group), a contact and:

- **Clocking interval** — how often the points should be clocked (minutes, hours or days).
- **Rest time** — extra grace added to the interval before warning.
- **Leeway percentage** — how many of the group's points must be clocked in that time (100 = all).
- **Start / End**, or **No start & end time (24 hours)** — when to watch.
- **Days** — which days to watch, in the contact's timezone (none picked means every day).

How it works: at the start time a check is queued for *interval + rest time* later. Each clocking inside the group counts the points clocked within that span; if at least the leeway percentage were, the check is pushed on another interval. If the check time arrives without that, the message is sent and the next check is queued.

Clockings only count from units that have the **automation** capability (a unit's status, set on **Bulk units** or the unit's page); the page warns about units that don't.

Time Between Clockings — the site has gone quiet

Like Missed Clockings but simpler: warn if the group goes longer than **Time between clockings** without two clockings.

Failed To Patrol

Creates a Missed Clockings notification and (with **Enable Time Between**) a Time Between one for the same group, contact and hours, in one form.

Reminders

A message you write, to one contact, either **once** at a date and time or **repeating** at a time (every day, one weekday, or one day of the month).

Reports on a schedule

On a report's page (Detailed clocking, Overview, In/Out, Guard evaluation, Stationary guard, Service interval...) choose the report's options, then **Automate** with a time: the same report is sent to the contact at that time every day, covering the period since it was last sent. **Notifications** → **Overview / Detailed Clocking Reports** list these for editing and duplicating.

The advanced page

List all → **New notification** (/notifications , permission menu_advanced_new_notification): step 1 choose any trigger and its settings, step 2 choose any message or report. Use it for combinations the simple pages don't offer — for example the guard evaluation report whenever a particular point is clocked.

Placeholders in messages

A message can use `{{name}}` placeholders. Which ones have a value depends on the trigger:

Placeholder	Filled with
<code>{{triggertime}}</code>	when it happened, in the contact's timezone
<code>{{groupname}}</code> , <code>{{groupclocked}}</code> , <code>{{groupabove}}</code>	the group, and the group above it
<code>{{lastuser}}</code> , <code>{{lastlocation}}</code>	who clocked and where
<code>{{description}}</code>	the eOB entry's text (eOB/shift notifications)
<code>{{unitserial}}</code> , <code>{{lastsite}}</code>	the unit and the last site it clocked
<code>{{signaldata}}</code> , <code>{{groupsbelongingtoo}}</code>	the unit's last signal values, and the groups it is linked to
<code>{{timebetween}}</code>	the missed-signal time
<code>{{groupsclocked}}</code> , <code>{{groupsnotclocked}}</code> , <code>{{numberofgroupsclocked}}</code> , <code>{{numberofgroupsnotclocked}}</code> , <code>{{percentageclocked}}</code>	for missed clockings: which points were and weren't clocked

When a notification doesn't arrive

1. **Is it enabled, and has it sends left?** List all.
2. **Was it sent?** View sent: if it's there, look at the answer (mail refused, Telegram blocked, held back by anti-spam — *resend suppressed*).
3. **Did the trigger happen?** For shift events, look at the site's eOB. For missed clockings / time between / missed signals, check its MissQueue row on Debug Info.
4. **Is the contact confirmed?** An unconfirmed Telegram contact receives nothing; see `LOGIN.md`.
5. **Clockings not counted?** The unit needs the automation capability (Missed Clockings / Time Between).
6. **On a dev machine?** Only Super Users receive.

Via the API?

- **Shifts can be read:** `GET /groups/<site>/shifts` shows a site's shifts, their state, when each is next checked and whether that check is overdue, and the recent patrol events. `GET /eob` gives each entry's priority. Creating or changing shifts is web-only.
- **Shift Notifications and Device Notifications can be set:** `GET /contacts`, then `POST /alerts` (`site_events`, `tamper`, `battery_low`, `not_communicating`) — see `docs/API.md`, *Alerts*. The other notification pages are web-only.