

REAPROPHET

Rev 4 Editor User Manual

Detailed Guide and Workflow Reference

Sequential Prophet-5/10 Rev4 • REAPER

Version 1.0



This manual focuses on installation, workflow, patch management, Layer A/B editing, saving, backups, and editor-specific controls. It is not a synthesis tutorial or a replacement for the Prophet hardware manual.

Contents

1	Installation and First Run
2	The Companion Patch Library Helper
3	Header Navigation and the Hardware & Sync Panel
4	Globals & Program Options
5	The Program Page: Editor-Specific Features
6	Patch Library: Hardware Browsing and Auditioning
7	Bank Sources, SysEx, Backups, and Drag-and-Drop Management
8	Stack/Split and Program A/B Workflows
9	Project Persistence, Restore Prophet, and Recovery
10	Help, UI Scaling, and Default Window Size
11	Practical Workflows, Troubleshooting, and Limitations
A	Appendix: Files and Storage Locations

FIVE CONCEPTS TO KEEP STRAIGHT

A stored program lives in Prophet memory. The edit buffer is the currently sounding, temporary hardware image. Upon project reload, the REAPER project is ReaProphet's saved local Normal or A/B state and may differ from the hardware until Restore Prophet is used. The local Patch Library and SysEx bank library are disk-based mirrors and collections; they do not overwrite Prophet memory unless you explicitly use a save, bank-send, or drag-and-drop write action. A "pair" is the term for a Stack or Split combination.

1. Installation and First Run

ReaProphet consists of two cooperating files: the JSFX editor and a companion Lua helper.

Install both and run together. The editor handles the interface, MIDI, NRPN, SysEx, hardware verification, and all Prophet writes. The helper handles disk access, file dialogs, local libraries, backups, logs, and saved UI defaults.

1.1 Files to install

File	Install location	Purpose
ReaProphet Rev 4 Editor.jsfx	REAPER resource path / Effects	The editor and all MIDI/hardware behavior.
ReaProphet Helper.lua	REAPER resource path / Scripts	Persistent patch library, SysEx banks, backups, logs, and UI default settings. Use the helper version supplied with the same ReaProphet release package.

1.2 Locate the REAPER resource path

1. In REAPER, open Options and choose Show REAPER resource path in Explorer/Finder.
2. Open the Effects folder and copy the JSFX file into it. A subfolder such as Effects/ReaProphet is fine if you prefer to keep custom effects organized.
3. Open the Scripts folder and copy the Lua helper into it.
4. In REAPER, open Actions > Show action list, choose ReaScript: Load, and select the helper Lua file.
5. Run the newly added helper action once. ReaProphet should report Patch library helper: online.

ONLY ONE HELPER

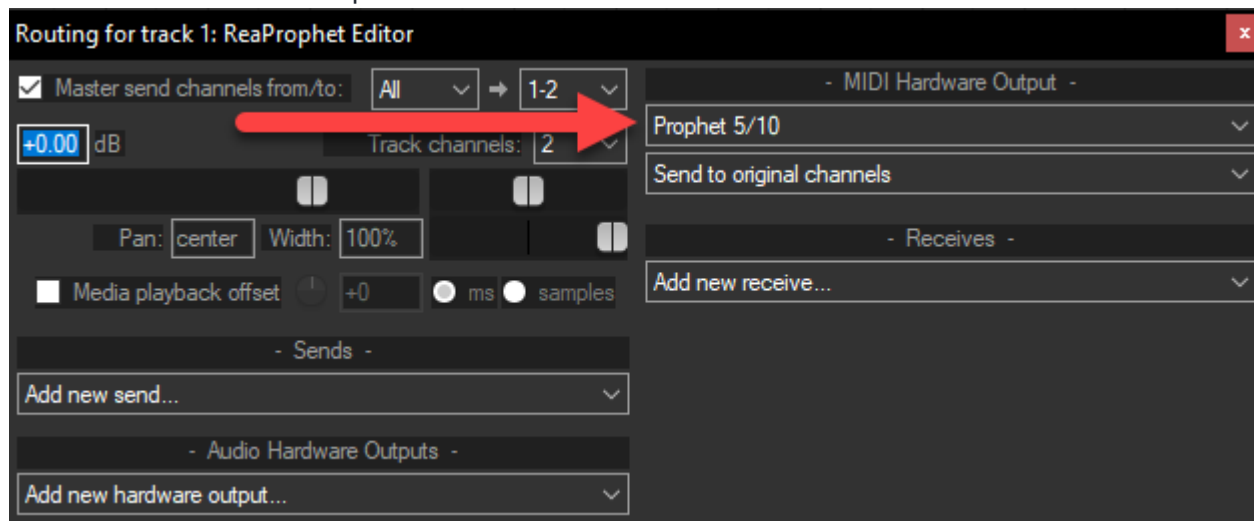
Do not run two differently named copies of the helper at the same time. They share the same bridge and can compete for the same disk-library commands.

1.3 Setup the REAPER tracks

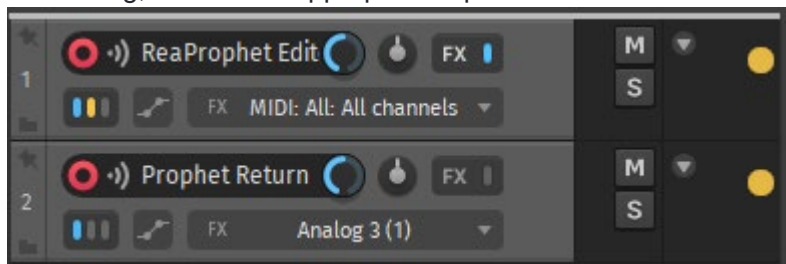
Ensure the Prophet MIDI Input and Output channels are enabled in Reaper, and the Prophet is turned on prior to Reaper starting.

1. Create a MIDI/editor track and insert ReaProphet as a JSFX (or drag ReaProphet directly from Reaper's FX Browser to the TCP). Set the MIDI input as desired.

- Open the track routing window and set a MIDI Hardware Output to the Prophet. Use the same MIDI channel selected in ReaProphet and the hardware.



- Create a separate audio-return track. Select the interface input receiving the Prophet, enable monitoring, and set an appropriate input level.



- Keep the editor track and audio-return track together. **Tip:** Once routing is confirmed, save them as a REAPER Track Template for future sessions.

1.4 Prophet hardware MIDI settings

Prophet setting	Required value	Why
Parameter Transmit/Receive	NRPN	ReaProphet's real-time parameter editing and receive mapping use NRPN.
MIDI Control	On	Allows remote control and program communication.
MIDI channel	Match ReaProphet	The editor and hardware must communicate on the same channel.
PRESET button	Lit	PRESET off places the Prophet in Live Panel mode, where the front panel can override the stored program behavior.

1.5 First synchronization

- Power on the Prophet and MIDI interface before starting REAPER. Load ReaProphet as outlined above, then start the helper and confirm that the Hardware & Sync panel shows it online.
- Press Receive Current Program to read the currently sounding Prophet edit buffer into ReaProphet.

3. Press Request Globals to synchronize supported global settings and read-only status values.
4. Press Refresh Bank for the selected 40-program bank, or Refresh All to read all 400 hardware programs. **When reading/writing banks, allow ReaProphet to finish before making changes on the hardware.**
5. Press Save Library after a refresh to persist the hardware mirror, favorites, and local category changes to disk.

RECOMMENDED STARTUP ROUTINE

Power on the Prophet and MIDI interface before starting REAPER. Run the helper automatically when REAPER starts. With SWS/S&M installed, place it in the global startup action; otherwise, add it to a startup custom action or toolbar.

2. The Companion Patch Library Helper

The helper is not a second editor and it never sends MIDI by itself. It is ReaProphet's disk and file-dialog service. Hardware communication remains inside the JSFX.

2.1 What the helper does

Service	What it provides
Hardware patch mirror	Loads and saves the 400-program patch_cache.json mirror used by the left Patch Library pane.
Favorites and categories	Persists ReaProphet favorites and local category edits.
SysEx bank library	Stores imported banks, backups, and Single Programs collections in a durable local library.
Import/export dialogs	Opens platform file pickers for program and bank SysEx import/export.
Bank backups	Creates timestamped backups before selected bank writes and stores backup banks in Bank Sources.
Save Pair recovery	The Helper writes a two-program recovery .syx and text manifest before Save Pair begins writing hardware memory.
MIDI receive logs	Writes the detailed receive/NRPN log and, when available, copies it to the clipboard.
UI scale defaults	Saves the default scale and updates the marked startup-size request. When first making changes to the scale, it's expected that you will need to manually resize the window. In some instances, saving a new default preset in ReaProphet Editor will help retain the window size.

2.2 What still works when the helper is offline

Normal editor controls, MIDI, NRPN, Receive Current Program, Globals receive, hardware program loading, and most direct hardware operations remain part of the JSFX. Disk-dependent features do not.

Works without helper	Requires helper online
Program editing, NRPN control, Receive Current Program, Globals receive, hardware program selection	Load Library, Save Library, imports, exports, favorites/category persistence
Save Program to the current location	Bank backups, Send Selected Bank safety backup, Save Pair recovery backup
Program A/B editing while the active pair is already available	Set current UI scale as Default and managed startup-size updates

2.3 Local data folder

The helper creates a ReaProphetEditor folder in the current user's Documents folder. It contains the hardware cache, durable SysEx library, backups, logs, settings, and UI preferences. See Appendix A for the full layout.

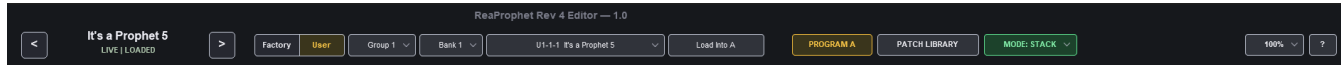
DO NOT MOVE INDIVIDUAL LIBRARY JSON FILES BY HAND

Use ReaProphet's Rename and Delete commands. The helper keeps the bank index and bank records consistent and creates .bak recovery files during transactional writes.

2.4 Helper status

- **Online:** disk-library commands are available. The helper automatically loads the saved hardware mirror when it connects.
- **Offline:** run the current Lua action, then retry the command. The helper status is serviced from the visible UI, so an active audio interface is not required merely to keep the helper online.
- **Incompatible version:** stop the old helper and replace the exact Lua file used by the REAPER action.

3. Header Navigation and the Hardware & Sync Panel

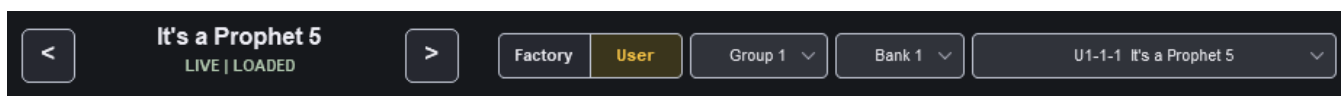


The top header is both navigation and a direct hardware program browser. In Normal mode, choosing a location immediately loads that stored Prophet program and synchronizes the editor. In an active pair, the same selectors become source selectors until you explicitly load the chosen sound into Program A or Program B.

3.1 Navigation tabs

Control	Normal behavior	Pair behavior
Program	Opens the Program (synth) editor.	The tab becomes Program A or Program B. Clicking it while already on the Program page toggles the active editable layer.
Patch Library	Opens the two-pane hardware and bank-source browser.	Rows become selection-only; use Load Into A or Load Into B to replace a layer.
Mode	Normal, Stack, or Split. Normal has no highlight color.	Also offers Swap Layer Sounds. Stack and Split retain the current A/B documents.
Scale	Choose 75%, 100%, 110%, 125%, or 150%.	Same behavior. The selector is immediately left of Help.
?	Opens Help & Quick Start. When active, pressing a second time returns to the Program page.	Same behavior.

3.2 Patch location controls



The header provides Factory/User, Group, Bank, and Program selectors. Factory appears on the left and User on the right. User remains the startup default, highlighted in yellow at U1-1-1. ReaProphet also displays compact locations such as U2-2-6 or F1-1-1.

LOCATION TERMINOLOGY

In the current header, Group selects the outer 1-5 hardware bank, Bank selects the 1-5 eight-program group inside it, and Program selects 1-8. The compact U/F location remains the clearest reference when confirming a destination.

- Factory/User changes the memory region and immediately loads the selected stored program in Normal mode. The visual order is Factory then User, but User remains the default region.
- Group and Bank open dropdown menus. Program shows the cached patch name when the hardware mirror is available.
- Previous and Next step through stored locations and synchronize the editor.
- In Program A/B editing, changing these selectors chooses a source. Press Load Into A or Load Into B to apply that source to the active layer.

- Right-click Load Into A/B to exit the pair and load the selected program as a complete Normal program.

3.3 Program identity and state

Indicator	Meaning
LIVE LOADED	The selected stored program is loaded in the Prophet edit buffer, matches its reference image, and its library data is current from the hardware.
CACHED LOADED	The selected stored program is loaded and matches its reference, but its displayed library information comes from the saved hardware mirror.
LIVE/CACHED EDIT BUFFER MODIFIED	The loaded program has unsaved edit-buffer changes. An asterisk appears after the name.
LIVE AUDITIONING	A library sound is temporarily sounding in the edit buffer. Stored Prophet memory has not changed.
LIVE/CACHED/MISSING NOT LOADED	The header selectors point to a program other than the program currently loaded in the edit buffer.
UNASSIGNED LOADED	A locationless program or layer is loaded and clean.
UNASSIGNED EDIT BUFFER MODIFIED	A locationless program or layer has unsaved changes.

Loaded does not mean saved.

LIVE | LOADED means the current edit-buffer image matches the reference image associated with the loaded stored program. It does not mean that a recent edit was written to Prophet memory. The edit buffer remains temporary until you explicitly use **Save Program**, **Save Pair**, a bank-write command, or another confirmed stored-memory write operation.

Additional document and synchronization indicators

The following indicators describe the active ReaProphet document or its relationship to the Prophet. They are separate from the LIVE, CACHED, LOADED, and AUDITIONING header status:

- **Program A / Program B** identifies the currently visible and editable layer of an active Stack or Split document. Each layer keeps its own program image, reference, location, and modified state.
- **Asterisk after a name** means that the Normal document or the indicated A/B layer differs from its own reference image. The change exists in the temporary edit buffer unless it is explicitly saved to Prophet memory.
- **Project sound: local only** means REAPER restored the saved ReaProphet document, but ReaProphet has not yet sent that sound to the Prophet. The document may be inspected and edited locally. Choose **Restore Prophet** to transmit and verify the complete saved Normal program or A/B pair.
- **Hardware state uncertain** means a protected operation may have changed the Prophet, but ReaProphet could not fully verify the final hardware state. MIDI output and host automation remain locked until project recall is retried, recovery is resumed, or the current Prophet state is reconciled.

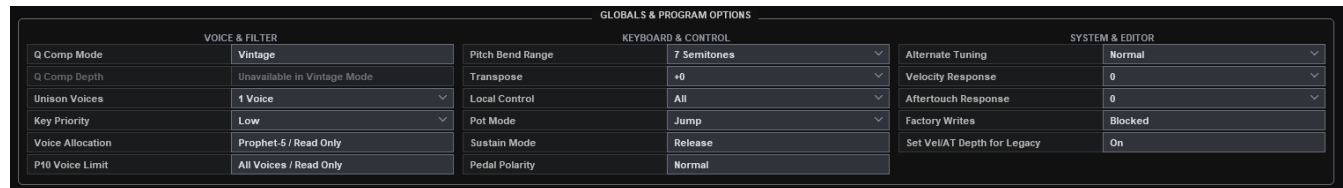
3.4 Hardware & Sync panel

Button or area	Function and use
Receive Current Program	Reads the current Prophet edit buffer and adopts it as the editor's active program. During Program A/B editing, ReaProphet warns before a readback would end the pair session and discard local-only layer work.
Send All Controls	Queues the complete current ReaProphet program image to the Prophet edit buffer. It does not save stored memory and cannot bypass the Project sound: local only recall boundary.
Clear Diagnostics	Clears receive/send counters and diagnostic state.
Request Globals	Reads the Prophet Global Dump and updates supported global and read-only values.
Refresh Bank	Reads the selected 40-program hardware bank into the local mirror. It never overwrites hardware.
Refresh All	Reads all 400 User and Factory programs. This can be cancelled.
Load Library	Loads the saved hardware mirror from disk through the helper.
Save Library	Validates and saves the current hardware mirror, favorites, and local category state.
Copy/Paste Program	Uses a volatile program clipboard. Paste changes the edit buffer only until Save Program or Save Pair.
Save Program / Save Pair	Normal programs save to the current location. Active pairs open the two-destination Save Pair workflow.
Restore Prior	Returns to the state captured before a browser audition or pair session when a valid runtime snapshot exists.
Export SysEx	Exports the current program as a nondestructive edit-buffer SysEx file.
MIDI Channel	Selects the channel used for Prophet communication.
Project sound: local only / Restore Prophet	A compact top-right status strip shown after REAPER restores a saved ReaProphet document. Restore Prophet performs deliberate, verified volatile recall; the strip disappears only after verification.

SEND ALL CONTROLS IS NOT SAVE

Send All Controls changes only the live edit buffer and does not save stored memory. While a saved project document is marked Project sound: local only, use Restore Prophet instead; Send All Controls does not bypass the protected recall boundary.

4. Globals & Program Options



The bottom panel deliberately combines true Globals, per-program options, editor preferences, and two read-only hardware values. The scope column below matters: program options travel with a patch, while Globals affect the instrument more broadly.

4.1 Voice & Filter

Control	Scope	Behavior
Q Comp Mode	Global	Vintage or Program. Q Comp Depth is only meaningful when Program mode is active.
Q Comp Depth	Per program	Eight stepped values. Disabled or visually unavailable when Q Comp Mode is Vintage.
Unison Voices	Per program	1-10 voices, Chord, or Poly Unison 2. Mirrors the Prophet's combined voices/type parameter.
Key Priority	Per program	Low, Last, Low Retrigger, Last Retrigger, High, or High Retrigger.
Voice Allocation	Read-only Global	Displayed from the Global Dump. Sequential confirmed no remote write path exists in the Prophet firmware, so this is read only. Change it on the hardware panel.
P10 Voice Limit	Read-only Global	Prophet-10 All Voices or 5 Voices. Sequential confirmed no remote write path exists in the Prophet firmware, so this is read only. Change it on the hardware panel.

4.2 Keyboard & Control

Control	Scope	Behavior
Pitch Bend Range	Per program	1-12 semitones. Copied, pasted, imported, exported, and saved with each patch.
Transpose	Global	Global keyboard transpose.
Local Control	Global	Off, Soft On, or All.
Pot Mode	Global	Relative, Passthru, or Jump.
Sustain Mode	Global	Release or Hold behavior.
Pedal Polarity	Global	Normal or Reverse.

4.3 System & Editor

Control	Scope	Behavior
Alternate Tuning	Global	Selects the Prophet alternate tuning table.
Velocity Response	Global	Selects the hardware velocity response curve.
Aftertouch Response	Global	Selects the hardware aftertouch response curve.
Factory Writes	Editor safety preference	Blocks Save Program, drag/drop, and bank-send writes to Factory memory unless deliberately enabled.
Set Vel/AT Depth for Legacy	Editor preference	For legacy patches with a velocity or aftertouch destination enabled and a stored depth of zero. This toggles setting the live velocity or aftertouch depth to maximum on load. It does not rewrite cached or durable files unless you save.
Split Point	Program A pair metadata	Appears only in Split mode. Drag the row to set the keyboard split; the note name and MIDI note are displayed.

FACTORY WRITE PROTECTION

Leave Factory Writes off for normal use. Enabling it allows permanent Factory-memory overwrites from Save Program, Save Pair destinations, bank sends, and drag-and-drop replacement. However, the option exists if you would like to overwrite or reorganize Factory patches.

5. The Program Page: Editor-Specific Features



This manual does not duplicate the Prophet synthesis controls. The Program page is designed to follow the panel, while adding editor-only functions, clearer program options, and Layer A/B tools.

5.1 Immediate editing and hardware feedback

- Dragging a knob or pressing a switch updates the live Prophet edit buffer through NRPN.
- Turning a knob or pressing a button on the Prophet will also be reflected in the editor.
- Editor labels turn yellow when the value differs from the currently saved patch.
- Velocity and Aftertouch destination switches cycle Off, first destination, second destination, and Both. Red, yellow, and green lamps distinguish those states.
- Filter revision uses red for Rev 1/2 and yellow for Rev 3. Filter keyboard tracking cycles Off, Half, and Full.
- When Project sound: local only is visible, knob and switch edits update the saved local document but intentionally send no MIDI. Immediate hardware editing resumes after verified Restore Prophet.

5.2 Oscillator Frequency octave shortcut

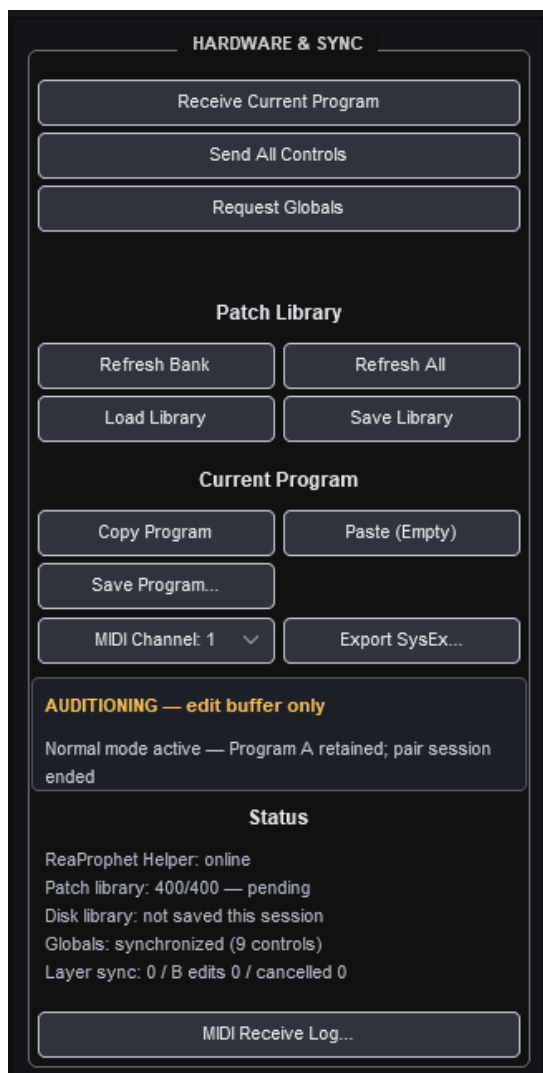
ALT/OPTION OCTAVE DRAG

Hold Alt on Windows or Option on MacOS while dragging Oscillator A Frequency or Oscillator B Frequency. The control moves in complete octaves instead of raw steps.

The shortcut is intentionally restricted to the two oscillator Frequency knobs. Normal dragging retains fine control, including the intermediate “+” positions used by the Prophet’s raw frequency values.

5.3 Unison and performance additions

- Unison Voices / Type exposes the complete 1-10 Voice, Chord, and Poly Unison 2 choices.
- Unison Detune is a per-program Rev4 value. It is copied, pasted, imported, exported, and saved with the patch.
- Velocity Depth and Aftertouch Depth expose the continuous per-program amounts available on newer firmware while retaining destination switches.
- When a pair is active, the Performance section adds Layer A Vol or Layer B Vol for the currently selected layer.



5.4 Program clipboard

- **Copy Program** captures the current complete program image into a volatile ReaProphet clipboard.
- **Paste Program** sends the clipboard image to the live edit buffer. Nothing is written to stored memory until Save Program or Save Pair.
- The clipboard is not persistent across a fresh REAPER session or script recompile.

5.5 Save Program and Export SysEx

- **Save Program** writes only the currently loaded location. The confirmation dialog allows a 20-character patch name and Category selection. Factory locations are blocked unless Factory Writes is enabled.
- **Export SysEx** creates an edit-buffer-form file. Loading or transmitting that file auditions the sound without directly overwriting a stored location.
- **Save Pair** used during Program A/B editing, rather than Save Program, to preserve the complete two-layer relationship.

6. Patch Library: Hardware Browsing and Auditioning



The left side of Patch Library is a searchable 400-program mirror of Prophet hardware memory. It is not a second hardware memory. Refresh reads the Prophet; Save Library persists the mirror to disk.

6.1 Refreshing and persistence

- **Refresh Bank** reads the currently selected 40-program User or Factory bank.
- **Refresh All** reads all 400 programs and displays modal progress. It can be cancelled safely.
- **Load Library** restores the last saved mirror from disk. The helper also loads it automatically when it starts and reconnects.
- **Save Library** validates the patch count and writes patch_cache.json with a backup file.

6.2 Browsing and the edit buffer

Action	Normal mode	Active Stack/Split pair
Click a normal row	Auto-auditions the program in the Prophet edit buffer. Stored memory is not changed.	Selects the row only so a layer is not replaced accidentally.
Click a Stack/Split Layer A row	Selects it and waits for Audition Pair.	Selects it as a possible source; a nested pair is never imported into one layer.
Load to Prophet	Explicitly loads the selected row to the live edit buffer and loads that patch location on the hardware.	Available as an explicit audition action, while Load Into A/B is used for layer replacement.
Restore / Restore Prior	Returns to the state captured before auditioning.	Returns to the state captured before the pair session when a valid snapshot exists.
Load Into A / Load Into B	Not shown because there is no active pair.	Replaces only the chosen layer and keeps the current pair metadata.

EDIT-BUFFER AUDITIONING IS SAFE

A normal library audition changes only the temporary edit buffer. It becomes permanent only if you deliberately save the program or use a confirmed hardware-write operation.

6.3 Search and filters

- Search hardware patches by location, patch name, and category text. **Example:** searching 'pad' will identify all patches with pad in the preset name, and all patches in the pads category.
- All ★ / ★ Only toggles between all patches and favorites only.
- All / User / Factory cycles the memory location filter.
- Click the star column to toggle a favorite. Favorites are ReaProphet metadata and require the helper to persist.
- Click the Category column to assign None, Keys, Synth, Pads, Organs, Strings, Lead, Sync, Bass, Bright Bass, Sub Bass, Brass, Vox, Bells, Percussion, or SoundFX.

6.4 Category behavior

Category is part of a Prophet program image, while Favorite is local library metadata. Changing the category of the currently auditioned or loaded program also updates the live edit-buffer image, so a later **Save Program** can write it to the Prophet. Category updates do not automatically resave Prophet presets, you must Save Program for the category to be written to the hardware. Changing the category of an imported bank edits ReaProphet's durable local copy; the original external .syx file is not modified.

6.5 Program clipboard on the library page

- **Copy Patch** copies the selected hardware or bank-source patch into the shared volatile program clipboard.
- The Hardware & Sync panel shows the copied patch name and source.
- **Restore** returns to the pre-audition edit buffer; Clear Clipboard empties only the volatile clipboard.

7. Bank Sources, SysEx, Backups, and Drag-and-Drop Management



The right side of Patch Library is a source browser. It can display cached hardware banks, imported SysEx banks, automatic backups, and the Single Programs collection. This makes the right pane useful both as a librarian and as a source for hardware reorganization.

7.1 Why hardware patches appear twice

INTENTIONAL REPETITION

The complete hardware catalog appears on the left as destinations and again on the right when a Hardware Library bank source is selected. The first ten Bank Sources are live views of cached User 1-5 and Factory 1-5. This repetition is intentional: it lets you keep a source bank open on the right while dragging patches to exact hardware destinations on the left for easy reorganization.

The right-side hardware entries are views of the existing cache, not additional stored copies. Imported banks and backups follow the ten hardware views in the Bank Sources menu.

7.2 Bank Sources controls

- Bank Source dropdown: choose a cached hardware bank, imported bank, backup, or Single Programs collection.
- Search selected bank source: filters its 40 slots by name, category, and displayed role/link information.
- All ★ / ★Only: filters the current source by favorite status.
- Layer Role / Link: identifies Stack/Split Layer A rows, their required Layer B location, and Layer B rows that serve another location.

7.3 Bank Actions

Action	Behavior
Import SysEx	Imports compatible Prophet Rev4 stored-program or edit-buffer SysEx. Single programs are grouped into a Single Programs collection; banks are stored as durable local bank records.
Send Selected Bank to Prophet	Chooses a User or allowed Factory destination, backs up the destination bank, then writes and verifies each source slot.
Back Up Selected Prophet Bank	Requires a fully cached hardware bank. Writes a timestamped .syx and adds the bank to the durable local library.
Export Selected Program	Exports the selected program as nondestructive edit-buffer SysEx.
Export Selected Bank	Exports all valid programs in the selected source as a stored-program .syx bank.
Rename Selected Bank	Renames a durable imported/backup bank. Disabled for live hardware source views.
Delete Selected Bank	Removes a durable bank from the local library. Original external files and backup .syx files are not deleted.

7.4 Drag-and-drop patch reorganization

1. Select a source bank on the right.
2. Click and drag a patch row. A normal click remains an audition; dragging begins only after the pointer moves beyond the small gesture threshold.
3. Move over the exact hardware destination row on the left. The valid target highlights red.
4. Release and confirm the replacement. ReaProphet verifies the stored write before reporting success.
5. Use Restore if you want to return the live edit buffer to the state that existed before the operation.

PERMANENT WRITE

Drag-and-drop replacement writes the selected hardware destination. Factory targets are unavailable unless Factory Writes is enabled. Always confirm the compact User/Factory destination location in the warning dialog.

7.5 Stack/Split banks and the Inhalt-style case

A Prophet Stack/Split Layer A program stores a pointer to a Layer B hardware location; it does not contain Layer B's sound. ReaProphet therefore treats imported combination banks differently from ordinary sound banks.

- Layer A rows are labeled Layer A: Stack or Layer A: Split and show the required Layer B location.
- Referenced Layer B rows are labeled Layer B and show which Layer A location uses them.
- Stable location labels are used because the location relationship is the important part of the combination.
- **A normal click does not auto-audition a Stack/Split Layer A.** To audition, select it and press **Audition Pair**.
- Audition Pair is available only when the required Layer B program is present in the same selected bank source and the current Prophet state can be captured.
- ReaProphet loads and verifies Layer B first, then loads Layer A last because Program A owns the combination metadata.

- When sending an entire bank to another hardware bank, links contained inside the source bank are remapped to the new destination bank. External or missing links retain their original locations and are reported before the write.

8. Stack/Split and Program A/B Workflows

ReaProphet treats a live Stack or Split as one editable two-document session. Program A and Program B keep independent sound images, references, locations, and dirty states. Program A also owns the pair metadata.

8.1 Creating a Stack or Split

Starting point	What ReaProphet does
Choose Stack or Split from Normal	Keeps the current sound as Program A, captures the prior state for Restore Prior, and initially loads Factory 1-1-1 as Program B. The intention is for users to select a new Program B layer manually.
Load an existing hardware Stack/Split	Captures Program A and Program B through verified readback, then opens the same editable A/B document used by imported pairs.
Audition Pair from a bank	Captures the current Prophet state, loads and verifies B first, then A, and offers Keep Pair Active or Restore Prior.

8.2 Switching and editing layers

- The Program tab becomes Program A or Program B. Clicking it while already on the Program page toggles layers.
- The Prophet's physical A/B layer buttons can also select the layer; ReaProphet follows the hardware selection.
- Before switching layers, ReaProphet commits pending edits and then sends the layer-select command, reducing the risk of an NRPN reaching the wrong layer.
- Each layer has an independent dirty indicator and reference image.
- When a Pair was restored from a REAPER project or FX preset and is still local only, Program A/B tab changes navigate the saved documents locally and do not send CC80 until Restore Prophet succeeds.
- Program A owns Mode, Split Point, Layer A/B volumes, and the linked Program B location.

8.3 Loading sounds into A or B

- Use the header source selectors or select a row in Patch Library, then press Load Into A or Load Into B.
- Only the chosen sound document is replaced. Program A's Stack/Split metadata remains intact.
- If the source itself is a Stack/Split Layer A, only its sound is imported. ReaProphet does not create nested pairs.
- Right-click Load Into A/B and choose Exit Stack/Split and Load Full Program when you intend to leave the pair and load the complete selected program.

8.4 Swap Layer Sounds

Swap Layer Sounds copies Program B's complete sound into Program A and Program A's complete sound into Program B. Mode, split point, layer volumes, linked locations, and Save Pair destination roles stay in place. The swap remains unsaved until Save Pair. This feature is particularly handy for quickly reversing a the low/high patches in a Split pair.

8.5 Save Pair

1. Choose **Save Pair** from the Current Program panel. If the pair came directly from hardware, ReaProphet first captures both active layers.
2. Review or edit the Program A and Program B names and choose separate destinations for both layers.
3. Confirm any User or Factory overwrites. Factory destinations require Factory Writes.
4. The helper creates a recovery .syx and manifest containing the original contents of both destinations.
5. ReaProphet writes and verifies Program B first. B is stored as a standalone Normal sound.
6. ReaProphet writes and verifies Program A last. A stores the Stack/Split mode, B location, split point, and layer levels.

WHY B IS WRITTEN FIRST

Program A points to Program B. Writing and verifying B first prevents A from being stored as a combination that immediately references an unprepared destination.

RECOVERY VERSUS ROLLBACK

Save Pair creates recovery backups, but a partial hardware-write failure is reported rather than automatically rolled back. Keep the generated recovery .syx until both destinations have been rechecked.

8.6 Restore Prior and returning to Normal

- **Restore Prior** is the pair-revert function. It discards the unsaved pair session and restores the exact state captured before the session began.
- The Restore Prior snapshot is runtime-only. Recompiling the JSFX or replacing the instance clears it. This is separate from project persistence, which saves the current coherent Normal or Pair document inside the REAPER project or FX preset.
- Returning to Normal preserves Program A in the live edit buffer, releases Program B, and does not write stored memory.
- If Program B contains unsaved edits, the Normal transition warns that those changes will be discarded.
- After returning to Normal, use Save Program only if you want to store Program A as a standalone Normal patch.

9. Project Persistence, Restore Prophet, and Recovery

ReaProphet saves the complete sound state with the REAPER project or FX preset. When that state is loaded, ReaProphet restores the controls, program names, locations, modified status, and any active Stack or Split setup inside the editor.

Loading a project does not automatically change the physical Prophet. Restoring the saved sound to the hardware is a separate, deliberate operation performed with **Restore Prophet** and confirmed through hardware readback.

9.1 What ReaProphet saves with the project

- A Normal sound preserves the complete current program, its original reference when available, and its source location.
- A Stack or Split preserves both Program A and Program B, their independent references and locations, the selected mode, split point, layer levels, and whichever layer was active when the project was saved.
- Modified status is rebuilt by comparing the saved sound with its original reference. Using Restore Prophet does not remove an asterisk or mark the sound as saved to Prophet memory.
- Temporary operating details—such as pending MIDI messages, transfer progress, transaction stages, recovery progress, helper activity, and temporary browser selections—are not restored as part of the saved project sound.
- Prophet Globals are not included in project sound recall.

THE SAVED PROJECT SOUND MAY NOT MATCH THE PROPHET YET

The saved sound can be viewed and edited even when the Prophet is powered off. It represents the sound the REAPER project intends to use, not confirmation that the physical Prophet currently matches it. **Choose Restore Prophet when you are ready to send and verify that sound on the hardware.**

9.2 Local-only state after project, preset, or host-state loading

- Opening a project, loading an FX preset, Undoing or Redoing a preset-state change, duplicating an FX, copying a track, or loading an FX chain may reconstruct ReaProphet state. These host actions do not automatically send project-recall MIDI.
- The compact top-right strip reads Project sound: local only (or Local only at narrow widths) and includes Restore Prophet.
- Normal and Pair documents remain fully inspectable. Local parameter edits are saved into the project document, and A/B tab navigation remains local, until hardware recall verifies.
- Send All Controls cannot be used to bypass this boundary. Incoming Prophet activity is not treated as permission to overwrite the saved project document.
- ReaProphet does not create REAPER Undo points for ordinary knob and switch gestures. REAPER may still Undo/Redo host preset-state changes that it created itself.

WHAT YOU HEAR BEFORE RESTORE IS NOT PROJECT RECALL

Immediately after reopening a project, the Prophet may still play whatever was last loaded on the hardware. The saved project sound has only been reconstructed locally until Restore Prophet is clicked and verified.

9.3 Restore Prophet

Click Restore Prophet only after the Prophet is powered on and the REAPER MIDI ports are available. The protected progress dialog is titled Restoring Project Sound.

Document	Verified volatile-recall sequence
Normal	Read-only edit-buffer preflight; establish Normal mode; select Program A; send the complete saved image; request and verify matching readback.
Stack or Split	Read-only preflight; establish the saved mode; send and verify B; send and verify A; re-select and verify B survived; return to the saved active layer; verify the final active-layer context.

- Restore Prophet changes only volatile edit buffers. It never writes a User or Factory program location.
- Known locations are reference metadata; recall does not issue a Program Change merely because a location is known.
- During the progress dialog, do not change programs, layers, MIDI routing, the Prophet panel, or another ReaProphet instance that uses the same hardware.
- After complete verification, the local-only strip simply disappears. There is no separate success banner.

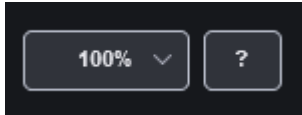
9.4 Project-recall status and modal reference

Message	Meaning	What to do
Project sound: local only	The saved document is loaded locally; hardware has not been recalled.	Inspect or edit locally, then click Restore Prophet when the device and ports are ready.
Restoring Project Sound	A protected preflight/send/readback sequence is active.	Wait. Do not touch programs, layers, routing, or the Prophet panel.
The Prophet Did Not Complete Project Recall	The read-only preflight received no valid edit-buffer response. No hardware changes were made.	Check power, ports, track output, MIDI channel, NRPN/SysEx settings, then Retry. Close returns to the local-only strip.
Project Recall Incomplete — Hardware State Uncertain	One or more volatile changes may have reached the Prophet, but the complete saved document did not verify.	Repair the connection, then Retry Project Recall. Return to Editor — Keep Hardware Locked is safe for inspection but leaves MIDI output and host automation locked.
Hardware state uncertain — MIDI output and host automation remain locked	A recovery workflow was suspended without claiming success.	Repair the connection or routing, then click Resume Recovery. Do not treat MODE: UNKNOWN or disabled hardware controls as a bug in this state.
Saved ReaProphet state was malformed / unsupported / not adopted over a live runtime	The incoming host state was rejected as a whole. The existing editor state was kept and no MIDI was sent.	Use the current build and a known-good project or preset. If an operation was active, let it finish and reload the host state in an idle instance.

9.5 Other protected-operation dialogs

Dialog	Recommended action
Save Stack/Split Pair / Confirm Save Pair	Choose different A and B destinations, review the current and replacement names, then Continue and Save Pair. Use Back to revise; Cancel starts no write.
Save Pair Failed / Partial Save Pair Failure	Close the dialog, keep the timestamped recovery files, and inspect both destinations. ReaProphet reports the strongest known partial-write state and does not automatically rewrite stored memory.
Layer-Sound Swap Failed	Use Retry Original Restore when available. If hardware is uncertain, Reconcile Current Hardware reads what is actually present. Return to Editor — Keep Hardware Locked leaves recovery available.
Pair Hardware Reconciled	Adopt Reconciled State only when you want the editor to continue from the verified current hardware images. Retry Original Restore attempts to return to the immutable pre-swap images.
Mode Change Failed / Hardware State Uncertain	If the previous mode was reasserted, Close. If hardware is uncertain, request or retry the edit-buffer readback. Returning to the editor keeps hardware locked.
MIDI Transmission Failed	Retry Pending Write only after checking the connection. Resync Before Retry or Abort and Resync discards unsafe pending work and requires a fresh readback.
Program Selection Failed	Retry Selection after repairing MIDI output. Abort and Resync abandons the unverified selection and requires a fresh readback.
Hardware Resync Required	Use the dialog's Request Edit Buffer button (the recovery-dialog name for the same basic readback used by Receive Current Program). If necessary, return to the editor with hardware locked and resume later.
End Program A/B Session?	Cancel preserves the local Pair. Request and End Session deliberately replaces it with a fresh hardware readback and discards unsaved layer-only work.
Exit Stack/Split?	Cancel preserves the Pair. Exit and Load leaves the Pair and loads the selected complete Normal program.

10. Help, UI Scaling, and Default Window Size



The Help page is a compact in-editor quick start. This manual is the longer workflow reference. The scale selector is immediately left of the Help icon and affects custom drawing, fonts, hit testing, scroll regions, and Retina/HiDPI rendering.

10.1 Scale choices

Scale	Typical use
75%	Laptop screens and smaller working areas. Some compact control labels are abbreviated to prevent overlap.
100%	The original logical layout and baseline size.
110%	A modest enlargement that often works well on larger desktop displays.
125%	Larger controls and text for high-resolution displays or increased viewing distance.
150%	Maximum built-in enlargement for large or very high-density screens.

Native REAPER/operating-system menus may keep the host's own text size, especially on Windows. ReaProphet scales menu placement and hit behavior, but native menu typography follows the host and OS.

10.2 Set the scale and startup window as the default

1. Choose the desired scale from the ReaProphet scale menu.
2. Open the scale menu again and choose Set [scale] as Default. The helper writes `ui_preferences.json` and updates the marked startup-size request in the installed scaling build.
3. Resize the floating FX window once if REAPER has retained an older wrapper size.
4. Open REAPER's FX preset menu and choose Save preset as default.
5. Close all ReaProphet instances, restart REAPER, and insert a fresh instance to confirm the selected scale and matching window dimensions.

THE REAPER PRESET STEP MATTERS

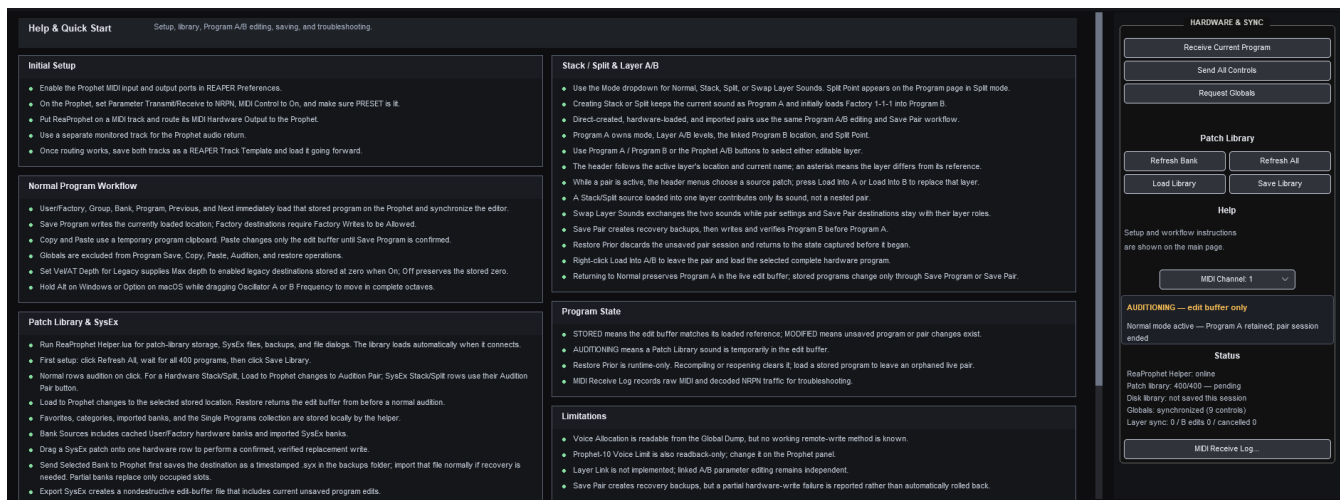
Set as Default configures ReaProphet's scale and requested `@gfx` size. Save preset as default tells REAPER to create fresh instances with that parameter state and allows the wrapper to open at the matching dimensions reliably.

10.3 Startup size requests

Scale	Requested initial size
75%	1410 × 578
100%	1880 × 770
110%	2068 × 847
125%	2350 × 963
150%	2820 × 1155

REAPER treats these values as requested sizes, so the available screen, docking state, and previously saved default preset can influence the actual wrapper. The five-step procedure above is the verified Windows workflow.

10.4 Help screen behavior



- Help uses box-bounded measured wrapping so text stays within each card on macOS, Windows, and scaled layouts.
- The Help page scrolls vertically when the cards exceed the available window height.
- The Hardware & Sync panel remains available and shows setup guidance while Help is open.

11. Practical Workflows, Troubleshooting, and Limitations

11.1 Recommended everyday startup

1. **Power on the Prophet and MIDI interface before starting REAPER.**
2. **Start REAPER, confirm the helper is running, and load the saved ReaProphet track template containing the MIDI/editor and audio-return tracks.**
3. **Confirm the Prophet PRESET light is on and the track MIDI Hardware Output is available on the correct channel.**
4. **If Project sound: local only appears, inspect the saved document and click Restore Prophet when ready. Otherwise, use Receive Current Program and Request Globals when the editor needs to follow the current hardware.**
5. **Use the header for fast stored-program navigation or Patch Library for search, favorites, categories, auditioning, and bank sources.**

11.2 Building a new Stack or Split

1. Load or design the sound you want as Program A.
2. Choose Stack or Split from the Mode menu. ReaProphet captures the prior state and creates an editable A/B session.
3. Select Program B and use the header or Patch Library to choose a source, then press Load Into B.
4. Adjust Layer A Vol, Layer B Vol, and Split Point if applicable.
5. Switch between Program A and Program B to edit both sounds independently.
6. Use Save Pair, confirm both destinations, and keep the recovery backup until the stored pair has been reloaded and checked.

11.3 Common problems

Symptom	Check
Helper shows Offline	Run the exact current Lua action. Stop duplicate helpers. Confirm the ReaProphetEditor folder is writable.
Controls do not affect the Prophet	First check whether Project sound: local only or a hardware-lock warning is visible. Otherwise verify MIDI input/output ports, track hardware output, matching channel, NRPN transmit/receive, and MIDI Control On.
Every patch sounds like the front panel	Press PRESET so its LED is lit. PRESET off is Live Panel mode.
No audio	ReaProphet does not pass audio and requires the Rev 4 hardware. Check the separate audio-return track input, monitoring, interface connection, and Prophet output level. Check the PRESET LED to ensure PRESET is enabled on the hardware.
Patch names are missing	Check the Helper status. If online, refresh the relevant hardware bank or Load Library so the cached names are available.
Factory target is unavailable	Leave it blocked unless intentional; enable Factory Writes only for a deliberate Factory overwrite.

Symptom	Check
Audition Pair is unavailable	Select a Stack/Split Layer A and confirm its required Layer B is present in the same selected bank source.
Restore Prior is unavailable	The runtime-only Restore Prior snapshot may not exist, or the instance may have been replaced/recompiled. This does not affect the separately persisted REAPER project document.
Scale loads but window size is old	Use Set as Default, resize once, then use REAPER's Save preset as default and test a fresh instance.
Project reopens but the Prophet still plays another sound	Expected until recall. Confirm the saved controls and names are present locally, then click Restore Prophet in the top-right local-only strip.
Restore Prophet reports no valid response or diagnostic 831	The read-only preflight received no valid edit-buffer dump, so no hardware change occurred. Check the MIDI return path, channel, power, and Prophet SysEx settings.
Prophet or MIDI port was turned off/closed while REAPER was open	Restart REAPER with the Prophet and MIDI interface already powered on. Closing a port, unplugging the interface, or power-cycling the Prophet can leave REAPER's existing MIDI connection unavailable.
Hardware state uncertain, MODE: UNKNOWN, or controls disabled	A protected operation may have mutated hardware without verification. Repair the connection and use Resume Recovery or the offered readback/reconciliation action. Return to Editor — Keep Hardware Locked intentionally leaves output disabled.
Saved state was not adopted over a live runtime	The existing document was kept and no MIDI was sent. Finish or close the active protected operation, then reload the project or FX preset in an idle instance.

11.4 Known limitations

- Prophet-10 Five Voice Limit and the Voice Allocation mode are readback-only and must be changed from the hardware panel.
- Restore Prior is volatile and cannot survive a fresh instance or recompile.
- Save Pair reports partial failures and retains recovery files; it does not automatically rewrite the old destination contents.
- The companion helper is required for persistent libraries, backups, logs, exports, and UI-default disk settings.
- Project persistence restores program documents, not Prophet Globals. Request Globals separately when needed.
- Automatic hardware recall during project, preset, Undo/Redo, duplication, or template loading is deliberately disabled; Restore Prophet is explicit.
- Transaction and recovery records remain runtime-only and are not resumed after a fresh plugin lifecycle.
- Ordinary synth knob and switch gestures do not create REAPER Undo points.
- After the Prophet or a MIDI port is closed or power-cycled while REAPER remains open, restart REAPER before expecting reliable hardware communication.

11.5 Release-safety habits

- Keep Factory Writes off by default.
- Refresh and save the hardware mirror before major bank reorganization.
- Back up any bank before a large write. Send Selected Bank does this automatically, but manual backups remain useful milestones.

- After Save Pair, reload both stored locations and confirm the pair, names, volumes, split point, and standalone B sound.
- Use compact U/F location codes in confirmations rather than relying on patch names alone.
- Power on the Prophet and MIDI interface before starting REAPER. If either is turned off or its port is closed mid-session, restart REAPER before continuing hardware work.
- During Restore Prophet, Save Pair, Pair Swap, mode changes, and bank writes, do not change programs, layers, routing, or the Prophet panel until verification finishes.
- Treat Project sound: local only as a deliberate safety boundary, not an error. Use Restore Prophet when the saved project document should become the live hardware state.

Appendix A. Files and Storage Locations

The helper stores user data under Documents/ReaProphetEditor. Exact platform paths differ only in the user-home prefix. The serialized Normal or Pair project document is stored by REAPER inside the project, FX preset, track template, or FX-chain state; it is not stored in the ReaProphetEditor helper folder.

Path inside ReaProphetEditor	Contents
settings.json	Device identity and paths used by the helper.
ui_preferences.json	Saved default scale and managed startup-size preference.
patch_cache.json	Saved 400-program hardware mirror, including names and local metadata.
patch_cache.json.bak	Previous hardware-cache version created during save.
library/bank_library.json	Index of durable imported banks, backups, and Single Programs collections.
library/banks/*.json	Validated bank records containing decoded program images and local favorites.
backups/*.syx	Hardware bank backups and Save Pair recovery files.
backups/*.txt	Save Pair recovery manifests describing the original destinations.
logs/midi_rx_log_latest.txt	Most recent exported MIDI/NRPN receive log.
logs/midi_rx_log_*.txt	Timestamped receive logs.

Appendix A.1 Typical full paths

Platform	Root folder
Windows	C:\Users\<name>\Documents\ReaProphetEditor
macOS	/Users/<name>/Documents/ReaProphetEditor

BACK UP THE ENTIRE FOLDER

Copying the complete ReaProphetEditor folder preserves the hardware mirror, favorites, categories, imported banks, backups, logs, and UI preferences together.