

GMRS-TTY

User Manual — Full Reference

Version v1.9.0

A TTY-style accessibility communicator for the General Mobile Radio Service (GMRS) and Family Radio Service (FRS), built for deaf, hard-of-hearing, and mute operators.

Live transcription · offline TTS · FCC Part 95 ID rules built in

Read the keyboard-shortcut reference on page 4 if you want to skip the tour and start operating immediately.

Contents

1. About this manual
2. System requirements
3. Installation — pre-built packages & from source
4. First run & configuration
5. Main window tour
- 5.11. Session Journals
- 5.12. Touch-screen mode
6. Configuration dialog
7. Contacts dialog
8. Quick Messages dialog
9. Add Station dialog
10. Keyboard shortcuts & mnemonics
11. GMRS vs FRS service mode
12. PTT (Push-to-Talk) modes
13. Receive (Rx) pipeline
14. Transmit (Tx) pipeline
15. Callsign detection & FCC verification
16. FCC Part 95 compliance features
17. Accessibility (WCAG 2.1 AA)
18. Off-grid operation
19. Troubleshooting
20. File reference (config.json, contacts.json)
21. Glossary

1. About this manual

This manual is the full operating reference for GMRS-TTY, a cross-platform desktop application that lets deaf, hard-of-hearing, and mute operators participate in two-way voice radio conversations over GMRS and FRS. The program is written in Python with PySide6, runs fully offline once installed, and ships every feature behind the FCC Part 95 rule set so operators can stay legal without memorizing the regulations.

The manual is organized as a reference rather than a tutorial. Section 5 walks every region of the main window in top-to-bottom order. Sections 6–9 document each dialog. Section 10 is the keyboard cheat sheet. Sections 11–16 cover the deeper behaviors (service mode, PTT modes, the audio pipelines, callsign verification, and FCC compliance). Section 19 is troubleshooting. Sections 20–21 give the on-disk file formats and a glossary.

If you only have five minutes (pre-built installer): install the `.deb` or `.msi` (section 3.1), drop one Piper voice into the `Voices/` subdirectory of the install folder, then launch GMRS-TTY and open **Settings** → **Configuration** to set your callsign and pick the voice. Press **Alt+L** to listen and type into the message box to transmit. **From source:** follow section 3.2 (clone, pip install, `bootstrap_models.py`, run).

2. System requirements

Operating system

- **Linux** — Debian 12+, Ubuntu 22.04+, Raspberry Pi OS (Bookworm). PortAudio dev libs required: `sudo apt install libportaudio2 portaudio19-dev`. On PipeWire/PulseAudio systems also install `pulseaudio-utils` (`sudo apt install pulseaudio-utils`) — provides `parec` and `pactl`, used for both microphone capture (preferred over PortAudio's PipeWire-via-ALSA bridge, which can silently deliver flat-zero audio on PipeWire 1.4) and System Audio Output (loopback) mode. Without it the app falls back to PortAudio for mic capture; loopback mode requires it.
- **Windows** — Windows 10 or 11. PortAudio ships in the wheel; no extra system libraries needed.
- **macOS** — Not formally targeted, but Python 3.11+ with PortAudio available through Homebrew usually works.

Hardware

- **Microphone** — Anything Linux/Windows recognizes — a USB headset, the built-in laptop mic, or a sound-card channel cabled from your radio's speaker output.
- **Speaker / radio audio path** — A speaker for live listening, or a USB sound card / Signalink / Digirig for feeding TTS audio directly into your radio.
- **Optional: PTT cable** — A USB-serial adapter (FTDI is the most common) for keying PTT via RTS/DTR. Not needed if your radio has VOX or you key manually.

Software

- **Pre-built installers (section 3.1)** — No Python or pip required — Python 3.13, all dependencies, and the offline STT models are bundled. ~1.5 GB disk for the installer and installed footprint.
- **From source (section 3.2+)** — Python 3.11 or newer (3.13 recommended). ~1 GB free disk for Python dependencies plus 75–150 MB for the Whisper STT model. Internet access once to install pip packages and fetch the model; the core radio workflow needs no network after that.

3. Installation

3.1 Pre-built installers (recommended for end users)

The pre-built installers bundle Python 3.13, all Python packages (CPU-only PyTorch, faster-whisper, PySide6, Piper, Silero VAD, and their dependencies), and the offline Whisper STT + speaker identification models. No Python installation or internet access is needed after downloading the installer.

Piper TTS *voice* models are **not** bundled — they are large, numerous, and user-chosen. Add them after install (see section 3.1.3 below).

3.1.1 Debian / Ubuntu (.deb)

Targets Debian 13 / Ubuntu 24.04+ on x86-64. System dependencies installed automatically by apt.

```
sudo apt install ./gmrs-tty_0.0.1_amd64.deb
gmrs-tty
```

The package installer (postinst) creates a Python virtual environment at `/opt/gmrs-tty/.venv`, installs all bundled wheels offline, and seeds an initial `config.json` from the example. Open **Settings** → **Configuration** on first launch to set your callsign, name, location, and Piper TTS voice.

- **Install location** — `/opt/gmrs-tty/` — writable by all users so the app can persist `config.json` and `contacts.json` in-place.
- **Launcher** — `/usr/bin/gmrs-tty` — runnable from a terminal or the desktop shortcut created at install time.
- **Uninstall** — `sudo apt remove gmrs-tty`. The venv and any locally-staged Piper voices under `/opt/gmrs-tty/Voices/` are preserved; only the package files are removed.

3.1.2 Windows (.msi)

Bundles Python 3.13 embeddable runtime. No separate Python installation required. Installs to `C:\Program Files\GMRS-TTY\` by default.

- Run the `gmrs-tty_0.0.1_x64.msi` installer and step through the wizard.
- A **GMRS-TTY** shortcut is added to the Desktop and Start Menu, pointing at the bundled `python\pythonw.exe main.py`.
- Open **Settings** → **Configuration** on first launch.
- Uninstall via Windows Settings → Apps.

The Windows installer is built by `scripts\build-msi.ps1` on a Windows machine using WiX v4. Source builds are not required to run the pre-built MSI.

3.1.3 Adding Piper TTS voices

Piper ONNX voices must be added manually after install. Drop each `.onnx + .onnx.json` pair into the `Voces/` subfolder of the install directory:

```
# Debian / Ubuntu
```

```
ls /opt/gmrs-tty/Voices/
```

```
# Windows – in Explorer:
C:\Program Files\GMRS-TTY\Voices\
```

Download voices from: <https://github.com/rhasspy/piper/blob/master/VOICES.md>. Most voices are MIT-licensed; en_US-libritts-high is CC BY 4.0 and requires attribution if redistributed.

After adding voices, open Configuration and pick one from the **Voice Model** dropdown. Click **Test** to preview before saving.

3.2 From source (developer / advanced install)

Five steps from a fresh clone to a working radio session.

3.2.1 Clone and create a virtual environment

```
git clone <repo-url> GMRS-TTY
cd GMRS-TTY

python3 -m venv .venv
source .venv/bin/activate          # Linux / macOS
# .venv\Scripts\activate          # Windows

pip install -r requirements.txt
```

3.2.2 Drop in a Piper voice

Download at least one Piper ONNX voice and its matching .json config file into a Voices/ directory at the project root. The Configuration dialog reads this folder at launch and lists every voice it finds.

```
Voices/
■■■ en_US-ryan-high.onnx
■■■ en_US-ryan-high.onnx.json
■■■ en_US-amy-medium.onnx
■■■ en_US-amy-medium.onnx.json
```

3.2.3 Bootstrap the speech-to-text model

The Whisper STT model is not bundled in the repo. Fetch it once on an internet-connected machine:

```
python bootstrap_models.py          # default: small.en
python bootstrap_models.py --model base.en    # smaller, faster, less accurate
python bootstrap_models.py --model medium.en  # higher accuracy, slower
```

This populates Models/STT/<model_name>/ with the faster-whisper CTranslate2 artifacts. The app loads from there on Listen and never attempts network access — if the directory is missing, listening fails fast with an instruction to run the bootstrap.

For air-gapped installs: run the bootstrap once on an internet-connected machine, then copy the entire Models/ directory (alongside the source) to the offline target. Silero VAD and Piper voices ship as local files already, so no other fetches are involved.

3.2.4 Configure

```
cp config.example.json config.json
```

```
$EDITOR config.json    # set your callsign, name, location, voice
```

You can also leave the file at its defaults and edit everything through the in-app Configuration dialog after first launch.

3.2.5 Run

```
source .venv/bin/activate
python main.py
```

4. First run & configuration

On first launch the app reads `config.json` from the working directory. If your callsign is still `YOUR_CALL` the header will display that literal string — open Settings → Configuration (or click the gear icon on the right side of the service toolbar). The dialog has six tabs; the minimum fields for a working session are:

- **Identity tab — Callsign** — Your FCC GMRS callsign (e.g. WSLZ233). Saved uppercased.
- **Identity — Name** — Your operator name. Appears in the callsign preface and the station-ID button output.
- **Identity — Location** — Free-form city / state. Used by the standalone **This is** ID announcement.
- **Audio tab — Input Device** — Microphone the Listen button captures from. *System Default* works for most setups. Select *System Audio Output (loopback)* to capture whatever is playing through the computer's speakers — open YouTube in a browser, play a podcast, or use any media player and the app will transcribe that audio. When this mode is selected a **Monitor Sink** dropdown appears so you can target a specific output device; *System Default* follows your OS default playback device. No extra tools required.
- **Audio — Output Device** — Where TTS audio is played — choose a USB sound card / Signalink / Digirig channel here to feed your radio directly.
- **Audio — VAD Threshold** — Silero VAD sensitivity (0.10–0.95). Lower is more sensitive (catches quiet signals but more false starts); higher is stricter.
- **Voice tab — Voice Model** — Pick the Piper voice you dropped into Voices/. Click **Test** to hear a sample before committing.
- **Voice — Speech Rate** — Slider from 0.70× (faster) to 1.50× (slower); 1.00× is the voice's native pace.
- **PTT tab — PTT Mode** — Manual, VOX, or USB FTDI / Serial. See section 12.
- **Behavior tab — Time Format** — 24-hour or 12-hour AM/PM timestamps on RX lines.
- **Behavior — Filter profanity** — On by default. Masks the f/s-words and similar with asterisks in both RX transcripts and outgoing TX before TTS speaks them.

Click **OK**. The header updates, the dropdown lists the new voice, and the listener restarts automatically if you changed the input device or VAD threshold.

Configuration is stored as plain JSON in `config.json` next to `main.py`. You can hand-edit it between runs; the in-app dialog and the file are interchangeable.

5. Main window tour

The main window is built on Qt's dockable-panel system: a movable **Service toolbar** at the top, a **Chat surface** in the center, four rearrangeable docked panels (**Station**, **Waterfall**, **Pending Stations**, **Quick Messages**), and a **Transmit** panel that stays pinned to the bottom by default but can also be moved or floated. The menubar holds Settings + View, and the status bar carries the live online/offline indicator on its right edge. The layout (dock positions, sizes, tabs, floats, window geometry) persists to `config.json` under `ui_layout` on close and restores on the next launch — see section 5.5 for customization details. Minimum window size is 720×520 to guarantee no clipping at high-DPI / large-font settings; the default is 960×720.

5.1 Service toolbar (top by default)

A movable `QToolBar`. Drag the left-edge handle to relocate it to any of the four toolbar areas (top, bottom, left, right). Hide via the View menu if you want to keep service-mode controls keyboard-only.

- **Service:** — Label.
- **GMRS / FRS** — Segmented radio buttons (Alt+G / Alt+F). Choose your operating service. The selection persists to `radio_service` in `config.json`. See section 11 for the full behavior diff.
- **Touch toggle (■ / ■)** — Leftmost icon in the cluster. Click ■ to enter touch-screen mode; click ■ to return to the desktop layout. In touch mode all dock panels hide automatically and the central panel switches to a large-button view designed for finger operation. Persists to `touch_mode` in `config.json`. See section 5.12 for full details.
- **Theme toggle (■ / ■■)** — Moon glyph in light mode, sun glyph in dark mode. Repaints the entire UI instantly: window background, conversation log background and text, header, dock title bars, menus, callsign pills, and status bar. The glyph shows the *destination* state — a moon means click for dark. Persists in `dark_mode`; stays enabled in both modes.
- **Q icon** — Bold capital Q. Opens the Quick Messages editor — same destination as Settings → Quick Messages.
- **Person icon** — Bust-in-silhouette glyph. Opens Contacts — same destination as Settings → Contacts or Ctrl+B. Disabled in FRS mode (no callsigns).
- **Notebook icon (■)** — Opens the Session Journals browser — same destination as Tools → View Session Journals or Ctrl+Shift+J. Enabled in both modes.
- **Gear icon** — Cog wheel. Opens Configuration — same destination as Settings → Configuration or Ctrl+,. Enabled in both modes.

5.2 Station panel (dock — Ctrl+Shift+S)

Docked at the top by default. Drag its title bar to move it to any side of the chat surface, float it as a separate window, or tab it with another panel. The panel content is a bold strip: in GMRS mode it reads Station: WSLZ233 | Operator: Benjamin | Location: Lansing, MI. In FRS mode the station segment is replaced with FRS Mode because FRS has no callsign requirement.

5.3 Chat area

A single horizontal **Listen strip** sits directly above the conversation log. It lives in the always-visible central widget rather than in any dock, so the RX controls and the chat they feed stay reachable independent of which docks are shown, moved, or floated.

- **Listen toggle** — Leftmost on the strip. Alt+L or Ctrl+L. Starts or stops microphone capture and live transcription. Loads the Whisper model from `Models/STT/<model>/` on first start; subsequent toggles are instant. The label flips to *Listening...* while active so the state is visible at a glance, and the accessible description updates for screen readers.
- **Listen only (RX-only safety)** — Sits immediately right of the Listen toggle. Mnemonic Alt+O. Checkable: when on, every TX path is blocked — Transmit, **This is**, Enter-to-send in the message box, the quick-message preset buttons, and the Ctrl+Return / Ctrl+I / Alt+1...Alt+9 global shortcuts all refuse to fire (the buttons grey out so the gate is visible). Microphone capture, transcription, callsign detection, callsigns detected, and the chat surface keep working normally. Persists to `config.json` under `listen_only`, so an operator who finishes a session in RX-only mode comes back up the same way. Toggling back off re-enables transmission instantly — and simultaneously disables the Monitor toggle (see below).
- **Monitor (audio monitor)** — Sits to the right of the Listen only toggle, separated by a small gap that signals it is a conditional sub-tool. Mnemonic Alt+M. Checkable: when on, incoming radio audio is routed unfiltered to the configured output device in real-time — only the 16 kHz → 48 kHz polyphase upsample is applied — so the operator hears the raw channel through speakers while the app simultaneously transcribes it. Transcription is not affected. **Available only when Listen-only mode is active** — enabling Monitor on a channel where transmissions are possible would risk audio feedback through the radio mic, so the button is greyed out whenever Listen only is off. When Listen-only is turned off the monitor stream stops automatically. Persists to `config.json` under `monitor_enabled`. A power-on default is also configurable from **Settings** → **Configuration** → **Monitor audio**.
- **Live input-level meter** — Thin horizontal bar stretching across the middle of the strip — real-time peak amplitude of the captured audio. Use it to verify your mic / cable / device is wired up; if it stays at zero while you key audio into the radio, the app isn't getting audio. Stays at zero when Listen is off.
- **Clear chat button** — Right-aligned on the strip. Ctrl+K from anywhere, or Tools → Clear Chat from the menu. Asks for Yes/No confirmation then wipes the log. Chat history is in-memory only — it cannot be recovered once cleared.
- **Conversation log** — Timestamped messages. Incoming lines are green [RX HH:MM:SS] :. Outgoing lines are blue [TX to <target>] :. Errors say Error: or Failed: in the text — state is never carried by color alone.
- **Auto-tail** — New messages keep the viewport pinned to the bottom while you're caught up. Scroll up to re-read older context and incoming traffic will *not* yank you back.
- **Callsign pills** — Any callsign that matches a saved contact is rendered with an amber, bold pill background. Hover for every operator linked to that callsign — useful for family-shared GMRS calls. A green ✓ appears immediately after the callsign when the contact has a confirmed FCC license match (hover the check for the FCC license verified tooltip).

5.4 Transmit panel (dock — Ctrl+Shift+T)

Pinned to the bottom by default. Movable and floatable but **not closable** — the input row is operationally critical, so an accidental dismiss cannot leave the operator with no TX path. The dock holds, left-to-right: **Target**, **Message box**, **Transmit**, and the standalone **This is** ID button. The Listen toggle and live input-level meter were previously bundled into this dock; they now live in the Listen strip above the chat (see Section 5.3) so RX feedback stays visible regardless of dock state.

- **Target dropdown** — Pick a contact callsign or *All*. Entries are sorted alphabetically; *All* is pinned at the top. Family-shared GMRS calls appear on separate rows like WSLZ233 (Eliza) / WSLZ233 (Jennifer), and the preface speaks the exact operator name from the row you selected.
- **Message box** — Text input. Type your message and press Enter or click Transmit. Placeholder reads *Type your message here....*
- **Transmit button** — Alt+T or Ctrl+Return / Ctrl+Enter. Sends the message through the TX pipeline: shorthand expansion, profanity masking, callsign framing, 15-minute ID check, PTT keying, TTS synthesis, and STT auto-pause until the unkey.
- **This is button** — Alt+I or Ctrl+I. Sends a standalone station ID — *This is <CALL>, <NATO phonetic>. <name> from <location>.* — without needing to type anything. Resets the 15-minute ID timer. Disabled in FRS mode (FRS has no ID rule); hover the disabled button for the explanation.

5.5 Customizing the layout

Drag any panel's title bar to dock it on the left, right, top, or bottom of the chat surface — or release it outside the main window to float it as a separate window. Drop one title bar onto another to tab two panels together. Drag the splitters between docked areas to resize.

Right-click a panel's title bar — or press the **Menu** key while the title bar has focus — for a keyboard-accessible **Move to Left / Right / Top / Bottom, Float / Re-dock**, and **Hide** menu. The keyboard path mirrors the mouse drag for operators who can't drag. **F6 / Shift+F6** walks keyboard focus across visible panel title bars so you can land on a panel and open its menu in two key presses.

Shortcuts: **Ctrl+Shift+S / P / Q / T** show or hide the Station / Pending / Quick Messages / Transmit panels (Transmit, having no Close, refocuses instead of hiding). **Ctrl+Shift+W** toggles the Waterfall (also at View → Show waterfall). **Ctrl+Shift+0** snaps everything back to the documented default arrangement while preserving your dark-mode and waterfall preferences.

The layout is saved to `config.json` under `ui_layout` on close — only on close, so dragging doesn't churn the file. If the saved state is missing, malformed, or from a different schema version, the default arrangement is used and re-written on next close. Forward-only migration: no user action needed when upgrading.

5.6 Pending Stations panel (dock — Ctrl+Shift+P)

Docked at the bottom by default, tabbed with Quick Messages. Hidden when no pills are pending so an empty titled frame doesn't sit on screen. Yellow pill buttons appear when a new GMRS callsign is detected on RX:

- **Click** a pill to open the Add Station dialog (section 9) pre-filled with the detected name and location.
- **Right-click** or long-press a pill to dismiss that one without adding it.

- **Dismiss all** (Alt+D) appears on the right edge whenever any pill is present — clears every pending pill at once.
- Pills wrap to additional rows up to a maximum of three; past that a vertical scrollbar appears so the chat area doesn't get squeezed.

5.7 Quick Messages panel (dock — Ctrl+Shift+Q)

Docked at the bottom by default, tabbed with Pending Stations. Hidden when the preset list is empty. Each button rides the standard TX pipeline. Curly-brace tokens like {N} in QSY to channel {N} prompt for a value before transmitting. The first nine buttons are bound to **Alt+1** through **Alt+9**. Edit the list from Settings → Quick Messages or the Q icon.

The seed list is: *Radio check, Loud and clear, Standing by, Acknowledged, Say again, QSY to channel {N}, Clear, Monitoring, Net check-in, Emergency traffic.*

5.8 Callsigns Detected panel (dock — Ctrl+Shift+A)

A roll-call grid that records every callsign detected during the current Listen session. Docked at the bottom by default, tabbed with Pending Stations and Quick Messages. **Off by default** — enable it from **View** → **Show callsigns detected** or **Settings** → **Configuration** → **Callsigns Detected** (Alt+D in the dialog). Persists at attendance.enabled in config.json. Disabled in FRS mode alongside every other callsign-dependent surface.

- **Columns** — **Callsign | Name | Location | GMRS | HAM**. Read-only; the grid is for reference, not editing. All columns autofit to their content each time a new callsign is recorded. Drag any column divider to adjust widths manually — the autofit on the next detection will refit all columns to content again.
- **Auto-population** — Unknown callsigns appear with only the Callsign column filled. The moment that callsign is added to (or already in) Contacts, the remaining four columns fill in automatically from the contact row — adding a station retroactively fills its callsigns-detected row.
- **Order** — Insertion order, deduplicated — the first station heard sits at the top. Re-hearing a callsign within the same session does not add a second row.
- **Persists across Listen cycles** — Toggling Listen off and back on does not clear the grid — callsigns accumulate across the entire operating session. Only the Remove and Clear controls below, or switching to FRS mode, reset the list.
- **Remove selected button** — Enabled when a row is selected. Removes that single callsign from the session list without touching Contacts. Right-clicking any row shows an equivalent context-menu option: *Remove CALLSIGN from session*. Removed callsigns can be re-detected and re-added if heard again.
- **Clear callsigns detected button** — Sits below the table to the right of Remove selected. Empties the entire grid immediately; future detections still log normally.
- **Save session button** — Stores the current grid as a timestamped net-session record under net_sessions/ for the attendance history (see Tools → Net Attendance History). Auto-save is available in Settings → Configuration → Behavior → Auto-save sessions: the grid is stored every time Listen stops, skipping sessions with no callsigns.

- **Export CSV button** — Saves the live grid to a CSV file with columns Callsign, Name, Location, GMRS, HAM.
- **Net Attendance History** — Tools → Net Attendance History opens a two-tab browser. **History** lists every saved session newest-first with its full roster, per-session CSV export, an export-all CSV (one row per station per session), and per-session delete. **Statistics** aggregates per-station attendance — total nets, attendance over the last 10, current streak, last seen — sorted busiest-first, with CSV export. A station is a callsign + name pair, so family members sharing one GMRS callsign count separately.
- **Touch-screen mode** — When the app is in touch mode the **Remove selected**, **Clear callsigns detected**, **Save session**, and **Export CSV** buttons scale to 44 px minimum height for reliable touch targets. The table itself is not affected.

5.9 Status bar

Carries a permanent **Online / Offline** indicator on the right edge (matching the OS taskbar convention). Green ● for online, amber ■ for offline. Updates every 30 seconds. The indicator is the user-visible side of the contract for opt-in network features — when offline, the FCC verification button disables and save-time verification is skipped. Hidden in FRS mode where FCC lookups don't apply. The left side of the status bar shows transient messages (Ready, STT status, waterfall activity).

5.10 Menubar

Three menus: **Settings** (Alt+S), **View** (Alt+V), and **Tools** (Alt+T).

Settings contains persistent configuration only — nothing destructive:

- **Configuration... (Alt+C or Ctrl+,)** — Opens the Configuration dialog (section 6).
- **Contacts... (Alt+N or Ctrl+B)** — Opens the Contacts dialog (section 7). Disabled in FRS mode.
- **Quick Messages... (Alt+Q)** — Opens the Quick Messages editor (section 8).

View contains a **Waterfall** submenu (Show waterfall Ctrl+Shift+W, Color map, Frequency range, Time window — all waterfall controls co-located), the callsigns-detected toggle (**Show callsigns detected**, Ctrl+Shift+A — keeps attendance .enabled in sync with the Configuration dialog checkbox so both surfaces are interchangeable), a **Panels** submenu carrying the show/hide checkboxes for Station, Pending Stations, Quick Messages, and Transmit, and a **Reset layout to default** action (Ctrl+Shift+O).

Tools contains session-level actions:

- **Generate Session Journal... (Ctrl+J)** — Sends the current conversation transcript and detected callsigns to Google Gemini, which generates a titled narrative summary. The entry is saved to journals/ as a timestamped JSON file. Requires a Gemini API key in Configuration. Shows an informative dialog if the key is missing or the transcript is empty. Generation runs in the background; the action disables until it completes.
- **View Session Journals... (Ctrl+Shift+J)** — Opens the non-modal Session Journals browser (section 5.11). Same destination as the ■ toolbar button.
- **Clear Chat (Ctrl+K)** — Erases every message from the conversation log after a Yes/No confirmation. Chat history is in-memory only and cannot be recovered once cleared.

5.11 Session Journals

The Session Journals feature sends the current conversation transcript and detected callsigns to **Google Gemini 3.5 Flash** and saves an AI-generated journal entry to disk. It requires a free Google Gemini API key configured in Settings → Configuration → Gemini API Key.

Generating a journal entry

Click **Tools** → **Generate Session Journal...** (Ctrl+J) or the **Generate log entry** button on the listen strip (visible when a Gemini API key is configured) while the conversation log has content. The app checks for a Gemini API key and a non-empty transcript; if either is missing an informative dialog explains what to do. When both are present, generation runs on a background thread — the action disables and the status bar shows *Generating journal entry via Gemini...* — so the UI stays responsive while the API call is in flight. On success, the status bar shows the saved file path for five seconds.

- **Title** — A concise session title — 10 words or fewer — generated by Gemini.
- **Summary** — A 2–4 paragraph narrative of the conversations and activities detected in the transcript.
- **Callsigns** — The list from the Callsigns Detected panel at the moment of generation.
- **Transcript** — The full conversation log text.
- **Exported at** — ISO-8601 timestamp of when the entry was generated.

Entries are saved under `journals/YYYYMMDD_HHMMSS.json` relative to the project root. The directory is created automatically on first use.

Browsing journal entries

Click **Tools** → **View Session Journals...** (Ctrl+Shift+J) or the  toolbar button to open the non-modal Journal browser. The dialog stays open while you listen, so you can review past sessions without interrupting the current one.

- **Entry list (left pane)** — All saved entries sorted newest first. Each row shows the export date and AI-generated title. Click any row to load its detail view.
- **Detail view (right pane)** — Shows the entry title, export timestamp, callsigns detected, and AI summary as formatted HTML.
- **Delete entry button** — Permanently deletes the selected entry after a Yes/No confirmation. Does not affect contacts or config. The list reloads automatically after deletion.

Journal generation requires an internet connection to reach the Gemini API. The rest of the app (RX, TX, contacts) is fully offline as always — journals are an opt-in cloud feature.

5.12 Touch-screen mode

Touch-screen mode replaces the normal docked-panel layout with a single full-panel view optimised for finger operation. It is designed for tablets, touch-enabled laptops, and Raspberry Pi units attached to a touchscreen display.

Entering and leaving touch mode

Click the  button (leftmost icon in the service toolbar icon cluster) to enter touch mode. The label flips to  while touch mode is active. Click  to return to the standard desktop layout. The preference persists to `touch_mode` in `config.json` — operators who leave the app in touch mode come back to the touch view on the next launch.

Touch view layout (top → bottom)

- **Pending-stations pill row** — The same pill buttons that appear on the Pending Stations dock are mirrored here at the top of the touch view. In touch mode the pills are larger — bolder text, extra padding, and 44 px minimum height — for reliable tap targets. Tap a pill to open the Add Station dialog; right-click or long-press to dismiss without adding.
- **Conversation log** — Full-height chat display — all RX transcripts, TX echoes, and callsign highlights stream in real time. The log stays in sync with the normal view; switching modes never drops a message. A round ▼ button (56 × 56 px) overlays the bottom-right corner whenever the log is scrolled up; tapping it jumps immediately to the latest message. The button disappears automatically when the log is already at the bottom.
- **Row 1 — primary radio controls (80 px tall)** — **Listen** (checkable, green) | **Listen Only** (checkable, amber). Both mirror the corresponding controls on the normal Listen strip — tapping either button on the touch view has exactly the same effect as clicking it on the desktop strip.
- **Row 2 — secondary controls (56 px tall)** — **Monitor** (checkable; enabled only when Listen-only is active, matching the desktop rule) | /  **Theme** (toggles dark/light mode) | **Callsigns** (floats the Callsigns Detected dock as an overlay without leaving touch mode — the Remove selected and Clear callsigns detected buttons inside that panel also scale to 44 px for touch use) | **Generate Log** (visible only when a Gemini API key is configured — generates an AI session journal entry) | **View Logs** (opens the Session Journals browser).

Dock behaviour

Entering touch mode automatically hides every dock panel (Station, Waterfall, Pending Stations, Quick Messages, Transmit). Their positions, sizes, and tab arrangements are saved internally. Exiting touch mode restores each panel to exactly the visibility it had before touch mode was activated — the desktop layout is preserved across the round trip.

The Transmit dock is hidden in touch mode because touch-screen operation is receive-focused. All RX controls are available in the touch view. If you need to transmit a typed message, exit touch mode, type in the Transmit dock, then re-enter.

State sync between views

The touch view always mirrors the normal-view state — no manual sync is needed when switching modes mid-session:

- Listen toggle state and label (*Listen / Listening...*) stay in step.
- Listen Only checked state stays in step.
- Monitor checked state and enabled state stay in step.
- Theme glyph (■ or ■■) updates whenever the theme changes, in either view.
- Generate Log visibility updates whenever the Gemini API key is saved or cleared in Configuration.
- Pending-station pills are added and removed in both views simultaneously.

6. Configuration dialog

Opened from Settings → Configuration, the gear icon, or Ctrl+,. Minimum width 420 px. Settings are organized into six tabs. All fields are mnemonic-linked (Alt+letter focuses the underlined field). The OK button is disabled until the background device-enumeration thread finishes; while loading you'll see *Loading devices...* in the device dropdowns.

Identity tab

Field	Type	Behavior
Callsign (Alt+C)	Text	Your FCC GMRS callsign. Saved uppercased and stripped.
Name (Alt+N)	Text	Operator name. Used in callsign preface and ID.
Location (Alt+L)	Text	City, state. Used by the standalone This is announcement.

Audio tab

Field	Type	Behavior
Input Device (Alt+I)	Dropdown	Microphone for capture. System Default plus every device PortAudio reports, plus <i>System Audio Output (loopback)</i> at the bottom of the list. Selecting loopback reveals a Monitor Sink sub-dropdown — pick which output device to capture (System Default follows the OS default playback device). Play audio in any browser or media player and the app transcribes it. On Linux uses <code>parec --device=.monitor</code> via PipeWire/PulseAudio; on Windows uses WASAPI loopback. The Monitor toggle is blocked when loopback is the input to prevent a feedback loop. Changing this restarts the listener.
Monitor Sink (Alt+M, loopback only)	Dropdown	Visible only when System Audio Output (loopback) is selected as the Input Device. Lists the available audio output devices (PulseAudio/PipeWire sinks on Linux; WASAPI output devices on Windows). System Default captures whatever is set as the OS default playback device. Stored as <code>system_monitor_sink</code> .
Output Device (Alt+O)	Dropdown	Where TTS audio plays. Pick a USB sound card to feed your radio directly. System Default uses the OS default sink.
Monitor audio (Alt+M)	Checkbox	Default off. When checked, the Monitor toggle on the main window activates automatically each time Listen-only mode is enabled, routing incoming radio audio unfiltered to the output device in real-time. The Monitor toggle is the live control; this checkbox sets the power-on default only.
VAD Threshold (Alt+D)	Spin 0.10–0.95 step 0.05	Silero VAD speech probability cutoff. Lower = more sensitive, higher = stricter. Default 0.50. Changing this restarts the listener.

Voice tab

Field	Type	Behavior
Voice Model (Alt+V)	Dropdown + Test	Lists every .onnx file in Voices/. Test button (Alt+T) synthesizes a short sample at the current speech rate, played on the currently selected output device.
Speech Rate (Alt+R)	Slider 0.70×–1.50×	Maps to Piper length_scale. 1.00× normal, lower = faster, higher = slower. Step 0.05. Stored as tts_length_scale.

STT tab

Field	Type	Behavior
Whisper Model (Alt+M)	Dropdown	Which locally staged Whisper model transcribes incoming audio. Only models already present under Models/STT are listed — run <code>bootstrap_models.py</code> to stage more. Larger models are more accurate but slower. Changing this restarts the listener.
Final-pass model (Alt+N)	Dropdown	Default Off (single pass). When set, each completed transmission is re-transcribed whole by this larger model on a background thread and the chat line is upgraded in place. Auto picks the best model staged under Models/STT. Stage one with <code>bootstrap_models.py --model small.en large-v3-turbo</code> .
Final-pass device (Alt+V)	Dropdown	Auto / GPU / CPU. GPU needs the optional <code>requirements-gpu.txt</code> extras; any GPU failure falls back to CPU automatically.
Final-pass max length (Alt+L)	Spin 5–600 s	Transmissions longer than this keep their streaming transcript instead of a possibly-truncated re-read. Default 60 s.
Gain mode (Alt+G)	Dropdown	Gain stage applied to each utterance before transcription. Dynamic AGC (default) levels weak and strong stations with fast-attack/slow-release smoothing; RMS normalize applies one flat gain to –20 dBFS; No gain leaves levels untouched. Stored as <code>stt_gain_mode</code> .
Noise profile (Alt+F)	Checkbox	Default off. Samples channel static while the squelch is closed and uses it as the denoiser's stationary noise estimate, instead of guessing from the speech itself. Can improve accuracy on consistently noisy channels.
Custom phrases (Alt+H)	Multi-line text	One phrase per line — names, landmarks, club jargon. Added to the Whisper vocabulary bias alongside built-in radio procedure words and contact callsigns, so the transcriber stops mishearing them. Stored as <code>saved_phrases</code> .
Max callsigns (Alt+X)	Spin 0–50	How many saved contact callsigns to include in the recognition vocabulary (newest win when over the limit). Each costs about 6 of the ~223 available prompt tokens. Set 0 to leave callsigns out entirely and give the whole prompt budget to procedure words and custom phrases.
Debug capture (Alt+B)	Checkbox	Default off. Records every utterance's raw, segmented, and processed audio plus transcripts for offline accuracy analysis with <code>python -m gmrs_tty.tools.eval_stt</code> . Captures grow quickly; leave off in normal use.
Debug directory (Alt+Y)	Text	Where debug captures are written. Default <code>debug/stt</code> , relative to the working directory. Only enabled while Debug capture is checked.

PTT tab

Field	Type	Behavior
PTT Mode (Alt+P)	Dropdown	Manual / VOX / USB FTDI / Serial. See section 12.
Serial Port (Alt+S)	Text	Only enabled in USB FTDI mode. e.g. /dev/ttyUSB0 or COM3.
Control Line (Alt+E)	Dropdown	Only enabled in USB FTDI mode. RTS or DTR.
VOX primer tone (Alt+O)	Checkbox	Default off. Plays a 1 kHz priming tone (then a short settle gap) before speech so a VOX-keyed radio is fully keyed before the first word — stops VOX attack from clipping the opening syllable.
Primer length (Alt+L)	Spin 50–2000 ms	Duration of the priming tone. Default 300 ms; radios with slow VOX attack may need longer.
Priming word (Alt+G)	Checkbox	Default off. Speaks a keyword before the actual message so the radio keys on a clear spoken word — an alternative or supplement to the tone.
Word (Alt+W)	Text	The spoken priming word. Default "transmit". Only enabled while Priming word is checked.

Behavior tab

Field	Type	Behavior
Time Format (Alt+F)	Dropdown	24-hour (14:32:15) or 12-hour (2:32:15 PM) for RX timestamps.
Filter profanity (Alt+Y)	Checkbox	Mask strong language with asterisks. Default on. Applies to both RX transcripts and TX messages before TTS speaks them.
Fuzzy callsigns (Alt+U)	Checkbox	Default off. When on, an incoming callsign that differs from a saved contact by exactly one same-class character (letter-for-letter or digit-for-digit) is rewritten in the chat to the canonical contact callsign and the pending-station pill is suppressed. Ambiguous near-misses (two contacts equally one character away) are left alone. See section 15.4.
Callsigns Detected (Alt+D)	Checkbox	Default off. Enables the Callsigns Detected dock — a roll-call grid of every callsign detected in the current Listen session. Persists at <code>attendance.enabled</code> . Also toggleable from View → Show callsigns detected (Ctrl+Shift+A). GMRS only.
Auto-save sessions (Alt+O)	Checkbox	Default off. Stores the Callsigns Detected grid as a net session record each time Listen stops (empty sessions are skipped). Records feed Tools → Net Attendance History. Persists at <code>attendance.autosave_sessions</code> .
Condition TX audio (Alt+X)	Checkbox	Default off. Band-limits synthesized speech to the 300–3000 Hz FM voice channel, compresses peaks, and normalizes the level so the voice modulates the radio consistently without clipping. Leave off if TTS plays through regular speakers.
Max TX length (Alt+L)	Spin 0–600 s	Hard cap on how long PTT may stay keyed for one transmission. A message that is already too long is cancelled before the radio is keyed, with a chat notice giving its length — shorten it and send again. Playback that still overruns is stopped and PTT released. Default 60 s; 0 (Off) disables. An Abort TX button (Esc) also appears in the Transmit row while a transmission is in progress.
Synthesis timeout (Alt+T)	Spin 0–300 s	How long to wait for Piper synthesis before abandoning the transmission. The radio is never keyed on this path. Default 30 s; 0 (Off) disables.
Gemini API Key (Alt+G)	Password text + Show/Hide	Google Gemini API key for AI-generated session journals. Leave blank to disable journal generation. The field uses password echo by default; toggle Show to reveal the key. Obtain a free key at https://aistudio.google.com/app/apikey . Stored in <code>config.json</code> as <code>gemini_api_key</code> .

On save the dialog returns a sanitized config object: callsign uppercased, name/location/serial-port stripped of leading and trailing whitespace, length scale rounded to two decimals.

7. Contacts dialog

Opened from Settings → Contacts, the person icon, or Ctrl+B. Minimum size 820×360. Six columns of editable data; **Verified** is read-only and reflects the FCC verification gate.

Columns

- **Callsign** — Saved uppercased. Required — rows with empty callsign are dropped on save.
- **Name** — Free-form operator name. Used by the callsign-match side of FCC verification (“Tim” matches “Surname, Timothy L” in the license record).
- **Location** — Free-form city / state. Auto-backfilled from the FCC city on a successful verification when the row had no location of its own; values you typed are never overwritten.
- **GMRS** — Cross-reference callsign — the operator’s GMRS call if the primary callsign is their HAM call. Auto-populated from FCC related records (service code ZA) on verification. Hand-editable. Uppercased on save.
- **HAM** — Cross-reference Amateur callsign. Auto-populated from FCC related records (service codes HA / HV). Hand-editable. Uppercased on save.
- **Verified** — Read-only. Green ✓ when the callsign is in the active FCC database AND the contact’s name matches a token in the licensee’s name. Hover for the licensee, GMRS / HAM cross-refs, and the timestamp of the last successful lookup.

Buttons

- **Add Contact** — Append a row with default values NEW_CALL / New Name. Edit in place.
- **Remove Selected** — Remove the currently selected row.
- **Sort by Suffix (Alt+S)** — View-only reorder by the last 3 digits of each callsign. Useful for spotting consecutive licenses or family clusters. *ALL* stays at the top. Clicking OK still saves the list alphabetically.
- **Verify all (Alt+V)** — Check every not-yet-verified row against the FCC database. Already-verified rows whose callsign and name match what was loaded are cached and skipped. Disabled when the app is offline; hover the disabled button for the reason.
- **Import... (Alt+I)** — Open a file-picker to import contacts from a **JSON** or **CSV** file. Incoming contacts are *merged* into the current list: rows matched by callsign + name are updated with any non-blank fields from the file while their FCC verification metadata is preserved; entirely new callsigns are appended. Works offline; FCC verification is not re-run on import.
- **Export... (Alt+X)** — Save the current contact list to a file. Choosing **JSON** exports all fields (including verification metadata) for a lossless round-trip between GMRS-TTY instances. Choosing **CSV** exports the five user-editable columns (Callsign, Name, Location, GMRS, HAM) for editing in a spreadsheet. The exported file can be imported back with Import...
- **OK / Cancel** — Standard. OK runs the same verification gate as Verify all (newly added rows, edits, and previously-failed lookups all get a fresh round trip; cached rows are skipped) and saves to `contacts.json`.

Verification semantics

- A row earns a green ✓ when (a) the callsign is in the active FCC database AND (b) the contact's name matches a token in the licensee's name. Family members on a shared GMRS callsign whose name doesn't match the licensee remain unverified.
- GMRS / HAM cross-references are *only* written when the name matches the licensee. A family-member row keeps its own GMRS / HAM fields untouched because those callsigns describe the licensee, not the family member.
- Offline behavior: Verify all disables, save-time verification is skipped, and previously-earned green checks are preserved untouched — a transient outage will never wipe verified state.

8. Quick Messages dialog

Opened from Settings → Quick Messages or the Q icon on the right side of the service toolbar. Minimum size 520×360. Single-column editable table of phrases plus four management buttons.

- **Add (Alt+A)** — Append a blank row and immediately enter edit mode on the new cell.
- **Remove Selected (Alt+R)** — Delete the currently selected row.
- **Move Up (Alt+U) / Move Down (Alt+D)** — Reorder the selected row. Order matters: the first nine rows get **Alt+1...Alt+9** shortcuts; everything past slot nine is mouse-only.
- **OK / Cancel** — OK saves the trimmed phrase list to `config.json` under `quick_messages`. Blank rows are dropped so the strip never shows an unlabeled button.

Placeholder tokens

Wrap a token in curly braces to prompt for a value at TX time. `QSY to channel {N}` opens a small input dialog asking for *N*; the substituted phrase then rides the standard TX pipeline. Multiple tokens are prompted in order. Cancelling any prompt aborts the transmission.

9. Add Station dialog

Opens when you click a yellow pill on the Pending Stations panel. Three-field form with an inline FCC lookup:

- **Callsign (Alt+C)** — Pre-filled with the detected callsign in canonical (compact, uppercase) form. A **Look Up** (Alt+U) button sits to the right of this field — see below.
- **Name (Alt+N)** — Pre-filled if the heuristic detected a name in the surrounding transcription. Left blank if no name was found; Look Up fills it from the FCC record if the field is still empty when you click.
- **Location (Alt+L)** — Pre-filled if a location was detected. Look Up fills it from the FCC city when empty.

Look Up button (Alt+U)

Queries the FCC crossref database for the entered callsign on a background thread so the UI stays live during the lookup (up to 5-second HTTP timeout). Behavior varies with what is already in the form:

- **Callsign only** — Name and Location are both empty. The lookup fetches the license record and pre-fills both fields from the FCC data. Status shows *Found: <name> — <city> (name didn't match)* because there was no contact name to verify against.
- **Callsign + Name** — The name-match gate runs (see section 16). A match earns a ✓ *Verified* status line; a mismatch shows *Found ... (name didn't match)*. Location is filled from FCC if still empty.
- **Callsign + Location** — Name field is blank; Location is already set. The lookup fills Name from FCC. The FCC location is shown in the status line so you can compare against what you typed.

The status line beneath the form uses color to communicate the outcome: green for verified, amber for found-but-not-verified, red for not found or error, gray for offline or in-progress. The button disables when the app is offline and while a lookup is running.

Click **OK** to append the new contact to `contacts.json`; the target dropdown is rebuilt and any historical RX lines mentioning that callsign retroactively gain the amber-pill highlight. The originating pill is removed automatically. **Cancel** leaves the pill in place so you can come back to it later.

10. Keyboard shortcuts & mnemonics

Global shortcuts

Action	Shortcut
Toggle Listen	Ctrl+L
Transmit message	Ctrl+Return / Ctrl+Enter
Send standalone station ID	Ctrl+I
Clear chat (with confirmation)	Ctrl+K
Open Configuration dialog	Ctrl+,
Open Contacts dialog	Ctrl+B
Send quick message preset 1–9	Alt+1 ... Alt+9
Toggle Waterfall panel	Ctrl+Shift+W
Toggle Callsigns Detected panel	Ctrl+Shift+A
Generate Session Journal	Ctrl+J
View Session Journals	Ctrl+Shift+J
Toggle Station panel	Ctrl+Shift+S
Toggle Pending Stations panel	Ctrl+Shift+P
Toggle Quick Messages panel	Ctrl+Shift+Q
Focus / re-show Transmit panel	Ctrl+Shift+T
Reset layout to default	Ctrl+Shift+O
Cycle keyboard focus across docks	F6 / Shift+F6

Main-window mnemonics

Widget	Mnemonic
GMRS radio	Alt+G
FRS radio	Alt+F
Clear chat button	Alt+C
Listen button	Alt+L
Listen only toggle (RX-only safety)	Alt+O
Monitor toggle (audio monitor, Listen-only only)	Alt+M
Transmit button	Alt+T
This is button	Alt+I
Dismiss all (pending pills)	Alt+D
Settings menu	Alt+S

Settings menu mnemonics

Item	Mnemonic
Configuration...	Alt+S, Alt+C
Contacts...	Alt+S, Alt+N
Quick Messages...	Alt+S, Alt+Q
Clear chat	Alt+S, Alt+R

Configuration dialog mnemonics

Field	Mnemonic
Callsign	Alt+C
Name	Alt+N
Location	Alt+L
Voice Model	Alt+V
Test voice	Alt+T
Speech Rate	Alt+R
Input Device	Alt+I
Output Device	Alt+O
VAD Threshold	Alt+D
Time Format	Alt+F
Filter profanity	Alt+Y
Fuzzy callsigns	Alt+U
Callsigns Detected	Alt+D
PTT Mode	Alt+P
Serial Port	Alt+S
Control Line	Alt+E

Tab order in the main window

Listen → Target dropdown → Message box → Transmit → This is. The order is explicitly set so keyboard-only operators get a predictable traversal.

11. GMRS vs FRS service mode

The top-of-window toggle controls which Part 95 subpart the app treats as canonical. GMRS (Subpart A) is licensed and callsign-bearing; FRS (Subpart B) is unlicensed and has no callsign requirement. Switching modes is instantaneous; saved contacts and the saved callsign are preserved across mode changes so switching back to GMRS restores prior state.

Feature	GMRS (default)	FRS
Header line	Station: CALL Operator Location	FRS Mode Operator Location
Outgoing TX preface	[CALL] [name] calling [target]	No preface — message is spoken as-is
15-minute ID rule	Enforced	Skipped
This is button	Enabled	Disabled (tooltip explains)
Target dropdown	Visible	Hidden
Contacts menu / icon	Enabled	Disabled (tooltip explains)
Pending-station pills	Detected and displayed	Detection suppressed
Callsign chat highlighting	Amber pills, FCC ✓	Suppressed
Online / Offline indicator	Visible (gates verification)	Hidden (no online features apply)
FCC callsign verification	On (opt-in via connectivity)	Not applicable

Why FRS turns features off: FRS operators are anonymous by regulation. Surfacing the 15-minute ID button or callsign verification on an FRS channel would be misleading at best and FCC-incorrect at worst. The toggle is therefore a hard behavioral switch, not a cosmetic relabel.

12. PTT (Push-to-Talk) modes

Selectable from Configuration → PTT Mode. The choice determines how the radio is keyed around each transmission.

Manual

Default. You press PTT on the radio yourself, the app plays audio through the configured output device, and you release PTT when the audio finishes. The app makes no attempt to key the radio.

Use when: you have a desk mic with a foot switch, you're operating a handheld and physically pressing the PTT button, or you're testing the app without any radio connected.

VOX (Voice-Operated Transmit)

Your radio is set to auto-key whenever it hears audio on its mic input. The app appends a short tail of silence after each transmission so the last syllable survives the VOX hang dropout.

Use when: your radio has a VOX setting and you've already tuned VOX sensitivity / hang time to your setup. Simplest wiring — just a cable from the output device to the radio's mic input.

USB FTDI / Serial

The app keys PTT through a USB-serial adapter's RTS or DTR line. The line drives an external transistor or opto-isolator wired to the radio's PTT pin. Short lead-in and tail silence are inserted around each transmission so the radio's keying ramp doesn't clip the audio.

- **Serial Port** — Device path — e.g. `/dev/ttyUSB0` on Linux or COM3 on Windows. Field is enabled only when this PTT mode is selected.
- **Control Line** — RTS or DTR. Choose whichever your interface uses. Most FTDI-based DIY PTT cables use RTS; Signalink-style boxes vary.

Cable safety: never wire a serial line directly to a radio's PTT pin. Use an opto-isolator or NPN transistor between the adapter and the radio so a stuck line or a host crash can't dump TTL voltage onto the radio.

13. Receive (Rx) pipeline

Pressing Listen starts a background thread that captures audio, gates it through Silero VAD, conditions each utterance through a DSP chain, and hands the result to faster-whisper for transcription. The pipeline is designed to skip kerchunks, static, and Whisper hallucinations so the chat log shows only genuine speech.

Audio capture

- On Linux with PipeWire the app prefers `parec` over PortAudio because PortAudio's PipeWire-via-ALSA bridge can silently deliver flat-zero audio on PipeWire 1.4. If `parec` is missing the app falls back to PortAudio.
- Anywhere else (or if you've selected a specific input device index in Configuration), PortAudio is used directly.

Squelch-open pre-trigger

A peak-amplitude edge detector watches every captured chunk so the leading syllables of a transmission survive Silero VAD's onset latency. The moment a remote operator's carrier opens — audio jumps above the noise floor for two consecutive ~32 ms chunks — the app starts buffering chunks in a rolling deque capped at roughly 2 s. When VAD then fires on real speech, the entire pre-voice buffer is prepended to the utterance before bandpass, denoise, and Whisper. When the carrier instead closes again (peak below threshold for ~500 ms) without VAD ever firing — a kerchunk, accidental key, or stray noise burst — the buffer is discarded and nothing reaches the chat.

The detector's thresholds and hysteresis are tuned defaults, not user-configurable. Operators should not need to think about this stage; it exists so the first word of a transmission isn't clipped.

Voice activity detection (VAD)

- **Silero VAD** — Neural VAD. Only audio it scores as speech is forwarded for transcription — kerchunks and static are dropped.
- **Threshold** — Tunable from 0.10 (very sensitive) to 0.95 (very strict) in Configuration. Default 0.50.
- **Auto-rebaseline** — After roughly 30 s of continuous silence the VAD is rebaselined so detection stays responsive on long-quiet channels.
- **TX auto-pause** — While the app is transmitting, the listener is paused so your own TTS isn't transcribed back. Listening resumes immediately on unkey, with VAD state reset so no in-progress speech bleeds across the boundary.

DSP conditioning

- **300–3000 Hz bandpass** — Matches the narrowband-FM voice band. Strips hum and out-of-band hiss. Applied both to the live monitor stream and per utterance before denoising.
- **Noise reduction** — Spectral gating applied per utterance after bandpass and before transcription.
- **RMS normalization** — After denoising, each utterance is normalized to -20 dBFS so weak or distant stations reach Whisper at a consistent input level.

- **Hallucination filter** — Drops utterances shorter than ~400 ms and common Whisper hallucinations on silence (“Thank you”, “Subtitles by...”, etc.).

Transcription

- **Engine** — faster-whisper (CTranslate2 backend). int8 CPU inference by default.
- **Default model** — `small.en`. Trade quality / speed via `bootstrap_models.py --model base.en|medium.en`.
- **Profanity masking** — When **Filter profanity** is on, strong language in the transcript is masked with asterisks before it lands in the chat log.

Streaming transcription for long utterances

Short transmissions are transcribed in one shot — the operator sees a single RX line drop into the chat at the unkey. Anything that runs longer than about 5 s is sliced and transcribed incrementally so the chat doesn't sit blank during a long monologue:

- Once the captured speech crosses ~5 s, the capture loop scans the next 500 ms for the quietest point and cuts there — slice boundaries land between words, not mid-syllable.
- Each slice is handed to a background Whisper thread under a shared utterance id so the capture loop never blocks waiting for inference.
- Partial arrivals in order and the chat renders them as a single growing line — `[RX 14:32:01]: hello bob how are you today...` — rather than as separate messages.
- Every partial still rides the full bandpass + denoise + hallucination-filter chain. Callsign-discovery scanning runs once over the full accumulated text — at the final segment, or immediately when the utterance is abandoned (PTT pressed mid-reception, Listen toggled off, or a new utterance starting before the previous one completes) — so the callsigns-detected panel always reflects what the chat shows.

Slice length and the cut-search window are tuned constants, not configuration knobs. The operator-facing contract is simple: short utterances arrive whole at unkey, long ones stream in as they're spoken.

Audio monitor

The **Monitor** toggle (Alt+M, on the listen strip) routes incoming radio audio to the configured output device in real-time, letting a deaf or HoH operator hear the channel through speakers simultaneously with the transcription pipeline.

- **When available** — Monitor is only enabled when **Listen-only** mode is on. Enabling it on a full-duplex session would risk audio feedback from the output device back into the radio mic, so the button is greyed out whenever Listen only is off. Turning Listen only off automatically stops the monitor stream.
- **Audio processing** — Before reaching the output device, audio passes through a 300–3000 Hz bandpass filter (matching the narrowband-FM voice band) and is upsampled from 16 kHz to 48 kHz via a polyphase sinc resampler so the device receives its native rate rather than relying on driver-level interpolation. TX mute and unmute transitions apply a 5 ms linear fade to eliminate clicks at keying boundaries.

- **Ring buffer** — A thread-safe bounded deque (~1 s) absorbs burst-capture spikes. When the buffer would exceed capacity, the oldest samples are dropped so playback never lags behind live audio.
- **Persistence** — The toggle state is saved to `monitor_enabled` in `config.json`. When the app starts with this flag set, the Monitor toggle activates automatically the next time Listen-only mode is turned on.
- **Configuration default** — Settings → Configuration → **Monitor audio** sets the power-on default; the in-strip toggle controls the live state.

13.9 STT calibration wizard

Tools → **Calibrate STT** (enabled while Listen is active) finds the most accurate STT settings for your radio and channel conditions in four steps:

- **1. Reference passage** — The wizard shows a fixed passage (the opening of the Declaration of Independence). Arrange for another station to read it over the air, or read it yourself on a loopback input.
- **2. Recording** — Click **Start recording**, have the passage read, then **Stop and analyze**. At least a few seconds of audio are required; the capture is bounded at three minutes.
- **3. Analyzing** — Every staged Whisper model × gain mode × noise profile combination transcribes the recording and is scored by word-error-rate against the passage. The progress bar reads "combination N of M"; on CPU a multi-model sweep takes minutes.
- **4. Results** — Combinations are ranked best-first with the recommended row preselected. **Apply selected** writes the model, gain mode, and noise profile to the configuration; a model change takes effect the next time Listen starts. Escape cancels cleanly at any step.

14. Transmit (Tx) pipeline

Each transmission (typed message, quick preset, or This is) passes through the same pipeline.

- **1. Placeholder substitution** — Curly-brace tokens in quick presets are filled by prompting the user.
- **2. Shorthand expansion** — TTY and ham-radio shorthand is rewritten to full words before TTS — GA → Go ahead, SKSK → end of conversation, 73 → best regards, QTH → location, etc. Matching is case-insensitive and word-bounded, longest-key-first, so QSO → radio contact while a lone Q inside another word passes through.
- **3. Profanity masking** — When the filter is on (default), strong language is asterisk-masked before TTS sees it. Mild PG-13 words (damn, hell, crap, bare ass) pass through unchanged. Word-bounded so substrings like classroom or Scunthorpe are never false-positives.
- **4. FCC framing (GMRS only)** — Prepends [CALL] [name] calling [target] when targeting a specific station; appends a station ID if more than 15 minutes have passed since last identification. *All*-targeted transmissions skip the preface but still pick up the ID append when the rule triggers. FRS skips both.
- **5. Spoken-callsign formatting** — Digits inside callsigns are split with spaces so TTS reads them as individual digits (WSLZ 2 3 3 rather than two hundred thirty-three).
- **6. PTT key-down** — Manual mode: no-op. VOX: append silence tail. USB FTDI: assert the configured control line and pad with lead-in / tail silence.
- **7. STT auto-pause** — The listener is paused so your own audio isn't transcribed back.
- **8. Piper synthesis** — TTS audio is rendered offline through the selected Piper voice at the configured length scale and played to the configured output device.
- **9. Unkey and resume** — PTT releases (in serial mode) after a tail of silence; STT resumes with VAD state reset.

15. Callsign detection & FCC verification

Recognized formats

- GMRS modern: WSLZ233.
- GMRS legacy: KAE1234.
- US amateur: K1ABC, KD9XYZ, W1AW.
- Compact: WSLZ233.
- Spaced: W S L Z 2 3 3.
- Separated: W.S.L.Z.233, WSLZ-233, WSLZ, 233.
- NATO phonetic: Whiskey Sierra Lima Zulu Two Three Three (also accepts *X-ray* and *X ray*).

Pending pills

Unknown stations earn a yellow pill on the Pending Stations panel. The “unknown” check considers all three callsign fields on every contact (`callsign`, `gmrs_callsign`, `ham_callsign`), so a HAM call detected over the air won't pill if the operator's GMRS call is already saved (and vice versa).

FCC verification (online, opt-in)

When the connectivity probe says the app is online, contacts are cross-referenced against the public FCC license database via the `ke8rxnwx` crossref API. A row earns a green ✓ in the Verified column when both conditions hold:

- The callsign is in the active FCC database, AND
- The contact's name matches a token in the licensee's name (“Tim” matches “Surname, Timothy L”).

Verified lookups also persist `gmrs_callsign` and `ham_callsign` fields on the contact, pulled from the FCC related cross-reference list (service codes ZA for GMRS and HA / HV for Amateur). For an operator licensed for both services, a HAM call entered as the primary will resolve its associated GMRS call and vice versa.

Cross-references are **only** written when the contact's name matches the licensee. A family-member row on a shared GMRS call (whose name doesn't match) keeps its own GMRS / HAM fields untouched because those callsigns describe the licensee, not the family member.

A verified lookup also backfills the contact's `location` field with the FCC city (title-cased) when the row had no location of its own. Any value the operator already typed is left alone.

Caching: rows whose `verified` flag is `True` with unchanged callsign and name are treated as cached. Both `Verify all` and `OK` skip them, so opening and closing Contacts doesn't churn the `verified_at` timestamp. Newly added rows, in-dialog edits, and previously-failed lookups all earn a fresh round trip.

15.4 Fuzzy callsign matching (opt-in)

Whisper occasionally mishears a single character of a callsign — WSLZ233 arriving as WSLZ235, or WSIZ233 with an *L/I* swap. The **Fuzzy callsigns** checkbox in Configuration (Alt+U; off by default) rewrites these near-misses to the canonical contact callsign so the chat doesn't fragment a single operator into half a dozen near-duplicates.

Match rules:

- Same length as a saved contact's callsign.
- Exactly one character differs.
- The differing characters are the same class — letter-for-letter or digit-for-digit. A digit misheard as a letter (or vice-versa) is rejected because it usually indicates a genuinely different callsign rather than a transcription slip.
- Unambiguous — if two saved contacts are equally one character away, the detected token is left alone so the operator can resolve it manually.

Effects when a rewrite fires:

- The chat line shows the canonical callsign and picks up the amber contact pill, FCC ✓ check, and hover tooltip.
- The pending-station pill is suppressed — no spurious “+ Add” prompt for a near-miss of a known operator.
- Toggling the checkbox mid-session retroactively rewrites near-misses in chat lines already on screen. Toggling it back off does not undo prior rewrites; the canonical form is already visible and accurate.

Fuzzy matching only ever rewrites *toward* a callsign you've already saved. It cannot invent a new callsign or merge two contacts. The opt-in default keeps the app's received-text contract literal by default — turn it on only if your STT environment regularly chews a single character.

16. FCC Part 95 compliance features

GMRS-TTY makes Part 95 GMRS compliance the default behavior:

- Outbound messages always carry your callsign and name when targeting a specific station.
- The 15-minute ID rule is enforced automatically — your callsign and name are appended when more than 15 minutes have passed since the last identification.
- Identification is appended even on short messages if the rule triggers.
- The standalone **This is** button (Alt+I or Ctrl+I) sends a complete station ID and resets the 15-minute timer.
- The PG-13 profanity filter (default on) masks strong language in both RX and TX so transmissions stay within Part 95 obscenity expectations. Toggle in Configuration if you operate on a private repeater with different norms.
- FRS mode (Subpart B) intentionally skips Part 95 Subpart A station-ID rules. FRS is unlicensed and has no callsign requirement, so callsign framing, the 15-minute timer, and the standalone-ID button are all disabled.

You are still responsible for legal operation. This software does not replace a valid FCC GMRS license, and the operator remains accountable for content, channel choice, and any local repeater rules. The app's compliance features are a convenience, not a guarantee.

17. Accessibility (WCAG 2.1 AA)

The application exists for users with disabilities, so accessibility is a hard design constraint. The UI targets WCAG 2.1 Level AA — the practical baseline DOJ and Section 508 reference for software ADA compliance.

- **Color contrast** — Text colors meet $\geq 4.5:1$ against the chat background (RX green #15803D, TX blue #1D4ED8, errors #B91C1C, warnings #92400E). UI borders such as the pending-station pills meet $\geq 3:1$.
- **State never by color alone** — Every RX line is prefixed [RX HH:MM:SS] :, TX lines [TX to ...] : or [TX ID] :, errors say *Error:* or *Failed:* in the text. Color is supplemental, never the only cue.
- **Full keyboard operation** — Explicit tab order (Listen → Target → Message → Transmit → This is). Mnemonics on every actionable label. Global shortcuts listed in section 10.
- **Screen reader support** — Every non-decorative widget has an `accessibleName` and (where useful) `accessibleDescription` for the Qt accessibility bridge — NVDA, JAWS, Orca, VoiceOver.
- **Font scaling** — No hard-coded font-size in stylesheets. Bold uses Qt's relative QFont API so the OS font-scale setting carries through.
- **Resizable layout** — Main window minimum 720×520 so high-DPI / large-font setups don't clip. All dialogs are resizable.
- **Focus visible** — Relies on the Fusion style's default focus indicator — Qt does not strip outlines and neither does the app.

18. Off-grid operation

Once the dependencies and models are in place, the radio workflow does not need a network. RX transcription, TX synthesis, PTT keying, and contact management all run locally.

- **Whisper STT** — Loaded from `Models/STT/<model>/`. Pre-staged via `bootstrap_models.py` or bundled by the installer; never re-fetched at runtime.
- **Silero VAD** — Ships as a local ONNX file inside the wheel.
- **Piper TTS** — Voices live under `Voices/`; you supply them once.
- **Online features** — Strictly opt-in via the connectivity probe. When offline: the Verify all button disables, save-time verification is skipped, the online indicator turns amber, and previously-earned verified checks are preserved untouched. The radio workflow keeps running.

Pre-built installers are already fully air-gapped for the radio workflow. The Whisper STT and speaker ID models are bundled inside the `.deb / .msi` — no bootstrap step, no network access needed after install. The only online feature ever used is FCC callsign verification, which is strictly opt-in.

Recommended deployment for a *source* install on an air-gapped target: clone the repo, install dependencies, and run `bootstrap_models.py` on an internet-connected machine. Then copy the entire source tree, `Models/`, and `Voices/` to the offline target. Reuse the same virtual environment if it's portable across the architecture, otherwise rebuild the venv from the same `requirements.txt` on the target.

19. Troubleshooting

Pre-built installer issues

- **gmrs-tty command not found (Linux .deb)** — The postinst may not have completed. Try `sudo dpkg --configure gmrs-tty` to re-run the postinst, then check that `/usr/bin/gmrs-tty` exists.
- **Python venv missing at /opt/gmrs-tty/.venv** — Reinstall the package: `sudo apt reinstall gmrs-tty`. The postinst will recreate and repopulate the venv from the bundled wheels.
- **App launches but the voice dropdown is empty** — No Piper voice files are present in `/opt/gmrs-tty/Voices/`. See section 3.1.3 for how to add voices.
- **Windows MSI: shortcut opens a console window** — The shortcut targets `pythonw.exe` which suppresses the console. If you see one, the shortcut may have been modified — use the Desktop shortcut created at install time, or recreate it to point at `pythonw.exe main.py` in the install directory.

Listen does nothing or fails immediately

- **“STT model not found” in chat** — Run `python bootstrap_models.py --model <name>` with the model named in the message.
- **Listen toggles but the level meter stays at zero** — The app isn't getting audio. Check the Input Device dropdown in Configuration, your microphone privacy permissions, and the physical cable. On PipeWire installs, confirm `pulseaudio-utils` is installed so `parec` is on PATH.
- **Lots of false starts on a noisy channel** — Raise VAD Threshold (try 0.65–0.75). Conversely, on a very quiet channel where speech is being missed, lower it (try 0.30).

Transmit is silent or clipped

- **Nothing audible at all** — Check Output Device. If you're feeding the radio directly, make sure you didn't accidentally leave Output Device on System Default — you'd hear it through the laptop speakers instead.
- **First syllable clipped on VOX** — VOX hang/sensitivity needs tuning on the radio; lengthen the radio's VOX attack if available.
- **Last syllable cut off on serial PTT** — Increase the tail-silence padding (recompile with longer pad) or switch to VOX where the tail is already extended.

FCC verification problems

- **Verify all is disabled** — Status bar shows Offline. Reconnect; the indicator refreshes every 30 s.
- **Row stays unverified even though the callsign is real** — Check that the Name field matches a token in the licensee's name. Hover the empty Verified cell — if the tooltip lists a licensee, the row was rejected for name-mismatch on purpose (family-shared GMRS call).

Quick messages

- **Strip is invisible** — You have no presets saved. Open the Quick Messages dialog and add at least one phrase.
- **Alt+1...Alt+9 fire the wrong preset** — Use Move Up / Move Down in the Quick Messages dialog to reorder — the shortcuts follow the saved order.

20. File reference

config.json

Lives in the working directory next to `main.py`. Plain UTF-8 JSON.

```
{
  "callsign": "WSLZ233",
  "name": "Benjamin",
  "location": "Lansing, MI",
  "voice": "Voices/en_US-ryan-high.onnx",
  "tts_length_scale": 1.0,
  "input_device": -1,
  "output_device": -1,
  "whisper_model": "small.en",
  "system_monitor_sink": "",
  "vad_threshold": 0.5,
  "time_format": "24h",
  "filter_profanity": true,
  "fuzzy_callsign": false,
  "listen_only": false,
  "monitor_enabled": false,
  "monitor_passthrough": false,
  "attendance": { "enabled": false },
  "gemini_api_key": "",
  "ptt_mode": "manual",
  "ptt_serial_port": "",
  "ptt_serial_line": "RTS",
  "radio_service": "GMRS",
  "touch_mode": false,
  "quick_messages": [
    "Radio check",
    "Loud and clear",
    "Standing by",
    "Acknowledged",
    "Say again",
    "QSY to channel {N}",
    "Clear",
    "Monitoring",
    "Net check-in",
    "Emergency traffic"
  ]
}
```

- **input_device / output_device** — -1 means system default. Any other integer is the PortAudio device index. The string "system_monitor" selects System Audio Output (loopback) — pair with `system_monitor_sink` to target a specific output device.
- **system_monitor_sink** — Which audio output to capture when `input_device` is "system_monitor". Empty string (default) uses the OS default playback device. On Linux, the PulseAudio/PipeWire sink name (e.g. `alsa_output.pci-0000.analog-stereo`). On Windows, the output device index as a string (e.g. "3"). Set via the Monitor Sink dropdown in Configuration.

- **tts_length_scale** — 0.70–1.50. Higher is slower.
- **vad_threshold** — 0.10–0.95. Higher is stricter.
- **listen_only** — Boolean. When true the app blocks every TX path at launch (matches the Alt+O toggle on the Listen strip). Microphone capture and transcription still run.
- **monitor_enabled** — Boolean. Default false. When true the Monitor toggle (Alt+M) activates automatically each time Listen-only mode is turned on. Has no effect when Listen only is off (the button is disabled in that state).
- **monitor_passthrough** — Boolean. Default false. When true the Passthrough toggle is pre-enabled when Monitor turns on, sending audio to the speaker without the bandpass filter. Does not affect VAD or Whisper transcription.
- **fuzzy_callsign** — Boolean. Off by default. When true a detected callsign that differs from a saved contact by exactly one same-class character is rewritten to the canonical form in chat. See section 15.4.
- **attendance** — Nested object. {"enabled": <bool>}. Default false. Controls the Callsigns Detected dock. Nested so future per-session options can land here without churning the top-level schema.
- **gemini_api_key** — String. Your Google Gemini API key. Empty string (default) disables journal generation. Set via Settings → Configuration → Gemini API Key.
- **ptt_mode** — manual, vox, or usb_ftdi.
- **ptt_serial_line** — RTS or DTR. Only used when ptt_mode is usb_ftdi.
- **time_format** — 24h or 12h.
- **touch_mode** — Boolean. Default false. When true the app launches directly into touch-screen mode (section 5.12). The  toggle on the service toolbar writes this key automatically.
- **radio_service** — GMRS or FRS. Default GMRS if missing or unknown.
- **quick_messages** — Ordered list of phrase strings; first nine get Alt+1...Alt+9.

contacts.json

Lives alongside `config.json`. Array of contact objects. The synthetic *All* row is always pinned at the top of the in-app dropdown and is preserved across saves.

```
[
  { "callsign": "All", "name": "Everyone" },
  {
    "callsign": "WSLZ233",
    "name": "Tim",
    "location": "Lansing, MI",
    "gmrs_callsign": "WSLZ233",
    "ham_callsign": "KD9XYZ",
    "verified": true,
    "verified_at": "2026-05-16T14:32:11Z",
    "license_name": "SURNAME, TIMOTHY L"
  }
]
```

- **callsign** — Required. Uppercased on save.
- **name** — Free-form.
- **location** — Free-form. Auto-backfilled from FCC city on successful verification if originally empty.
- **gmrs_callsign / ham_callsign** — Cross-references. Auto-populated by FCC verification when the name matches; hand-editable otherwise.
- **verified** — Boolean. True only when callsign is active in the FCC database AND name matches the licensee.
- **verified_at** — ISO-8601 UTC timestamp of the last successful lookup.
- **license_name** — Raw licensee name returned by the FCC — stored so the Verified-cell tooltip can show *FCC license is held by ...*

21. Glossary

Term	Definition
FCC Part 95	The section of US federal regulations covering the Personal Radio Services. Subpart A is GMRS; Subpart B is FRS.
FRS	Family Radio Service. Unlicensed, no callsign, lower power, fixed antennas only on most channels.
GMRS	General Mobile Radio Service. Licensed (one license covers an immediate family), callsign required, up to 50 W on repeater inputs.
kerchunk	A brief unmodulated key-up with no voice. Common when checking repeater access. Silero VAD is designed to ignore these.
NATO phonetic	The ICAO/NATO spelling alphabet (Alfa, Bravo, Charlie...). Used over voice radio to make letters unambiguous.
Piper	An offline neural TTS engine using ONNX voices.
PTT	Push-to-Talk. The control line that keys the radio into transmit mode.
QSL / QSO / QTH / QRZ	Ham-radio Q-signals. QSL = acknowledge. QSO = radio contact. QTH = location. QRZ = who is calling me?
Silero VAD	An open-source neural voice-activity detector. Decides which audio chunks contain speech.
STT	Speech-to-text. The Whisper transcription stage.
TTS	Text-to-speech. The Piper synthesis stage.
TTY	Teletypewriter — the device, and by extension the abbreviation style (GA, SK, 73, ILY), that deaf operators have used over phone and radio for decades.
VAD	Voice activity detection. See Silero VAD.
VOX	Voice-operated transmit. A radio's auto-key on detected audio.
Whisper	OpenAI's speech-recognition model. faster-whisper is the CTranslate2-accelerated CPU runtime used by this app.