



OUTPOST

USER MANUAL

Manual v1.1 · App 0.9.7 · Firmware 0.9.1

October 2026

OFFGRID TACTIX

OFF-GRID MESH COMMUNICATIONS

CONTENTS

1. What Outpost Is	3
2. First Setup.....	5
2.1 Flash the T-Deck Firmware	5
2.2 Set the Region (Only If It's Unset).....	5
2.3 Name Your Radio	5
2.4 Then, in Order	6
3. The Console (Desktop).....	7
3.1 Connect Tab.....	7
3.2 Nodes Tab	7
3.3 Chat Tab (Rooms and DMs Together)	7
3.4 Profile Tab	8
3.5 Map Tab	8
3.6 Backup Tab	8
3.7 Settings Tab — And the Danger Zone.....	8
3.8 Firmware Tab and Test Bench	9
4. The Android App	10
4.1 Connect First: SETUP → Connect.....	10
4.2 Home — The Dashboard	10
4.3 Nodes Tab.....	11
4.4 Chat and Rooms Tabs	11
4.5 Profile Tab.....	11
4.6 SETUP	11
4.7 Tools	12
5. The T-Deck's Own Screens.....	13
6. The Watch	14
6.1 Install It (One Time, from a PC).....	14
6.2 Pick a Mode (First Launch).....	14
6.3 SEND TO — Choosing the Room (Direct Mode).....	15
6.4 Watch Limits, Stated Plainly	15
7. Concepts: Nodes, Profiles, Rooms, Friends.....	16
7.1 Nodes and "Represents"	16

7.2 Profiles and the Five Audiences	16
7.3 Hosted Profiles (For Stock Radios).....	17
7.4 Rooms: Public Vs. Private	17
7.5 Friends	18
8. SOS and Check-Ins	19
8.1 SOS	19
8.2 Check-Ins	19
8.3 Message Receipts — What ✓ Actually Means.....	19
9. Troubleshooting.....	21
10. Stay Connected	22

Outpost turns ordinary Meshtastic radios into a coordination system for a group traveling off-grid: named rooms instead of raw channels, profiles that say what each radio represents, friends, polls, waypoints, check-ins, and SOS — on the radio itself, on your phone, on your watch, and on a desktop console. No cell service, no account, no cloud. If two radios can hear each other (directly or through relays), Outpost works.

This manual is written for the person who owns the gear. Steps are short and plain. Where a feature has a real limit, the manual says so.

1. What Outpost Is

Outpost is four pieces that work together:

Piece	What it is	What it's for
Firmware	Outpost software that runs on the radio itself (a Meshtastic fork with the Outpost module inside)	Stores your profile on the radio, serves it to the group, runs the deck's own screens
Console	A single HTML file you open in Chrome or Edge on a PC, connected to a radio by USB cable	Setup, flashing firmware, backup, settings, the full radio log — the "workbench"
Android app	The phone companion	Day-to-day use: connect, chat, rooms, maps, profiles, friends
Watch app	A Wear OS app	Check-ins, quick replies, and SOS from your wrist

How they fit together:

- **The radio is the hub.** Your identity — your profile, your rooms, your friend list — lives on the radio, not in any one app. The phone app and the console are editors and windows into it. Unplug them and the radio keeps working.

- **Everything speaks one language.** All Outpost features ride on the standard Meshtastic mesh as small text packets (the "OVL1" protocol), so a plain stock Meshtastic radio still forwards Outpost traffic without understanding it — and can be a full member of the mesh with a hosted profile (see §7).
- **No internet is required for the mesh itself.** Maps can use cached tiles, the console is one local file, and the watch talks straight to a radio. Internet only helps for downloads and map tiles.

Example cast used throughout this manual: **Alex Rivers** is the operator. Alex's handheld radio is a T-Deck called **OGT1**. Alex also runs a small tracker called **OPD1**. The group's rooms are **CONVOY PUB** (public) and **CONVOY Private** (private).

[← Back to contents](#)

2. First Setup

2.1 Flash the T-Deck Firmware

1. Open the Console (§3) in Chrome or Edge and connect the T-Deck by USB.
2. Go to the **Firmware** tab, pick the firmware file (`outpost-tdeck-0.9.1.bin`), and flash. (Alternative: `esptool` users flash the app image at `0x10000`.)
3. The deck reboots into the Outpost boot screen: black background, the Outpost emblem, "OUTPOST MESHNET" in amber, and the build number in white at the very bottom.

Update vs. full erase — read this once:

- A normal **update flash** keeps everything that matters: your LoRa region, your rooms and their keys, and your owner name all survive. This is the flash you want, every time.
- The Firmware tab has a "**Full chip erase**" checkbox. Leave it **off** (it is off by default). A full erase is a factory reset: it wipes the region, rooms, and owner along with the firmware. Only use it deliberately, on a radio you intend to set up from scratch — and make a backup first (§3.6).

2.2 Set the Region (Only If It's Unset)

A radio with **no region set cannot transmit at all** — the screen or log will say "Region unset" or "LoRa TX disabled." This only happens on a brand-new or fully-erased radio; updates never clear it.

- **On the deck itself:** open the **SETUP** tab. If the region is unset, the region row offers **Set** for your region (for example **US**). Tap it; the radio reboots with the region applied. The deck only offers this while the region is unset — it is a recovery action, not a settings playground.
- **From the console:** the Settings tab has a single-purpose "**Restore region: US**" button for the same situation.
- The phone app **cannot** set the region. Region is a regulatory setting, so Outpost keeps it on the radio and in the recovery tools only.

2.3 Name Your Radio

Every radio has a **long name** and a **short name** (4 characters). Example: long name `OGT-Outpost-TDeck1`, short name `OGT1`. The short name is what people see in lists, and the radio's Bluetooth name becomes `OGT1_` plus the last four characters of its node ID — handy when several radios are in scan range.

Set the name in any of these places (they all write to the same radio): the deck's **SETUP** tab, the app's **SETUP** → **Radio**, or the console's **Settings** tab.

2.4 Then, in Order

1. Install the Android app (§4) and connect to your radio.
2. Fill in your profile (§7.2).
3. In the console's **Profile** tab, use **Save profile to radio** and wait for the confirmation (§3.4). Your radio now answers for you even when your phone is off.
4. Create or join your rooms (§7.4).

Tanmatsu owners: the Tanmatsu build is a full factory image — flash `tanmatsu-outpost-firmware.bin` at address `0x0`. It is headless (no Outpost screens on the badge); drive it from the app or console. Flashing it replaces the badge's hosted WiFi/BLE firmware; the stock firmware can be restored at recovery.tanmatsu.cloud.

[← Back to contents](#)

3. The Console (Desktop)

The console is one file, `outpost-console.html`. Open it in **Chrome or Edge** (it needs WebSerial, which those browsers provide) and keep using that same copy — the console stores its data (rooms, profile, settings) in the browser, tied to that file's location. When a new console version arrives, save it **over the old file, in place**, and your data carries across. The console's version is printed under the OUTPOST CONSOLE title in the header — check it there, not by filename.

3.1 Connect Tab

Plug the radio in by USB, click **Connect**, and pick the serial port. This tab also shows the **radio log** (the firmware's own debug output — the ground truth when anything misbehaves) and a plain readout of the radio's configuration, including region and Bluetooth state.

3.2 Nodes Tab

Every node the radio knows about, as cards: name, device model, signal, last heard, distance, battery, and — once profiles are shared — a **Represents** line saying what the node is (Person, Vehicle, Outpost, Basecamp, Repeater, Tracker) with a short profile summary.

- **Search** filters live by name, node ID, or profile handle.
- **Sort** by last heard, favorites, distance, hops, signal, or name; **Basic/Full** switches card density; click the ★ to favorite.
- **Right-click a card** for actions: direct message, request position, request profile, show on map, favorite, copy node number.

3.3 Chat Tab (Rooms and DMs Together)

Chat and rooms are one screen. A rail on the left lists every conversation — **Public Mesh**, each room, and each direct-message thread — and the panel shows the selected thread. The composer names its destination before you send ("Send to CONVOY Private"), so a message never goes somewhere the screen didn't say.

- **Manage rooms** (the ⚙ in Chat) creates rooms, joins by invite code, and shares invites — including a QR code a phone can scan.
- **Polls** live in a panel inside Chat: post a question with 2–4 options to the current room and watch votes tally live.
- **Reactions**: right-click or long-press a message for the emoji picker. Reactions use the standard Meshtastic format, so they also work with the official app.
- **Receipts** are honest — see §8.3 before you rely on them.

3.4 Profile Tab

This is the editor for the profile stored **on your radio**: your handle, what your node represents, the details for that kind of node, and the audience pickers (§7.2) for the profile and for its medical block.

Save profile to radio pushes the profile to the connected radio and waits for the radio itself to confirm. When you see "**Profile stored on radio ✓**", the radio holds it and will answer profile requests on its own. If the confirmation doesn't arrive, the console says the save is **unconfirmed** — it never pretends. (Radios running firmware older than 0.9.0 can't confirm; the profile still lives in the console and app.)

3.5 Map Tab

Nodes and waypoints on a real map. Layers: **CARTO dark** street map, **Esri satellite**, **Google** (needs your own Google API key, stored only in your browser), and an **Offline plot** that needs no network at all — positions are plotted relative to each other. Tiles you've viewed are cached; the plot layer always works in the field.

3.6 Backup Tab

Exports everything — rooms **including private keys**, your profile, and settings — as one **encrypted** file (PBKDF2-SHA256 + AES-GCM-256; your passphrase is never stored). Restore imports it back, and the console can then **re-provision** a radio in one click, replaying your rooms onto it — the fast way back after a full erase or onto a replacement radio. Treat the backup file like keys to a house: the encryption is only as strong as your passphrase.

3.7 Settings Tab — And the Danger Zone

Settings holds the radio's owner name, channel table, position settings, WiFi and Bluetooth toggles, reboot, the region recovery button (§2.2), and the console's own Units & display preferences (miles/km, °F/°C).

At the bottom sits the **Danger Zone**, guarded on purpose — both actions are two-step: the first click **arms**, the second click within 5 seconds **fires**:

- **Delete node database** — the radio forgets every node it has learned. Owner, rooms, and configuration are kept.
- **Factory reset device** — wipes everything on the radio. This is the nuclear option; backup first (§3.6).

3.8 Firmware Tab and Test Bench

The Firmware tab flashes `.bin` files over USB with a progress bar and log (see §2.1 — leave full erase off). The Test Bench tab is the workbench's test gear: a live decoded packet log and tools for sending test traffic — mostly useful when diagnosing with the radio log beside it.

[← Back to contents](#)

4. The Android App

On launch you get the Outpost splash — black, emblem, "OUTPOST MESHNET" — for a full three seconds. It's deliberate, not a hang. The app's version is stamped under the OUTPOST logo ("APP v0.9.7"); when anyone asks "which build is that?", that's the number.

Seven tabs run along the bottom: **HOME · NODES · CHAT · SOS · ROOMS · PROFILE · SETUP**. SOS sits in the center, red, on purpose. The map is not a tab — it opens as an overlay from Home.

4.1 Connect First: SETUP → Connect

Connection lives in **SETUP** → **Connect** (not on the Nodes tab).

- **Transport:** choose **Bluetooth** or **WiFi/TCP**.
- *Bluetooth:* scan, then tap your radio in the list. Radios show their friendly names (e.g. `0GT1`), not raw Bluetooth names.
- *WiFi/TCP:* for a radio with WiFi enabled on your network, enter its **host/IP** and **port** (default **4403**), optionally turn on **auto-reconnect**, and connect. WiFi transport is **new in 0.9.6** — it speaks the identical protocol as Bluetooth but is still awaiting its first full bench confirmation, so if it misbehaves, Bluetooth is the known-good path.
- **Recent Connections:** the last 5 radios you've connected to, newest first. Tap one to reconnect — no rescan needed.

While connected, the masthead shows a connection cluster: node, transport, and how long you've been connected, ticking live.

4.2 Home — The Dashboard

Home is a panel of **clickable stat tiles**. Each tile answers one question and takes you to the place that acts on it:

- **Our node** — your radio's name and ID → Profile.
- **Signal** — the real Bluetooth link signal → Connect.
- **Our GPS** — the position your radio is broadcasting → Map.
- **Last node discovered** and **Last heard** → Nodes.
- **Last message** — sender and a snippet → that chat thread.
- **Radio** — battery % and voltage.
- **Nodes / Friends / Favorites counts** → the Nodes list, already filtered to match.

A compact actions row carries **Check-in** and **Map**, and a **tools grid** (also mirrored in SETUP) opens the Radio Log, Traceroute, Field Guide, Almanac, Trip Log, and Backup & Data (§4.7).

4.3 Nodes Tab

The mesh as your radio knows it: search, sort orders, ★ favorites, and cards with the device model and Represents line. The radio you're connected to is highlighted. **Long-press any node** to open its **full profile page**: the public profile, plus actions — favorite, **friend request**, direct message, request position / telemetry / profile, **traceroute** (the route drawn as node names), share a room with them, hide node (this phone only), copy node ID.

4.4 Chat and Rooms Tabs

Chat is self-sufficient: a **dropdown at the top switches threads** — rooms and DMs side by side — and the composer names where the message will go. **Polls live inside Chat**: open the polls panel to vote, or post a new poll to the room you're in. Room management (create, join, invite, settings) is dialogs inside Chat; you are never bounced to a separate manager screen. Long-press a message to react.

The **Rooms tab** is the conversation index: every room with its **last-message preview** (time · sender · text), then a **DIRECT MESSAGES** divider with DM threads in the same format. It's the fastest way to see what you missed; tapping a row opens that thread in Chat.

4.5 Profile Tab

Your profile editor (§7.2): what your node represents, the fields for that kind, and the audience pickers. Saving stores the profile on the connected radio too — watch for "**Profile stored on radio ✓**"; if the radio can't confirm, the app says so plainly.

Device type is read-only here — it describes your hardware and comes from the radio, not from the editor.

4.6 SETUP

- **Connect** — §4.1.
- **Notifications** — a matrix: new nodes, public-room messages, private-room messages, and DMs, each on/off — plus a per-room switch list so one busy private room can notify (or not) on its own. Private-room switches default off.
- **Display & Units** — night mode (red-on-black, easy on night vision) and units: miles · °F or km · °C.
- **Radio** — rename the connected radio (owner), radio admin, and a **Danger Zone** with the same two guarded actions as the console (delete node database, factory reset — arm, then confirm).
- **Profile extras** — incoming **friend requests** to accept or decline, and the list of nodes you've hidden, to unhide.
- **About** — app version and build info.

4.7 Tools

- **Radio Log** — a live list of the packets and events the app sees: what arrived, from whom, on which port. First stop when something's odd.
- **Traceroute** — pick a node and ask the mesh how packets reach it; the route comes back as node names.
- **Field Guide** and **Almanac** — offline reference (knots, water, first aid, stars) and sun/moon times computed on the phone. No network needed.
- **Trip Log** — your recorded track, check-ins, waypoints, and SOS events, exportable.
- **Backup & Data** — exports your profile, rooms, and friends as a JSON file you can keep or move. (The console's backup, §3.6, is the encrypted full-system one; this is the quick shareable export.)

[← Back to contents](#)

5. The T-Deck's Own Screens

With firmware 0.9.1 the deck is fully Outpost: seven tabs along the bottom — **HOME**, **NODES**, **CHAT**, **SOS** (red, center), **ROOMS**, **PROFILE**, **SETUP** — and no door back to the stock Meshtastic screens. Advanced settings deliberately live in the console; the deck carries what you need in the field.

- **HOME** — status at a glance (radio, region, battery, GPS), your current room, quick tiles, and a **LAST MESSAGE** strip fed by live traffic.
- **NODES** — every node the deck knows. Open one for its detail page, including **Request position**, which asks that node to report where it is.
- **CHAT** — read and send room messages on the **physical keyboard**. A room selector at the top switches rooms; keep messages short — the composer caps at 120 characters. Polls for the current room are one tap away.
- **SOS** — arm or cancel an SOS (§8). The screen also shows your stored medical note the way responders will receive it.
- **ROOMS** — the conversation index: each room the radio holds, with its last message.
- **PROFILE** — read-only: what this radio has stored and serves — your audiences, any hosted profiles (§7.3), and pending profile requests. Edit the profile from the app or console.
- **SETUP** — writes straight to the radio: owner name, the **region row** (§2.2 — Set appears only while the region is unset), Bluetooth on/off, position broadcast interval, screen timeout, night mode, and About. The last line says it plainly: *advanced settings live in the console*.

[← Back to contents](#)

6. The Watch

The watch app does three things from your wrist: **check in**, send **quick replies**, and raise/cancel **SOS**. The status screen shows your node name, battery, how many nodes are known, the last message, and SOS state. Buttons buzz (haptics) so you know a tap landed.

6.1 Install It (One Time, from a PC)

The watch app is sideloaded with Android's `adb` tool:

1. On the watch: Settings → About watch → Software info → tap **Build number** 7 times to unlock Developer options.
2. Settings → Developer options → turn on **Wireless debugging**. Note the IP address and port it shows. Tap **Pair new device** and note the pairing code and its (different) port.
3. On the PC, `adb.exe` is a **command-line tool** — **do not double-click it** (a window that opens and instantly closes is normal; it has nothing to show you). Instead, open the folder containing it, click in File Explorer's address bar, type `cmd`, press Enter, and run the commands in that window. No admin needed.
4. Copy the watch APK into that same folder, then: - `adb pair <watch-IP>:<pairing-port>` — enter the 6-digit code. - `adb connect <watch-IP>:<wireless-debugging-port>` - `adb install outpost.watch.0.2.0.apk`
5. **Check the file before you install:** the APK on your PC must be **exactly the delivered size** (the 0.2.0 file is **774,977 bytes** — right-click → Properties). A different size means the download was mangled; delete it and download again. If your browser saved it as `.zip`, rename it back to `.apk` — do not unzip it. (§9 has the matching error and fix.)

Ports change when the watch's Wi-Fi reconnects, so read fresh ones each session. Updates install the same way, over the old copy.

6.2 Pick a Mode (First Launch)

The watch asks once; you can change it later on the status screen.

- **DIRECT TO RADIO** — the watch pairs to a radio **itself** over Bluetooth: scan, tap the radio, done. It remembers the radio and reconnects on its own (with one quiet retry after a drop). No phone involved. A **FORGET SAVED RADIO** row on the status screen clears the pairing.
- **THROUGH PHONE** — the watch links to the **phone app** instead, and the phone does the sending. The phone app must be **running** — the relay lives in the app, so if the app is closed, linked mode is dark until you open it again.

6.3 SEND TO — Choosing the Room (Direct Mode)

In direct mode, a **Send to:** row sits above CHECK-IN. Tap it and pick any channel the radio holds — **Public (primary)**, CONVOY PUB, CONVOY Private, and so on. Check-ins and quick replies ride the room you picked, and the choice is **remembered per radio**, so the deck and the tracker can each have their own default. Incoming messages on the status line name their room ("CONVOY Priv · Alex · At camp"). In linked mode there is no picker — the phone sends on the room active in the app.

SOS is the exception, everywhere: it always goes on the **public** channel, in both modes. The picker says so on its face. See §8.

6.4 Watch Limits, Stated Plainly

- A watch-sent SOS carries **no GPS position of its own** — the alert goes out, and your radio's own position broadcasts are what place you on the map.
- Direct mode tracks the SOS state **it** set; the phone's view of the world is the fuller one.
- There's no ambient (always-on) watch face mode yet.

[← Back to contents](#)

7. Concepts: Nodes, Profiles, Rooms, Friends

7.1 Nodes and "Represents"

A **node** is one radio. It has a node number (written like `!a1b2c3d4`), a long and short name, and a position on the mesh. Every node also declares what it **represents** — its function:

Represents	Means	Profile carries
Person	A human carrying the radio (Offroader, Camper/RVer, or Hiker)	Handle, party size, your details — and optionally a link to your vehicle node
Vehicle	A node mounted in/on a vehicle; it speaks for the rig	Vehicle type: Car, Offroader, Overlander, Camper/RV, Van, Motorcycle, UTV/Side-by-Side, Trailer, Recovery, or Medical
Outpost	A fixed station — home, or a long-term camp	Power source, antenna height, services
Basecamp	A station set up for this trip's camp	Power, antenna height, services
Repeater	A coverage relay, mostly unattended	Power, antenna height, services
Tracker	A position beacon — its job is to be tracked	Just a handle and party size

Node lists everywhere (app, console, deck) show the Represents line, so a glance tells you "that's Alex," "that's Alex's truck," and "that's the tracker on the trailer."

7.2 Profiles and the Five Audiences

Your **profile** is the card other nodes can request from your radio: handle, what you represent, the details for that kind, and a **medical** block (allergies, conditions, meds — whatever you'd want a rescuer to know).

The profile **lives on the radio**. The app and console edit it and push it to the radio (§3.4, §4.5); the radio stores it and answers requests itself, even with your phone off.

Who gets to see it is set by **audiences** — the same five tiers everywhere in Outpost:

Tier	Who receives your profile when they ask
Public	Anyone on the mesh

Tier	Who receives your profile when they ask
Friends	Nodes on your friend list (\$7.5)
Room members	Requests arriving through one of your rooms (the private ones are the meaningful scope)
Ask	Nobody automatically — requests are logged, and you approve case by case
Off	Nobody; the radio stays silent

The **medical block** has its own **audience picker** on the same five tiers, usually set tighter than the profile itself. Two deliberate exceptions: when you send an **SOS**, your medical note rides with it regardless of the medical audience — that's the point of an SOS — and medical is **never** included in public profile beacons.

Set audiences with a clear head: encryption hides a private room's *content*, but your presence on the mesh is not hidden — any radio in range can hear that *a* node is transmitting, and node lists show every node heard.

7.3 Hosted Profiles (For Stock Radios)

A tracker like **OPD1** runs stock firmware — it can't store or serve an Outpost profile itself. An Outpost radio can **host** one for it: the profile is saved against the tracker's node number on the host (say, OGT1), and the host answers requests for it. Other clients file it under the tracker and mark it "**via OGT1**" so you know who's speaking for it. The hosted profile's audiences are enforced by the host, same as its own.

7.4 Rooms: Public Vs. Private

A **room** is a named conversation on one of the radio's channel slots (1–7; slot 0 is the open primary mesh, labeled **OUTPOST Public**). Rooms have a category — **CONVOY** (offroad groups), **BASECAMP** (campers/RVers; bulletin-first — the latest bulletin pins at the top of the room), or **TRAIL** (hikers; safety-first).

- A **public room** (like CONVOY PUB) uses the mesh's standard default key. Anyone in range can read it. That's not a flaw — it's for coordination with whoever's around.
- A **private room** (like CONVOY Private) has its own random 32-byte key. Only nodes holding that key can read a word; everyone else just relays sealed packets. Joining is by **invite**: a code (starting `OUTPOST:`) you paste, or a QR code from the console that a phone scans. Whoever you hand the code to can read the room — hand it out like a key, because it is one. To lock someone out, make a new room and re-invite the rest.
- The radio stores channel names truncated to **11 characters** ("CONVOY Priv" on the radio itself). Your app and console keep the full name and use it everywhere they display the room.

Creating a room from any client (app, console, or the deck's own + **Room** flow) writes the channel to the radio immediately — no reboot, no separate "set on radio" step to remember.

7.5 Friends

Friends are nodes you've vouched for. From a node's profile page, send a **friend request**; it travels over the mesh itself. The other side accepts or declines (requests wait for them in their app's SETUP → Profile). If you both request each other at once, the requests cross and complete automatically. Unfriending is one tap on the profile page.

Why bother: the **Friends audience tier** (§7.2). Your friend list also syncs to your connected radio, so the radio can answer a friend's profile request *as* a friend even when your phone is off. Friends get profile answers; medical still follows its own, separate audience — friendship alone doesn't unlock it.

[← Back to contents](#)

8. SOS and Check-Ins

8.1 SOS

SOS is the same on every surface — app SOS tab, deck SOS tab, watch, console:

1. **Arm it** (on the watch: tap SOS, then tap again to confirm — two taps, so a pocket can't send one).
2. Your radio broadcasts an **SOS beacon with its position** on the **public channel**, repeats it about every minute, and nearby Outpost nodes relay it once to help it punch through a sparse mesh.
3. Attached to the SOS is your **profile card — medical included**, whatever your medical audience says. In an emergency, responders get the note. That's deliberate and labeled wherever you write the note.
4. **Cancel** stands it down everywhere.

Why public? An SOS is meant to be heard by *strangers too* — the convoy two ridges over may be closer than your own group. So SOS rides the public channel **by design, in every app, on the watch, and on the deck**, even when everything else you send goes to a private room. A watch-sent SOS carries no position of its own; your radio's position broadcasts place you (§6.4).

A related, quieter tool: **Assist** asks your room for non-emergency help (a tug, a jump, tools, other) with a calm alert — no siren, no public broadcast.

8.2 Check-Ins

A **check-in** is one tap that broadcasts "*I'm OK*" to your current room (on the watch: to your SEND TO room, §6.3). Use it on a schedule or whenever you change plans. Both the app and the deck can also check in automatically on an interval (30/60/120 minutes) — set it and forget it. And the **overdue** safety net works the other way: arm a return time, and if you don't check in or move by the deadline plus a grace period, your group is notified with your last known position.

8.3 Message Receipts — What ✓ Actually Means

Mesh radio has no postman, so Outpost's receipts only ever claim what they can prove:

- **Room messages:** "✓ **relayed**" means exactly that — another node was *heard rebroadcasting* your message, proof it entered the mesh. If no relay is heard within ~15 seconds, the app retries **once** automatically. After that the message settles as amber "**— no relay heard · tap to resend.**" Amber does **not** mean failed — it means *unproven*. It may well have arrived; nobody repeated it where you could hear. Tap to resend manually if it matters.
- **Direct messages** are stronger: a DM is addressed to one node whose radio answers with an acknowledgment, so "✓ **delivered**" is real delivery proof. "**— no ACK · tap to resend**" after ~20 seconds means the peer never confirmed.

- **The two-node bench trap:** on a test bench with just your radio and one other node, a room message may *never* show ✓ — there's no third node to relay it, and some nodes (trackers, set to the Tracker role) **don't relay at all**. That's the system being honest, not broken. Add a third node and the ✓ appears.

[← Back to contents](#)

9. Troubleshooting

The radio won't transmit — "Region unset" / "LoRa TX disabled." The region was never set, or a full-erase flash cleared it. Set it from the deck's SETUP tab (the region row offers Set only while unset) or the console's "Restore region: US" button (§2.2). Normal update flashes never cause this — if it happened, an erase was involved.

Node counts look wrong — "250 nodes"?! or ghosts that never leave. Old firmware counted *slots*, not nodes; update the deck to firmware 0.9.1, which reads the real count. For genuinely stale entries: the app has **Forget stale nodes** (by age: 24 h / 7 d / 30 d), and the console's **Delete node database** (Danger Zone, §3.7) makes the radio forget everyone and re-learn the live mesh.

Watch install fails: `INSTALL_PARSE_FAILED_NOT_APK`. Your downloaded APK was mangled on its way to the PC — this is a file problem, not a watch problem. Check the size (Properties): it must match the delivered size **exactly** (watch 0.2.0 = 774,977 bytes). If it doesn't — or if the file arrived as `.zip` — delete it, download a fresh copy, rename `.zip` back to `.apk` if needed (never unzip it), and install again (§6.1).

Two files with the same name — which one do I have? Never trust a filename. Check the **version inside the thing itself**: the app stamps it under the OUTPOST logo and in About; the console prints it in its header; the deck shows its build on the boot screen and in SETUP → About; the watch stamps its status screen. This bit Outpost itself once — a 0.9.0 deck file circulated the same night as a different 0.9.0 build, which is why the deck UI release is **0.9.1** and why changed content never reuses a version number again. When in doubt, compare file sizes too.

Receipts never show ✓. See §8.3. On a two-node bench it's expected (nothing to relay; trackers don't relay). In the field, check the obvious: the message went to the room you think it did (the composer names its destination — read it), and at least one relay-capable node is in range.

The app won't connect over WiFi. WiFi transport is new in 0.9.6 (§4.1). The radio must have WiFi enabled and be on the same network as the phone, and the port is 4403 unless you changed it. If it still won't come up, use Bluetooth — it's the proven path — and note what the app reported.

The console opened blank tabs / lost my data. Always open the **same file in the same place** — console data lives in the browser, tied to the file's path, so moving or re-downloading the file into a different folder starts you empty. Update by overwriting the file **in place**. Restore from a Backup export (§3.6) if you have one.

A private room shows a short name ("CONVOY Priv") on the radio. Normal. Radios cap channel names at 11 characters; your app and console keep and display the full name. Nothing is lost.

[← Back to contents](#)

10. Stay Connected

Please follow us on social media for the latest updates and software release announcements. Visit our website at www.offgridtactix.com for more information and downloads.

Need Support? Reach out to us on social media (links below), or email us at support@offgridtactix.com.

- **Website:** www.offgridtactix.com
- **Instagram:** [@offgrid.tactix](https://www.instagram.com/offgrid.tactix)
- **Facebook:** [OFFGRID TACTIX on Facebook](#)
- **Email:** support@offgridtactix.com

[← Back to contents](#)

Manual version: 1.1 — tracks App 0.9.7 / Console 0.9.11 / Watch 0.2.0 / Firmware 0.9.1. This manual is a living document: any release that changes a screen it covers updates it in the same release.