



Access Controller

End-user and installer manual

2plus1 · 2026-09-08 · Generated from manual/src

Contents

01 Introduction and concepts

02 Getting started

03 Doors

04 Alerts (Telegram)

05 Controllers and hardware

06 People and fobs

07 Access levels and schedules

08 Room bookings and the resident portal

09 Phone access (dial release)

10 Reception screen

11 Log and reporting

12 Users, roles and permissions

13 Settings reference

14 Multi-tenant / provider guide

15 Security and privacy

16 Installation and hosting

17 Troubleshooting and FAQ

18 Glossary

Introduction and concepts

Access Controller is a cloud-managed door access-control system made by 2plus1. It runs gyms, studios, offices, workshops, apartment blocks and holiday lets from one web panel: people and their fobs, doors and the small controllers that drive them, access levels and weekly schedules, release-by-phone, bookable rooms with their own resident portal, a live reception screen, a full event log and Telegram alerts when a door is forced or left open.

Its starting point is that you should not have to rip anything out. Access Controller is a **drop-in replacement for any Wiegand system**: the readers, locks, cabling and fobs already on site stay exactly where they are, and only the controller box changes.

This manual is written for three readers:

- **Owners and office managers** who add people, hand out fobs, set opening hours and look at the log. Chapters 2, 4, 6, 7, 8, 10 and 11 are for you.
- **Installers** who fit readers, locks and controllers and get them talking to the panel. Chapters 3, 5, 13 and 17 are for you.
- **Providers and IT** who host the panel, create tenants and look after security. Chapters 12, 14, 15 and 16 are for you.

A drop-in upgrade for existing door hardware

Most door systems already on the wall (Paxton, Kantech, Honeywell, ZKTeco, unbranded kits, old PC-based panels) use the same building blocks: a Wiegand reader on the outside of the door, a 12 V electric strike or maglock, an exit button on the inside and, sometimes, a door contact. Access Controller reuses all of that. You keep the readers, the locks, the cabling and the fobs already in people's pockets; only the old controller box is replaced by an Access Controller unit, and everything is then managed from the panel.

- Any Wiegand 26-bit or 34-bit reader works: proximity, MIFARE, keypad-with-fob, QR readers that output Wiegand.
- Existing fobs and cards keep working, because the code is read straight off the reader and stored as it arrives.
- Standard relay output for strikes and maglocks, exit-button and door-contact inputs, fail-safe or fail-secure per door.
- One-door and two-door controllers, PoE (wired) or Wi-Fi: one board can drive two independent doors, each with its own reader, relay, exit button and door contact.
- Sites can migrate door by door; every door keeps working on its own if the network drops.

The building blocks

The panel is organised around a small number of ideas. Once these are clear, every page in the panel makes sense.

TERM	WHAT IT MEANS
Person	Someone who may open doors: a member, a staff member, a resident, a cleaner. Has a name, optional email and phone, a photo, notes, and a status of active or inactive. Called "People" in the panel.
Fob	A physical credential (fob, card, tag) with a unique code read by the door reader. A person can hold several fobs. Each fob has a status of active or revoked. Codes are shown as uppercase hexadecimal, for example <code>A1B2C3</code> .
Door	A physical door. Has a name, a description, and a fail behaviour (fail-safe or fail-secure). A door is driven by one controller channel.
Controller	The small ESP32 unit fitted near the door that reads the reader, fires the lock relay, watches the exit button and door sensor, and talks to the panel. A single-door controller drives one door; a dual-door controller drives two (channel 1 and channel 2).
Schedule	A weekly pattern of when access is allowed: for each day of the week either no access, all day, or a time window (which may run overnight, such as 22:00 to 06:00).
Access level	A named bundle of doors plus one schedule, for example "Members: front entrance and gym floor, 06:00 to 22:00". A person holds one or more access levels. A level with no schedule is 24/7.
Tenant	One customer organisation with its own people, doors, controllers, settings, staff logins and branding. A single-site customer has one tenant. A provider hosting many customers has many.
User	A staff login for the panel (not to be confused with a person who opens doors). Every user has a role: super admin, admin, viewer, or a custom role.
Room	A door that has been made bookable: a residents' lounge, shared gym, laundry or meeting room. Residents reserve a slot on the resident portal and the door opens for them for that slot. A room needs a door module and an exit button; a fob reader is optional.
Booking	One person's reservation of a room for one slot. A confirmed booking is another reason the door may open, alongside a fob and a phone call.
Event	One line in the log: an access attempt (granted or denied with a reason), a door event (door opened, door closed, exit button pressed), an alert from a door (forced, held open) or a system event (controller registered, enrolment started, a person moved to the trash, a booking confirmed, an import).
Alert	An event the door raises on its own rather than in answer to a fob: <i>door forced</i> (opened with no grant) or <i>door held open</i> (open longer than the time set on the door). Alerts are always logged and can also be sent to a Telegram chat per tenant.

How a decision is made

When a fob is presented, the door grants access only if all four of these are true:

1. the fob is known and its status is active;
2. the person holding it is active;
3. one of the person's access levels includes this door; and
4. the current local day and time fall inside that level's schedule (or the level has no schedule).

If any test fails, the door stays locked and the log records why (unknown fob, fob revoked, person inactive, no access level for door, outside schedule). The same rule is used by the "Check access" tool on the dashboard, so you can test your set-up before anyone walks up to the door.

A **confirmed room booking** adds a fifth possibility: it gives that person's fobs a time window on that room's door, from a few minutes before the slot until just after it ends (chapter 8).

Other paths open doors without a fob at all. The **exit button** on the inside always releases the door (free egress). **Release by phone** lets a recognised caller, or a guest with an apartment number and PIN, open the door from the access line. The **resident portal** gives someone whose booking is running an "Open the door" button on their phone. All are logged. Where a door contact is fitted the controller also logs every opening and closing of the door, and raises an alert if the door is forced or held open.

How the controller keeps working offline

Access Controller is designed so that a broken internet connection never locks anyone out or leaves a door open.

- The panel compiles a **keyset** for every door: the complete list of fobs that may open that door, each with its allowed time windows. The keyset is pushed to the controller over MQTT as a *retained* message and the controller saves it to its own flash memory.

- The controller makes every grant or deny decision **locally** from that keyset. It does not ask the panel. A scan is answered in milliseconds whether or not the panel is reachable.
- Events that happen while offline are queued on the controller (up to about 100 KB, oldest dropped first) and sent to the panel when the link returns, so the log catches up.
- Whenever you change something that affects a door (add a fob, revoke one, change a level or schedule, set someone inactive) the panel pushes a fresh keyset within seconds. It also re-pushes every few hours so daylight-saving changes are covered.
- The controller keeps the clock over the network (SNTP) and can carry an optional battery-backed real-time clock. If the controller has no trusted time and a fob is time-restricted, the firmware fails open and logs it, in keeping with the product's fail-safe stance.

Revocation therefore takes effect within seconds while the controller is online, and no later than the next periodic push after an outage.

What "cloud-managed" means here

The panel is a single program that contains the web interface, the MQTT broker the controllers connect to, and the database. 2plus1 hosts it for you, or you can run it on your own server or a small cloud machine (see chapter 16). Controllers only need to reach the panel's address on the MQTT port; they never need inbound connections, and the panel never needs to reach into your site network.

Where things are in the panel

The left sidebar is the whole map of the product. Which entries you see depends on your role (chapter 12).

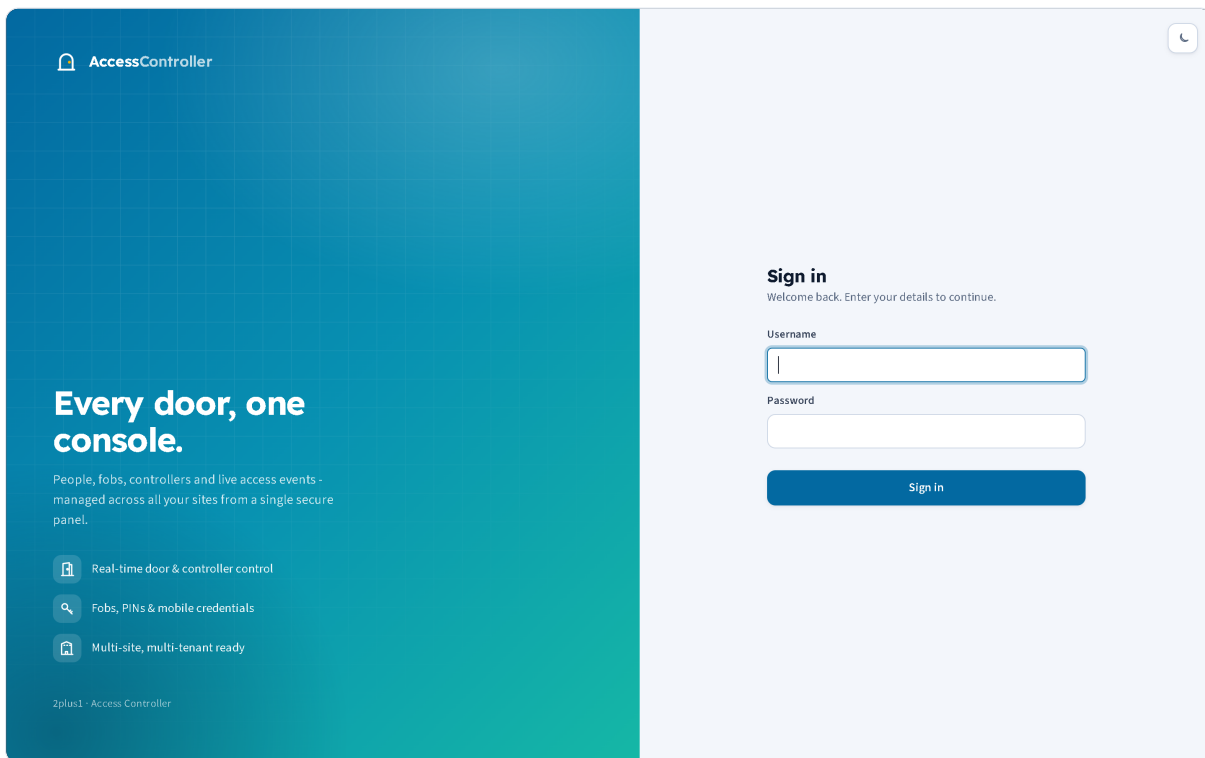
Sidebar Entry	What it is for	Chapter
Dashboard	Counts, controller health, recent activity, Check access tool	2
People	People, fobs, photos, access levels, phone release, bulk changes, the trash	6
Enrolment	Starting enrolment mode at chosen doors, assigning captured fobs, import from CSV or an AC8000 backup	6
Access levels	Which doors, on which schedule	7
Schedules	Weekly time windows	7
Bookings	Room bookings made by residents or by staff, and marking them paid	8
Rooms	Turning doors into bookable rooms: hours, slots, extras, price	8
Log	Every event, with filters	11
Doors	Doors, fail behaviour, reader/relay mapping, reception screen link	3, 10
Controllers	Onboarding, status, push keys, unlock, update, re-point	5
Flash & adopt	Flashing a new controller and sending it its settings	5
Users	Staff logins, roles	12
Branding	Logo and accent colour	2
Tenants	Provider only: create and switch between customers	14
Settings	Controller Wi-Fi, controller type, pin maps, bookings and the resident portal, Telegram alerts, the site number for phone enrolment, adoption address, phone API token	4, 8, 13

Getting started

This chapter walks a new user through their first session: signing in, reading the dashboard, finding your way round on a phone, switching between customers (providers only) and giving the panel your own branding.

Signing in

Open the panel address you were given (for a hosted service this is typically `https://doors.<provider-domain>`; for a self-hosted panel see chapter 16). Enter your username and password and press **Sign in**.

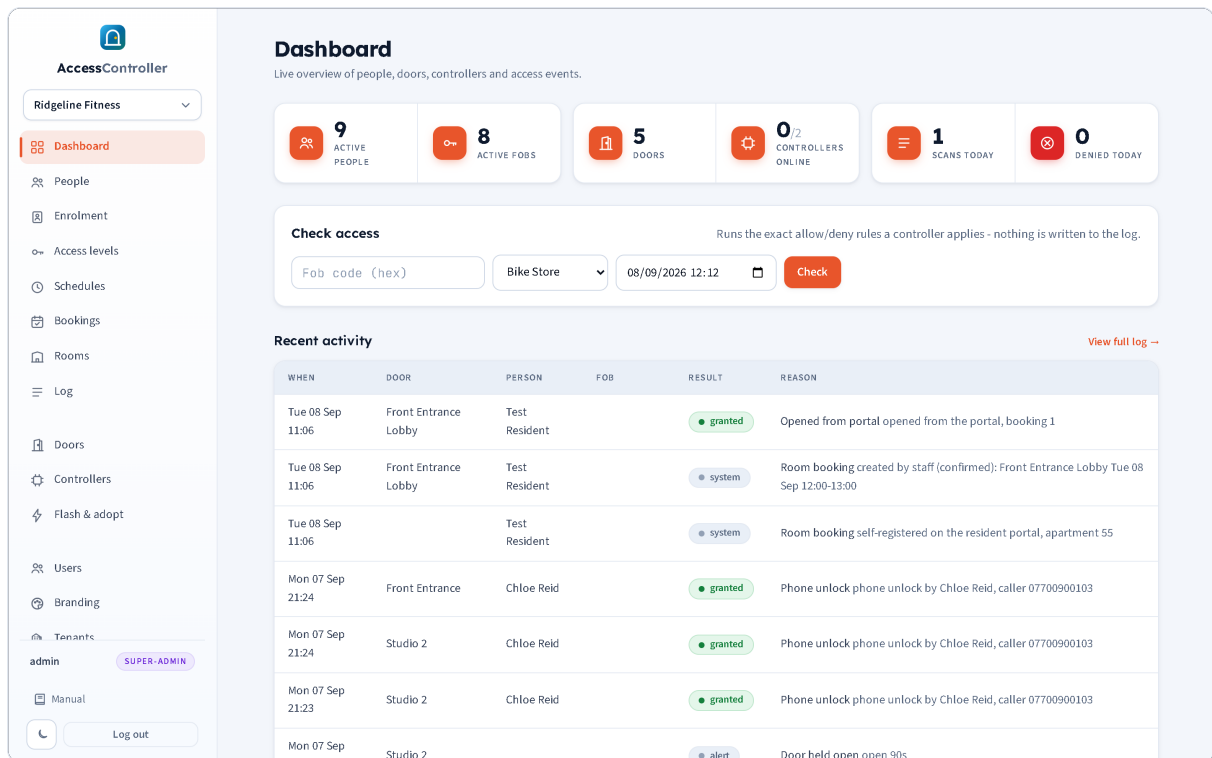


The sign-in page. The left panel carries the product branding; the form on the right takes a username and password.

- Your account was created either by the person who hosts the panel (the super admin) or by an admin of your organisation on the **Users** page. There is no self-registration and no "forgot password" link: an admin resets your password for you (chapter 12).
- A wrong username or password shows "Invalid username or password" and the panel deliberately pauses for half a second before answering, which makes password guessing slow.
- A session lasts seven days. **Log out** is at the bottom of the sidebar. Closing the browser does not end the session, so on a shared computer always log out.
- The sun/moon button in the sidebar footer (top-right on the sign-in page) switches between light and dark themes. The choice is remembered on that browser.

The dashboard

After signing in you land on the dashboard. It answers the question "is everything all right today?" at a glance.



The dashboard: three tile groups, the Check access tool and the most recent activity.

The top row has three tile groups. Each number is a link to the page it counts.

TILE	MEANING
Active people / Active fobs	People whose status is active, and fobs whose status is active and belong to someone in this tenant.
Doors / Controllers online	Number of doors, and how many controllers have reported in within the last 10 minutes out of the total registered. 2/2 is healthy; 1/2 means one controller has gone quiet.
Scans today / Denied today	Access events since midnight (panel timezone) and how many of them were denied. Door open/close and exit-button events are not counted. The denied tile turns red when the count is above zero and links straight to the filtered log.

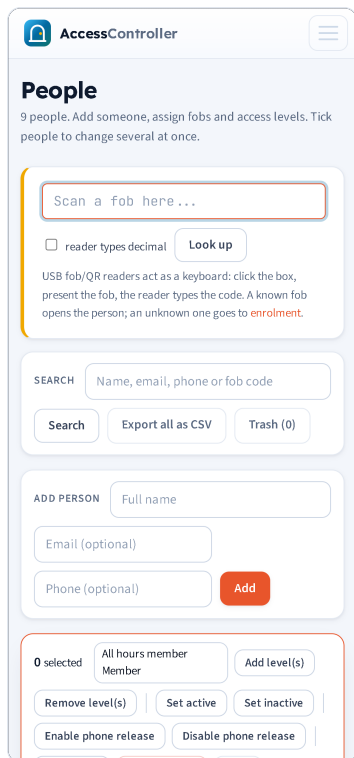
Below the tiles:

- **Captured fobs waiting** appears as an amber banner when fobs have been captured by enrolment and not yet assigned to a person (chapter 6). Click **assign** to deal with them.
- **Controllers awaiting onboarding** appears when a new controller has registered itself but has not yet been named, mapped to a door and set active (chapter 5).
- **Check access** runs the exact allow/deny rule a controller applies, for any fob code, door and date/time, without writing anything to the log. Use it to confirm a new person's set-up or to see why someone was refused at 05:55 yesterday. Enter the fob code (hexadecimal, as shown on the person's page), choose a door, adjust the time if you want and press **Check**. The result shows granted or denied, the person matched, and the reason in plain English.
- **Recent activity** lists the last 15 events. **View full log** opens the Log page with filters (chapter 11).

If enrolment mode is running for your organisation, a banner is shown at the top of every page with a **Stop** button (chapter 6).

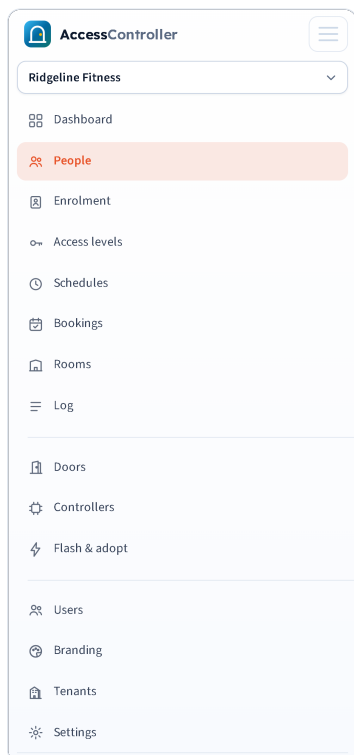
Using the panel on a phone

The panel is the same on a phone as on a desktop; the sidebar simply folds away. Below about 880 pixels wide, which covers every phone and most tablets held upright, the sidebar becomes a bar across the top with the logo on the left and a **menu button** (three lines) on the right.



The panel on a phone: the header bar with the logo and the menu button, and the page below it.

Tap the menu button and the whole sidebar drops down underneath it: the tenant switcher (or the tenant name), every page you have permission to see, and at the bottom your username and role, the light/dark toggle and **Log out**. Tapping any link closes the menu and takes you there.



The mobile menu open, showing the tenant name, the full list of pages and the footer with the dark-mode toggle and log out.

Wide tables scroll sideways inside their own panel rather than stretching the page, so the log and the people list stay readable. Everything an admin can do at a desk can be done from a phone, which matters most for the jobs you do standing at a door: assigning a captured fob, unlocking a door, checking why someone was refused.

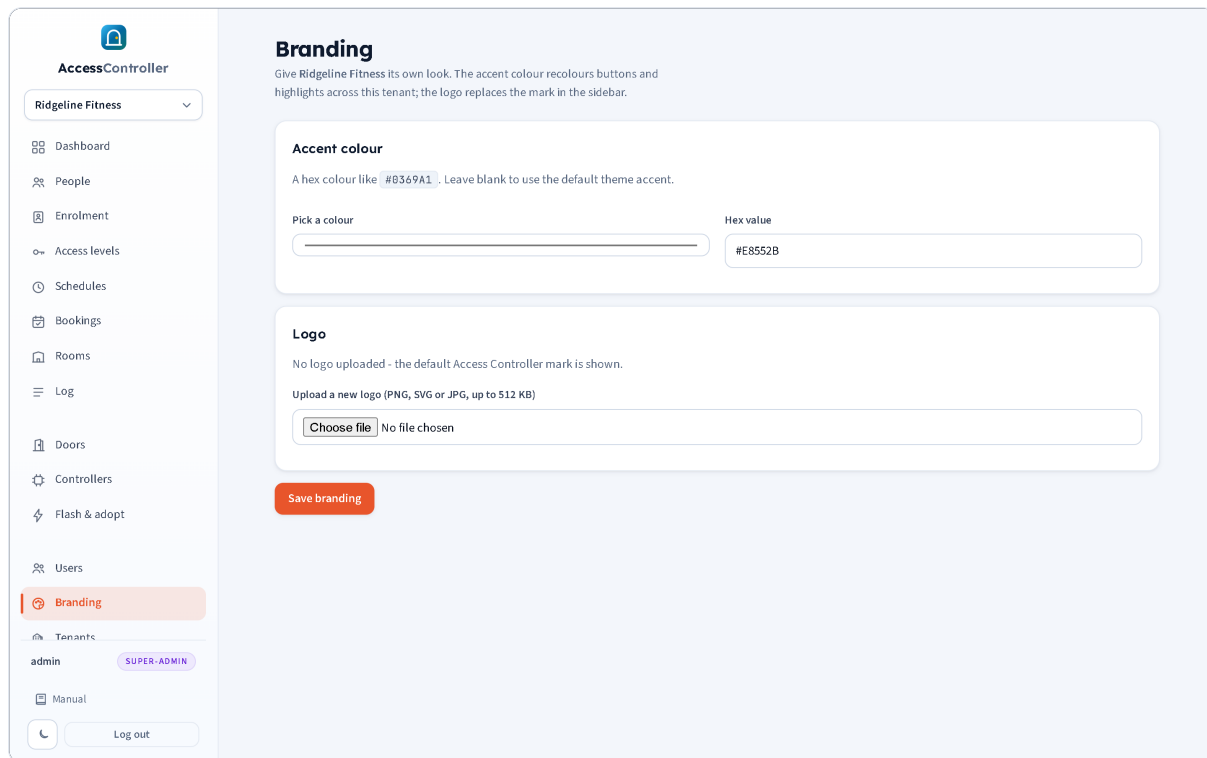
Switching tenants (providers)

If you are a super admin hosting several customers, the drop-down at the top of the sidebar lists every tenant. Choosing one "switches you in": the whole panel then shows only that tenant's people, doors, controllers, log, users and settings, and anything you add is created in that tenant. The provider's own tenant is marked "(provider)". The Tenants page (chapter 14) lists them all with a **Switch in** button per row.

Admins and viewers belong to one tenant and see its name in the sidebar instead of a drop-down; they cannot switch.

Branding

Each tenant can carry its own look. Open **Branding** in the sidebar.



The Branding page: an accent colour picker with hex value, and a logo upload.

- **Accent colour:** pick a colour or type a hex value such as `#0369A1`. It recolours the buttons and highlights throughout the panel for that tenant, and the reception screen uses it too. Leave it blank to return to the default blue.
- **Logo:** upload a PNG, SVG or JPG of up to 512 KB. It replaces the Access Controller mark at the top of the sidebar. Tick **Remove the current logo** and save to go back to the default mark.

Press **Save branding**. The change applies immediately for everyone in that tenant. Branding is a "change" permission on its own page, so admins and custom roles with the Branding tick can do this; viewers cannot.

A sensible order for a new site

If you are setting a site up from nothing, this order avoids back-tracking:

1. **Doors** (chapter 3): add each door by name and choose fail-safe or fail-secure.
2. **Settings** (chapter 13): enter the controller Wi-Fi (unless every controller is wired), the controller type and any non-default wiring.
3. **Flash & adopt and Controllers** (chapter 5): flash each controller, let it register, name it, map it to its door, set it active.
4. **Schedules** (chapter 7): create the weekly patterns you need, for example "Opening hours".
5. **Access levels** (chapter 7): bundle doors and a schedule into "Members", "Staff" and so on.
6. **People** (chapter 6): add people, attach fobs (by typing, scanning at the desk, or enrolling at a door), tick their access levels.
7. **Check access** on the dashboard: try a fob against a door before anyone tests it for real.
8. **Users** (chapter 12): give colleagues their own logins with the least permission they need.

9. Optional: **Reception screen** (chapter 10) and **Phone access** (chapter 9).
10. Optional: **Rooms** and **Bookings** (chapter 8) if the site has shared spaces residents should book, and then publish the resident portal under Settings.

Doors

A door in the panel is the thing people walk through. It has a name, an optional description, a fail behaviour, a held-open alert time, and a mapping to the controller channel that physically drives it. Everything else (who may open it and when) is decided by access levels, not on the door itself.

AccessController

Ridgeline Fitness

Dashboard

People

Enrolment

Access levels

Schedules

Bookings

Rooms

Log

Doors

Controllers

Flash & adopt

Users

Branding

Tenants

admin **SUPER-ADMIN**

Manual

Log out

Doors

Each door is opened by one controller; Fail behaviour is per-door - fire-escape routes normally must fail open; confirm against the building's fire requirements.

Reception screen

A full-screen page for a laptop/monitor that shows people's photos as they fob in. Open it on the screen, then tick which doors it watches. The link contains a secret token - anyone with it can view live entries, so share it carefully.

<https://localhost:8087/reception/838f38bb4d7dc11179900064> Copy Open screen Regenerate

ADD DOOR Fail-safe (unlocks on fault) ▾ Add

DOOR	DESCRIPTION	FAIL MODE	CONTROLLERS
Bike Store	Rear yard bike store	fail-secure	none assigned
Front Entrance	Main street door, fob or phone	fail-secure	Front door controller
Gym Floor	Inner door to the gym floor	fail-safe	Gym dual controller
Staff Room	Staff only	fail-secure	none assigned
Studio 2	Class studio, members in session hours	fail-safe	none assigned

The Doors page: the reception screen link panel, the add-door toolbar and the list of doors with fail mode and assigned controllers.

Adding a door

Open **Doors** in the sidebar. In the **Add door** toolbar:

1. Type a **Door name**, for example "Front Entrance", "Gym Floor", "Studio 2". Names are shown in the log, on the reception screen, in phone menus and in access levels, so keep them short and unambiguous.
2. Optionally add a **Description** such as "Main street door, maglock".
3. Choose the **fail mode**: fail-safe or fail-secure (see below).
4. Press **Add**.

The door appears in the list. The **Controllers** column shows which controller drives it, or "none assigned" until you map one (chapter 5).

Click a door's name to open its detail page.

The screenshot shows the 'AccessController' web interface. On the left is a sidebar with navigation links: Dashboard, People, Enrolment, Access levels, Schedules, Bookings, Rooms, Log, Doors (highlighted), Controllers, Flash & adopt, Users, Branding, Tenants, admin (with a 'SUPER-ADMIN' badge), and Manual. At the bottom of the sidebar are a 'Log out' button and a user icon.

The main content area is titled 'Studio 2' with a 'Back to doors' link. It contains several form fields:

- Name:** Studio 2
- Description:** Class studio, members in session hours
- Fail behaviour:** Fail-safe - unlocks on power/network fault (selected from a dropdown)
- Alert if held open for N seconds (0 = off):** 60

 Below these fields is a note: 'Needs a door position sensor wired to the controller. When the door stays open longer than this the controller raises a "door held open" alert in the log and, if configured under Settings, a Telegram message.' and a 'Save' button.

Below the main form is a 'Reader / relay mapping' section with the text: 'Which controller and channel drives this door. A dual-door controller has two channels; pick channel 2 for its second reader/relay.' It includes two dropdown menus: 'esp32-7b22e4 (Gym dual controller) - dual' and 'Channel 2', followed by a 'Save mapping' button.

At the bottom of the main content area is a 'Delete door' button.

A door's detail page: name, description, fail behaviour and held-open alert time, the reader/relay mapping panel, and the delete button.

Fail-safe or fail-secure

Every door records how its lock behaves when power or the controller fails:

- **Fail-safe:** the lock releases when power is lost. This is how a maglock works: it holds only while energised. The door is open in a fault.
- **Fail-secure:** the lock stays locked when power is lost. This is how most electric strikes work: the latch is held until the strike is energised. The door is closed in a fault.

Warning: fire escape routes. A door on a fire escape route normally must fail open so that people can always get out. Choosing fail-secure for such a door, or wiring a maglock without a break-glass or a fire-alarm interface, can trap people in a fire. The panel records your choice so that it is visible to everyone, but it cannot check your building. Confirm every door's fail behaviour against the building's fire risk assessment and the local fire authority's requirements before the site goes live, and keep the exit button and any break-glass unit on the inside of every fail-safe door.

The fail mode setting is a **record of intent**. The actual behaviour is fixed by how the installer wires the relay (normally-open versus normally-closed contact) and by the lock type. The controller's relay is de-energised at power-up and between pulses, so:

- for a **maglock** (fail-safe) the relay contact carries the lock supply and the lock drops when the relay pulses or the power fails;
- for a **strike** (fail-secure) the relay pulse energises the strike to release it.

The relay switches the lock's own power supply; it must never carry the controller's supply. Lock power should come from a separate, battery-backed supply. Fit a flyback diode across maglock coils. Chapter 5 has the wiring detail.

Held-open time and door pulse

Two timings matter for a door:

- **Relay pulse:** how long the lock is released after a granted scan, exit button or remote unlock. The default is 4000 ms (4 seconds). It is set per channel under **Settings > Channel 1 wiring** and **Channel 2 wiring** (chapter 13) and written to the controller when it is provisioned or re-pointed.
- **Held-open time:** how long the door may stay open before the controller raises a *door held open* alert. It is set per door on the door's detail page as **Alert if held open for N seconds (0 = off)**. The default is 60 seconds; **0** switches the held-open alert off for that door; the maximum is 3600. Press **Save** and the value is pushed to the door's controller within seconds inside the next keyset (it does not need re-provisioning), and the controller keeps it across power cuts. A door position sensor must be wired to the controller for this to work.

Note: the per-door held-open time is honoured by the current firmware. A controller running an older build ignores the value and keeps its compiled-in 60 seconds (or whatever `heldOpenMs` an installer set in its pin map over USB). Press **Update** on the Controllers page (chapter 5) to bring it up to date. On a dual-door controller the current firmware watches the channel 1 door only, so the held-open time on a channel 2 door has no effect yet.

After a granted scan the controller also allows a grace window (the pulse length plus 10 seconds) in which the door may open without being reported as *forced*. Opening the door outside any grant, exit press or remote unlock raises a *door forced* alert. Both alerts go to the Log and, if it is set up, to Telegram (chapter 4).

Door open and close logging

When a door position sensor is fitted, the controller reports every change of the contact: the Log shows a **Door opened** line when the door opens and a **Door closed** line when it shuts, each stamped with the controller's own time. Exit-button presses are reported in the same way as **Exit button**. These are *door* events (chapter 11): they are neither grants nor denials, so they are not counted in the dashboard's scan and denied totals and do not appear on the reception screen, and the Log has its own **Door open/close, exit button** filter for them. Together with the access rows they show who opened the door, how long it stood open and whether it closed again.

The controller ignores contact changes that follow another within 200 milliseconds, so a rattling reed switch does not produce a burst of open/close lines. Without a sensor (door pin set to `-1` in Settings) no open/close events are produced; without an exit button on the REX pin there are no exit-button events. Both need the current firmware.

Reader/relay mapping

The **Reader / relay mapping** panel on the door detail page says which controller and channel drives this door.

1. Pick the **controller** from the list. Controllers are shown by their ID (for example `esp32-3f9a1c`), their friendly name if set, and "- dual" if they are configured as dual-door.
2. Pick **Channel 1** or **Channel 2**. Single-door controllers only use channel 1. On a dual-door controller, channel 1 is the first reader and relay and channel 2 is the second set (chapter 13 has the pin maps).
3. Press **Save mapping**.

A door can be driven by only one channel. Mapping a door to channel 2 also marks that controller as dual-door. The same mapping can be made from the **Controllers** page (the **Door** drop-down there sets channel 1); the door page is where you set channel 2.

Choose "(none)" to unmap a door.

Editing and deleting a door

Change the name, description, fail behaviour or held-open alert time on the detail page and press **Save**. Renaming a door renames it everywhere, including in old log entries' door name where the name was stored at the time.

Delete door removes it. Access levels that referenced it lose that door, any controller mapped to it becomes unmapped, and if the door was a bookable room the room and its bookings go too (chapter 8). The log keeps the historic entries. You are asked to confirm.

Doors that are bookable rooms

Any door can be turned into a **room**: a residents' lounge, a shared gym, a laundry, a meeting room. Residents then reserve slots on the resident portal and the door opens for whoever has the slot. This is set up on the **Rooms** page and covered in chapter 8; nothing on the door's own page changes.

It is worth knowing at the wiring stage, because a room is the least demanding door on a site. It needs the lock relay and, ideally, a door position sensor, and it needs an **exit button** on the inside so that nobody is ever shut in. It does **not** need a fob reader: residents come in through the portal's "Open the door" button or by ringing the access line from the mobile they booked with. Fit a reader only if you also want people to fob in during their slot.

The reception screen link

At the top of the Doors page is the **Reception screen** panel: a link to a full-screen page for a laptop or wall monitor at the front desk that shows people's photos as they fob in. **Copy** copies the link, **Open screen** opens it in a new tab, and **Regenerate** replaces the secret token in the link so the old one stops working. Chapter 10 covers setting a screen up.

Alerts (Telegram)

The panel can send a Telegram message to your staff the moment a door is **forced** (opened without a fob, exit button or remote unlock) or **held open** longer than the time you set on the door. The message arrives on the phones of whoever is in the chat within a couple of seconds of the door reporting it, so a propped fire door or a forced back door is noticed while somebody can still do something about it.

Each tenant (each gym, site or customer) has its own bot and its own destination chat, so one site never sees another site's alerts. Every alert is always written to the **Log** (chapter 11) whether or not Telegram is set up; Telegram is a second copy that reaches people who are not looking at the panel.

What is alerted

ALERT	RAISED WHEN	MESSAGE LOOKS LIKE
Door forced	The door position sensor sees the door open with no grant, exit-button press or remote unlock in the last few seconds (the relay pulse plus 10 seconds).	DOOR FORCED: Anytown Gym: Front Entrance opened without a grant or exit press (controller esp32-3f9a1c)
Door held open	The door stays open longer than the Alert if held open for N seconds value on that door's page (chapter 3). 0 switches the alert off for that door.	Anytown Gym: Front Entrance open for over 60s (controller esp32-3f9a1c)

Both need a **door position sensor** (reed switch or door contact) wired to the controller; without one the controller cannot tell that the door is open, and neither alert is ever raised. Chapter 5 covers the wiring.

Nothing else is sent to Telegram in the current release: denied scans, offline controllers, failed firmware updates and system events stay in the Log only.

You need two things, both from Telegram itself:

1. A **bot token**: a long string that identifies your bot.
2. A **chat ID**: the number of the chat, group or channel the bot should post into.

Both are pasted into **Settings > Alerts (Telegram)** in the panel. The whole job takes about five minutes and needs no technical knowledge; follow the parts below in order.

Part A: create the bot with @BotFather

1. Open Telegram (phone or desktop) and, in the search box at the top, type `BotFather`. Pick the one with the blue verified tick and the name **BotFather**. Press **Start** if you see a Start button.
2. Type `/newbot` and send it.
3. BotFather asks for a **name**. This is the display name people will see, for example `Anytown Gym Doors`. Send it.
4. BotFather asks for a **username**. It must end in `_bot` and contain no spaces, for example `anytowngym_doors_bot`. If it says the username is taken, try another one. Send it.
5. BotFather replies with a message that starts "Done! Congratulations on your new bot" and contains a line like:

```
Use this token to access the HTTP API:
123456789:AAH4kZ9yq2f_ExampleExampleExample
```

The part after "HTTP API:" (numbers, a colon, then letters) is the **bot token**. Tap it once in Telegram to copy it.

Warning: keep the token private. Anyone with the token can send messages as your bot. Do not paste it into group chats or tickets; if it leaks, send `/revoke` to BotFather for that bot and paste the new token into the panel.

You can get the token again at any time by sending `/mybots` to BotFather, choosing your bot, then **API Token**.

Part B: get the chat ID

Telegram does not show chat IDs anywhere in its normal screens, which is why this step trips people up. There are three ways, easiest first.

Way 1 (recommended): the panel's Find my chat ID button

1. Decide where alerts should go:
 - **A private chat with one person:** that person opens Telegram, searches for the bot's username (for example `@anytowngym_doors_bot`), opens it and presses **Start**, then sends any message, for example `hello`.
 - **A group** (several staff): create the group (or use an existing one), add the bot as a member (Group info > Add member > search for the bot's username), then send any message in the group, for example `hello`.
 - **A channel** (announcement style, nobody replies): open the channel's info > Administrators > Add administrator > search for the bot's username and add it. Then post one message in the channel.
2. In the panel go to **Settings**, open the **Alerts (Telegram)** section, paste the bot token into **Bot token** and click **Find my chat ID**.
3. A table headed **Chats this bot can see** appears listing every chat the bot has noticed, with its name, type (private, group, supergroup or channel) and ID, most recent first. Click **Use this** next to the right one. The chat ID is saved and the panel confirms with "Chat ID ... saved - tick Enable, then Send test message".
4. If the panel says the bot has not seen any chats yet: send another message to the bot, or in the group or channel, wait a few seconds, and click **Find my chat ID** again. If it is a group, also read "Privacy mode" below.

Way 2: read it from the bot's update feed in a browser

1. Send a message to the bot (or in the group or channel it is in) as above.
2. In a web browser open this address, replacing `TOKEN` with your bot token (keep the word `bot` directly in front of the token, no space):

```
https://api.telegram.org/botTOKEN/getUpdates
```

For example `https://api.telegram.org/bot123456789:AAH4kZ9yq2f_Example/getUpdates`.

1. You will see a block of text. Look for `"chat":{"id":` and the number right after it is the chat ID, for example:

```
"chat":{"id":987654321,"first_name":"Sam","type":"private"}
```

or, for a group:

```
"chat":{"id":-1001234567890,"title":"Front desk","type":"supergroup"}
```

1. Copy that number (including the minus sign if there is one) into **Chat ID** in the panel.

If the page shows `{"ok":true,"result":[]}` the bot has not received anything yet: send a message and refresh the page.

Way 3: rules of thumb for groups and channels

- **Private chat:** a positive number, for example `987654321`.
- **Group:** add the bot to the group and send a message. Group IDs are **negative**. Normal groups look like `-123456789`; larger groups ("supergroups") look like `-1001234567890`. Telegram may upgrade a group to a supergroup when you change certain settings, which changes its ID: if alerts stop after you reorganised a group, run **Find my chat ID** again.
- **Channel:** the bot must be added as an **administrator** (a plain member of a channel cannot post). Post one message in the channel. The ID always starts with `-100`.

Privacy mode: why a bot in a group may see nothing

By default a bot added to a group only sees messages that are commands (starting with `/`) or that mention it. That means **Find my chat ID** and `getUpdates` may show nothing even though you have been chatting in the group. Two fixes; either works:

- **Make the bot a group administrator** (Group info > Administrators > Add administrator). Admin bots see every message.
- **Turn privacy mode off**: send `/mybots` to BotFather, pick the bot, choose **Bot Settings > Group Privacy > Turn off**. If the bot is already in the group you must remove it and add it again for the change to take effect.

Alternatively just type a command in the group, for example `/start`, which the bot always sees.

For sending alerts none of this matters: the bot can always post into a group it is a member of. Privacy mode only affects what it can *read*, which is only needed to discover the chat ID.

Part C: put it in the panel and test

Open **Settings** and expand the **Alerts (Telegram)** section (it shows a green **on** badge once alerts are enabled). Viewers cannot see Settings; admins and custom roles with the Settings permission can (chapter 12). A super admin sets it up per tenant after switching in.

FIELD	MEANING
Enable Telegram alerts	The master switch. Nothing is sent while it is unticked, whatever else is filled in.
Bot token	The token from Part A. Once saved, the field shows a green set badge and stays blank on purpose: the panel never shows the token again. Leave it blank to keep the stored token, or paste a new one to replace it. Until a token is saved the badge reads not set .
Forget the stored token	Appears once a token is saved. Tick it and save to erase the token from the panel (for example when the bot is retired or the token was leaked).
Chat ID	The destination from Part B: a positive number for a private chat, negative for groups and channels.
Door forced	Send a message when a door is forced. On by default.
Door held open	Send a message when a door is held open longer than its own held-open time. On by default.
Save alert settings	Saves everything above.
Find my chat ID	Saves, then asks the bot which chats it has seen and lists them with a Use this button each (Part B, way 1).
Send test message	Saves, then posts "Test from Access Controller (your site name)" to the chat ID using the stored token.

All three buttons save what is typed first, so nothing you entered is lost when you click one of the helpers.

Step by step:

1. Paste the **bot token** into **Bot token**.
2. Put the **chat ID** in **Chat ID**, or use **Find my chat ID** and click **Use this** on the right row.
3. Tick **Enable Telegram alerts**, and check that **Door forced** and **Door held open** are ticked as you want them (both are on by default).
4. Click **Send test message**. Within a couple of seconds the message "Test from Access Controller (your site name)" should appear in the chat. The panel shows a green "Test message sent - check Telegram", or a red "Test message NOT sent:" followed by the reason (see the troubleshooting table below).
5. Click **Save alert settings**.

Then open **Doors**, click each door that has a door sensor, and set **Alert if held open for N seconds** to suit that door (chapter 3). The default is 60; `0` switches the held-open alert off for that door. The value is sent to the door's controller automatically within seconds.

Tip: test the real thing once, not just the test button. Open a sensor-equipped door with the exit button and hold it past its held-open time; the *Door held open* message should arrive and the Log should show the same alert. Then open it from outside without a fob (with the relay pulse long finished) to see *Door forced*.

Cooldown and what is logged

- Repeated alerts of the same type for the same door are limited to **one Telegram message per minute**, so a bouncing sensor or a door that is opened and closed repeatedly cannot flood the chat. The Log still records every alert; only the Telegram copy is suppressed, and the panel's server log notes "suppressed (cooldown)" when that happens.
- The message is sent in the background, so a slow or unreachable Telegram never delays the controller or the Log.
- If a send fails (wrong chat, bot removed from the group, no internet from the panel) the failure is written to the panel's server log as "telegram send failed"; nothing is shown in the web interface, which is why the **Send test message** button exists. Re-test after any change to the bot, the group or the chat.
- The bot token is stored in the tenant's settings in the panel database and is never rendered back into a page. Chapter 15 lists it with the other secrets.

The panel host needs outbound HTTPS to `api.telegram.org` (chapter 16). Controllers do not talk to Telegram; only the panel does.

Troubleshooting

The exact message shown in red after **Send test message** or **Find my chat ID** is the best thing to pass on to whoever supports your panel. The common ones:

WHAT YOU SEE	WHAT IT MEANS	WHAT TO DO
401 Unauthorized / "rejected the bot token"	The token is wrong, has a typo, or was revoked.	Send <code>/mybots</code> to BotFather, pick the bot, API Token , copy it fresh and paste it again. Make sure there are no spaces at either end.
404 Not Found / "could not find that bot"	The token is incomplete (usually cut off when copying).	Copy the whole token, including the numbers before the colon.
400 chat not found	The chat ID is wrong, or the bot has never been in that chat.	Run Find my chat ID and pick from the list. For groups, check the bot is still a member; for channels, that it is an administrator. Remember the minus sign.
"bot was blocked by the user"	The person blocked or never started the bot.	That person opens the bot in Telegram and presses Start (or unblocks it).
"bot was kicked from the group"	Someone removed the bot.	Add it back to the group.
"not enough rights" / "no permission to post there"	The bot is in a channel but not an administrator.	Channel info > Administrators > add the bot.
"webhook is active" / getUpdates shows nothing even after messaging	Someone set a webhook on this bot (a different integration). Telegram then refuses <code>getUpdates</code> .	Open <code>https://api.telegram.org/botTOKEN/deleteWebhook</code> once in a browser (replace TOKEN), then try Find my chat ID again. Sending alerts works regardless.
Find my chat ID finds nothing for a group	Privacy mode: the bot cannot read normal group messages.	See "Privacy mode" above: make the bot an admin or turn privacy off, or type <code>/start</code> in the group.
"Saved, but no bot token is set" / "no chat ID is set"	You pressed a helper button before filling the field it needs.	Paste the token first for Find my chat ID ; both token and chat ID for Send test message .
"That chat ID does not look right"	The value is not a whole number.	Chat IDs are digits only, with an optional leading minus sign.
Test works but real alerts never arrive	Alerts are not enabled, or the type is unticked, or the door has no sensor, or its held-open time is 0.	Check the Enable tick box and the two alert tick boxes, then the door's held-open time and the controller's door sensor wiring. Look at the Log : if <i>Door held open</i> or <i>Door forced</i> rows are there, the panel saw the alert and the problem is on the Telegram side; check the panel's server log for "telegram send failed".
"could not reach Telegram"	The panel's server cannot reach the internet (firewall, proxy, DNS).	Allow outbound HTTPS from the panel host to <code>api.telegram.org</code> .
Alerts arrive once and then stop for a minute	Normal: repeats for the same door and type are rate-limited to one per minute.	Nothing to do.
Alerts stopped after the group was reorganised	The group became a supergroup and its ID changed.	Run Find my chat ID again and pick the new ID.

Controllers and hardware

This chapter is for the installer. It covers the supported controller boards, what connects to them, the default wiring for each board type, flashing and adopting a new controller from the browser, onboarding it into a tenant, over-the-air updates, moving a controller to a new server, and reading controller health.

Supported boards

The controller firmware runs on three families of board. The same firmware image is used for every unit of a family; nothing is compiled per site.

BOARD	NETWORK	NOTES
Classic ESP32 devkit (WROOM-32 / WROVER)	Wi-Fi	The general-purpose choice. Cheap, widely available.
Olimex ESP32-POE-ISO	Wired Ethernet with PoE	One cable for power and data, isolated. Ignores Wi-Fi settings and uses DHCP. Recommended where a network point is available.
ESP32-S3 (for example ESP32-S3-N16R8)	Wi-Fi	A different chip family from the classic ESP32 with its own firmware build. The flash wizard picks the right build automatically.

There are two firmware kinds:

- **Networked controller** (the usual choice): adopts to the panel over MQTT, gets its keyset from the panel, reports events. Everything in this manual assumes networked controllers.
- **Standalone controller**: no panel at all. It runs its own protected Wi-Fi admin page where fobs are enrolled with the reader, named and deleted. Useful for a single door run independently. It is flashed from the same wizard but is otherwise outside this manual.

What connects to a controller

Per door (per channel on a dual-door unit):

- **Wiegand reader**: any 26-bit or 34-bit Wiegand reader. Data 0 (usually the green wire) and Data 1 (usually white) go to the controller; the reader's 0 V is common with the controller's; the reader takes its power from the lock/reader supply (typically 12 V), not from the controller. The controller enables internal pull-ups on D0 and D1, so external pull-up resistors are usually unnecessary. Frames other than 26 or 34 bits are ignored and noted on the serial console.
- **Lock relay**: one output pin drives a relay module. The relay contact switches the lock's own supply. See the fail-safe warning in chapter 3.
- **Exit button (REX)**: a normally-open push button between the REX pin and 0 V. Pressing it always releases the door (free egress) and is logged as an exit-button event. Set the pin to `-1` if there is no button.
- **Door position sensor**: a reed switch or door contact between the door pin and 0 V, closed when the door is closed. Enables *door forced* and *door held open* alerts (chapters 3 and 4) and the *door opened* / *door closed* lines in the Log. Set the pin to `-1` if there is no sensor.
- **Real-time clock (optional)**: a DS3231 module on the SDA/SCL pins keeps accurate time through power cuts without network. Without it the controller gets time from the network (SNTP) after boot.
- **Power**: 5 V USB or the board's own regulator input; PoE on the Olimex board. Keep controller power separate from lock power.

A **dual-door controller** is one board with two of everything above, wired to the channel 1 and channel 2 pins.

Default GPIO wiring

The firmware ships with per-chip defaults so that a freshly flashed board works before anything is configured. The defaults are chosen to be free on both a generic devkit and the Olimex POE board (whose Ethernet PHY uses GPIO 12, 17, 18, 19, 21, 22, 23, 25, 26 and 27). Every pin can be changed in **Settings** (chapter 13); the values are sent to the controller when it is provisioned.

Classic ESP32 and Olimex ESP32-POE-ISO, channel 1

SIGNAL	GPIO	NOTES
Wiegand D0	32	reader green wire
Wiegand D1	33	reader white wire
Lock relay	4	active high by default (energise = HIGH); switchable to active low
Exit button (REX)	14	to 0 V, internal pull-up; -1 if absent
Door position sensor	15	reed to 0 V, closed = low, internal pull-up; -1 if absent
RTC SDA / SCL	13 / 16	optional DS3231
Setup button	0 (BOOT)	hold low at power-up to force setup mode

Classic ESP32 dual-door, channel 2 (panel defaults)

The Settings page pre-fills channel 2 with these values. They must not clash with channel 1. On an Olimex POE board GPIO 25, 26 and 27 are used by the Ethernet PHY, so choose other free pins for channel 2 there.

SIGNAL	GPIO
Wiegand D0	25
Wiegand D1	26
Lock relay	27
Exit button (REX)	13
Door position sensor	2

ESP32-S3, channel 1

The S3 has reserved pins that must be avoided: 26 to 32 (SPI flash), 33 to 37 (OPI PSRAM on -R8 parts; touching these hangs the chip), 19 and 20 (USB), 43 and 44 (UART0), and 0, 45 and 46 (strapping). The firmware defaults for the S3 all sit in the always-safe 1 to 18 range.

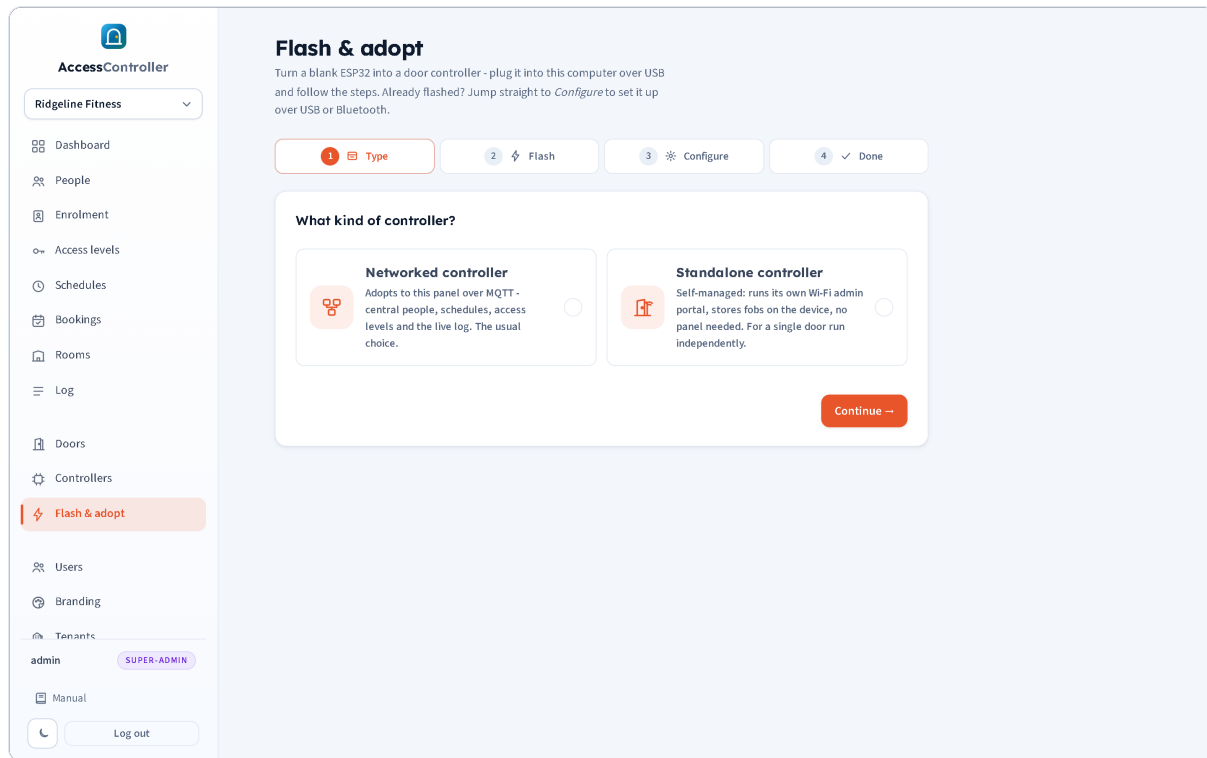
SIGNAL	GPIO	NOTES
Wiegand D0	4	
Wiegand D1	5	
Lock relay	6	active high by default
Exit button (REX)	7	to 0 V, internal pull-up
Door position sensor	15	reed to 0 V, closed = low
RTC SDA / SCL	8 / 9	optional DS3231
Setup button	0 (BOOT)	hold at power-up for setup mode

Note: the pin values held in the panel's Settings page are what actually get written to a controller when it is provisioned. The Settings defaults are the classic ESP32 values (D0 32, D1 33, relay 4, REX 14, door 15). If you are provisioning an ESP32-S3 from the panel, change the channel 1 pin map in Settings to the S3-safe values above **before** sending settings, otherwise the S3 will be told to use GPIO 33 and may hang.

Other fixed behaviour: the relay pulse defaults to 4000 ms, the held-open alert to 60 s until the door's own **Alert if held open** value arrives in its keyset (chapter 3), the door contact is debounced over 200 ms, and a stalled controller reboots itself after 30 s (hardware watchdog).

Flash and adopt: step by step

The **Flash & adopt** page turns a blank board into a door controller entirely from the browser. No installer software is needed. You need a laptop with Chrome or Edge, a USB cable to the board, and the panel open on a secure page.



The Flash & adopt wizard: four steps across the top (Type, Flash, Configure, Done) and the choice of controller kind.

The four steps across the top are **Type**, **Flash**, **Configure** and **Done**. They are also buttons: click any of them to jump straight there. That matters most for a board that is already flashed, where you want to go directly to **Configure** and send it its settings over USB or Bluetooth without flashing it again. Jumping ahead without choosing a firmware type first assumes **Networked controller**, which is what you want in all but a handful of cases.

Note: Web Serial and Web Bluetooth only work on a secure page. Use the panel's `https://` address (hosted panels are already HTTPS; a self-hosted panel also listens on port 8443 with a self-signed certificate, so accept the browser warning once), or open `http://localhost:8080/flash` on the panel machine itself. Safari on iPhone and iPad supports neither Web Serial nor Web Bluetooth.

Before you start

- On **Settings** (chapter 13) enter the controller Wi-Fi network and password (not needed for the Olimex wired board), the controller type (single or dual door), the pin maps if they differ from the defaults, and check the **Adoption address** and **MQTT port** the controllers will connect to. These are what the wizard sends to the board.
- Plug the board into the laptop over USB. Most boards need no driver on current Windows, macOS or Linux; if the port does not appear, install the USB-serial driver for the board's chip (typically CP210x or CH340).

Step 1: Type

Choose **Networked controller** (adopts to this panel) or **Standalone controller**. Press **Continue**.

Step 2: Flash

The page shows, for the chosen kind, which chip builds are staged on the panel ("ready", a partial count, or "not built"). Press the **install** button, pick the serial port in the browser's chooser, and let it run. The right build for the connected chip is selected automatically (classic ESP32 and ESP32-S3 have different bootloader offsets; this is handled for you). A first install takes about a minute.

If the board will not connect: hold **BOOT**, tap **RST** (reset), release **BOOT**, then retry. Some boards need this every time to enter download mode.

Upload firmware binaries lets an administrator stage new builds on the panel for a chip (four `.bin` files per chip: bootloader, partitions, boot_app0 and firmware). Most users never need this; 2plus1 stages builds on hosted panels.

As soon as flashing finishes, the wizard automatically reopens the USB port and sends the settings from step 3 to the board. Watch the log box under the flash button. If it reports that the port could not be reopened, go to step 3 and press the send button yourself.

Step 3: Configure

This step sends the Wi-Fi, the adoption address and port, the MQTT credentials and the pin map to the controller, which saves them, reboots and registers with the panel. First choose **Doors on this controller**: single door or dual door (two readers and relays). Then choose how to reach the board:

USB cable (default). A summary shows the Wi-Fi, adoption address and pins that will be sent, with a **Change in Settings** link. Press **Send settings to device**, pick the port again if asked, and wait for `ACSCFG OK` in the log. The board restarts and should appear under Controllers within a minute. **Query device** reads the board's current settings back (passwords masked) without changing anything; use it to check what a board believes.

If the board does not answer, press its **RESET** button and click send again. The board listens for settings over USB at any time, including in normal operation, so this also works later to re-provision a unit on the bench.

Bluetooth. No cable needed after flashing. Put the controller into setup mode by holding **BOOT** while powering it on. Press **Scan & connect** and pick the controller from the chooser: it advertises as `ACS-Cfg-` followed by the last six characters of its ID. Once connected, the Wi-Fi and address fields are pre-filled from Settings and can be overridden here; **Read current** shows what the board holds, **Write settings to controller** sends the new values and the board reboots onto Wi-Fi. Bluetooth configuration is only available while the controller is in setup mode; it is switched off during normal running.

Web Bluetooth works in Chrome and Edge on desktop and Android. It does not work in Safari or on iOS.

Step 4: Done

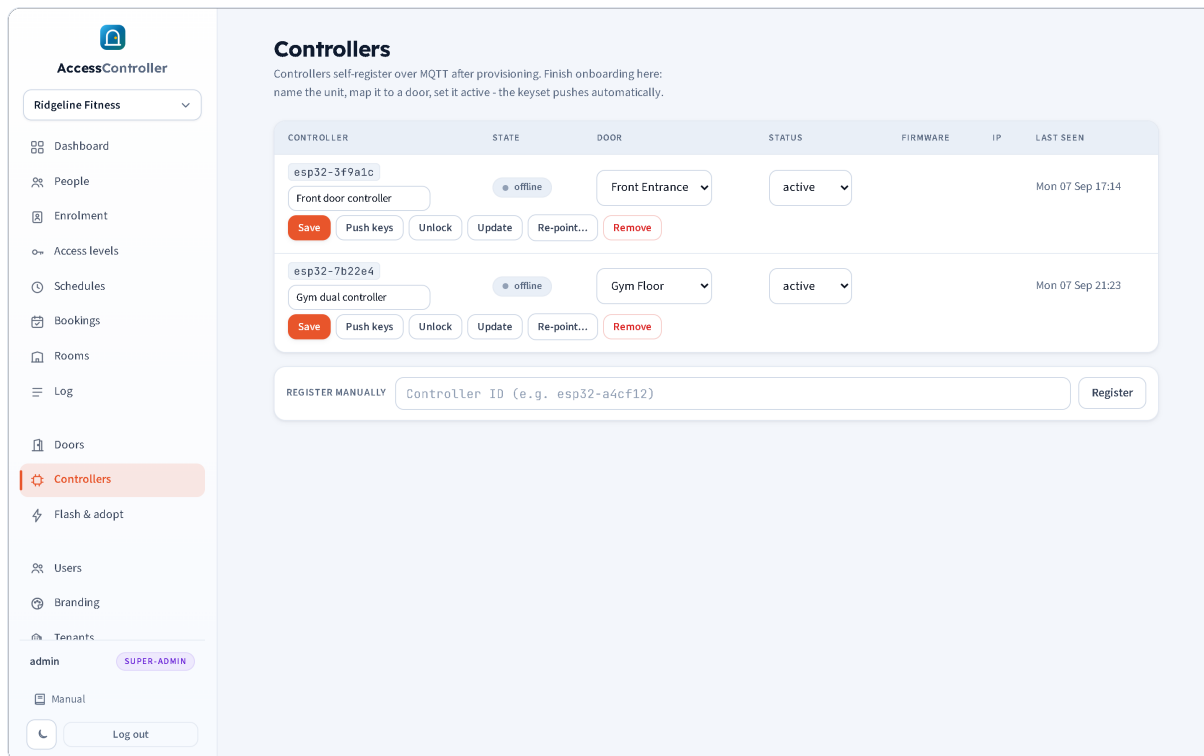
For a networked controller, go to **Controllers** to finish onboarding. For a standalone controller, connect a phone to the board's own Wi-Fi network (`ACS-Door-` followed by its ID, default password `access1234` unless you changed it in step 3) and its admin page opens automatically.

The setup portal fallback

A controller with no saved settings, or one powered up with **BOOT** held, starts its own Wi-Fi access point named `ACS-Setup-` followed by its ID with a captive setup page at `http://192.168.4.1`. The page takes the Wi-Fi network and password, the panel address and MQTT port, and optional MQTT username and password, then saves and restarts. Use this when there is no laptop to hand. The same setup mode is also where Bluetooth configuration is available.

Adopting a controller into a tenant

After provisioning, the controller connects to the panel and announces itself. The panel registers it automatically and logs a *Controller registered* system event. Finish onboarding on the **Controllers** page:



The Controllers page: each controller with its ID, friendly name, online state, door, status, firmware, IP and last-seen time, plus its action buttons.

1. Give it a **friendly name** ("Front door controller").
2. Choose its **Door** (this maps channel 1; for the second door of a dual controller use the door's own detail page and pick channel 2, as in chapter 3). A controller set to dual door is listed there with "- dual" after its ID, so you can see which units have a second channel to assign.
3. Set **Status** to **active**.
4. Press **Save**. The panel compiles the keyset for that door and pushes it; the message "Controller saved - keyset pushed" confirms it.

Status values: **pending** (registered, not yet onboarded; it receives no keyset), **active** (in service), **disabled** (kept in the list but ignored; no keyset is pushed and its events are not attributed to a door).

Which tenant does a new controller land in?

- If the controller's ID was **registered manually** first (the **Register manually** box at the bottom of the Controllers page, using the ID printed on the board's serial console or on its setup network name, for example `esp32-a4cf12`), it is created in the tenant you are switched into and simply comes online when it connects. This is the recommended path for a provider commissioning a customer's door: switch into the customer, register the ID, then flash and adopt.
- If an unknown controller connects first, it is placed in the **unassigned pool** (owned by no tenant) so that no customer can see another customer's hardware. A super admin then assigns it to a tenant. The current release's Controllers page does not yet show a pool section, so use manual registration in the right tenant before the controller first connects.

A controller only ever receives the fobs of the tenant that owns it.

Controller status and health

Each row on the Controllers page shows:

- **State:** **online** if the controller has reported within the last 10 minutes, otherwise **offline**. Controllers send a status heartbeat about once a minute and re-announce themselves on every reconnect.
- **Firmware** version and chip family (`esp32` or `esp32s3`).
- **IP** address on the site network and **Last seen** time.

The dashboard tile "Controllers online" shows the same information as a count. A controller that shows offline is still opening its door from its saved keyset; what it cannot do is receive new fobs or send events until it reconnects. Chapter 17 has the offline checklist.

Action buttons

- **Push keys:** recompile and re-send the door's keyset now. Use it after fixing a mapping or if you suspect the controller has an old list. The panel also pushes automatically on every relevant change and on a periodic timer.
- **Unlock:** fire the relay once (same pulse as a granted scan). Logged as *Remote unlock* with no person.
- **Update:** send the latest staged firmware for this controller's chip to it over the air. See below.
- **Re-point...:** move the controller to a different panel. See below.
- **Remove:** delete the controller from the panel. It will re-register in the pool if it connects again.

Over-the-air (OTA) updates

Update tells the controller to download the firmware image staged on the panel for its chip and flash it into its spare firmware slot, then reboot into it. The download rides HTTP on the panel's own port, or HTTPS through the reverse proxy if **Public HTTPS URL** is set in Settings (chapter 13). An *ota_start* event is logged when the download begins; a failure raises an *ota_failed* alert with the reason.

Things to know:

- The controller must be online.
- The firmware has two slots but **no automatic roll-back**. An image that does not boot leaves the unit in a boot loop until it is re-flashed over USB. Test an update on one unit before pushing it to a site.
- Units flashed with very old firmware (before the two-slot layout) must be flashed once over USB before OTA works.

Re-pointing a controller to a new server

Re-point... opens a small form on the controller's row: new **server address**, **MQTT port** (1883 by default), and optional **MQTT user** and **password**. Send it and the controller saves the new address, reboots and connects to the new panel, where it registers as a new controller. No site visit or USB cable is needed.

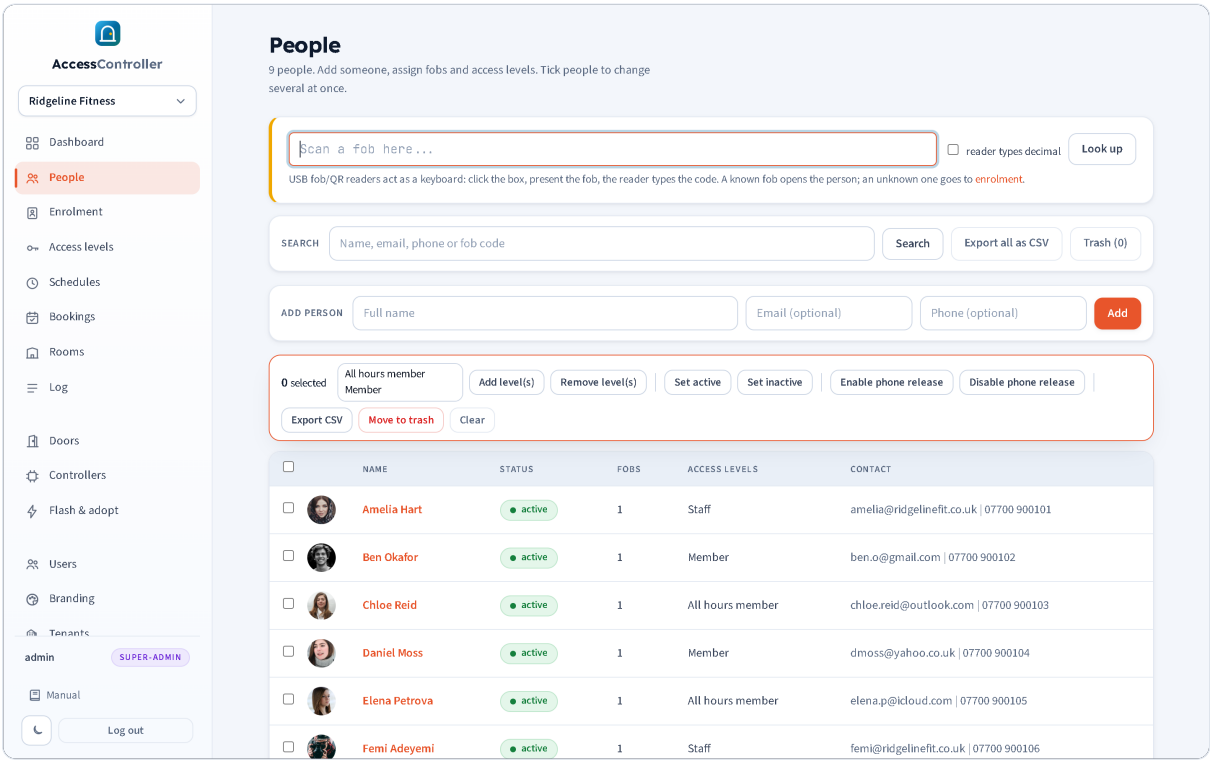
- The controller must be online with the current panel to receive the command.
- Leave the user and password blank to carry over *this* panel's MQTT credentials. If the new server uses different credentials, type them here. Sending blank credentials to a server that requires different ones leaves the controller unable to connect, and it cannot then be re-pointed remotely (chapter 17 explains the recovery).
- The controller's firmware must support re-pointing; if the Re-point button has no effect, update the firmware over the air first, then re-point.

Hardware installation checklist

1. Mount the controller in the same enclosure the old panel came out of, or a small IP-rated box, within cable reach of the reader, lock and exit button.
2. Wire the reader (D0, D1, 0 V, 12 V from the reader supply), relay (to the lock supply, NO or NC to suit fail-safe or fail-secure), exit button and door contact per the GPIO table for the board.
3. Fit a flyback diode across a maglock coil. Put the lock supply on a battery-backed PSU. Keep a break-glass or fire-alarm interface in the lock circuit where required.
4. Power the controller (USB 5 V or PoE) and confirm the reader's LED and the serial banner (115200 baud) if you have a console.
5. Flash and adopt from the panel, or use the setup portal.
6. On the Controllers page: name, door, active, Save.
7. Present a known fob: the log should show *OK* and the relay should click. Press the exit button: the log should show *Exit button*. Open and close the door: the log should show *Door opened* then *Door closed* (if a sensor is fitted). Open the door without a grant: the log should show *Door forced*.
8. Note the controller ID on the door's paperwork.

People and fobs

People are the members, staff, residents and contractors who open doors. Each person can hold several fobs and one or more access levels. This chapter covers adding people, photos, finding people, managing fobs, the four ways of getting fob codes into the system, changing many people at once, the trash, importing from a spreadsheet or from an AC8000 backup, the check-a-fob tool and what "inactive" means.



The People page: the desk scan box, search, the add-person toolbar and the list with photos, status, fob count, levels and contact details.

Adding a person

Open **People**. In the **Add person** toolbar type the full name and, optionally, an email address and phone number, then press **Add**. The person is created as active with no fobs and no access levels, and their page opens.

The phone number matters if the person will release doors by phone (chapter 9): the caller ID of their calls is matched against it, on the last nine digits, so 07911 123456 , +44 7911 123456 and 447911123456 all match.

The person page

AccessController

Ridgeline Fitness

Dashboard

People

Enrolment

Access levels

Schedules

Bookings

Rooms

Log

Doors

Controllers

Flash & adopt

Users

Branding

Tenants

admin

SUPER-ADMIN

Manual

Log out

Back to people

Ben Okafor

Details

Choose file No file chosen Upload

Webcam Remove

Name

Ben Okafor

Email

ben.o@gmail.com

Phone

07700 900102

Status

Active

Notes

Save

Fobs

CODE STATUS

7A8B9C8D active Revoke Delete

Fob code (hex, e.g. A1B) Add fob

or from a door: Bike Store Wait for fob at door

Access levels

☐ All hours member

☒ Member

☐ Staff

Save levels

Door release by phone off

If enabled, this person can release the ticked doors by calling the access line from their stored Phone number (set it in Details above). One door releases and hangs up; several offer a keypad menu. Every call is written to the Log.

The top of a person's page: photo and details on the left, fobs and access levels on the right. The door release by phone panel sits below (chapter 9).

Details

Name, email, phone, status (active or inactive) and free-text notes. Press **Save** after editing.

Photos

A photo is shown on the People list, on the person's page and, most usefully, on the reception screen the moment the person fobs in.

- **Upload:** choose an image file (JPEG, PNG, GIF or WebP, up to 4 MB) and press **Upload**.
- **Webcam:** press **Webcam**, allow the browser to use the camera, position the person and press **Capture**. The picture is cropped square and uploaded as a JPEG. Webcam capture needs a secure page (HTTPS or localhost), which hosted panels already are.
- **Remove** deletes the current photo. People without a photo are shown with their initials.

Fobs

The **Fobs** panel lists every fob on the person with its code and status.

- **Add fob:** type the code (hexadecimal, for example `A1B2C3` ; spaces, dashes and colons are ignored) and press **Add fob**. A code can belong to only one person within this tenant; the same code may exist independently in another tenant on the panel.
- **Revoke** sets a fob to revoked. It stops working at every door within seconds (the doors get a new keyset). The fob stays on the person's page so you can see it was theirs.
- **Re-activate** returns a revoked fob to active.
- **Delete** removes the fob entirely (asks for confirmation).
- **Wait for fob at door:** pick a door and press the button; the panel listens at that door's reader for two minutes and attaches the next unknown fob presented there to this person. See "Enrolling a fob at a door" below.

A person's active fobs open doors according to their access levels; revoked fobs are refused with *Fob revoked*.

Access levels

Tick the levels the person should hold and press **Save levels**. A person with no levels is refused everywhere with *No access level for door*. Chapter 7 explains levels.

Door release by phone

Whether the person may release doors from their phone, which doors, and their apartment number and PIN for guests. Chapter 9.

Move to trash

At the bottom of the page. **Move to trash** takes the person out of service straight away: their fobs stop opening doors within seconds, they disappear from the People list, the keysets, phone release and the resident portal. Nothing is destroyed. They sit in the trash for **30 days**, where they can be restored complete with their fobs and access levels, and are then removed automatically. See "The trash" below.

If the person may return sooner, or you only want to suspend them, set them **inactive** instead: that leaves them on the list and is a single click to undo.

Searching

The **Search** box on the People page matches name, email, phone or fob code (partial matches allowed). Press **Search**; **Clear** shows everyone again. The list shows a photo or initials, name, status, number of active fobs, access levels and contact details. Click a name to open the person.

Export all as CSV in the toolbar downloads everyone as a spreadsheet named `people-<date>.csv`, with the columns Name, Email, Phone, Status, Fobs, Access levels and Apartment. When a search is active the button becomes **Export matches as CSV** and exports only what the search found. People in the trash are never exported. Viewers may export as well as admins.

Changing several people at once

Every row on the People page has a tick box, and the box in the header ticks everyone. As soon as anything is ticked a bar appears showing **N selected** and the actions that can be applied to all of them:

ACTION	WHAT IT DOES
Add level(s)	Adds the access levels chosen in the box beside the buttons to everyone ticked, keeping what they already hold. Ctrl-click to pick more than one level.
Remove level(s)	Takes those levels away from everyone ticked.
Set active / Set inactive	Changes the status of everyone ticked.
Enable phone release / Disable phone release	Turns caller-ID release on or off for everyone ticked (chapter 9). It does not change which doors they may release.
Export CSV	Downloads just the ticked people as a spreadsheet.
Move to trash	Moves everyone ticked to the trash, after a confirmation that names the number of people.
Clear	Unticks everything. It changes nothing.

The message afterwards counts the people actually changed, so **Set active** applied to a mixed selection reports only the ones that were inactive. Keysets are re-pushed once at the end. The search is kept, so you can search "Smith", tick all, add a level, and carry on.

This is the quickest way to run a membership change: search for the group, tick, add or remove the level.

The trash

The **Trash** button in the People page toolbar shows how many people are in it, for example **Trash (3)**. (The same page is linked from the hint under a person's delete button.) It lists everyone who has been deleted and not yet purged.

COLUMN	MEANING
Name	The person, with their email underneath.
Fobs	How many fobs went with them. They are kept but inert.
Deleted on	When they were moved to the trash.
Removed in	How many days are left of the 30. <code>next purge</code> means they are due to go at the next sweep.

- **Restore** puts the person back exactly as they were: same fobs, same access levels, same phone-release doors. Their doors work again within seconds. It opens their page.
- **Delete permanently** removes them and everything attached to them now, after a confirmation. This cannot be undone.

The automatic sweep runs when the panel starts and every six hours after that, so someone may sit in the trash a few hours past their 30 days before disappearing. Each sweep writes one line to the Log per tenant (*Trash purged*), and moving to the trash, restoring and deleting permanently are each logged too (chapter 11).

The trash is what makes deleting a person safe: the usual "did we just delete the wrong Smith?" is a click to undo. Log entries are never deleted by any of this; they keep the name recorded at the time.

Getting fob codes into the system

There are four ways to attach a fob code, from the quickest for a single fob to the best for a whole site.

1. Type it

If the code is printed on the fob or known from the old system, type it into **Add fob** on the person's page. Codes are uppercase hexadecimal. Old systems often show the same fob as a decimal "card number"; use the desk scan box or the CSV import with the decimal option to convert.

2. Scan it at the desk

A USB fob or card reader that acts as a keyboard (a "keyboard wedge" reader) can be plugged into the office computer. On the People page, click into the **Scan a fob here** box, present the fob, and the reader types the code and presses Enter for you. Tick **reader types decimal** if your reader outputs decimal numbers rather than hexadecimal.

- A known fob opens that person's page (the message says whose it is).
- An unknown fob is sent to the **Enrolment** page with an assignment form ready, marked "desk".

This is also the fastest way to answer "whose fob is this?" for a fob found on the floor.

3. Enrol it at a door

You do not need a desk reader; any door reader will do.

- **From the person's page:** choose a door under "or from a door" and press **Wait for fob at door**. The page shows "Listening at (door) until (time)". Present the new fob at that door's reader within two minutes. The panel captures the code, attaches it to the person, pushes the new keyset and logs *Fob enrolled at door*. The page refreshes itself when it happens. **Stop listening** cancels.
- **From the Enrolment page:** choose a door and a listening time (2, 5 or 15 minutes) and press **Wait for a fob at this door**. The next unknown fob at that reader is captured into the list below and the page jumps to an assignment form for it.

While a door is listening, the fob presented does **not** open the door (it is still unknown at that moment) and the scan is logged as *Captured for enrolment* rather than as a denial, so it does not count towards "denied today". Only one door listens at a time per tenant.

4. Enrolment mode (a batch of fobs at chosen doors)

Enrolment mode is the migration path from a standalone system (Paxton Net2 without an export, an AC8000 whose backup has been lost, and similar) where nobody knows the codes. If you still have the old system's export or backup, use the import wizard instead ("Importing from a legacy system" below). Switch it on at the doors people will badge and every unknown fob presented **there is accepted (the door opens) and captured**, so members keep coming in as normal while the panel harvests their fobs. Then assign names to the captured codes.

Enrolment mode is started on the **Enrolment** page, not in Settings. In the **Enrolment mode** panel:

1. **Tick the doors** people will badge. Only those doors accept and capture unknown fobs; every other door on the site is unaffected and keeps refusing them. The doors you used last time are already ticked, so re-arming is one click.
2. Choose the **default access level** the people you enrol should end up with, for example *Access level: Members*. Choose **Access level: none** to decide later.
3. Choose how long: **for 15 minutes**, **for 30 minutes** (the default), **for 60 minutes** or **for 120 minutes**.
4. Press **Start enrolment**.

A banner then appears at the top of every page for the tenant reading, for example:

Enrolment mode on at Front, Gym until 14:35 (level: Members) - 3 captured to assign

with a **Stop** button. The Enrolment page's own panel turns green and shows the same information with **Stop enrolment now**. The Settings page also shows a green **Enrolment mode** panel while it is running, with a stop button, but starting is only ever done from the Enrolment page.

Enrolment ends automatically at the time shown, or when you press Stop. Stopping flashes *Enrolment stopped - unknown fobs are denied again* and keeps your door ticks and level choice ready for next time. Starting and stopping are logged as system events (*Enrolment started*, *Enrolment stopped*), and scans accepted this way are logged as granted with the reason *Enrolment (auto-accepted)*.

Warning: at the doors you ticked, **any** fob of the right format opens the door while enrolment mode is on, including fobs from other buildings and fobs that were revoked on the old system. That is the point of it, but it means you should tick only the doors you are actually standing at, keep the window as short as the job allows, and switch it off as soon as you are done. Picking doors is what limits the exposure: the front entrance can be in enrolment mode while the plant room is not.

The Enrolment page

The screenshot shows the 'Enrolment' page of the AccessController. On the left is a sidebar with a navigation menu including Dashboard, People, Enrolment (highlighted), Access levels, Schedules, Bookings, Rooms, Log, Doors, Controllers, Flash & adopt, Users, Branding, and Tenants. The main content area is titled 'Enrolment' and contains three sections: 'Enrolment mode' with checkboxes for Bike Store, Front Entrance, Gym Floor, Staff Room, and Studio 2, and dropdowns for 'Access level: none' and 'for 30 minutes', followed by a 'Start enrolment' button; 'Enrol a fob at a door' with a dropdown for 'Bike Store', a dropdown for 'for 2 minutes', and a 'Wait for a fob at this door' button; and 'Captured fobs' which currently shows 'Nothing captured yet. Turn enrolment mode on at a door above and have people badge it, or scan fobs at the desk from the People page.' At the bottom of the main area is a section for 'Import people and fobs'. The sidebar also shows a 'Log out' button at the bottom.

The Enrolment page: the enrolment mode panel, the enrol-a-fob-at-a-door panel, the captured fobs list with assign and dismiss controls, and the import section.

Captured fobs lists every code captured by enrolment mode, by listening at a door, by phone enrolment or by an unknown desk scan. The count is shown beside the heading.

COLUMN	MEANING
Fob	The code, in hexadecimal.
Seen	How many times that fob has been presented. Re-presenting a fob bumps this rather than adding a second row.
Last seen	When it was last presented.
Where	The door it was last presented at. A fob that came from the office reader is marked desk .
Assign to	The controls to give it to somebody.

To assign a fob: choose an existing person, **or** leave - **new person** - selected and type a name; choose an access level (the enrolment level is already selected); press **Assign**.

The access level is applied the same way whether the fob went to a new person or an existing one, and it is **added** to whatever the person already holds. Assigning a fob therefore never takes access away: enrol a long-standing member at the front door and they keep everything they had plus the enrolment level.

Dismiss drops a captured code you do not want (a visitor's fob, a fob from another site). Captured fobs stay in the list until they are assigned, dismissed or matched by an import; nothing expires them. The dashboard shows an amber "captured fobs waiting to be assigned" banner while the list is not empty.

Tip: the whole point of the default access level is that you never have to come back. Tick the doors, set the level to what a new member gets, and every fob you assign during the window arrives fully set up.

Importing from a legacy system

If the old system can give you its data, bring people and fobs across in one go from the **Import people and fobs** section of the Enrolment page. Two sources are supported; pick one with the two large buttons at the top of the section:

- **CSV / spreadsheet export:** a user export from Paxton Net2, or any CSV with a name column and a card-number column.
- **AC8000 / iCCard3000 backup (.mdb):** the database backup made by the AC8000 PC software. This brings across people, fob codes, door names and, optionally, the swipe history.

Whichever source you use, nothing is written to the panel until you press **Run import** on the preview page, and a fob that already exists is never overwritten.

Import from CSV

1. Export users to CSV from the old software.
2. Choose **CSV / spreadsheet export**, then either choose the file or paste its contents into the box, and press **Preview import**.
3. On the preview page, check the detected columns: **First name**, **Surname** (or a single **Full-name** column when there is no first/surname split) and the **Token / fob** column. Correct any guess with the drop-downs.
4. Leave **Tokens are decimal card numbers** ticked for Net2 and most systems that show card numbers in decimal; untick it if the column already holds hexadecimal codes.
5. Optionally tick the access levels to give everyone imported (see "The preview page" below).
6. Check the sample rows (the first 12, with any per-row problem flagged) and press **Run import**.

Format rules:

- The file may be comma, semicolon or tab separated; the delimiter is detected.
- The first row is treated as headers. Column names are matched case-insensitively: token columns are recognised by "token", "card number", "card no", "fob", "credential" or "badge"; first names by "first name", "forename" or "given"; surnames by "surname", "last name" or "family"; a whole name by "full name", "user name" or "name".
- Tokens may contain spaces, dashes, colons or dots; these are stripped. Decimal tokens are converted to hexadecimal. The result must be 2 to 16 hexadecimal digits.
- Rows with an empty name or an invalid token are counted as failed and skipped.
- A fob that already exists in the tenant is **skipped, never overwritten**; its row does not create a duplicate person.
- Up to 4 MB per upload.

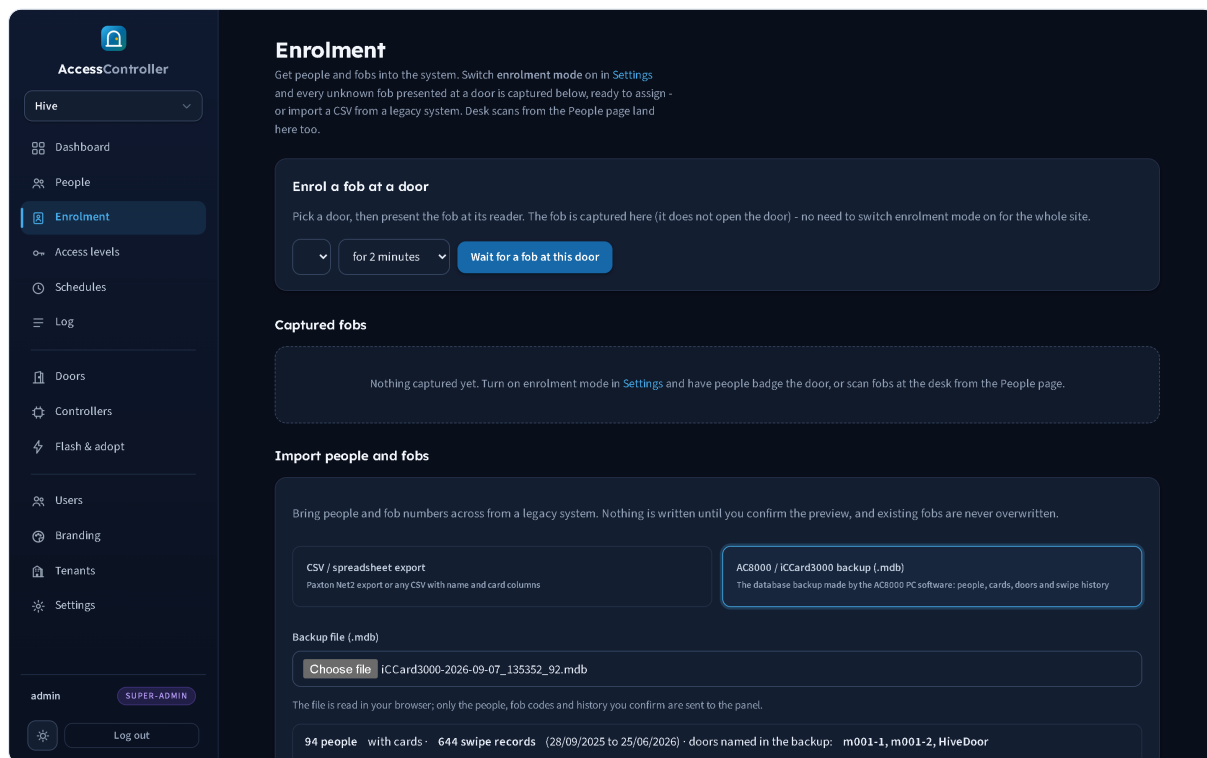
A minimal file looks like this:

```
First name,Surname,Token number
Alice,Example,12345678
Bob,Sample,12345679
```

Import from an AC8000 / iCCard3000 backup

Sites running the AC8000 family of standalone controllers (the PC software is called iCCard3000 or AC8000 depending on the version) usually have no export function, but the software keeps its whole database in a Microsoft Access file and can back it up. That backup contains everything the wizard needs: the people and their card numbers, the readers and doors, and every swipe the system recorded.

Where the backup comes from. In the AC8000 PC software use its backup (or database backup) function; it writes a file with an `.mdb` extension, typically named after the software and the date, for example `iCCard3000-2026-09-07_135352_92.mdb`. Copy that file to the computer you use for the panel. If you cannot find a backup, the live database the software uses is the same kind of file and can be copied while the software is closed.



The import wizard after reading a backup: the AC8000 / iCCard3000 source is selected, the file has been read in the browser, and the summary shows how many people and swipe records were found and which doors are named.

The steps:

1. Open **Enrolment**, scroll to **Import people and fobs** and choose **AC8000 / iCCard3000 backup (.mdb)**.
2. Under **Backup file (.mdb)** choose the file. It is read in your browser, not uploaded: only the people, fob codes and history you go on to confirm are sent to the panel. After a moment a summary appears: how many people with cards were found, how many swipe records and the dates they span, and the door names found in the backup. If any people had a keypad PIN in the old system, the summary says so and reminds you that PINs are not imported.
3. Choose the **Reader format at your doors: Wiegand 34 (8 hex digits) or Wiegand 26 (6 hex digits)**. This decides how the decimal card numbers in the backup are written as fob codes, and it should match what the readers on the new controllers send. If the backup contains a card number above 16,777,215 (the largest value a 26-bit reader can send) the wizard knows the site's readers must be 34-bit and sets 8 digits for you, with a note in the summary. Otherwise the default is 8 digits; change it to 6 if your readers are Wiegand 26.
4. Leave **Include the swipe history** ticked to bring the old log across, or untick it to import people and fobs only.
5. Leave **Rows with the same name are one person with several fobs** ticked unless you want every card to become its own person. See "One person, several fobs" below.
6. Press **Continue to preview**.

What is read from the backup:

- **People and their names.** Every record that has a card. A record with no name is imported as "Unnamed".

- **Card numbers**, converted from the decimal number the AC8000 shows to a hexadecimal fob code of the chosen width, padded with leading zeros (decimal `20002473` becomes `013136A9` at 8 digits).
- **Door names**, worked out from the reader names. The AC8000 names readers per side of the door, such as `HiveDoor-In` and `HiveDoor-Exit`; the wizard strips the `-In`, `-Out` or `-Exit` suffix and treats what is left as the door. Only readers that appear in the history or in an access rule are counted, so the placeholder readers the AC8000 lists for unused door slots are ignored.
- **Swipe history**: every swipe record with its time, card, reader and whether the old system granted or denied it. The most recent 20,000 records are kept if there are more.

What is not imported:

- **Keypad PINs**. The panel's PINs are per-apartment for phone access (chapter 9) and are set on each person afterwards if needed.
- Access rules and schedules from the old system. Give access with the tick list on the preview page instead.

Status, valid-to dates and notes

The wizard carries across three things beyond the name and the card, and they are worth understanding because they decide who arrives switched off:

- **Status**. The AC8000's per-person "Access Control" flag becomes the panel's active/inactive status. Somebody whose access was turned off in the old system arrives **inactive** on the panel, so a lapsed or barred member does not silently regain access on the day you cut over.
- **Valid to**. The old system's end date. **A valid-to date that has already passed also imports the person as inactive**, for the same reason. The AC8000 uses a date in 2099 to mean "no expiry"; the wizard recognises that and ignores it.
- **Notes**. Each person's notes receive their AC8000 record number, for example `AC8000 record 0042`, and the valid-to date where one is set. That record number is the thread back to the old system if you ever have to check something, so keep it.

Everyone else arrives **active**. Review the inactive people after the import: some will be genuine leavers to move to the trash, others will be members whose renewal never got recorded in the old software.

One person, several fobs

Standalone systems commonly hold one record per **card**, so a member who was issued a replacement fob appears twice with the same name. The wizard's **Rows with the same name are one person with several fobs** option, which is **on by default**, collapses those into a single person holding both fobs. It is the behaviour you almost always want: one person on the People page, both cards working, one set of access levels.

Untick it (in the wizard, or on the preview page) if the duplicate names are genuinely different people. Each card then becomes its own person, and the AC8000 record number is appended to the name to tell them apart, for example "J Smith 0042" and "J Smith 0117". Rename them afterwards.

The same fob at more than one site

A fob code is unique **within a tenant**, not across the panel. The same backup can therefore be imported into a second tenant (a test tenant, or a second building of the same operator) and the cards attach there too, independently. A resident who holds one fob for two buildings you manage is a person in each tenant, and revoking them in one has no effect on the other.

The preview page

AccessController

Hive

Dashboard

People

Enrolment

Access Levels

Schedules

Log

Doors

Controllers

Flash & adopt

Users

Branding

Tenants

Settings

admin SUPER-ADMIN

Log out

Back to enrolment

Confirm import

94 rows found in the AC8000 / iCCard3000 backup. Check the column mapping and the sample below, choose the access to grant, then run the import. Existing fobs are never overwritten; people who already hold a fob just get the access levels added.

Columns

First name column: (none)

Surname column: (none)

Full-name column (used when first/surname are none): Name

Token / fob column: Token

☐ Tokens are decimal card numbers, convert to hex

Run import (94 rows)

Access for everyone imported

Tick the access levels to grant to every imported person. People whose fob already exists get these added to what they have.

☐ Hive members Imported from AC8000

History

644 swipe records found (28 Sep 2025 to 25 Jun 2026).

☒ Import the swipe history into the Log

☒ Create the doors named in the backup if they do not exist: HiveDoor

Sample (first 12 rows, with auto-detected columns)

PERSON	FOB (HEX)	PROBLEM
Ash	013036A9	
B4U 2	01303BCD	
B4U 3	01303BDF	
B4U 4	01303661	

The preview for a backup import: the columns are already mapped, access levels can be ticked for everyone imported, and the History panel offers to import the swipe records and create the door named in the backup.

The preview page is the same for both sources. It says how many rows were found (and, for a backup, that they came from the AC8000 / iCCard3000 backup), and shows:

- **Columns.** For a backup the mapping is already right: **Full-name column** is Name and **Token / fob column** is Token, and **Tokens are decimal card numbers** is unticked because the wizard has already converted them. For a CSV, check and correct the guesses.
- **Access for everyone imported.** Tick the access levels to grant. New people get exactly these levels. People whose fob already exists on the panel are not re-created and their fob is left alone, but the ticked levels are **added** to whatever they already hold, so you can safely run the import twice or use it to grant a level in bulk. If the tenant has no levels yet you can import now and assign access later, or create a level first (chapter 7).
- **Rows with the same name are one person with several fobs**, carried over from the wizard and still changeable here.
- **History** (only when the backup had swipe records and you left the history ticked). It shows how many records and the date range, with two boxes: **Import the swipe history into the Log**, and **Create these doors if they do not exist**. Under the second is **one editable text box per door name found in the backup**. This is where you tidy up what the old system called things: change `HiveDoor` to `Studio 2` and the door is created under the sensible name, or type the name of a door that already exists on the panel and the history is attached to that door instead of creating a duplicate. Matching is case-insensitive. Untick the box entirely and the history is imported with no door for those names.
- **Sample:** the first 12 rows as they will be imported, under the headings **Person**, **Fob (hex)** and **Problem**. A row with a problem (`empty token` , `no name` , or a code that is not valid) is flagged in red and will be counted as failed.

Press **Run import (N rows)**. Until then nothing has been written; you can go **Back to enrolment** and start again with different options.

What the import does

- Creates each person and attaches their fob. A fob that already exists in the tenant is skipped (its person keeps their name and other fobs) and only gets the ticked levels added.
- Clears any captured fob on the Enrolment page that matches an imported code.
- Writes the swipe history into the Log if asked, with each record's **original date and time**, the door, the fob and, where the card belongs to an imported (or existing) person, the person's name. Records the old system granted are logged as granted; the rest as denied with *Unknown fob*. Every imported row carries "AC8000 history, reader (reader name)" in its detail so it can be told from the panel's own events (chapter 11).
- Creates any missing doors named in the history if asked, described as "Imported from AC8000 backup", with no controller mapped. Map a controller to each on the Doors page (chapter 3) when the new hardware goes in.
- Pushes the new keysets to every door straight away.

The result message on the People page reads, for example, *Import done: 94 people created, 94 fobs attached, 0 skipped, 0 failed, 644 history records added to the Log*. The same numbers are written to the Log as a system event (CSV *import*, reason `import`) whichever source was used.

Tip: the panel matches a fob whether it is stored with 6 or 8 digits (`0136A9` and `000136A9` are the same fob), so a code imported at the wrong width still opens the door when the reader sends the other form. What no setting can fix is a Wiegand 26 reader presented with a card above 16,777,215: it physically cannot send the high bits. See "Imported fobs are denied at the door" in chapter 17.

Check a fob

Two tools tell you what a fob is and what it does:

- **Desk scan** (People page): whose fob is this? A known fob opens the person; an unknown one goes to enrolment.
- **Check access** (Dashboard): would this fob open this door at this time, and if not, why? It runs the exact decision rule the controller uses and writes nothing to the log. Enter the code, choose a door, optionally change the date and time, press **Check**.

Active and inactive

Setting a person's **Status** to **Inactive (all access suspended)** stops every fob they hold from opening any door, immediately, without touching the fobs or their levels. The log shows *Person inactive* for their attempts. They also cannot release doors by phone. Use it for lapsed memberships, suspended accounts and leavers who may come back. Set them back to **Active** to restore everything exactly as it was.

Inactive people are not counted in the dashboard's "Active people" tile but do appear in the People list with an "inactive" badge.

Access levels and schedules

Who may open which door, and when, is expressed with two things: a **schedule** (a weekly pattern of allowed hours) and an **access level** (a set of doors plus one schedule). People hold access levels. This keeps the rules in one place: change the schedule and every person on that level follows.

Schedules

Open **Schedules**. Type a name in **Add schedule** ("Opening hours", "Staff hours", "Cleaners early morning") and press **Add**, then click the name to open the editor.

The guided schedule editor: presets across the top, a row per day with access mode and times, a timeline bar per day, and the live summary.

The guided editor

Each day of the week (Monday to Sunday) has:

- **Access:** **No access**, **Time window** or **All day**.
- **From** and **To** times in 24-hour format, used only for a time window. Times step in 5-minute increments.
- A **timeline bar** showing the open hours for that day at a glance, from 00:00 on the left to 24:00 on the right. An amber marker shows the current time.
- -> **Mon-Fri** copies that day's setting to Monday to Friday; -> **All** copies it to every day.

Conveniences:

- **Quick set** presets fill the whole week in one click: **24 / 7**, **Weekdays 09:00-17:00**, **Mon-Sat 06:00-22:00**, **Weekdays only, all day**, and **Clear all** (no access every day). Start from the nearest preset and adjust.
- Switching a day to **Time window** fills 09:00 to 17:00 if the times are blank; typing a time switches the day to Time window automatically.
- A **To** time earlier than the **From** time runs overnight into the next day: 22:00 to 06:00 means from ten at night until six the next morning. The timeline draws this as two segments.
- The summary line under the grid reads the schedule back in words, grouping identical days: "Mon-Fri 06:00-22:00 . Sat-Sun 08:00-18:00".

Press **Save schedule**. Times are in the panel's timezone (set by the host; Europe/London by default), and daylight-saving changes are handled for you.

Delete schedule removes it; any access level that used it becomes 24/7, so check the levels afterwards.

Access levels

Open **Access levels**. In **Add level** type a name and optional description, choose a schedule (or "No schedule - 24/7") and press **Add**. Click a level to edit it.

The screenshot shows the 'Access levels' page in the AccessController interface. On the left is a sidebar with a navigation menu including Dashboard, People, Enrolment, Access levels (highlighted), Schedules, Bookings, Rooms, Log, Doors, Controllers, Flash & adopt, Users, Branding, Tenants, and Manual. The main content area is titled 'Access levels' with a subtitle: 'A level defines which doors open and on what schedule. People hold one or more.' Below this is an 'ADD LEVEL' form with fields for 'Level name (e.g. Standard access)', 'Description (optional)', and a dropdown for 'No schedule - 24/7', followed by an 'Add' button. A table lists existing access levels:

LEVEL	SCHEDULE	DOORS	PEOPLE	DESCRIPTION
All hours member	Round the clock	0	2	Round-the-clock gym access
Member	Opening hours	0	4	Gym floor and studios during opening hours
Staff	Staff hours	0	2	All doors, staff hours

The Access levels page: each level with its schedule, door count, people count and description.

On a level's page:

- **Name** and **Description**.
- **Schedule**: one schedule, or **No schedule - 24/7** meaning the doors open at any time.
- **Doors this level opens**: tick every door the level should include.

Press **Save level**. **Delete level** removes it and every person holding it loses that access.

A person can hold several levels; the doors and hours add together. If any one of their levels allows the door at that time, they get in.

Worked examples

The three levels below cover most gyms and offices. Adjust the names and hours to suit.

Example 1: "Members 06:00-22:00"

Ordinary members may use the front entrance and the gym floor from six in the morning until ten at night, every day.

1. **Schedules** > **Add schedule** "Opening hours". Open it, press the **Mon-Sat 06:00-22:00** preset, then set Sunday to **Time window** 06:00 to 22:00 (or press -> **All** on Monday's row). Save.
2. **Access levels** > **Add level** "Members", schedule "Opening hours". Open it and tick **Front Entrance** and **Gym Floor**. Save.
3. On each member's page tick **Members** under Access levels and save.

A member scanning at 05:55 is refused with *Outside schedule*; at 06:00 they are let in.

Example 2: "Staff"

Staff may use every door, including the office and the plant room, from an hour before opening until an hour after closing.

1. Schedule "Staff hours": every day **Time window** 05:00 to 23:00 (set Monday, press -> **All**). Save.
2. Level "Staff", schedule "Staff hours", tick all doors. Save.

3. Give staff the **Staff** level. They do not also need Members.

Example 3: "24/7"

Owners, the cleaning contractor's supervisor and the alarm keyholder need every door at any time.

1. Level "24/7 all doors", schedule **No schedule - 24/7**, tick all doors. Save. (No schedule needs to be created.)
2. Give it only to the few people who need it.

Example 4: an overnight window

Night cleaners work 22:00 to 02:00. Schedule "Night clean": Monday to Friday **Time window** From 22:00 To 02:00. The summary reads "Mon-Fri 22:00-02:00 overnight". Note that a window starting on Friday night runs into the early hours of Saturday, which is what you want; you do not need to add anything on Saturday.

Example 5: a class studio with restricted hours

Studio 2 is only for the yoga membership, on class evenings. Level "Yoga", schedule "Yoga evenings" (Tue and Thu 18:00 to 21:00, everything else No access), door Studio 2. Yoga members hold both **Members** (front door and gym floor) and **Yoga** (the studio on those evenings).

What happens when you change something

Every change to a schedule, a level, a person's levels, a fob or a person's status causes the panel to recompile the keyset for each affected door and push it to that door's controller. This usually takes a second or two. You will see the effect on the next scan.

- A controller that is offline at the time receives the new keyset as soon as it reconnects (the last keyset is held for it on the panel's broker), and the panel also re-pushes all keysets every few hours as a safety net.
- Use **Check access** on the dashboard to confirm the effect of a change without walking to the door.
- Deleting a schedule turns any level that used it into 24/7; deleting a level removes that access from everyone who held it. Both ask for confirmation.

Room bookings and the resident portal

Some doors are not "who may come in?" but "whose turn is it?": a residents' lounge, a shared gym, a laundry, a meeting room, a treatment room. Access Controller handles these as **rooms**. A room is an ordinary door that has been made bookable. Residents reserve a slot on the **resident portal**, a small mobile web page of their own, and a confirmed booking opens that door for them for the length of the slot and nothing else.

Nothing here changes how the rest of the panel works. A room is still a door, a resident is still a person, and a booking is simply another reason the door may open, logged like any other.

When to use a room

Use a room when all of these are true:

- more than one household or team wants the same space at different times;
- you want the door to open for the person who booked it and stay shut otherwise;
- you may want to charge for it, or at least record who had it.

If everyone with a fob may walk in whenever the door is unlocked, you want an access level and a schedule (chapter 7), not a room.

What a room door needs

A bookable room is the simplest door on the site to fit, because nobody needs to present anything at it:

- a **door module** (the controller channel: the lock relay, and ideally the door position sensor);
- an **exit button** on the inside, so anyone in the room can always get out;
- **no fob reader is required**. Residents come in through the portal's **Open the door** button or by ringing the access line from the mobile they booked with.

Fit a reader anyway if you want residents to be able to fob in during their slot: a confirmed booking is written into the door's keyset as a time window, so their existing fob works for exactly that period. But a room with no reader at all is a supported and cheaper build, and it is what the phone and portal paths exist for.

Note: the panel does not enforce this. Any door on the tenant can be turned into a room, reader or no reader. Wiring it without a reader is a choice you make on site; the software simply never needs one.

Warning: an exit button (or free mechanical egress) on the inside of a bookable room is not optional. Somebody whose slot has just ended must always be able to get out.

Setting a room up

Open **Rooms** in the sidebar. Rooms and Bookings both appear there for anyone whose role includes the **Bookings** permission (chapter 12); there is no separate Rooms permission.

AccessController

Ridgeline Fitness

- Dashboard
- People
- Enrolment
- Access levels
- Schedules
- Bookings
- Rooms**
- Log
- Doors
- Controllers
- Flash & adopt
- Users
- Branding
- Tenants
- admin SUPER-ADMIN
- Manual
- Log out

Rooms

A room is a bookable door. Residents reserve a slot on the portal, and a confirmed booking opens that door for them - fob or phone - from a few minutes before the slot until just after it ends.

Resident portal

Residents book at the address below. Share it, print it as a QR code by the entrance, or link it from the building's own site.

Copy Open portal

ADD ROOM

Pick a door...

Add

ROOM	DOOR	SLOT	PRICE	BOOK AHEAD	CANCEL BY	DOOR OPENS	STATUS
Front Entrance Lobby	Front Entrance	60 min	Free	30 days	1 h before	120 min early	bookable

The Rooms page: the resident portal address panel at the top, the add-room toolbar, and the list of rooms with slot length, price, booking window and status.

To create one, pick a door from the **Add room** drop-down (only doors that are not already rooms are offered), optionally type a **Room name** (it defaults to the door name) and a **Description**, and press **Add**. The room is created open Monday to Sunday, 08:00 to 20:00, and you adjust it from there.

The list shows every room with its **Door**, **Slot** length, **Price**, **Book ahead** window, **Cancel by** cut-off, how early the **Door opens** and a **bookable** or **closed** status badge. Click a room's name to edit it.

Room settings

AccessController

Ridgeline Fitness

- Dashboard
- People
- Enrolment
- Access levels
- Schedules
- Bookings
- Rooms**
- Log
- Doors
- Controllers
- Flash & adopt
- Users
- Branding
- Tenants
- admin SUPER-ADMIN
- Manual
- Log out

Rooms

Front Entrance Lobby

Bookings on this room open Front Entrance. Settings here decide the slot grid, the price and how early the door lets someone in.

Settings

Room name

Slot length (minutes)

Description

Gap between bookings (minutes)

Taking bookings

Yes, residents can book it

Price per slot

Order on the portal

Currency

Book up to (days ahead)

Cancel until (hours before)

Door opens early (minutes)

A price of 0.00 makes the room free: those bookings confirm straight away. Priced rooms stay pending until the money arrives (card or bank transfer). The door stays open to the person for five minutes after the slot ends.

Save room

A room's page: the settings grid, the seven-day opening-hours table and the extras panel.

SETTING	WHAT IT DOES
Room name	What residents see on the portal. Required.
Description	A line of explanation shown under the name on the portal.
Taking bookings	Yes, residents can book it or No, hide it from the portal . Hiding a room leaves existing bookings alone; it just stops new ones.
Order on the portal	A number. Rooms are listed by this, then by name.
Slot length (minutes)	The size of one booking, 5 to 1440. The whole day is cut into slots of this length inside the opening hours.
Gap between bookings (minutes)	A buffer applied on both sides of every booking, for changeover or cleaning. A 15-minute buffer round a 60-minute booking blocks the quarter-hour either side of it.
Price per slot	Pounds and pence, for example <code>12.50</code> . A price of 0.00 makes the room free , and free bookings confirm immediately.
Currency	A three-letter code, <code>GBP</code> by default. GBP, EUR and USD are shown with their symbol; any other code is shown as the code followed by the amount.
Book up to (days ahead)	How far into the future residents may book. 30 by default.
Cancel until (hours before)	A resident may cancel their own booking up to this many hours before it starts. 24 by default. Staff can cancel at any time.
Door opens early (minutes)	How long before the slot the door starts letting the person in. 10 by default.

Press **Save room**. The door's controller is re-sent its keys straight away, because changing the lead time changes access windows that have already been granted.

The door also stays open to the person for **five minutes after the slot ends**, so nobody is locked in while they gather their things. That grace period is fixed and is not a setting.

Opening hours

The **Opening hours** panel is a row per day, Monday to Sunday. Tick **Closed** for a day with no bookings, or give an **Open from** and **until** time. The second pair of times (**and from ... until**) covers a split day, such as a lunchtime close. Times are in five-minute steps. Slots are cut from these ranges and aligned to the slot length, so a room open 08:00 to 20:00 with 90-minute slots simply runs out of room at 18:30 and the last part of the evening is unbookable. Press **Save hours**.

Extras

The **Extras** panel adds optional items a resident can tick when booking: a projector, a set of weights, a cot, a cleaning charge. Each extra has a **Name**, a **Price** and a **Stock**. Leave the stock blank for unlimited; a number is the total held across every booking that has not finished yet, so `3` chairs means at most three are out at once. Set an extra to **Hidden** to retire it without disturbing bookings that already include it. **Remove** deletes it; bookings that already include it keep the price they were quoted.

Extras appear only on the portal. A booking made by staff from the panel cannot include them.

Removing a room

Remove room at the bottom of the room's page deletes the room and its bookings. **The door itself is untouched** and goes back to being an ordinary door. Deleting the underlying door on the Doors page removes the room with it.

Publishing the resident portal

The portal is off until you switch it on. Open **Settings**, expand **Bookings and portal** and tick **Publish the resident portal**. The **Portal address** is the last part of the web address residents use; leave it blank to use the tenant's own short name. It may contain lowercase letters, digits, dashes and underscores, up to 40 characters, and must be different from every other customer's on the panel.

The address is then shown on the Rooms page with **Copy** and **Open portal** buttons. It looks like:

`https://doors.example.com/p/riverside/`

Share it however suits the building: a link from the block's own website, a line in the welcome pack, or a printed QR code by the entrance.

There is also a **board** at the same address with `board` on the end, for example `https://doors.example.com/p/riverside/board`. It shows today's confirmed bookings by room and time with names reduced to initials, and needs no login at all. It is meant for a screen or a printout by the room door.

Warning: the board is genuinely public. Anyone who has the address, or who guesses it, can see what is booked today. It shows initials and times only, never full names, numbers or fob codes, but if that is still more than the building wants to publish, simply do not share the board address.

Text messages

Residents sign in with a code sent to their mobile, so the tenant needs an SMS gateway. Under **Settings > Bookings and portal > SMS gateway**, give the **Gateway URL**, the **Sender name or number** and the **API key**. The panel sends the gateway a JSON `POST` of `to`, `from` and `text` with an `Authorization: Bearer` header, which most bulk-SMS providers accept directly or through a one-line adapter.

With no gateway URL set, login codes are not sent anywhere. They are written to the panel's server log and shown on the Bookings page in a **Login codes sent** panel marked *development only*, so that a new site can be tested before the SMS account exists. That is a commissioning aid. Never run a live building on it: staff would have to read every resident's code out to them.

The resident's experience

The portal is a small mobile page in the tenant's own logo and accent colour. Everything below is what a resident sees on their phone.

Ridgeline Fitness

Book a room

Enter your mobile number and we will text you a 6-digit code.

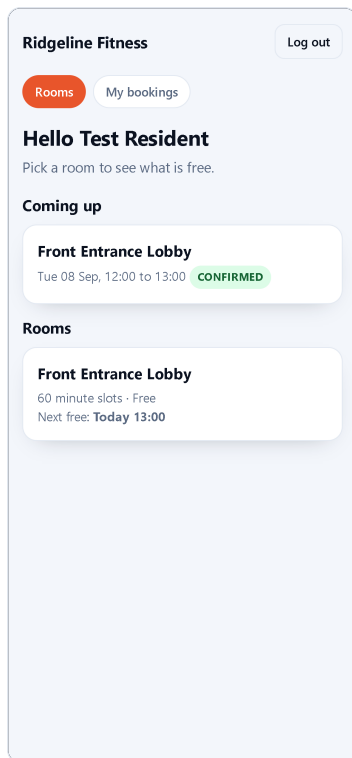
Mobile number

07700 900123

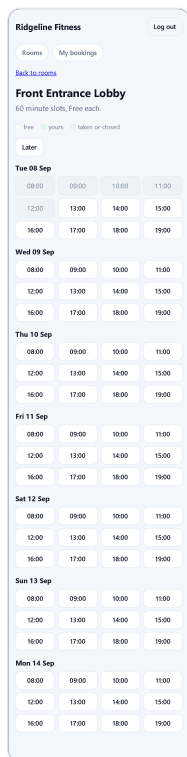
Send me a code

Only residents of Ridgeline Fitness can book. If we do not recognise your number we will ask for your name and apartment.

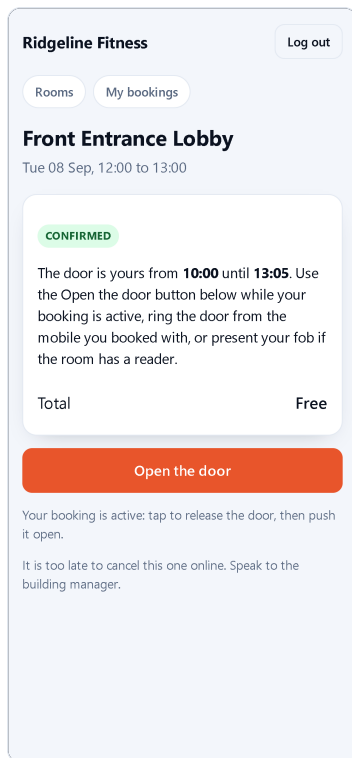
The resident portal on a phone: the building's name, the mobile number field and the Send me a code button.



The portal home page after signing in: a greeting, the resident's next booking, and each room with its slot length, price and next free slot.



A room on the portal: the slot length and price, a key for free, yours and taken, then a grid of times for the next seven days.



A confirmed booking: the slot times, the total, and the Open the door button that works while the booking is active.

Signing in

1. **Book a room** asks for their **Mobile number**. They press **Send me a code**.
2. **Check your texts**: they type the 6-digit code. It lasts ten minutes, allows five attempts and can only be used once.
3. Then one of three things happens:
 - If the number matches a person who has an **apartment number** on file, **One more check** asks them to confirm it. Three wrong answers lock that number out for 15 minutes.
 - If the number matches a person with no apartment number recorded, they are signed straight in.
 - If the number matches nobody, **Nice to meet you** asks for their name and apartment number and creates them as a person in the tenant, noted as *Self-registered via the resident portal* and recorded in the Log. Staff see the new person on the People page and can attach a fob and access levels in the usual way.

People who are inactive, or in the trash, never match, so they cannot sign in. The session lasts 90 days on that phone, so in practice a resident signs in once.

Note: a resident who changes their mobile number cannot sign in and cannot self-register, because their apartment already exists. Update the number on their person page.

Booking a slot

The home page greets them by name and lists their **Coming up** bookings, then every bookable room with its slot length, price and the **Next free** time. Tapping a room opens a seven-day grid.

Each day shows its slots colour-coded **free, yours** or **taken or closed**; **Earlier** and **Later** page through the weeks up to the room's booking window. Tapping a free slot opens **Confirm your booking**, which shows the room and time, offers any **extras** with the number left in stock, and totals the price. The button says **Book it** for a free room and **Book and pay** for a priced one.

- **A free room confirms immediately** and the door is theirs.
- **A priced room is held, pending payment**, and they go to the payment page.

The slot rules are re-checked at the moment they confirm, so two residents racing for the same slot cannot both get it; the loser is told *That slot is already booked* and sent back to the grid.

Paying

The payment page shows the total and offers whichever methods the building has set up.

- **Pay by card** takes them to Stripe Checkout and back. The booking confirms as soon as the payment succeeds.
- **Pay by bank transfer** shows the building's bank details and a reference of the form `BK-RIVERSIDE-42`. They quote it with the transfer; the booking confirms when a member of staff marks it paid on the Bookings page.

An unpaid booking does not hold its slot forever. Card bookings are held for **30 minutes** and bank-transfer bookings for **120 minutes** by default (both are settings), after which the booking expires and the slot goes back on sale. The resident is not shown a countdown, so make the bank-transfer window generous.

During the booking

While the slot is running (from the lead time before it starts until five minutes after it ends), the booking page carries a full-width **Open the door** button. Tapping it releases the door there and then, and the Log records **Opened from portal** with the booking number. The same person can also:

- **ring the access line** from the mobile they booked with, which releases the room door for the length of the booking even if they have no phone-release doors ticked at all (chapter 9); or
- **present their fob**, if a reader is fitted, because the booking is compiled into the door's keyset as an absolute time window.

Outside the window the button is not shown, and pressing it by other means is refused with *The door can only be opened during your booking*.

My bookings lists everything they have booked, newest first. **Cancel this booking** is offered while the room's cancellation cut-off has not passed; after that they are told to speak to the building manager.

Note: fob access during a booking needs the current controller firmware. A controller running an older build ignores the booking windows in its keyset and refuses the scan. Update every controller that fronts a bookable room before bookings go live (chapter 5). The portal button and the phone path work whatever the firmware, because the panel sends the unlock itself.

Managing bookings from the panel

Open **Bookings** in the sidebar.

AccessController

Ridgeline Fitness

- Dashboard
- People
- Enrolment
- Access levels
- Schedules
- Bookings**
- Rooms
- Log
- Doors
- Controllers
- Flash & adopt
- Users
- Branding
- Tenants
- admin
- Manual
- Log out

Bookings

What is booked, and what is waiting to be paid for. Confirming a booking is what opens the door: the room's controller is re-pushed the moment anything here changes.

SHOW Upcoming All rooms

WHEN	ROOM	PERSON	EXTRAS	AMOUNT	STATUS	ACTIONS
Tue 08 Sep 12:00 to 13:00	Front Entrance Lobby	Test Resident		Free	confirmed	Cancel

Book a room for someone

Residents book at <http://localhost:8087/p/ridgeline/>. Today's board, with no login at all, is at <http://localhost:8087/p/ridgeline/board>.

The Bookings page: the filter bar, and the table of bookings with time, room, person, extras, amount, status and actions.

Show filters to **Today**, **Upcoming** (the default), **Past** or **Everything**, and a second drop-down filters to one room. The table has **When**, **Room**, **Person**, **Extras**, **Amount**, **Status** and the actions available on that row.

STATUS	MEANING
confirmed	Paid for, or free. The door opens for that slot.
awaiting payment	Made but not paid for. Holds the slot until its expiry window runs out. Grants nothing.
cancelled	Cancelled by the resident or by staff. Grants nothing.
expired	Not paid for in time. The slot went back on sale.

What staff can do:

- **Mark paid** appears on any booking that is awaiting payment. Type the reference the bank actually showed (or leave it blank to record the panel's own `BK-` reference) and press it. The booking confirms and the door opens for that slot from then on.
- **Cancel** works on a pending or a confirmed booking and takes the door access with it. Staff are not bound by the room's cancellation cut-off.
- **Book a room for someone** is a panel at the bottom of the page for a resident who has walked up to the desk. Choose the room, the person, a start time and an optional note. The start must land on the room's slot grid inside its opening hours, or the booking is refused with the reason. A free room confirms at once; a priced one is created awaiting payment with a reference to quote. Staff-made bookings cannot include extras.

Note: there is **no refund function**. Cancelling a booking that was paid for by card does not refund it: do that in Stripe. The panel records access, not money movements.

Every change writes a line to the Log as a **Room booking** system event, so the history of a booking (created, confirmed, cancelled, expired) is auditable alongside the door events (chapter 11).

Settings reference: Bookings and portal

All of these are on the **Settings** page, in the **Bookings and portal** section, per tenant. It has its own **Save bookings settings** button; the main **Save settings** button does not save it.

SETTING	MEANING
Publish the resident portal	Master switch. While unticked the portal address returns "not found".
Portal address	The short name in the portal's web address. Blank uses the tenant's own. Lowercase letters, digits, dash and underscore, up to 40 characters, unique across the panel.
Bank details shown on the payment page	Free text shown to residents paying by transfer: account name, sort code, account number. Blank means no bank option is offered.
Publishable key (Stripe)	Stored for completeness. Payments work from the secret key alone.
Secret key (Stripe)	<code>sk_...</code> . This is the one that matters: with it set, Pay by card appears on the payment page. Shown as a set badge; the field is always blank. Tick Remove the stored secret key and save to erase it.
Webhook signing secret (Stripe)	<code>whsec_...</code> , from the webhook you create in Stripe. Handled the same way as the secret key.
Hold a bank transfer booking for (minutes)	Default 120.
Hold a card booking for (minutes)	Default 30.
Gateway URL, Sender name or number, API key	The SMS gateway that delivers login codes.

Setting Stripe up

1. In the Stripe dashboard, take the **secret key** for the account (test keys while you are trying it out) and paste it into **Secret key**.
2. Add a **webhook endpoint** in Stripe for the event `checkout.session.completed`, pointing at the portal address with `stripe/webhook` on the end, for example `https://doors.example.com/p/riverside/stripe/webhook`.
3. Paste the webhook's signing secret into **Webhook signing secret**.
4. Press **Save bookings settings** and make a test booking.

The webhook is a belt-and-braces measure: a booking also confirms when the resident is returned from Checkout to the panel, so a site with no webhook still works. The webhook covers the case where the resident closes the browser before the redirect completes.

How bookings reach the door

For anyone who wants to know what is actually happening at the lock:

1. Confirming a booking writes an absolute time window, from the lead time before the slot until five minutes after it, against every active fob the person holds.
2. The panel recompiles the room door's keyset and pushes it. Up to seven days of bookings ahead are carried, and up to eight windows per fob.
3. The controller decides locally, as always. It does not ask the panel when someone fobs in during a booking, so a booked room still works if the internet is down when the slot comes round.
4. A minute-by-minute check on the panel re-pushes whenever a window opens or closes, and every confirm, cancel, expiry, staff booking and room setting change pushes immediately.
5. The portal's **Open the door** button and a phone release do not use the keyset at all: the panel sends the controller an unlock command over its existing connection. Those need the panel to be reachable at that moment; the fob path does not.

Limits worth knowing

- **One booking per slot per room.** There is no capacity, group booking or recurring booking.
- **No emails.** The system sends one kind of text message, the login code. There are no booking confirmations, reminders or cancellation notices; the portal is the record.
- **One timezone for the whole panel** (`Europe/London` unless the host changed it), not one per tenant.
- **Opening hours allow two ranges a day** in the editor.
- **Staff bookings cannot include extras**, and **no refunds** are issued from the panel.

Phone access (dial release)

Phone access lets someone open a door by ringing a phone number: no fob, no app. One inbound number serves every site and tenant on the panel. There are two ways a call can open a door:

- **Caller-ID release** for known people. The caller's number is recognised; if they may open exactly one door it opens straight away and the call ends; if several, a short keypad menu asks which.
- **Apartment mode** for guests and anyone whose number is not recognised. The caller keys in an apartment number and then a PIN; the doors ticked for that apartment's resident open. This suits holiday lets, Airbnb, tradespeople and deliveries: give the guest a PIN, change it after they leave.

The same phone line does one more job. A known caller can **enrol a new fob over the phone**: press **#** at the start of the call, key in the site's number, and they have ten minutes to walk to a door and present the new fob, which is then attached to them. That is described under "Phone enrolment" below.

A confirmed room booking also opens a door by phone for the length of the slot, even for someone with no phone-release doors ticked at all (chapter 8).

Every call, allowed or refused, is written to the Log with the number and the outcome (chapter 11).

The phone side runs on a PBX (phone system). MiRTA PBX and other Asterisk-based systems are supported out of the box with the script supplied; anything that can make an HTTPS request can use the same small API.

Setting it up per person

On the person's page, in the **Door release by phone** panel:

The screenshot shows the 'AccessController' web interface. On the left is a sidebar with navigation links: Dashboard, People (selected), Enrolment, Access levels, Schedules, Bookings, Rooms, Log, Doors, Controllers, Flash & adopt, Users, Branding, Networks, and Manual. The main content area is titled 'Elena Petrova' and has a 'Back to people' link. It contains three main panels: 'Details', 'Fobs', and 'Door release by phone'. The 'Details' panel has fields for Name, Email, Phone, Status, and Notes. The 'Fobs' panel shows a table with columns 'code' and 'status', and buttons for 'Add fob', 'Wait for fob at door', and 'Access levels'. The 'Door release by phone' panel has a toggle switch set to 'on', a list of doors to release with checkboxes, and an 'Apartment mode' section with fields for 'Apartment number' and 'Set apartment PIN'. At the bottom, there are buttons for 'Save phone release' and 'Clear PIN', and a 'Move to trash' button.

The person page, showing the door release by phone panel with the allow tick, door ticks, and apartment mode fields.

1. Make sure the person's **Phone** is filled in under Details. Caller IDs are matched on the last nine digits, so the format does not matter.
2. Tick **Allow this person to release doors by phone (caller ID)**.
3. Tick the **doors they may release**. One door means instant release; two or more means the keypad menu.
4. Press **Save phone release**.

The panel heading shows an **on** or **off** badge so you can see at a glance whether the person has phone release.

For **apartment mode** on the same panel:

1. Enter the **Apartment number** (digits only, up to 10). This is what the guest keys in first.

2. Enter a **PIN** of 4 to 10 digits in **Set apartment PIN**. The PIN is stored hashed; the panel never displays it again, so write it down for the guest before leaving the page. The badge shows **PIN set** or **no PIN**.
3. Tick the doors under "Doors they may release" (the same door list is used for both paths). The caller-ID tick is **not** needed for apartment mode.
4. Press **Save phone release**.

To change a guest's PIN, type a new one and save. **Clear PIN** removes it (after confirmation) so the apartment number no longer opens anything. If several people share an apartment number, each has their own PIN and any of them works.

An inactive person cannot release doors by caller ID.

Menu numbering

When a caller may open more than one door the PBX plays a prompt, then speaks the option numbers ("one", "two", ...). Door names themselves are not spoken (the PBX has no text-to-speech). Options are numbered in **alphabetical order of door name**, which is the order the doors appear on the person's page, so tell people "1 is Front Entrance, 2 is Gym Floor".

Phone enrolment

Handing a new fob to somebody usually means a trip to the office. Phone enrolment removes it: the member rings the access line from their own mobile, keys in the building's number, walks to a door and presents the new fob. It is attached to them, with the site's enrolment access level, without anyone touching the panel.

The site number

Because one inbound number serves every building on the panel, the caller has to say which building they are at. That is what the **site number** is for.

Set it on **Settings** under **Site number (phone enrolment)**: 2 to 6 digits, and it must be different from every other site's on the panel (the panel refuses a number already in use and tells you so). Leave it empty and phone enrolment is switched off for that site. Press **Save site number**.

Pick something the residents will not have to look up: the building number, the last digits of the postcode, anything printed by the door.

What the caller does

1. **Ring the access line** from the mobile number recorded on their person page.
2. The line answers and, because it recognises them, offers "*To enrol a new fob, press hash*" and waits about three seconds.
3. **They press #** . (Doing nothing instead lets the call carry on and open the door as normal, so the offer costs a regular caller a three-second pause and nothing else.)
4. The line asks for the site number. **They key it in**. The door is **not** released on this call.
5. **They walk to a door and present the new fob**. The next unknown fob presented at any of the site's enrolment doors, within ten minutes, is attached to them.

The Log records *Phone enrolment armed* when the window opens, and *Fob enrolled at door* with "by phone" in the detail when the fob lands.

What has to be set up first

Phone enrolment arms a window only if the site has **enrolment doors picked on the Enrolment page** (chapter 6). Those ticks are what decides where a caller may present a fob.

- If enrolment mode is already running, the caller gets their ten minutes at the doors it is running at.
- If enrolment mode is **off** but doors are still ticked from the last time, the **#** path switches it on for those doors for ten minutes. It never changes the doors or the access level an administrator chose, and it never shortens or extends a window that is already running longer.
- If no doors are ticked at all, the caller hears the refusal prompt and nothing happens.

The access level the fob arrives with is the **default access level** set in the Enrolment mode panel, and it is added to whatever the person already holds, so a member who is enrolling a replacement fob keeps everything they had.

Safety of it

Phone enrolment is deliberately narrow:

- It only ever attaches a fob to **the person whose number rang in**. It cannot create people and cannot enrol anyone else.

- It **never opens a door**. The call that enrolls is not the call that lets you in.
- It only works at doors an administrator has already put into enrolment mode, and only for ten minutes.
- The site number stops a caller enrolling themselves at somebody else's building. It is not a password, and it should not be treated as one: the protection comes from the caller ID and from the enrolment doors, not from the digits.

Because caller ID can be spoofed (see "Security notes" below), treat phone enrolment the way you would treat caller-ID release: fine for a gym or an apartment block, not for a door where a fob is a security control in its own right. Leaving the site number empty switches it off entirely.

The shared API token (Settings)

The PBX authenticates to the panel with one shared token. It is on **Settings** under **Phone unlock (dial release) API**, visible to super admins only:

- The **endpoint URL** (ending `/api/phone/doors` ; the base of it is what the PBX script needs). **Copy URL** copies it.
- The **token**. **Copy token** copies it.
- **Regenerate** issues a new token and invalidates the old one immediately; the PBX script must then be updated. Do this if the token may have leaked.

One token serves all tenants: it identifies your PBX, not a customer.

Setting up MiRTA PBX (Asterisk AGI)

MiRTA PBX is Asterisk-based and runs custom call logic as **AGI scripts** attached to a **Custom Destination**. The supplied script `door_release.php` (in the product's `docs/pbx` folder) does the whole call: answers, prompts, collects digits, calls the panel, plays the result and hangs up. It is written for the PHP 5.x found on MiRTA boxes.

1. **Upload** `door_release.php` with WinSCP (or scp) into the PBX's AGI folder, the same one your existing AGI scripts live in, normally `/var/lib/asterisk/agi-bin/`. Make it executable (`chmod 755 door_release.php` , or permissions 0755 in WinSCP). It must keep Unix line endings.
2. **Edit the two lines** near the top of the script: `$TOKEN` = the token from Settings; `$BASE` = your panel's public URL, for example `https://doors.example.com` , with no trailing slash. Always use the `https` address.
3. In the PBX tenant, **Custom Destinations > add**: type **AGI script**, choose `door_release.php` . Leave the parameters empty (set "Number of Parameters" to 0); the script reads the caller's number from the AGI environment.
4. **DIDs**: open the inbound number you want to use for door release and set its destination to that Custom Destination. Save.
5. **Test**: ring the number from a phone whose number is set on a person with a door ticked. It should answer, play the granted prompt (or "thank you") and open the door. Then ring from an unknown phone: it should ask for an apartment number and PIN.

The script also sets the channel variable `AGIRESULT` to TRUE or FALSE, so you can optionally follow it with a MiRTA **Condition** (for example, route failures to reception). Not required.

For debugging, the script writes to `/tmp/door_release.log` on the PBX and to the Asterisk console (`asterisk -rvvv` , lines starting `[door_release]`). PHP with curl is normal on a MiRTA box; if curl is missing the script falls back to PHP streams.

Recorded prompts

The script plays custom recordings if present and falls back to built-in Asterisk sounds otherwise. Record 8 kHz, mono, 16-bit PCM WAV files (or GSM) and put them in `/var/lib/asterisk/sounds/door_release/` :

FILE	SUGGESTED WORDING	WHEN IT PLAYS
<code>granted.wav</code>	"Access granted. The door is now open."	A door was released.
<code>choose.wav</code>	"Press the number for the door you want to open."	Before the option numbers are spoken.
<code>denied.wav</code>	"Sorry, that is not recognised. Please contact the office."	Refused, locked out, rate limited, panel unreachable or bad token.
<code>enrol_offer.wav</code>	"To enrol a new fob, press hash."	A known caller, before the door is released.
<code>enter_site.wav</code>	"Enter your site number, then hash."	After the caller has pressed <code>#</code> .
<code>enrol_armed.wav</code>	"Go to the door and present the new fob within ten minutes."	The enrolment window has been armed.
<code>enter_apartment.wav</code>	"Please enter your apartment number, then hash."	Start of the guest path.
<code>enter_pin.wav</code>	"Now enter your PIN, then hash."	Guest path.
<code>wrong.wav</code>	"That apartment number or PIN was not recognised. Please try again."	Guest path retry (up to three attempts).
<code>goodbye.wav</code>	"Goodbye."	Timeouts and after the last failed attempt.

Keep `enrol_offer.wav` short. It plays to every recognised caller before their door opens, and the script waits about three seconds afterwards for a keypress, so a long recording makes every ordinary call feel slow. The prompt can be interrupted: a caller who knows the routine can press `#` while it is still playing.

If a site does not use phone enrolment, record `enrol_offer.wav` as a moment of silence, or leave the site number empty (the panel then never arms a window) and accept the short pause.

Other PBXs: the API

Any phone system that can make an HTTPS POST can drive door release. All calls are `POST` with a JSON body (form encoding is also accepted) and the header `Authorization: Bearer <token>`.

ENDPOINT	BODY	RESULT
<code>/api/phone/doors</code>	<code>{"caller":"+447911123456"}</code>	<code>{"result":"granted","door":"..."}</code> (door opened); <code>{"result":"choose","doors":[{"opt":1,"name":"..."},...]}</code> ; with code <code>unknown_caller</code> or <code>no_doors</code> ; 401 bad token; 429 <code>rate_limited</code> ; 503 <code>no_controller</code> .
<code>/api/phone/doors</code>	<code>{"caller":"...","dry":true}</code>	Identify only, release nothing. <code>{"result":"known","name":"...","doors":[{"opt":1,"name":"..."}</code> or 403 <code>unknown_caller</code> . Call this first if you want to offer enrol before the lock clicks.
<code>/api/phone/release</code>	<code>{"caller":"...","opt":2}</code>	<code>granted</code> , or 403 (<code>unknown_caller</code> , <code>bad_option</code>).
<code>/api/phone/apartment</code>	<code>{"apartment":"12","pin":"4913","caller":"...","opt":N}</code> (<code>opt</code> optional)	<code>granted</code> , <code>choose</code> (then repeat with <code>opt</code>), 403 <code>bad_pin</code> or <code>no_locked_out</code> with <code>retry_after</code> seconds.
<code>/api/phone/enrol</code>	<code>{"caller":"...","site":"4021"}</code>	<code>{"result":"armed","minutes":10,"doors":["Front","Gym"]}</code> ; 4 <code>unknown_caller</code> , <code>bad_site</code> (no site has that number) or <code>wrong</code> number belongs to a different site from the caller's; 409 <code>no_e</code> 429 <code>rate_limited</code> .

Suggested call flow: answer; call `/doors` with `"dry":true` to identify the caller; if the result is `known` , play the enrolment offer and wait about three seconds for a `#` ; if `#` was pressed, collect the site number and call `/enrol` , play the armed prompt and hang up **without** opening anything; otherwise call `/doors` for real, and on `granted` play the granted prompt, on `choose` play the menu, collect a digit and call `/release` ; on any 403 fall into the apartment path: collect apartment and PIN, call `/apartment` (again handling `choose`), allow three attempts.

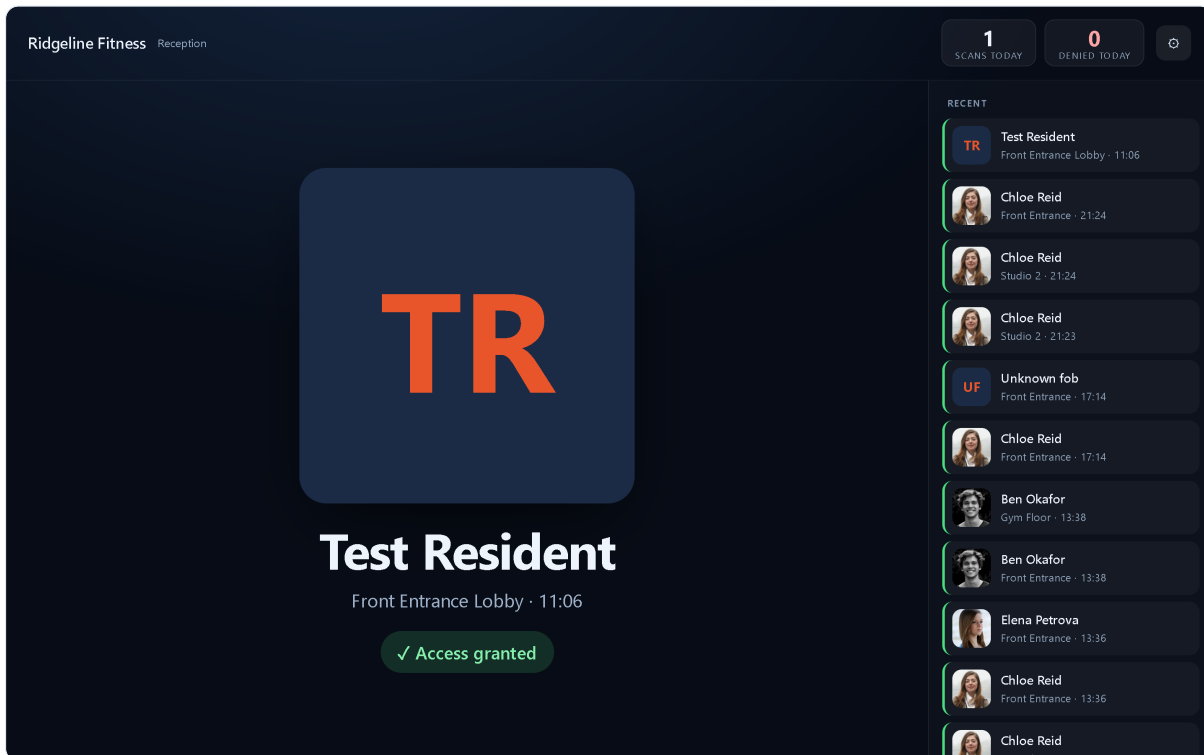
Note: collect the enrolment keypress with a "wait for one digit" primitive, not with a digit-collection primitive. Most systems, Asterisk included, treat `#` as the terminator of a digit string, so a `#` pressed on its own is reported as "nothing was pressed" and the offer appears to be ignored.

Security notes

- **Caller ID can be spoofed.** Some telephony providers let the caller present any number. Caller-ID release is therefore convenience-grade: right for a gym member or an employee, not for a high-security door. Where a real secret matters, use apartment mode (a PIN) or a fob. Both paths are logged.
- **Apartment PINs** are stored as bcrypt hashes and never shown. After five consecutive wrong PINs for an apartment the panel locks that apartment for 60 seconds, doubling with each further failure up to 16 minutes; a correct PIN during a lockout is still refused. Unknown apartment numbers take the same time to answer as wrong PINs so a caller cannot tell them apart.
- **Phone enrolment** attaches a fob only to the person whose number rang in, only at doors already in enrolment mode, only for ten minutes, and it never opens a door. The site number keeps a caller out of another building's enrolment window; it is not a secret and is not a substitute for the caller ID check. Leave the site number empty to switch phone enrolment off for a site.
- **Rate limiting:** the phone API allows a burst of 10 requests and then about 2 per second per source IP address. A PBX making normal calls never hits this.
- **The token** authenticates the whole PBX. Keep it out of tickets and email; if it leaks, **Regenerate** it on Settings and update `$TOKEN`.
- Always point the PBX at the `https` address so the token and numbers are not sent in clear.
- A release by phone always sends the unlock to the door's active controller; if the door has no active controller the call is refused and logged (`no_controller`).

Reception screen

The reception screen is a full-screen web page for a laptop, tablet or wall monitor at the front desk. It shows each person's photo and name the moment they fob in, granted or denied, with the last dozen entries down the side and today's scan and denied counts at the top. Staff can greet people by name and spot a refused scan or an unfamiliar face immediately.



The reception screen: tenant name and today's counts at the top, the latest person large in the middle with a granted or denied verdict, recent entries on the right.

How it works

- The screen is a **public page with a secret link**. Nobody logs in on the front-desk machine; the link itself is the key. Anyone with the link can watch live entries and see people's photos, so treat it like a password.
- It updates every 2.5 seconds and shows *access* events only (not door open/close, exit-button, alert or system events). Fob scans and phone releases show the person; a remote unlock from the panel has no person and appears as a granted "Unknown fob" entry, as does a genuinely unknown fob. Exit-button presses are door events and are not shown.
- Photos come from the People pages (chapter 6); people without a photo show their initials.
- It uses the tenant's accent colour from Branding and always uses a dark theme suited to a screen that is on all day.

Setting up a screen

1. Sign in to the panel on the reception computer, open **Doors**, and in the **Reception screen** panel press **Open screen** (or **Copy** the link and paste it into the browser). Then sign out of the panel; the screen keeps working.
2. Put the browser into full-screen or kiosk mode (F11 in most browsers). On Windows you can create a shortcut that starts the browser with `--kiosk <link>` so it opens automatically.
3. Press the **gear** button at the top right of the screen to choose which doors it watches (see below), then **Done**.
4. Turn off screen sleep on the machine, or accept that staff will wake it.

The page needs to reach the panel over the network. Nothing else is required on the machine.

Door filter

A screen watching every door of a large site can be noisy. The gear button opens **Show entries from...** with a tick box per door. Untick the doors that should not appear (for example the staff back door) and press **Done**. The choice is saved on that computer only (in the browser), so two screens can watch different doors from the same link, and the list resets after **Done** so it starts clean with the new filter.

The counts at the top ("scans today", "denied today") follow the same door filter.

Regenerating the link

On **Doors > Reception screen**, press **Regenerate** (a change permission on the Doors page). A new secret link is created immediately and the old one stops working: any screen still on the old link shows nothing and must be reopened with the new link. Regenerate when a laptop that had the link is lost or leaves the business, when a member of staff who knew it leaves, or on a routine schedule.

What the screen shows for a denial

A denied scan shows the person (if the fob is known) or "Unknown fob", in red, with the reason code: `unknown_fob`, `fob_revoked`, `member_inactive`, `no_access_level` or `outside_schedule` (chapter 11 explains each). This is enough for the desk to say "your membership needs renewing" without opening the panel. The fob code itself is not shown on the screen; look it up in the Log if needed.

Log and reporting

Everything the system does is written to the **Log**: every scan at every door, every phone release, every opening and closing of a door that has a contact, every exit-button press, every alert from a door, and every administrative event that affects access. Each tenant sees only its own log.

AccessController

Ridgeline Fitness

Dashboard

People

Enrolment

Access levels

Schedules

Bookings

Rooms

Log

Doors

Controllers

Flash & adopt

Users

Branding

Tenants

admin

SUPER-ADMIN

Manual

Log out

Access log

Every scan, alert and system event - most recent first.

FILTER

All kinds

Granted + denied

All doors

Apply

WHEN	KIND	DOOR	PERSON	FOB	RESULT	REASON
Tue 08 Sep 11:06	access	Front Entrance Lobby	Test Resident		granted	Opened from portal opened from the portal, booking 1
Tue 08 Sep 11:06	system	Front Entrance Lobby	Test Resident			Room booking created by staff (confirmed): Front Entrance Lobby Tue 08 Sep 12:00-13:00
Tue 08 Sep 11:06	system		Test Resident			Room booking self-registered on the resident portal, apartment 55
Mon 07 Sep 21:24	access	Front Entrance	Chloe Reid		granted	Phone unlock phone unlock by Chloe Reid, caller 07700900103
Mon 07 Sep 21:24	access	Studio 2	Chloe Reid		granted	Phone unlock phone unlock by Chloe Reid, caller 07700900103
Mon 07 Sep 21:23	access	Studio 2	Chloe Reid		granted	Phone unlock phone unlock by Chloe Reid, caller 07700900103
Mon 07 Sep 21:23	alert	Studio 2				Door held open open 90s
Mon 07 Sep 17:14	access	Front Entrance			granted	Lock released by command
Mon 07 Sep 17:14	access	Front Entrance	Chloe Reid		granted	Phone unlock phone unlock by Chloe Reid, caller 07700900103, controller confirmed release
Mon 07 Sep 13:47	door	Front Entrance				Door closed

The Log page: filters for kind, result and door, and the event table with time, kind, door, person, fob, result and reason.

Reading the log

Open **Log** in the sidebar. The newest events are at the top; the page shows the most recent 200 matching the filter. Each row has:

COLUMN	MEANING
When	Date and time in the panel's timezone. For scans this is the controller's own timestamp at the moment of the scan (so events that were queued while offline still show the right time); if the controller had no valid clock the arrival time is used. Rows imported from an AC8000 backup keep the original date and time from the old system.
Kind	access , door , alert or system (see below).
Door	The door name, or the controller ID if the controller is not mapped to a door.
Person	The person the fob belonged to at the time, if known. Blank for unknown fobs and for exit-button and remote unlocks.
Fob	The fob code, in hexadecimal.
Result	granted or denied for access events; blank for door events, alerts and system events.
Reason	The reason in plain English, followed by any detail (for example the caller's number for a phone unlock, or how long a door was held).

The **dashboard** shows the last 15 events in the same format, plus today's counts.

Filters

The toolbar at the top filters the list; press **Apply**.

- **Kind:** All kinds, Access, Door open/close, exit button, Alerts or System.

- **Result:** Granted + denied, Granted only, Denied only. (Only meaningful for access events.)
- **Door:** All doors or one door.

The dashboard's red "Denied today" tile links straight to the log filtered to denied events. Filtering by person is done from the person's fobs: search the log page in your browser for the fob code, or use the People search to confirm whose fob a code is.

Event kinds

- **Access:** an attempt to open a door, whether by fob, remote unlock or phone. Always granted or denied with a reason. These are the events the dashboard's scan and denied counts and the reception screen are built from.
- **Door:** the door contact changing state (door opened, door closed) and exit-button presses. Neutral: no result, not counted as a scan or a denial, not shown on the reception screen. Only produced by controllers with a door sensor or exit button wired and the current firmware.
- **Alert:** something the door reported on its own: forced open, held open, or a failed update. Forced and held-open alerts can also be sent to Telegram (chapter 4).
- **System:** administrative events: a controller registered, enrolment started or stopped, a phone enrolment window armed, a fob captured or enrolled, an import from CSV or an AC8000 backup, a person moved to or restored from the trash, and every change to a room booking.

Imported history

When a site is migrated from an AC8000 / iCCard3000 system with the import wizard (chapter 6), the old system's swipe records can be brought into the Log. They appear as ordinary **access** events, granted or denied, and are told apart from the panel's own events by their detail, which reads **AC8000 history, reader (reader name)**, for example *AC8000 history, reader HiveDoor-In*. The reader name is the one the old system used for that side of the door; the **Door** column shows the door the wizard derived from it (or created).

Imported rows carry the **original timestamps** from the backup, so they sort into the Log at the dates and times the swipes really happened, not at the time of the import; filter by door or result and they behave like any other event. Where the card belongs to a person on the panel the **Person** column shows their name; a swipe by a card that was not in the backup's people list shows no person, and if the old system had refused it the row is *denied* with the reason *Unknown fob*. The import itself is recorded once as a system event with the counts.

Every reason code

The panel shows reasons in words. This table gives the code the controller and the API use, the words shown, and what to do about it.

Access decisions

CODE	SHOWN AS	MEANING AND WHAT TO DO
ok	OK	Granted: the fob is active, the person is active, one of their levels includes this door and the time is inside its schedule.
unknown_fob	Unknown fob	Denied: the code is not on any person in this tenant. A new member's fob has not been added, a visitor tried their fob from elsewhere, or the reader sent a different format. Use the desk scan or Enrolment to attach it, or Check access to test it.
fob_revoked	Fob revoked	Denied: the fob is on a person but its status is revoked. Re-activate it on the person's page if that was a mistake.
member_inactive	Person inactive	Denied: the person's status is inactive. Everything else is fine; set them active to restore access.
no_access_level	No access level for door	Denied: the person is active but none of their access levels includes this door (or they hold no levels). Add the right level.
outside_schedule	Outside schedule	Denied: a level includes the door but the current day and time are outside that level's schedule. Check the schedule, and check that the panel timezone is right if it seems off by an hour.
enrol	Enrolment (auto-accepted)	Granted: enrolment mode was on and an unknown fob was let in and captured. Assign it on the Enrolment page.
remote_unlock	Remote unlock (panel)	Granted: someone pressed Unlock on the Controllers page. No person is recorded. When the controller confirms it carried the unlock out, that confirmation is folded into this same row rather than logged twice.
cmd_unlock	Lock released by command	Granted: a controller reported that it released the lock on command, but the panel had no matching row to fold it into. In practice this means a release the panel did not initiate, or one whose originating row was more than about twenty seconds old.
phone_unlock	Phone unlock	Granted (or denied) via the phone line. The detail says who and from which number, or why it was refused: unknown caller, no release doors, wrong PIN, locked out, no active controller.
portal_unlock	Opened from portal	Granted: a resident pressed Open the door on the booking portal during a confirmed slot (chapter 8). The detail gives the booking number. If no controller was online the row is denied instead.

Door events

CODE	SHOWN AS	MEANING
door_opened	Door opened	The door position sensor saw the door open. Stamped with the controller's time. If there was no grant or exit press just before, a <i>Door forced</i> alert follows.
door_closed	Door closed	The door position sensor saw the door close again. The gap between this and the preceding <i>Door opened</i> is how long the door stood open.
rex	Exit button pressed	The exit button on the inside was pressed and the door released. Free egress; no person is recorded.

Contact changes within 200 ms of the previous one are ignored by the controller, so a rattling reed switch does not fill the log.

Alerts from the door

CODE	SHOWN AS	MEANING AND WHAT TO DO
door_forced	Door forced	The door-position sensor saw the door open with no grant, exit press or remote unlock in the last few seconds. Someone forced it, propped it, or the exit button is bypassed. Also raised if the sensor is faulty or wired the wrong way round. Sent to Telegram if enabled (chapter 4).
door_held_open	Door held open	The door stayed open longer than the door's Alert if held open time (60 seconds by default, set per door in chapter 3; 0 = off). Someone is holding it, it is wedged, or the closer is weak. The detail says how long. Sent to Telegram if enabled (chapter 4).
ota_failed	ota_failed	An over-the-air update failed; the detail has the reason. The controller keeps running its old firmware.
tamper , power	tamper, power	Reserved in the protocol for enclosure tamper and power alerts; not raised by the current firmware.

System events

CODE	SHOWN AS	MEANING
registered	Controller registered	A controller announced itself for the first time; its firmware and IP are in the detail. Onboard it on the Controllers page.
captured	Captured for enrolment	A door was listening (chapter 6) and an unknown fob was captured there. It is on the Enrolment page. Not a denial.
fob_enrolled	Fob enrolled at door	A fob captured at a listening door was attached straight to a person.
enrol_started	Enrolment started	Enrolment mode was switched on; the detail gives the duration.
enrol_stopped	Enrolment stopped	Enrolment mode ended (manually or by timer).
enrol_armed	Phone enrolment armed	A recognised caller pressed # and keyed in the site number, opening a ten-minute window to present a new fob (chapter 9). The detail names the person, the doors and the calling number.
trashed	Moved to trash	A person was moved to the trash, singly or in a bulk action (chapter 6).
trash_restored	Restored from trash	A person was restored from the trash.
trash_deleted	Deleted permanently	A person in the trash was deleted for good.
trash_purged	Trash purged	The automatic 30-day sweep removed people from the trash. The detail gives the count.
booking	Room booking	A room booking was created, confirmed, cancelled or expired, or somebody self-registered on the resident portal (chapter 8). The detail says which, with the room and slot.
import	CSV import	An import ran (from a CSV or from an AC8000 backup; the label says CSV either way); the detail gives the counts created, attached, skipped and failed, and the number of history records added.
ota_start	ota_start	A controller began downloading a firmware update.
setcfg	setcfg	A controller received a re-point command and is restarting on its new server.

Using the log

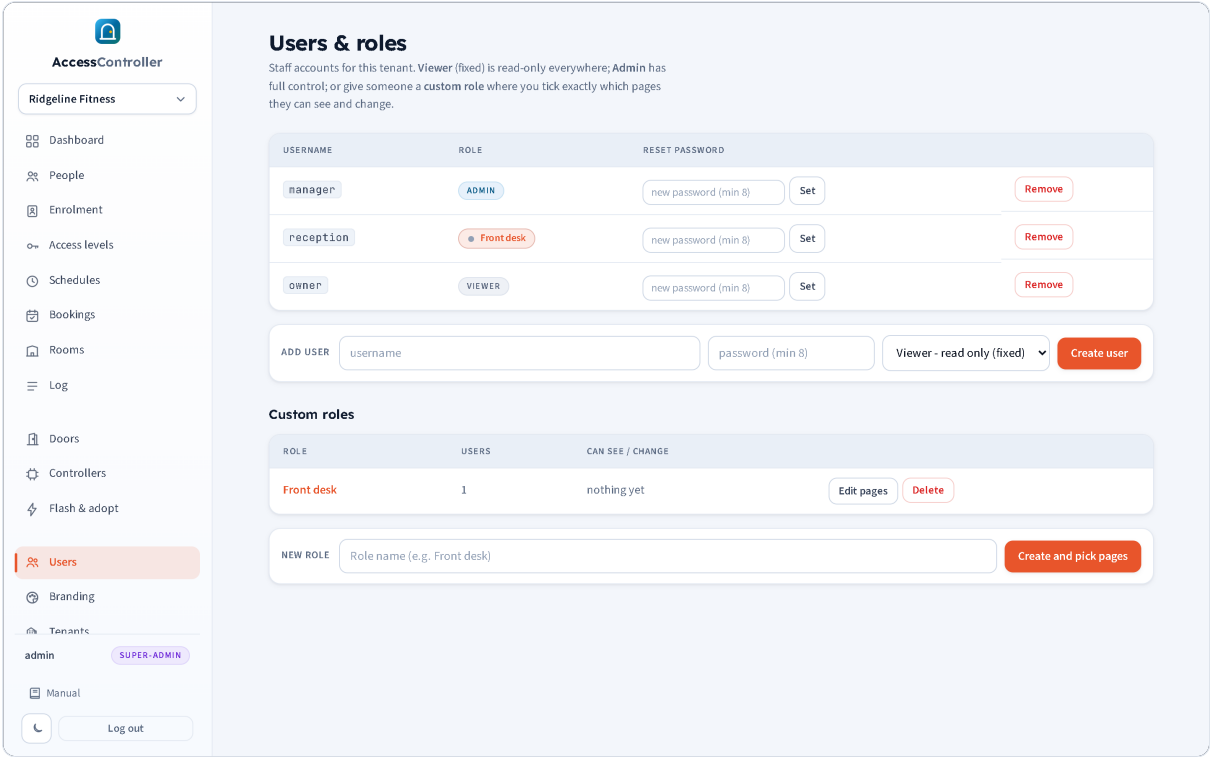
- **Who was in the building?** Filter Kind = Access, Result = Granted, pick the door. Names and times are in order.
- **Why was someone refused?** Filter Result = Denied. The reason column tells you exactly which rule failed; the person's page is a click away via People search.

- **Is a door being propped?** Filter Kind = Alerts. Repeated *Door held open* on one door points to a closer or a habit. Switch to Kind = Door open/close, exit button and pick the door to see each *Door opened* / *Door closed* pair and how long the door actually stood open.
- **Who let someone out?** Filter Kind = Door open/close, exit button. *Exit button pressed* lines show every release from the inside, with the door and time.
- **Who used a booked room?** Filter Kind = System and read the *Room booking* lines for the booking's history, then Kind = Access on that door for *Opened from portal*, *Phone unlock* and fob entries during the slot (chapter 8).
- **Audit an administrator's actions:** System events record enrolment windows, imports, controller registrations, deletions to the trash and restorations, with times. Panel logins and edits to a person's details are not written to this log in the current release (see chapter 15).

Events are kept indefinitely; the database grows by a few hundred bytes per event. The host can archive or trim old events at the database level (chapter 16).

Users, roles and permissions

A **user** is a staff login for the panel. Users belong to one tenant and have a role that decides which pages they can see and which they can change. People who open doors (chapter 6) are not users and never log in.



The Users page: the user list with role badges and password reset, the add-user toolbar, and the custom roles section.

The built-in roles

ROLE	CAN SEE	CAN CHANGE	NOTES
Super admin	Everything in every tenant, plus the Tenants page	Everything	The provider's account. Switches between tenants. Created only from the server command line (chapter 16), never from the panel.
Admin	Every page in their own tenant	Everything in their tenant	The normal account for an owner or office manager. Can create other admins, viewers and custom roles.
Viewer	Every operational page in their tenant (not Settings)	Nothing	Read-only. Edit controls are hidden, not merely disabled, and any attempt to post a change is refused. Good for front-desk staff who only need to look people up and watch the log.
Custom	Exactly the pages ticked in the role	Exactly the pages ticked as "change"	Built by an admin on the Users page.

The role of the signed-in user is shown as a badge at the bottom of the sidebar.

Creating a user

On **Users**, in **Add user**: type a username, a password of at least 8 characters, choose **Viewer**, **Admin** or one of your custom roles, and press **Create user**. Usernames must be unique across the whole panel. The new user can sign in straight away.

Resetting a password

There is no self-service reset. In the user's row type a new password (at least 8 characters) into **Reset password** and press **Set**. Tell the user the new password out of band and ask them to change it. (Users change their own password the same way, on their own row.)

Removing a user

Press **Remove** on the row (not available on your own account). Any session the user has is ended.

Custom roles

A custom role is a tick list of pages. Open **Users**, type a role name in **New role** ("Front desk", "Membership team", "Installer") and press **Create and pick pages**.

PAGE	VIEW	CHANGE
Dashboard	☐	-
People	☐	☐
Enrolment & import	☐	☐
Doors	☐	☐
Controllers	☐	☐
Access levels	☐	☐
Schedules	☐	☐
Bookings	☐	☐
Log	☐	-
Flash & adopt	-	☐
Users & roles	-	☐
Branding	-	☐

The role editor: the page tree with a View and a Change tick per page, and quick-set buttons.

The editor shows the pages as a tree in sidebar order. For each page:

- **View**: the user can open the page and see its contents. The page appears in their sidebar.
- **Change**: the user can also add, edit and delete on that page. Ticking Change ticks View as well; unticking View unticks Change.

Some pages behave differently:

- **Dashboard** and **Log** are view-only; there is nothing on them to change, so they only have a View tick. (The Check access tool on the dashboard is allowed with View, since it changes nothing.)
- **Flash & adopt**, **Users & roles** and **Branding** are all-or-nothing action pages: they only have a Change tick, and having it means full use of the page.
- **Enrolment & import** is a child of **People**; it can be granted on its own.
- **Bookings** covers both the **Bookings** and the **Rooms** pages (chapter 8). There is no way to grant one without the other: anyone who can see bookings can also change the rooms they are made against.
- **Tenants** cannot be granted to anyone but a super admin.

Quick set buttons tick all View, tick all Change, or untick everything. Press **Save role**. Pages not ticked disappear from the user's sidebar, and opening them by address redirects the user to the first page they may see with the message "You do not have access to that page". Any attempt to post a change to a page without the Change tick is refused with "read-only account".

Assign a custom role when creating a user (the role appears in the role drop-down as "custom role"). On the Users page the role name is a link to its editor, and the **Custom roles** table shows how many users each role has and a summary of what it grants. **Delete** on a role turns its users into viewers.

Suggested roles

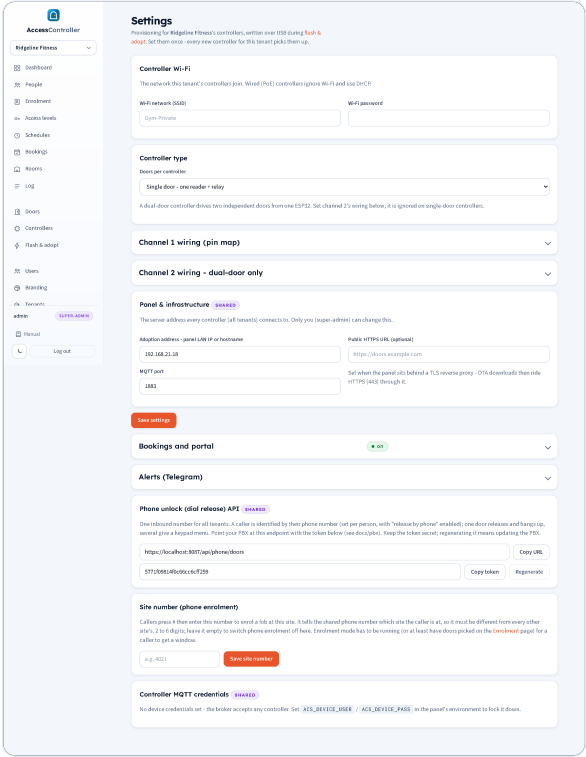
ROLE	VIEW	CHANGE	FOR
Front desk	Dashboard, People, Log	(none)	Reception staff who look people up and read the log. Or simply use Viewer.
Membership team	Dashboard, People, Log	People, Enrolment & import	Staff who add members and fobs but must not touch doors or schedules.
Installer	Doors, Controllers	Doors, Controllers, Flash & adopt, Settings	A contractor commissioning hardware who should not see people.
Manager	everything except Users and Settings	People, Enrolment, Access levels, Schedules, Doors	A site manager who runs the day to day but does not administer accounts.
Concierge	Dashboard, People, Log, Bookings	Bookings	Front-desk staff for a block with bookable rooms: they can take a booking at the desk, mark a bank transfer paid and cancel, but cannot change people or doors.

What permissions do not cover

- Per-door or per-person restrictions (for example "may only edit members, not staff") are not available; permissions are per page.
- Viewers and custom roles are always confined to their own tenant. Only super admins cross tenants.
- Super admins cannot be created or promoted from the panel. The provider creates them with the `create-admin` command on the server (chapter 16).

Settings reference

The **Settings** page holds the values that are written into every new controller for the tenant, the tenant's booking, portal and Telegram alert settings, the site number for phone enrolment, and (for super admins) the shared infrastructure settings that apply to every tenant on the panel. Set them once; every controller flashed and adopted afterwards picks them up. Changing a pin map or Wi-Fi setting later does **not** reach controllers already in service until they are re-provisioned over USB or Bluetooth, or re-pointed from the Controllers page.



The Settings page: controller Wi-Fi, controller type, the two wiring accordions, panel and infrastructure, the phone API token, the site number and the MQTT credential note. The Bookings and portal and Alerts (Telegram) accordions sit below Panel & infrastructure.

Viewers cannot see Settings. Admins and custom roles with the Settings tick see the tenant sections; the sections marked **shared** are visible and editable by super admins only.

Enrolment mode

There is **no enrolment control on this page any more**. Enrolment mode is started on the **Enrolment** page, where you tick the doors it should run at, choose the default access level and pick 15, 30, 60 or 120 minutes (chapter 6).

While it is running, a green **Enrolment mode** panel appears at the top of Settings showing where it is running, when it ends and which level new fobs get, with a **Stop enrolment now** button. That is all Settings does with it. When enrolment is off, the panel is not shown at all.

Controller Wi-Fi

FIELD	MEANING
Wi-Fi network (SSID)	The wireless network the tenant's controllers join. Use a private, staff-only network with a strong password rather than guest Wi-Fi. 2.4 GHz only (the ESP32 has no 5 GHz radio).
Wi-Fi password	Its password. Stored on the panel and written to the controller at provisioning.

Wired (PoE) controllers ignore these two fields and use DHCP on the cable.

Controller type

FIELD	MEANING
Doors per controller	Single door - one reader + relay or Dual door - two readers + relays . The default for controllers provisioned from this tenant. The Flash & adopt wizard has a "Doors on this controller" selector that overrides this per unit. A dual-door unit drives two independent doors from one board using the channel 1 and channel 2 pin maps.

Channel 1 wiring (pin map)

The GPIO numbers the controller uses for its first (or only) door. Defaults suit a generic classic ESP32 devkit and the Olimex ESP32-POE-ISO. Chapter 5 has the per-board tables, including the different safe pins for an ESP32-S3.

FIELD	DEFAULT	MEANING
Wiegand D0 (green)	32	Reader data-0 line.
Wiegand D1 (white)	33	Reader data-1 line.
Relay	4	Output pin that drives the lock relay.
Relay active level	Active high	Active high: the pin goes HIGH to energise the relay. Active low: the pin goes LOW to energise. Match the relay module: many opto-isolated modules are active low. Getting this wrong leaves the relay energised at rest (a strike permanently released or a maglock permanently dropped).
Exit (REX) button	14	Input for the exit button, to 0 V. -1 if there is no button.
Door position sensor	15	Input for the door contact, to 0 V, closed when the door is closed. -1 if there is no sensor; this also disables forced and held-open alerts and door opened/closed logging for the door.
Relay pulse (ms)	4000	How long the relay stays energised after a grant, exit press or remote unlock. 3000 to 6000 ms is usual.

The held-open alert time is not on this page: it is set per door as **Alert if held open for N seconds** on the door's own page (chapter 3) and pushed to the controller with its keyset.

Channel 2 wiring - dual-door only

The same seven fields for the second reader and relay of a dual-door controller. Ignored on single-door controllers. The pins must not clash with channel 1. Panel defaults: D0 25, D1 26, relay 27, REX 13, door 2, active high, 4000 ms. On an Olimex POE board GPIO 25, 26 and 27 belong to the Ethernet PHY; choose others.

Panel & infrastructure (shared, super admin only)

FIELD	MEANING
Adoption address - panel LAN IP or hostname	The address every controller (all tenants) connects to for MQTT, written into controllers at provisioning. For a hosted panel this is its public hostname (for example <code>doors.example.com</code>); for a panel on the site LAN it is the panel machine's IP. The panel pre-fills its own LAN IP on a fresh install. It is also used to build the download address for over-the-air updates when no public HTTPS URL is set.
MQTT port	The port controllers connect to, 1883 by default.
Public HTTPS URL (optional)	Set when the panel sits behind a TLS reverse proxy such as Caddy (chapter 16), for example <code>https://doors.example.com</code> . Over-the-air firmware downloads then ride HTTPS on port 443 through the proxy instead of the panel's plain HTTP port.

Press **Save settings** to save everything above. The sections below each have their own save button and are not covered by it.

Bookings and portal

A collapsible section, per tenant, that publishes the resident booking portal and holds its payment and messaging settings. It shows a green **on** badge while the portal is published. Chapter 8 covers rooms, bookings and the portal in full.

FIELD	MEANING
Publish the resident portal	Master switch. While unticked the portal address returns "not found" and nobody can book.
Portal address	The short name in the portal's web address, for example <code>riverside</code> in <code>https://doors.example.com/p/riverside/</code> . Blank uses the tenant's own short name. Lowercase letters, digits, dash and underscore, up to 40 characters, and unique across every tenant on the panel.
Bank details shown on the payment page	Free text (account name, sort code, account number) shown to residents paying a priced booking by transfer. Blank means no bank option is offered.
Publishable key (Stripe)	Stored for completeness; card payments work from the secret key alone.
Secret key (Stripe)	The <code>sk_...</code> key. With it set, Pay by card appears on the resident's payment page. Shown as a set badge; the input is always blank, and blank keeps the stored key. Tick Remove the stored secret key and save to erase it.
Webhook signing secret (Stripe)	The <code>whsec_...</code> value from the Stripe webhook. Handled the same way as the secret key. Point a Stripe webhook for <code>checkout.session.completed</code> at the portal address with <code>stripe/webhook</code> on the end.
Hold a bank transfer booking for (minutes)	How long an unpaid bank-transfer booking keeps its slot before expiring. Default 120.
Hold a card booking for (minutes)	The same for a card booking. Default 30.
Gateway URL, Sender name or number, API key (SMS)	The gateway that delivers portal login codes. The panel sends it a JSON <code>POST</code> of <code>to</code> , <code>from</code> and <code>text</code> with an <code>Authorization: Bearer</code> header. With no URL set, login codes are only written to the server log and shown on the Bookings page , which is for commissioning only and must never be how real residents receive codes.
Save bookings settings	Saves this section.

Site number (phone enrolment)

FIELD	MEANING
Site number	2 to 6 digits. A caller who presses # on the access line keys this in to say which building they are at, so it must differ from every other site's on the panel; the panel refuses a number already in use and says which. Leave it empty to switch phone enrolment off for this site.
Save site number	Saves it.

The site number on its own does nothing: phone enrolment also needs enrolment doors ticked on the Enrolment page. Chapter 9 has the caller's side and chapter 6 the enrolment doors.

Alerts (Telegram)

A collapsible section, per tenant, that sends a Telegram message when a door is forced or held open. It shows a green **on** badge in its heading while alerts are enabled. Chapter 4 has the step-by-step set-up, how to get a chat ID and the troubleshooting table.

FIELD	MEANING
Enable Telegram alerts	Master switch. Nothing is sent while unticked.
Bot token	The token from @BotFather. Shown as a set / not set badge; the input is always blank. Blank keeps the stored token, a value replaces it.
Forget the stored token	Shown once a token is stored. Tick and save to erase it.
Chat ID	The chat, group or channel to post into. Positive for a private chat, negative for groups and channels (often starting <code>-100</code>).
Door forced	Send on <i>door forced</i> . Default on.
Door held open	Send on <i>door held open</i> . Default on.
Save alert settings	Saves the section (it has its own button; Save settings above does not save it).
Find my chat ID	Saves, then lists every chat the bot has seen with a Use this button per row.
Send test message	Saves, then sends "Test from Access Controller (tenant name)" to the chat and reports success or the reason for failure.

Repeat alerts of the same type for the same door are rate-limited to one Telegram message per minute. These settings take effect immediately and are not written to controllers.

Phone unlock (dial release) API (shared, super admin only)

FIELD	MEANING
Endpoint URL	The address the PBX posts to, ending <code>/api/phone/doors</code> . Copy URL copies it. It is derived from the public URL when set.
Token	The shared bearer token that authenticates the PBX. Copy token copies it.
Regenerate	Issues a new token; the old one stops working at once and the PBX script must be updated.

Chapter 9 covers the PBX side and the security notes.

Controller MQTT credentials (shared, super admin only)

This section is informational. It shows whether the panel's broker requires a username and password from controllers:

- If `ACS_DEVICE_USER` and `ACS_DEVICE_PASS` are set in the panel's environment (chapter 16), the username is shown here and the credentials are written into every controller at provisioning and carried over by re-point unless you type different ones.
- If they are not set, the broker accepts any controller. That is acceptable on an isolated LAN and not acceptable on the internet: set the variables to lock the broker down.

The credentials cannot be changed from the panel; they belong to the server's configuration.

Where each setting takes effect

SETTING	WRITTEN TO CONTROLLERS...
Wi-Fi, adoption address, port, MQTT credentials, pin maps, channels	At Flash & adopt (USB or Bluetooth), via the setup portal, or by Re-point on the Controllers page. Never pushed silently to a running controller.
Enrolment mode (started on the Enrolment page)	Immediately, inside the next keyset push to the doors it runs at.
Bookings and portal	Not written to controllers. A confirmed booking is written into the room door's keyset as a time window and pushed straight away.
Site number	Not written to controllers. Used by the panel when the PBX calls the enrolment endpoint.
Held-open alert time (per door, on the door's page)	Immediately, inside the next keyset push to that door's controller. Needs the current firmware.
Alerts (Telegram)	Not written to controllers. Used by the panel the moment an alert arrives from a door.
Public HTTPS URL	Used by the panel when you press Update on a controller.
Phone API token	Used by the panel when the PBX calls; the PBX needs the new value.

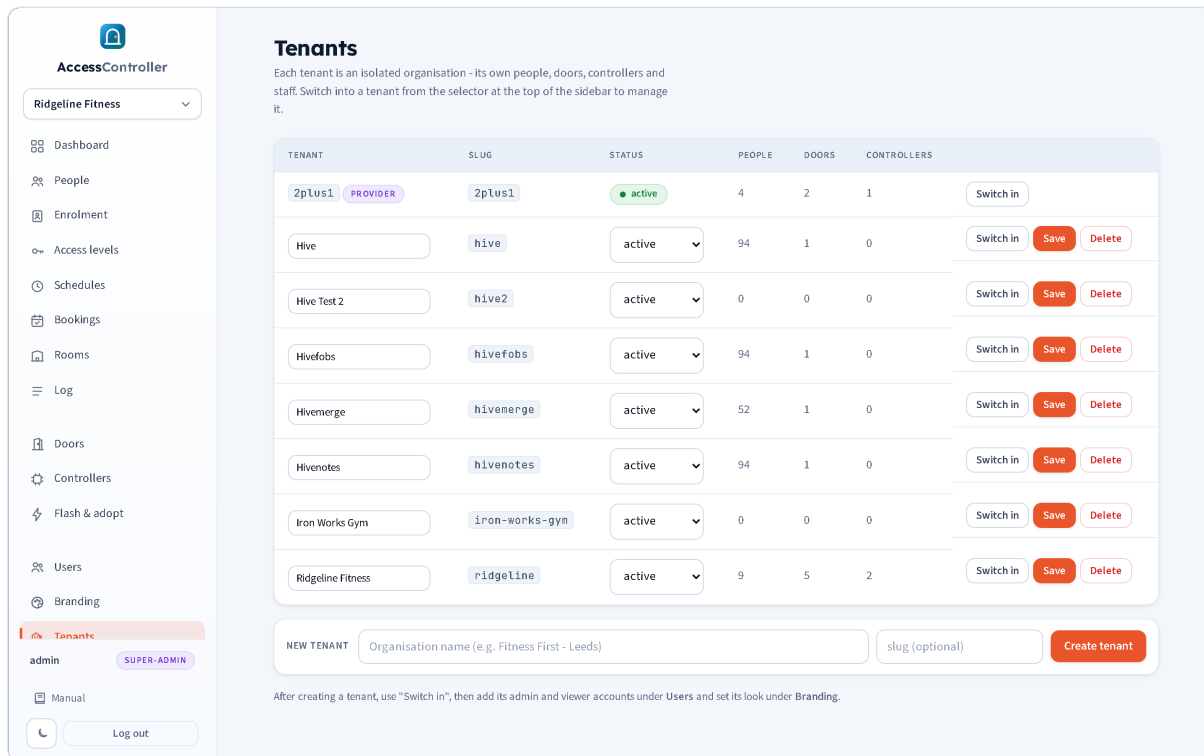
Multi-tenant / provider guide

One Access Controller panel can serve many customers. Each customer is a **tenant**: an isolated organisation with its own people, fobs, doors, controllers, access levels, schedules, log, staff logins, settings and branding. A **super admin** (the provider) sees every tenant and can switch into any of them; a tenant's own admins and viewers see only their tenant and do not know the others exist.

The provider's own organisation is also a tenant, marked **provider** on the Tenants page. It was created when the panel was first installed and owns any data that existed before multi-tenancy.

The Tenants page

Open **Tenants** (super admins only).



Tenants

Each tenant is an isolated organisation - its own people, doors, controllers and staff. Switch into a tenant from the selector at the top of the sidebar to manage it.

TENANT	SLUG	STATUS	PEOPLE	DOORS	CONTROLLERS	
2plus1 PROVIDER	2plus1	active	4	2	1	Switch in
Hive	hive	active	94	1	0	Switch in Save Delete
Hive Test 2	hive2	active	0	0	0	Switch in Save Delete
Hivefobs	hivefobs	active	94	1	0	Switch in Save Delete
Hivemerge	hivemerge	active	52	1	0	Switch in Save Delete
Hivenotes	hivenotes	active	94	1	0	Switch in Save Delete
Iron Works Gym	iron-works-gym	active	0	0	0	Switch in Save Delete
Ridgeline Fitness	ridgeline	active	9	5	2	Switch in Save Delete

NEW TENANT Organisation name (e.g. Fitness First - Leeds) slug (optional) **Create tenant**

After creating a tenant, use "Switch in", then add its admin and viewer accounts under Users and set its look under Branding.

The Tenants page: every tenant with slug, status, people, door and controller counts, Switch in, Save and Delete, and the new-tenant toolbar.

Each row shows the tenant's name, its **slug** (a short identifier derived from the name, letters, digits and dashes only), its status, and how many people, doors and controllers it has.

Creating a tenant

In **New tenant** type the organisation name (for example "Iron Works Gym"), optionally a slug (otherwise one is made from the name), and press **Create tenant**. Slugs must be unique.

Then, in this order:

1. **Switch in** to the new tenant (the button on its row, or the drop-down at the top of the sidebar).
2. **Users**: create the customer's first admin account (and any viewers). Chapter 12.
3. **Branding**: upload their logo and set their accent colour. Chapter 2.
4. **Settings**: enter their controller Wi-Fi, controller type and pin maps. Chapter 13.
5. **Doors**: add their doors. Chapter 3.
6. **Controllers**: register the IDs of the controllers you will fit, then flash and adopt them. Chapter 5.

Everything you create while switched in belongs to that tenant.

Suspending and renaming

Change the name in the row and press **Save**. Set **Status** to **suspended** to mark a tenant as not in service (for example an unpaid account); the status is recorded and shown here. Set it back to **active** to restore.

Deleting a tenant

Delete removes the tenant and **all** of its people, fobs, doors, levels, schedules, users and log entries after a confirmation. Its controllers are not deleted: they return to the unassigned pool so they can be reused. This cannot be undone; take a backup first (chapter 16). The provider tenant cannot be deleted.

Switching in

The drop-down at the top of the sidebar lists every tenant. Choosing one scopes the entire panel to it until you choose another; the choice is remembered for your session. The Tenants page row buttons do the same. A message "Now viewing (tenant)" confirms the switch.

While switched in, you act with super-admin rights inside that tenant: you can see and change everything, including its Settings and Users. Changes are logged in that tenant's log where the action is logged at all.

Controllers and the pool

Controllers belong to exactly one tenant, and a controller only ever receives the fobs of its own tenant. New hardware reaches a tenant in one of two ways:

- **Register first, then adopt** (recommended): switch into the customer, open **Controllers**, type the controller's ID into **Register manually** (the ID is printed on the board's serial console at start-up and forms the name of its setup network, `ACS-Setup-xxxxxx = esp32-xxxxxx`). When the controller later connects it is already the customer's and just goes online.
- **Adopt first**: a controller that connects before anyone registered it lands in an unassigned pool owned by no tenant, invisible to every customer. The panel has an assign action for super admins to move a pooled controller into a tenant; the current release's Controllers page does not yet show the pool section, so prefer registering first. A controller that was pre-registered in the wrong tenant can be removed there and registered in the right one.

Physical fob codes are unique across the whole panel (the same fob cannot be attached to people in two tenants), which matches real hardware where a fob has one code.

Per-tenant and shared settings

PER TENANT	SHARED ACROSS THE PANEL
People, fobs, doors, controllers, access levels, schedules, log	The panel address controllers adopt to, and the MQTT port
Controller Wi-Fi, controller type, channel 1 and 2 pin maps	The public HTTPS URL
Enrolment mode window	The phone-unlock API token (one inbound number serves all tenants)
Users and custom roles	Controller MQTT credentials (from the server environment)
Branding: logo and accent colour	Timezone (from the server environment)
Reception screen link	Firmware images staged for Flash & adopt

Phone access identifies the caller across all tenants from one number, and releases doors in the caller's own tenant. Apartment numbers are matched across tenants too, so two tenants may both have an "apartment 12"; the PIN decides which resident is meant.

Isolation and what to tell customers

- A tenant's admins can see only their own tenant; there is no page that lists other tenants.
- A tenant's controllers hold only that tenant's keyset.

- All controllers connect to the same broker with the same shared device credentials, which is appropriate while the provider owns and provisions all the hardware. If a customer will provision their own controllers on an untrusted network, treat the shared credentials as something you may need to rotate for everyone (chapter 15).
- The reception link and the phone token are secrets; the reception link is per tenant, the phone token is shared.

Commissioning checklist for a new customer

1. Create the tenant, switch in.
2. Create the customer admin; agree who holds it.
3. Branding.
4. Settings: Wi-Fi (or note that the site is PoE), controller type, pin maps for the board you are fitting.
5. Doors with fail modes confirmed against the fire assessment.
6. Register controller IDs, flash and adopt, map to doors, set active. Test a fob, the exit button and the door sensor at each door.
7. Schedules and access levels.
8. Import or enrol the customer's people and fobs (chapter 6). If they are migrating from a standalone system with no export, plan an enrolment-mode window during staffed hours.
9. Optional: reception screen on their front desk; phone access and PBX routing.
10. Hand over: the panel address, their admin login, this manual, and the chapter 17 checklist for "controller offline".

Security and privacy

This chapter describes how the product protects the panel, the controllers and the data, and what the operator should do on their side. It is written so that an IT contact or a data-protection lead can read it without the rest of the manual.

Transport

- **Web panel:** the hosted service is served over HTTPS with automatically issued certificates (Let's Encrypt via the reverse proxy; chapter 16). A self-hosted panel also listens on HTTPS with a self-signed certificate on port 8443, because the Flash & adopt page needs a secure page to use the browser's serial and Bluetooth features. The panel honours the proxy's forwarded-protocol header so it knows it is being served securely.
- **Controllers:** talk MQTT to the panel's embedded broker on port 1883. The broker requires a username and password when `ACS_DEVICE_USER` and `ACS_DEVICE_PASS` are set on the server, which every internet-facing installation should do. Note that MQTT on 1883 is not encrypted: fob codes and door names cross the network in clear. The keyset is a list of fob codes and time windows, not names. Restrict port 1883 at the firewall to the site addresses that need it where you can (chapter 16). TLS MQTT is not offered by the current firmware.
- **Firmware downloads** for over-the-air updates are public files served without authentication (they contain no secrets). When a public HTTPS URL is set, controllers fetch them over TLS; the controller does not verify the certificate chain, which is the accepted trade-off for a device without a certificate store.
- **Phone API:** the PBX must use the HTTPS address; the bearer token and callers' numbers would otherwise travel in clear.

Accounts and sessions

- Passwords are stored as **bcrypt** hashes; the panel never stores or displays a password. Minimum length is 8 characters.
- Sign-in failures answer after a deliberate half-second delay, which blunts online guessing. There is no account lockout in the current release, so choose strong passwords and keep the number of admin accounts small.
- Sessions are random tokens stored hashed on the server, delivered in an `HttpOnly`, `SameSite` cookie, and expire after 7 days. Logging out destroys the session on the server. Deleting an account ends its sessions.
- Every form carries a per-session **CSRF** token; posts without it are refused.
- Roles are enforced on the server for every request, not just hidden in the interface: a viewer or a custom role that posts a change to a page it may not change gets "read-only account"; a page it may not view redirects.
- Super admin accounts are created only from the server console, never from the web.

Secrets held by the panel

SECRET	STORAGE	HOW TO ROTATE
User passwords	bcrypt hash	Reset on the Users page.
Apartment PINs	bcrypt hash, never displayed	Set a new PIN or Clear PIN on the person's page.
Session tokens	SHA-256 hash	Log out; expire after 7 days.
Reception screen link token	Per tenant, plain (it is the link)	Regenerate on the Doors page.
Phone API bearer token	Shared, plain (the PBX must present it)	Regenerate on Settings and update the PBX.
Telegram bot token	Per tenant, plain (the panel must present it to Telegram); never shown again in the interface	Send <code>/revoke</code> to @BotFather for the bot, paste the new token under Settings > Alerts (Telegram), or tick Forget the stored token .
Stripe secret and webhook secrets	Per tenant, plain (the panel must present them to Stripe); never shown again in the interface	Roll the key in Stripe, paste the new one under Settings > Bookings and portal, or tick Remove the stored secret key .
SMS gateway API key	Per tenant, plain; never shown again in the interface	Roll it at the gateway and paste the new one, or tick Remove the stored API key .
Portal login codes	6 digits, SHA-256 hash salted with the mobile number, 10-minute life, five attempts, single use	Expire on their own; they cannot be read back from the panel.
Portal session cookie	Random, stored as a SHA-256 hash, 90-day life	The resident presses Log out ; deactivating or trashing the person ends it at their next request.
Controller MQTT credentials	Server environment file	Change the environment variables, restart, and re-point or re-provision every controller with the new values.
Controller Wi-Fi password	Panel settings and each controller's flash	Change on Settings and re-provision controllers.

Abuse protection on the phone API

- Wrong apartment PINs: after 5 consecutive failures the apartment is locked for 60 seconds, doubling per further failure to a cap of 16 minutes. A correct PIN during a lockout is still refused. The response carries a retry-after time.
- A PIN attempt for an apartment that does not exist takes the same time as a wrong PIN, so an attacker cannot enumerate apartment numbers by timing.
- Per-source-IP rate limit on all phone endpoints: a burst of 10, then about 2 requests per second.
- Every attempt, granted or refused, is logged with the caller ID as presented.
- Caller ID is not a secret; treat caller-ID release as convenience-grade (chapter 9).

Audit trail

The Log (chapter 11) records every access decision with the fob, person, door, time and reason; every door alert; and the system events that change who can get in (enrolment windows, imports, fob capture, controller registration). Log entries are never edited by the panel; deleting a tenant deletes its log.

Not recorded in the current release: panel sign-ins, and edits to people, levels or schedules as individual audit lines (their effect is visible in subsequent access decisions). Operators who need a change history should restrict edit rights to named accounts and keep periodic database backups.

Controllers

- A controller stores its Wi-Fi password, the panel address, the MQTT credentials and its keyset in its own flash. Anyone with physical access to the board and a USB cable can read the configuration (passwords are masked in the status query, but the flash itself is not encrypted) and could present

a different keyset over USB. Mount controllers on the **secure side** of the door in a locked enclosure, never beside the reader.

- Reader lines (D0/D1) carry fob codes in clear; this is inherent to Wiegand and is true of every Wiegand system. Keep reader cabling inside the wall and use readers with tamper switches where the threat warrants it.
- Free egress: the exit button releases the door unconditionally by design.
- Firmware is updated only when an administrator presses **Update**; controllers do not fetch updates on their own.
- The watchdog restarts a stalled controller within 30 seconds; the keyset survives restarts and power cuts.

Data protection (UK GDPR / Data Protection Act)

The panel holds personal data. As the operator you are the controller of that data (in the legal sense); 2plus1 acts as processor where it hosts the panel. Points to cover in your records of processing:

- **What is held:** names, optional email and phone numbers, optional photographs, free-text notes, fob codes, apartment numbers, which doors a person may open and when, and a log of every door use with time and outcome. The lawful basis is normally the contract with the member or employee and the legitimate interest in securing the premises.
- **Photos:** are optional. Their purpose is identification at the reception screen. Tell people a photo will be taken and shown to reception staff; let them decline. Photos are stored in the database and served only to signed-in users of the tenant and to the reception screen link.
- **Access log:** reveals when individuals entered the premises. Decide a retention period that matches your purpose (many operators keep 3 to 12 months), and have the host trim older events (chapter 16). Restrict who can view the log with roles.
- **Phone numbers:** used only for caller-ID matching; the last nine digits are compared. Numbers of unrecognised callers to the access line are written to the log.
- **Notes** are free text; do not record special-category data (health, etc.) there.
- **Subject access:** a person's page and a filtered log export (chapter 16, database queries) give everything held about an individual.
- **Erasure:** moving a person to the trash suspends everything at once but keeps the record for 30 days; **Delete permanently** in the trash removes their record, photo and fobs immediately (chapter 6). Either way their historic log entries keep the name as recorded at the time. If full erasure is required, the host can anonymise those rows in the database. Note the 30-day trash window when answering an erasure request: use **Delete permanently** rather than waiting for the sweep.
- **Reception screen:** displays names and photos to anyone who can see the monitor. Position it so that the public cannot read it, and keep the link private.
- **Room bookings and the resident portal** (chapter 8) add: the resident's mobile number, their apartment number, what they booked and when, and any payment reference. Card payments are taken by Stripe, which holds the card details; the panel never sees or stores a card number. Login codes are texted through whatever SMS gateway the tenant configures, so that gateway is a processor of the resident's mobile number and should appear in your records.
- **The public board** at the portal address shows today's confirmed bookings by room and time with names reduced to initials, and needs no login. Anyone with the address can read it. Treat sharing that address as a publication decision, and simply do not share it if the building would rather not.
- **Self-registration:** an unrecognised mobile can create a person record with a name and apartment number from the portal. Those records are marked "Self-registered via the resident portal" and logged. Review them; they are the one route by which personal data enters the panel without a member of staff typing it.
- **Multi-tenant hosting:** each tenant's data is isolated at the application layer; all tenants share one database on one server operated by the provider.

Recommendations for operators

1. Use HTTPS only; never expose the plain HTTP port to the internet.
2. Set the controller MQTT credentials and firewall port 1883 to known site addresses.
3. Give each member of staff their own login with the least role needed; remove leavers promptly.
4. Regenerate the reception link and the phone token when staff or hardware leave.
5. Keep the enrolment-mode window short and staffed.
6. Mount controllers on the secure side in locked enclosures.
7. Back up the database (chapter 16) and test a restore.

Installation and hosting

This chapter is for whoever runs the panel: 2plus1 for the hosted service, or your own IT for a self-hosted installation. Access Controller is one binary with an embedded database and MQTT broker, so hosting is deliberately simple. Where this chapter describes a layout that is conventional rather than fixed by the software, it is marked **typical**.

Note: hosted customers can skip this chapter. If 2plus1 hosts your panel, there is nothing here for you to do. The panel is installed, backed up, monitored and **kept up to date by 2plus1**; new releases are applied for you, and the only thing that ever needs your attention is updating the firmware on your own controllers, which is done from the Controllers page (chapter 5) at a time that suits the site. The rest of this chapter is for self-hosters.

What you get

- One executable, `accesscontroller` (Linux) or `accesscontroller.exe` (Windows), containing the web interface, the MQTT broker, the keyset compiler and the database engine. No runtime, no separate database server, no message broker to install.
- A data directory holding the SQLite database file `acs.db` (plus its `-wal` and `-shm` companions while running), a `tls` folder for the self-signed certificate, and a `firmware` folder with the controller images used by Flash & adopt and over-the-air updates.

Requirements: a small Linux virtual machine or on-premise box (1 vCPU, 512 MB RAM and a few GB of disk are ample), a hostname with an A record pointing at it, and the ability to open three inbound ports.

Quick start (any platform)

```
accesscontroller create-admin admin YOURPASSWORD
accesscontroller
```

The first command creates (or resets) a super-admin login in the provider tenant. The second starts the server: web interface on `http://localhost:8080` and `https://localhost:8443` (self-signed), MQTT broker on port 1883, database at `data/acs.db` relative to the working directory. Sign in and follow chapter 2.

Run `accesscontroller help` to see the options.

Environment variables

All configuration is by environment variable; there is no config file.

VARIABLE	DEFAULT	MEANING
ACS_DB	data/acs.db	Path of the SQLite database file. The directory is created if missing. The TLS certificate is kept in a <code>tls</code> folder beside it.
ACS_ADDR	:8080	Web interface listen address (plain HTTP). Behind a reverse proxy bind it to localhost, for example <code>127.0.0.1:8081</code> .
ACS_TLS_ADDR	:8443	HTTPS listen address with a self-signed certificate. Set to <code>off</code> when a reverse proxy provides TLS.
ACS_MQTT_ADDR	:1883	MQTT broker listen address for controllers. Must be reachable from every site.
ACS_TZ	Europe/London	The panel timezone used for schedules, "today" counts and displayed times.
ACS_DEVICE_USER / ACS_DEVICE_PASS	empty	Username and password controllers must present to the broker. Empty means an open broker, acceptable only on an isolated LAN. Also written into controllers at provisioning.
ACS_SYNC_HOURS	3	How often keysets are re-pushed to every controller as a safety net (also carries daylight-saving offsets).
ACS_FW_DIR	data/firmware	Where controller firmware images live: <code><kind>/<chip>/</code> with four <code>.bin</code> files each, for example <code>networked/esp32/</code> and <code>networked/esp32s3/</code> .

Typical Linux layout (systemd)

The following is how 2plus1 deploys the hosted service and is a sensible template for self-hosting. Adjust paths and names freely.

- Binary and data under `/opt/accesscontroller/`, owned by a dedicated system user `acs`.
- Environment in `/etc/accesscontroller.env` (mode 0600, root-owned):

```
ACS_ADDR=127.0.0.1:8081
ACS_TLS_ADDR=off
ACS_MQTT_ADDR=:1883
ACS_DB=/opt/accesscontroller/data/acs.db
ACS_FW_DIR=/opt/accesscontroller/data/firmware
ACS_TZ=Europe/London
ACS_DEVICE_USER=doorctl
ACS_DEVICE_PASS=<long random secret>
```

- A unit file `/etc/systemd/system/accesscontroller.service` (**typical**):

```
[Unit]
Description=Access Controller panel
After=network-online.target
Wants=network-online.target

[Service]
User=acs
Group=acs
WorkingDirectory=/opt/accesscontroller
EnvironmentFile=/etc/accesscontroller.env
ExecStart=/opt/accesscontroller/accesscontroller
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
```

Then:

```
sudo systemctl daemon-reload
sudo systemctl enable --now accesscontroller
sudo journalctl -u accesscontroller -f
```

The log line `web ui on 127.0.0.1:8081` confirms the panel is up. A warning "no admin accounts" means you still need `create-admin` (run it as the `acs` user with the same `ACS_DB` so it opens the same database).

Reverse proxy and certificates (Caddy)

Put a TLS reverse proxy in front of the panel so that users get HTTPS with an automatically issued Let's Encrypt certificate. Caddy does this in two lines (**typical**), in `/etc/caddy/Caddyfile` :

```
doors.example.com {
    reverse_proxy 127.0.0.1:8081
}
```

Caddy obtains and renews the certificate, redirects HTTP to HTTPS, and passes `X-Forwarded-Proto: https`, which the panel uses to know it is on a secure page (needed for Flash & adopt and webcam capture). Prerequisite: the DNS A record `doors.example.com` must already point at the server before Caddy can issue the certificate. Nginx or Apache with certbot work equally well; forward to the panel's `ACS_ADDR` and pass the forwarded-proto header.

After the proxy is up, set **Public HTTPS URL** on the Settings page to `https://doors.example.com` so that over-the-air firmware downloads use it, and set the **Adoption address** to `doors.example.com` (chapter 13).

Ports and firewall

PORT	PROTOCOL	WHO CONNECTS	NOTES
443	HTTPS	Staff browsers, the reception screen, the PBX, controllers fetching OTA images	Via the reverse proxy.
80	HTTP	Let's Encrypt validation, redirect to 443	Via the reverse proxy.
1883	MQTT (TCP)	Controllers at every site	The panel's embedded broker binds this itself. Restrict to site addresses where possible. Plain text.
22	SSH	Administrators	Standard.

Outbound from the panel host: HTTPS (443) to `api.telegram.org` if any tenant uses Telegram alerts (chapter 4). Nothing else leaves the panel.

The panel binary binds `ACS_ADDR` (localhost behind a proxy) and `ACS_MQTT_ADDR` (all interfaces) itself; the proxy binds 80 and 443. If the panel is on a site LAN with no proxy, users connect to 8080 or 8443 directly instead.

Note: at each site the controllers need outbound TCP to the panel on 1883 (and 443 for updates). Most routers allow outbound traffic by default. On a filtering firewall, add a pass rule for the door network to the panel address on 1883. If a rule appears correct but shows zero traffic, check that the firewall actually loaded its rule set: one deployment lost days to a pfSense box whose filter reload was silently failing because of an oversized block list, so every new rule sat idle and the default deny won. Chapter 17 has the diagnosis steps.

Firmware images

Flash & adopt and OTA serve images from `ACS_FW_DIR`. Copy the built images for each chip into `networked/esp32/` and `networked/esp32s3/` (and `standalone/...` if you offer standalone units): `bootloader.bin`, `partitions.bin`, `boot_app0.bin` and `firmware.bin`. The Flash & adopt page also accepts uploads per chip, and shows per chip whether a build is complete. 2plus1 supplies the images with each release.

Backups

The whole system state is the SQLite database file. To back it up:

- **Simplest:** stop the service, copy `acs.db`, start the service. Stopping first guarantees a consistent file and folds the write-ahead log (`acs.db-wal`) into the main file.

- **Without downtime:** use SQLite's online backup, for example `sqlite3 acs.db ".backup /backups/acs-$(date +%F).db"`, which is safe while the panel runs.

Keep the environment file (it holds the device credentials) and the `data/firmware` folder with the backup. Photos are stored inside the database, so the one file really is everything.

Restore: stop the service, remove `acs.db`, `acs.db-wal` and `acs.db-shm`, copy the backup in as `acs.db`, make sure it is owned by the service user, start the service. Run `create-admin` if you need to reset a login. After a restore the panel re-pushes keysets to every controller on its next connection, so doors pick up the restored state automatically.

Updates

On the hosted service, 2plus1 applies updates; nothing below applies to you.

Self-hosting, replace the binary and restart:

```
sudo systemctl stop accesscontroller
sudo install -m 0755 -o acs -g acs ./accesscontroller /opt/accesscontroller/accesscontroller
sudo systemctl start accesscontroller
```

Database schema changes are applied automatically at start-up and are additive, so a newer binary opens an older database. Take a backup before updating anyway. Copy any new firmware images into `ACS_FW_DIR` at the same time, then use **Update** on the Controllers page to roll firmware out to controllers one at a time (chapter 5).

Moving to a new server

1. Install the new panel, restore the database backup onto it, set the same `ACS_DEVICE_USER` and `ACS_DEVICE_PASS` (or be ready to type the new ones), and put the proxy and DNS in place.
2. If the hostname stays the same and DNS is switched over, controllers simply reconnect to the new address when their existing connection drops.
3. If the address changes, use **Re-point** on the old panel for each controller while it is still online, giving the new address and, if they differ, the new MQTT credentials. Chapter 5 and chapter 17 cover the credential pitfall.

Housekeeping queries (optional)

The database is standard SQLite and can be inspected with the `sqlite3` tool while the service runs (read-only queries are safe). Two examples for data-protection requests and retention (**typical**; take a backup first before any write):

```
-- everything logged about one fob
SELECT datetime(ts,'unixepoch'), door_name, decision, reason FROM events WHERE fob_code='A1B2C3' ORDER BY ts;

-- trim access events older than a year for one tenant
DELETE FROM events WHERE tenant_id=3 AND kind='access' AND ts < strftime('%s','now','-1 year');
```

Windows and development use

The binary also runs on Windows (`accesscontroller.exe`) with the same variables, which is handy for a bench set-up or a single-site installation on an existing office PC. Open port 1883 in Windows Firewall for the door network and use the PC's LAN IP as the adoption address. For anything more than a bench, host it on Linux behind a proxy as above.

Troubleshooting and FAQ

Work down the section that matches the symptom. Most problems show up in the **Log** (chapter 11) or on the **Controllers** page (chapter 5), so start there.

A controller shows offline

"Offline" means the panel has not heard from the controller for 10 minutes. The door is still working from its saved keyset; what has stopped is updates and event reporting.

1. **Power.** Is the board lit? Is the PoE switch port up? A controller that reboots every 30 seconds (watchdog) may have a wiring fault shorting a pin.
2. **Network.** For Wi-Fi: is the SSID still the same, was the password changed, is the signal adequate at the door, and is it 2.4 GHz? For PoE: link light, and does the site DHCP hand out an address?
3. **Reach the panel.** From a laptop on the same site network, can you connect to the panel address on port 1883 (for example `Test-NetConnection doors.example.com -Port 1883` on Windows)? "False" while 443 is "True" means a firewall is blocking 1883 outbound at the site or inbound at the panel.
4. **Firewall rule loaded?** If a pass rule exists but its counters stay at zero and traffic still hits the default deny, the firewall may not have loaded its rule set. On pfSense, a failing filter reload (for example caused by an oversized pfBlockerNG list) makes every new rule inert while the interface looks normal. Fix the reload error, then re-test. Useful diagnosis order: packet capture on the panel (no packets arriving = dropped upstream), then the firewall's state table and firewall log (which rule matched).
5. **Credentials.** If the panel's broker requires a username and password and the controller has the wrong ones, it connects and is refused (serial console shows `mqtt connect failed rc=5`). See "A re-pointed controller never came back" below.
6. **Serial console.** Plug the controller into a laptop and open a serial monitor at 115200 baud. It prints `ACS-WAIT uid=... mqtt=host:port link=down` every 5 seconds while it cannot reach the broker, and `net: mqtt connected` when it can. `ACSCFG?` typed into the console dumps its settings.
7. **Re-provision.** On Flash & adopt, step 3, use **Query device** then **Send settings to device** over USB (or Bluetooth in setup mode) to rewrite the Wi-Fi, address and credentials.

A fob is denied

Look at the reason in the Log or use **Check access** on the dashboard with the fob code.

REASON	FIX
Unknown fob	The code is not on anyone in this tenant. Attach it: desk scan, Enrolment page, or Add fob on the person. If the fob <i>is</i> on the person, the reader may be sending a different format from what was typed: compare the code in the log with the one on the person's page.
Fob revoked	Re-activate it on the person's page, or give them a new fob.
Person inactive	Set the person's status to Active.
No access level for door	Tick a level that includes this door on the person's page, or add the door to one of their levels.
Outside schedule	The level's schedule does not allow this day/time. Check the schedule and whether the door should be in a different level. If everything is off by an hour, check the panel timezone (<code>ACS_TZ</code>) and that the controller has time (see below).
No reason, nothing in the log	The scan never reached the panel: the controller is offline (above) or the reader is not wired to the pins the controller expects (chapter 5). Watch the serial console for <code>wiegand: 26 bits -> ...</code> lines when a fob is presented.

A fob that is granted at one door and refused at another is nearly always an access-level question: the second door is not in any of the person's levels.

Problems after an import

Imported fobs are denied at the door

The log says *Unknown fob* for people who were imported from a CSV or an AC8000 backup, yet their fob is on their page.

- **Compare the two codes.** Open the person's page and note the fob code, then look at the code in the log row. The panel matches a fob whether it was stored with 6 or 8 hexadecimal digits (leading zeros do not matter), so `0136A9` on the person will match a reader sending `000136A9`. If the digits themselves differ, the wizard was told the wrong reader format or the CSV's decimal box was set wrongly: re-run the import with the other setting (existing people are skipped, so nothing is duplicated) or correct the code on the person.
- **Check the reader type.** A **Wiegand 26** reader sends 24 bits of card number and cannot read a card whose number is above 16,777,215 (hexadecimal `FFFFFF`): it sends a truncated code that will never match. If the backup contained numbers that large the wizard will have said so and set 8 digits; the readers at the door then need to be **Wiegand 34** (or the same model the old system used). A reader that sends 26 bits shows `wiegand: 26 bits -> ...` on the serial console when a fob is presented; 34 bits for the wider format.
- **Nothing in the log at all:** the scan never reached the panel; see "A fob is denied" above.

The wizard says it cannot read the file

"Could not read that file" under the file box means the browser could not find the AC8000 tables in what you chose. The wizard needs the **database backup made by the AC8000 / iCCard3000 PC software**, which is a Microsoft Access file with an `.mdb` extension. It cannot read a printed report, a spreadsheet, a CSV or any other export from that software: for those, use the **CSV / spreadsheet export** source instead. If the file is genuinely the backup and still fails, make it again from the old software (a backup taken while the software was writing can be incomplete), and try in a current browser: the file is parsed by the page itself, so an old browser or a content blocker that stops the page's scripts will also produce this message. "parser not loaded" means exactly that: reload the page and choose the file again.

History imported but people show as unknown

Imported history rows are matched to people by fob code. A row with no name and the reason *Unknown fob* means the card that made that swipe was **not in the backup's people list** when the backup was taken: a card that had been deleted from the old system, a visitor's card, or a card the old system itself refused. That is what the old system recorded, so the rows are correct as they stand. If the card does belong to someone, add it to them (chapter 6): future scans will show the name, but history rows already written are not re-linked.

The door does not release

- The log shows **granted** but the lock does not move: the relay is not wired to the expected pin, the **relay active level** is wrong (relay energised at rest and de-energised on grant), or the lock supply is off. Press **Unlock** on the Controllers page and listen for the relay click. Check the pulse length is not zero.
- The lock releases for the exit button but not for a fob: access-level or schedule problem; see above.
- The lock releases but only for a moment: normal; the pulse is 4 seconds by default. Lengthen it in Settings and re-provision, or fit a door closer that does not need longer.
- The maglock never holds: the relay contact is on the wrong side (NO versus NC) for a fail-safe lock, or the active level is inverted. Recheck the wiring against chapter 3.
- The door releases on power-up: the relay module is active low and the panel says active high (or the reverse), so the board's "off" state is the relay's "on" state. Change **Relay active level** in Settings and re-provision.

Phone unlock says "not recognised"

1. **Is the caller's number on the person?** Under Details, the Phone must match the caller ID on the last nine digits. Withheld numbers cannot match.
2. **Is release by phone allowed?** The tick "Allow this person to release doors by phone" and at least one door must be set, and the person must be Active.
3. **Is the PBX pointed at the right panel with the right token?** `[door_release] ... http=401` in the Asterisk console (or `/tmp/door_release.log`) means the token is wrong; `http=0` means the PBX cannot reach `$BASE`.
4. **Does the door have an active controller?** A release to a door with no active controller is refused with `no_controller` and logged.
5. **Apartment mode:** the apartment number is digits only, the PIN is 4 to 10 digits, and the doors are ticked on the resident's page. After five wrong PINs the apartment is locked for a rising period; wait, or set a new PIN (which resets the lockout).
6. **Rate limited:** a burst of more than 10 calls from the PBX address in a few seconds is refused; wait a few seconds.

The Log has a line for every call with the outcome in the detail column.

Held-open and forced alerts keep appearing

- **Door held open** on every opening: the door's held-open time (60 s by default) is shorter than the time the door is genuinely open, for example a loading door. Fit a stronger closer, or open the door's page and raise **Alert if held open for N seconds** (or set it to 0 to switch the alert off for that door). The new value reaches the controller within seconds if it runs the current firmware; older firmware ignores it and keeps 60 s (or the `heldOpenMs` in its pin map) until it is updated from the Controllers page.
- **Door forced** on every opening, even legitimate ones: the door sensor is wired the wrong way (open when the door is closed), or the sensor is on a different pin from the one in Settings. The controller expects the contact closed (to 0 V) when the door is closed.
- Both alerts constantly with the door shut: a loose or broken sensor cable. Set the door pin to `-1` in Settings and re-provision to silence the alerts until it is fixed.
- No alerts ever: no sensor fitted, or the pin is `-1`, or (held open only) the door's held-open time is 0. Check the Log for *Door opened / Door closed* lines first: if they appear, the sensor works and the alert settings are the problem.
- Alerts are in the Log but no Telegram message arrives: see below.

Telegram test message fails

The red message after **Send test message** or **Find my chat ID** says what Telegram objected to; chapter 4 has the full table. The three most common:

- **401 Unauthorized / rejected the bot token**: the token has a typo, is incomplete, or was revoked. Send `/mybots` to @BotFather, choose the bot, **API Token**, copy it again and paste it with no spaces at either end. A 404 means the same thing with the front of the token missing.
- **chat not found**: the chat ID is wrong or the bot has never been in that chat. Message the bot (or add it to the group, or make it a channel administrator and post once), then use **Find my chat ID** and **Use this** rather than typing the number. Groups and channels are negative numbers; keep the minus sign.
- **Find my chat ID finds nothing**: the bot has not received anything yet, or it is in a group with privacy mode on and cannot read ordinary messages. Send `/start` in the group (commands are always visible to the bot), or make the bot a group administrator, or turn privacy mode off in @BotFather (remove and re-add the bot afterwards). If the bot is also used by another integration that set a webhook, `getUpdates` is refused: open `https://api.telegram.org/botTOKEN/deleteWebhook` once in a browser.

"could not reach Telegram" is the panel host itself: allow outbound HTTPS to `api.telegram.org`. If the test works but real alerts never arrive, check **Enable Telegram alerts** and the two alert tick boxes are on, that the door has a sensor and a non-zero held-open time, and that the alert rows actually appear in the Log; then look at the panel's server log for "telegram send failed". Remember repeats for one door are limited to one message per minute.

Door opened / closed does not appear in the log

The controller reports door open and close transitions only when it has a door contact and the current firmware.

- **No door contact wired**: fit a reed switch or contact between the door pin and 0 V, closed when the door is closed (chapter 5).
- **Door pin is `-1`**: the pin map in Settings (or the pin map an installer sent over USB) says there is no sensor, so the controller never reads it. Set the right GPIO and re-provision or re-point the controller so it picks the pin map up.
- **Old firmware**: builds before the door-event change do not publish *Door opened / Door closed* or *Exit button* lines (older builds logged the exit button as a granted access event called *rex*). Press **Update** on the Controllers page, or re-flash from Flash & adopt, and check the firmware version shown on the Controllers page afterwards.
- **Only opens appear, never closes** (or the reverse): the contact is wired the wrong way round or is intermittent; *Door forced* on every legitimate opening is the usual companion symptom. Watch the serial console while opening and closing the door.
- **Nothing for the second door of a dual-door controller**: the current firmware watches the channel 1 door contact only.

If the rows are there but you cannot see them, choose **Door open/close, exit button** in the Kind filter: they are not access events and are excluded from the granted/denied views.

The reception screen is blank or stuck

- Nothing at all: the link has been **regenerated**; get the new link from Doors and reopen it.

- "Waiting for the next entry" and nothing appears when people fob in: press the gear and check the door filter includes the doors in use; the browser may have an old filter saved. Check the screen machine can reach the panel address.
- Photos show as initials: no photo on the person, or the screen machine cannot fetch images from the panel (proxy or content blocker).
- Counts are zero at the top: no access events today on the selected doors.
- The screen went to sleep: adjust the machine's power settings; the page itself never times out.

Lost admin password

Any admin of the tenant can reset a user's password on the **Users** page. If nobody can sign in:

- On a hosted panel, contact 2plus1; the super admin resets it.
- On a self-hosted panel, on the server run `accesscontroller create-admin <username> <newpassword>` with the same `ACS_DB` as the service (stop the service first if the database is locked). Running it for an existing username resets that password; running it for a new name creates a new super admin in the provider tenant.

A re-pointed controller never came back

Re-point carries the *current* panel's MQTT credentials when the user and password fields are left blank. If the new server requires different credentials the controller connects, is refused (`rc=5` on the serial console), and cannot be reached remotely because it is no longer connected anywhere. Two ways out:

- Temporarily allow the new server's broker to accept the controller (clear its device credentials, restart), let the controller connect, re-point it **with** the correct credentials, then re-lock the broker; or
- Visit the controller with a laptop and send the credentials over USB from Flash & adopt (step 3, **Send settings to device**) or via the setup portal. The USB path merges: it keeps the Wi-Fi settings and only changes what you send.

Avoid the problem by using the same device credentials on old and new panels, or by typing the new ones in the Re-point form.

An OTA update failed or the controller now boot-loops

- *ota_failed* in the log with a reason: usually the controller could not fetch the image (no public HTTPS URL set and the panel's HTTP port not reachable from the site, or no image staged for that chip). Fix and press Update again.
- The controller restarts endlessly after an update: there is no automatic roll-back. Re-flash it over USB from Flash & adopt with a known-good build. Always test an update on one unit first.

The Flash & adopt page says it needs a secure page

Web Serial and Web Bluetooth require HTTPS or localhost. Use the panel's `https://` address (accept a self-signed certificate warning on a self-hosted panel) or open the page on the panel machine as `http://localhost:8080/flash`. iPhones and iPads cannot use either feature; use a laptop.

The board will not flash

Hold **BOOT**, tap **RST**, release **BOOT**, then start the install again. Try a different USB cable (many are charge-only) and port. Close any other program holding the serial port (a serial monitor, another browser tab). If the port never appears, install the USB-serial driver for the board.

An ESP32-S3 hangs after provisioning

The panel's default pin map uses GPIO 33 for Wiegand D1; on an S3 with OPI PSRAM (N16R8 and similar), GPIO 33 to 37 are reserved and driving them hangs the chip. Set the S3-safe pins in Settings (D0 4, D1 5, relay 6, REX 7, door 15) and re-provision over USB with BOOT held at power-up to reach setup mode.

Times in the log are wrong

- Everything is out by a fixed hour or more: the panel timezone (`ACS_TZ`) is wrong for the site.

- Only events from one controller are wrong, or dated 1970: that controller has no valid time (no network time yet, no RTC). Events without a valid controller time are stamped on arrival at the panel. Fit a DS3231 RTC for sites with poor connectivity.

Phone enrolment does not arm

A caller who presses # and keys in the site number hears the refusal prompt rather than the "present the fob" prompt. The prompts cannot tell the reasons apart by ear, so read the Log:

1. **Is a site number set?** Settings, **Site number (phone enrolment)**. Empty means phone enrolment is switched off for that site.
2. **Did they key in the right one?** The number belongs to the building, not to the person. A caller who keys in another site's number is refused, and the Log records "site code belongs to another site".
3. **Are enrolment doors ticked?** On the Enrolment page, at least one door must be ticked. With none, there is nowhere to present the fob and the call is refused. Enrolment mode itself does not have to be running: the # path switches it on for ten minutes at the ticked doors.
4. **Is the caller recognised?** The same caller-ID rules as dial release apply; see "Phone unlock says not recognised" above.
5. **Did the offer play at all?** If callers never hear it, the PBX script is collecting the keypress with a digit-collection primitive rather than "wait for one digit", so the # is being swallowed as a terminator (chapter 9).

When it works, the Log shows *Phone enrolment armed*, and *Fob enrolled at door* with "by phone" in the detail once the fob is presented.

A booking does not open the door

1. **Is the booking confirmed?** *Awaiting payment* grants nothing. Mark it paid on the Bookings page, or take the payment.
2. **Is it actually the slot?** Access runs from the room's lead time before the start until five minutes after the end. Outside that, nothing opens.
3. **Fob refused, portal button works?** The controller's firmware is too old to understand booking windows. Update it from the Controllers page (chapters 5 and 8). The portal button and the phone path work regardless, because the panel sends the unlock itself.
4. **Portal button says the controller is offline?** The room's door has no active controller mapped, or the controller has dropped off. Check the Controllers page.
5. **Is the door mapped to the right channel?** On a dual-door controller, a room on the second door must be mapped to channel 2 on the door's page (chapter 3), or the unlock fires the wrong relay.

A resident cannot sign in to the portal

1. **Is the portal published?** Settings, **Bookings and portal**, **Publish the resident portal**. Unpublished, the address returns "not found".
2. **Did the code arrive?** With no SMS gateway configured, codes are not sent at all: they appear in the **Login codes sent** panel on the Bookings page, marked *development only*. That is for commissioning. Configure the gateway before residents use the portal.
3. **Wrong apartment number?** Someone whose person record has an apartment number is asked to confirm it. Three wrong answers lock them out for 15 minutes. Check the number on their person page.
4. **Are they inactive or in the trash?** Neither can sign in. Restore or reactivate them.
5. **Have they changed their mobile?** They cannot self-register, because their apartment already exists. Update the phone number on their person page.

Frequently asked questions

Does the door work if the internet goes down? Yes. The controller decides from its own copy of the access list. Changes made in the panel during the outage reach it when it reconnects.

How many fobs can a door hold? About 1,500 per door in the current firmware. Larger sites should ask 2plus1 before deployment.

Can a fob open doors at two of our sites? Yes. If both sites are in the same tenant, add both doors to the person's levels. If they are separate tenants, the same fob code can be registered in each one (codes are unique within a tenant, not across the panel), but they are two independent people: revoking one has no effect on the other.

Can two people share a fob? No. A fob code belongs to one person. Issue a second fob.

Can I use my existing readers and fobs? Yes, if the readers are Wiegand 26-bit or 34-bit, which nearly all are. The fobs' codes are read as they are.

Can someone have access at some times to one door and all the time to another? Yes: give them two levels, one per door with its own schedule.

Why did "denied today" go up when I enrolled a fob at a door? It should not: captures at a listening door are logged as system events, not denials. If it did, the scan happened before you pressed "Wait for fob" or at a different door.

How do I temporarily let a contractor in? Add them as a person with a fob and a level, and set them inactive (or revoke the fob) when the job ends. Or, if they will phone, use apartment mode with a PIN you clear afterwards.

Can I see who is in the building right now? The log shows entries, not exits (unless exit readers are fitted as separate doors). Occupancy counting is not a feature of the current release.

Can I be told when a door is left open or forced? Yes. Fit a door contact, set the held-open time on the door, and set up Telegram alerts under Settings (chapter 4). Alerts are also always in the Log.

Why does the log show an exit-button press as a door event rather than granted? Because nobody was checked: the exit button always releases the door, so it is recorded as a neutral *Exit button pressed* line and left out of the scan and denied counts. Older firmware logged it as a granted access event named *rex*.

I deleted the wrong person. Can I get them back? Yes, for 30 days. Open the trash from the People page, find them and press **Restore**; their fobs, access levels and phone-release doors all come back (chapter 6).

Can I add an access level to 200 members at once? Yes. Search for them on the People page, tick the header box, pick the level and press **Add level(s)** (chapter 6). It adds to what they already hold rather than replacing it.

Do I need a card reader on a room we only let residents book? No. A room needs the lock relay and an exit button; residents come in with the portal's **Open the door** button or by ringing the access line from the mobile they booked with. Fit a reader only if you also want fob entry during a slot (chapter 8).

Can we take money for the residents' lounge? Yes, by card through Stripe or by bank transfer with a quoted reference. Set both up under Settings, Bookings and portal (chapters 8 and 13). The panel does not issue refunds: cancel the booking here and refund in Stripe.

A member is standing at the door with a new fob and nobody is in the office. What now? If the site has a site number and enrolment doors ticked, tell them to ring the access line, press # , key in the site number and present the fob within ten minutes (chapter 9).

What is the difference between the site number and an apartment PIN? The site number says which *building* a caller is at, for enrolling a fob, and is not a secret. An apartment PIN says which *person* is calling, for opening a door, and is stored hashed. They are unrelated.

Does enrolment mode open every door on the site? No, not any more. It runs only at the doors you tick when you start it; every other door keeps refusing unknown fobs (chapter 6).

Glossary

AC8000. A family of standalone door controllers and the Windows PC software that manages them. Its database backup (an `.mdb` file) can be imported into the panel: people, fob codes, door names and swipe history. Chapter 6.

Access level. A named set of doors plus one schedule. People hold one or more. A level with no schedule is 24/7. Chapter 7.

Active / inactive (person). A person's status. Inactive suspends every fob and phone release they have without deleting anything. Chapter 6.

Active / pending / disabled (controller). A controller's onboarding state. Only active controllers receive a keyset. Chapter 5.

Active / revoked (fob). A fob's status. Revoked fobs are refused everywhere but stay on the person's page. Chapter 6.

Add-on / extra. An optional item a resident can include when booking a room (a projector, a set of weights). Priced, with optional stock. Chapter 8.

Adoption address. The panel hostname or IP that controllers connect to over MQTT; written into each controller at provisioning. Chapter 13.

AGI script. An Asterisk Gateway Interface program that a PBX runs to control a call. The supplied `door_release.php` is one. Chapter 9.

Alert. An event a door raises on its own: *door forced* or *door held open* (plus a failed firmware update). Always logged; the first two can also be sent to Telegram. Chapters 4 and 11.

Apartment mode. Phone-access path for unrecognised callers: apartment number then PIN. Chapter 9.

Bcrypt. The slow, salted hash used to store passwords and PINs so that they cannot be read back. Chapter 15.

Board (resident portal). A public page at the portal address with `board` on the end, showing today's confirmed bookings by room and time with names as initials. No login. Chapter 8.

Booking. One person's reservation of a room for one slot. A confirmed booking opens that room's door for the slot. Statuses: awaiting payment, confirmed, cancelled, expired. Chapter 8.

BOOT button. The button on an ESP32 board that, held during power-up, puts the controller into setup mode (setup portal plus Bluetooth configuration) or, with RST, into firmware download mode. Chapter 5.

Bot token. The secret string from @BotFather that lets the panel send messages as your Telegram bot. Stored per tenant, never shown again. Chapter 4.

Bulk actions. Changing many people at once from the People page: add or remove access levels, set active or inactive, turn phone release on or off, export or move to the trash. Chapter 6.

Caller ID. The phone number presented by a caller. Used to recognise people on the access line; matched on the last nine digits. Chapter 9.

Channel. One reader-and-relay set on a controller. Single-door controllers have channel 1; dual-door controllers have channels 1 and 2. Chapter 3.

Chat ID. The number of the Telegram chat, group or channel that alerts are posted into. Positive for a private chat, negative for groups and channels. Found with **Find my chat ID**. Chapter 4.

Check access. The dashboard tool that runs the door decision rule for a fob, door and time without logging anything. Chapter 2.

Controller. The ESP32 unit at the door that reads the reader, drives the lock, watches the exit button and door sensor, and talks to the panel. Chapter 5.

Cooldown (alerts). The one-minute limit between repeat Telegram messages for the same door and alert type. Chapter 4.

CSRF token. A per-session value included in every form so that another website cannot submit actions on your behalf. Chapter 15.

Custom role. A staff permission set built by ticking pages as view or change. Chapter 12.

Dial release. Another name for phone access. Chapter 9.

Door event. A neutral log line from the door contact or exit button: *door opened*, *door closed*, *exit button*. Not a grant or a denial; not counted as a scan. Chapter 11.

Door forced. Alert raised when the door sensor sees the door open without a grant or exit press. Chapter 11.

Door held open. Alert raised when the door stays open longer than the door's held-open time. Chapter 11.

Door position sensor. A reed switch or contact that tells the controller whether the door is open. Enables forced and held-open alerts. Chapter 5.

Dual-door controller. One board driving two doors via channels 1 and 2. Chapter 5.

Enrolment (at a door). Listening at one door's reader for the next unknown fob and capturing it, without opening the door. Chapter 6.

Enrolment doors. The doors ticked in the Enrolment mode panel. Only these accept and capture unknown fobs while enrolment mode runs, and they are where a phone enrolment window applies. Chapter 6.

Enrolment level. The default access level chosen when enrolment mode is started. It is added to every person a captured fob is assigned to, new or existing. Chapter 6.

Enrolment mode. A tenant-wide window during which unknown fobs are accepted at every door and captured for assignment. Chapter 6.

Event. One line in the Log: access, alert or system. Chapter 11.

Exit button (REX). A push button on the inside of the door that always releases it. "Request to exit". Chapter 5.

Fail-safe. A lock that releases when power fails (maglock). **Fail-secure.** A lock that stays locked when power fails (most strikes). Chapter 3.

Fob. Any physical credential read by the door reader: fob, card, tag. Identified by its code. Chapter 6.

Fob code. The number read from a fob, shown as uppercase hexadecimal, for example `A1B2C3` (26-bit) or `0BADF00D` (34-bit). Chapter 6.

GPIO. A numbered input/output pin on the controller board. The pin map says which GPIO does what. Chapter 5.

Held-open time. The per-door **Alert if held open for N seconds** value; 0 switches the alert off. Pushed to the controller with the keyset. Chapter 3.

Hexadecimal. Base-16 notation using digits 0 to 9 and letters A to F. Fob codes are stored this way; many old systems show the same value in decimal.

iCCard3000. The older name of the AC8000 PC software; its backups are the same `.mdb` format and import in the same way. Chapter 6.

Keyboard-wedge reader. A USB fob reader that types the code like a keyboard. Used with the desk scan box. Chapter 6.

Keyset. The complete list of fobs and time windows that may open one door, compiled by the panel and stored on the controller for offline decisions. Chapter 1.

Lead time (room). How many minutes before a booking starts the door begins letting the person in. Ten by default. The door also stays theirs for five minutes after the slot ends. Chapter 8.

MQTT. The lightweight messaging protocol between controllers and the panel's embedded broker, on port 1883. Chapter 16.

OTA (over the air). Updating a controller's firmware from the panel without a cable. Chapter 5.

Panel. The Access Controller web application and everything inside it (interface, broker, database).

PBX. A phone system. Routes calls on the access line to the door-release script. Chapter 9.

Person. Someone who opens doors. Called "People" in the panel; not a panel user. Chapter 6.

Phone enrolment. A recognised caller presses `#` on the access line, keys in the site number, and has ten minutes to present a new fob at an enrolment door, which is then attached to them. It never opens a door. Chapter 9.

PIN (apartment). A 4 to 10 digit secret a guest keys in after the apartment number. Stored hashed. Chapter 9.

Pool (unassigned). Where a controller that connects before being registered waits until a super admin assigns it to a tenant. Chapter 14.

Portal (resident portal). The mobile web page where residents sign in with their mobile number and book rooms, at `/p/<portal address>`. Chapter 8.

Provider. The organisation hosting the panel for several tenants. Its own tenant is marked "provider". Chapter 14.

Provisioning. Writing Wi-Fi, panel address, credentials and pin map into a controller, over USB, Bluetooth or the setup portal. Chapter 5.

Reception screen. The public, token-linked full-screen page that shows people as they fob in. Chapter 10.

Relay pulse. How long the lock is released after a grant, in milliseconds. Chapter 13.

Re-point. Telling an online controller to move to a different panel address. Chapter 5.

Role. What a panel user may see and change: super admin, admin, viewer or custom. Chapter 12.

Room. A door that has been made bookable. Needs a door module and an exit button; a fob reader is optional. Chapter 8.

Schedule. A weekly pattern of allowed hours, one entry per day: no access, all day or a time window (which may run overnight). Chapter 7.

Setup portal. The controller's own Wi-Fi network and web page (`ACS-Setup-xxxxxx`, `http://192.168.4.1`) for entering settings without a laptop. Chapter 5.

Site number. The 2 to 6 digit number, unique across the panel, that a caller keys in to say which building they are enrolling a fob at. Set in Settings; empty switches phone enrolment off for that site. Chapters 9 and 13.

Slot. One bookable period on a room, cut from its opening hours and aligned to its slot length. Chapter 8.

Slug. A tenant's short identifier made of letters, digits and dashes. Chapter 14.

Standalone controller. A firmware variant that manages its own fobs from its own web page with no panel. Chapter 5.

Super admin. The provider's account: sees all tenants, switches between them, and is created from the server console. Chapter 12.

Telegram bot. A Telegram account operated by software rather than a person. The panel posts alerts through one bot per tenant. Chapter 4.

Tenant. One customer organisation with its own isolated data, users and branding. Chapter 14.

Token (reception, phone). A long random secret in the reception link or presented by the PBX. Regenerate to revoke. Chapter 15.

Trash. Where a deleted person goes for 30 days. Their fobs stop working at once; restoring them puts everything back. Swept automatically. Chapter 6.

User. A staff login for the panel. Chapter 12.

Viewer. The fixed read-only role. Chapter 12.

Wiegand. The two-wire (D0/D1) signalling standard used by nearly all door readers. 26-bit and 34-bit frames are supported. Chapter 5.

Wiegand 26 / Wiegand 34. The two common frame lengths. Wiegand 26 carries 24 bits of card number, shown as 6 hexadecimal digits, with a highest value of 16,777,215; Wiegand 34 carries 32 bits, shown as 8 digits. The import wizard asks which your readers use. Chapters 5 and 6.