

P12 Librarian

User Guide

Back up, browse, rename and restore the programs
of a Prophet 12 over MIDI SysEx

qutrit LOGIC

Version 1.0.1 · October 2026
Tested with Prophet 12 Keyboard, OS 1.4.2.3

Contents

1 Introduction 1
1.1 Tested setup 1
2 Getting started 1
2.1 Installing 1
2.2 Settings on the synth 2
2.3 Connecting 2
2.4 The window 3
2.5 Slots 3
2.6 Selecting 3
2.7 Sorting 3
3 Backing up 4
3.1 Reading programs 4
3.2 Backing up everything 4
3.3 Saving a selection 4
3.4 Bank sheets 4
4 Writing patches to the synth 5
4.1 Opening a file 5
4.2 Writing 5
5 Editing patch lists 5
5.1 Comparing the synth with a file 6
5.2 Duplicates 6
5.3 Rearranging patches on the synth 6
5.4 Audition 7
6 Renaming 7
6.1 Rename 7
6.2 Cleanify names 7
7 Layer B 8
8 Initializing slots 8
9 Searching 8
10 Tags and favourites 9
11 Safety and backups 9
11.1 Undo and restore points 10
12 Settings 10
12.1 Zoom, font and theme 10
12.2 Where the app keeps things 10
13 Keyboard shortcuts 11
14 Troubleshooting 12
15 Command-line tool 12
16 Technical notes 12
17 About 13
18 Disclaimer 13

1 Introduction

P12 Librarian is a desktop app for managing the programs (patches) stored in a Dave Smith Instruments Prophet 12. It talks to the synth over MIDI SysEx and lets you:

- **back up** all User banks, or any selection, to standard `.syx` files;
- **browse and search** every program by name, and find them again with **tags and favourites**;
- **audition** patches from a file without storing them;
- **write** patches to any User slot, including into **Layer B** of an existing program;
- **rearrange** patches on the synth: swap them, or move them and shift the rest;
- **compare** the synth with a backup, and find duplicates;
- **edit** `.syx` patch lists without a synth;
- **rename** patches, and **clean up** names typed in capitals on the synth;
- **initialize** slots with the synth's Basic Patch;
- **print** bank sheets, or export them for a spreadsheet.

Every write is protected the same way: the slots about to change are read and saved first, and every written program is read back from the synth and compared with what was sent. Any change can be undone (Section 11).

Use at your own risk. Writing to the synth replaces the programs stored there, and data can be lost. Keep your own backups. The app asks you to accept the disclaimer the first time it starts (Section 18).

1.1 Tested setup

Synthesizer	Prophet 12 Keyboard
Synth OS	1.4.2.3, the final release from DSI
Connection	USB, directly to the computer
Computers	Linux Mint 22.3 (Cinnamon, X11) and Windows 11, both 64-bit

> **Note**
> The **Prophet 12 Module** is expected to work the same way, but has not been tested. There is no macOS version.

What was verified on the hardware above, beyond what the manual documents:

- where the program name is stored inside a program (the manual doesn't say);
- that writing, renaming and initializing store exactly what was sent (every write is verified);
- that lowercase letters in names display correctly;
- that the synth's own **Copy A -> B** copies a whole layer, which is what Layer B copying is based on.

2 Getting started

2.1 Installing

Download P12 Librarian from qutrit.se/p12-librarian. Installing a newer version over an older one keeps your settings and backups.

Linux Ubuntu 24.04, Linux Mint 22, Debian 13 or newer, 64-bit. Install the `.deb` package from a terminal in the folder you downloaded it to, then start P12 Librarian from the applications menu:

```
sudo apt install ./p12-librarian_<version>_amd64.deb
```

Other Linux Fedora 40, Arch, openSUSE Tumbleweed and other 64-bit distributions from 2024 on. Download the AppImage, make it executable and start it; nothing is installed. If it doesn't start, install FUSE 2 (`libfuse2`, on Ubuntu 24.04 `libfuse2t64`).

```
chmod +x P12-Librarian_<version>_x86_64.AppImage
./P12-Librarian_<version>_x86_64.AppImage
```

Windows Windows 11, 64-bit. Run the downloaded `setup.exe`. Windows shows *"Windows protected your PC"* because the installer isn't signed: click **More info**, then **Run anyway**. Accept the licence, and start P12 Librarian from the Start menu. It installs for your user account only and needs no administrator rights; uninstall it in **Settings → Apps**.

The command-line tool `p12` (Section 15) is only included in the Linux `.deb` package for now.

2.2 Settings on the synth

The Prophet 12 ignores SysEx until it's enabled. Press **Global** and set:

Global setting	Value	Why
10 MIDI SysEx Enable	On	Lets the synth receive and send SysEx.
11 MIDI SysEx Cable	USB	Use **MIDI Port** instead if you connect through a MIDI interface's DIN ports.

2.3 Connecting

Connect the Prophet 12 by USB and start the app. It looks for MIDI ports with "Prophet 12" in the name and connects automatically. The header then shows the model and OS version, e.g. ● Prophet 12 Keyboard · OS 1.4.2 (the synth reports its OS version without the last part; the Global screen shows all of it).

If the synth is unplugged or switched off, the app notices within a few seconds and says so. When it comes back, the app connects again by itself, unless you clicked **Disconnect** **Connect** turns this back on.

If it doesn't connect, choose the ports in the **In** and **Out** lists, click **Connect**, and use ↻ to rescan. Connecting checks that a Prophet 12 really answers; if it doesn't, see Section 14.

> **Note**
> On **Windows**, only one program at a time can use a MIDI port. Close any DAW, editor or other MIDI software that uses the Prophet 12 before connecting.

2.4 The window

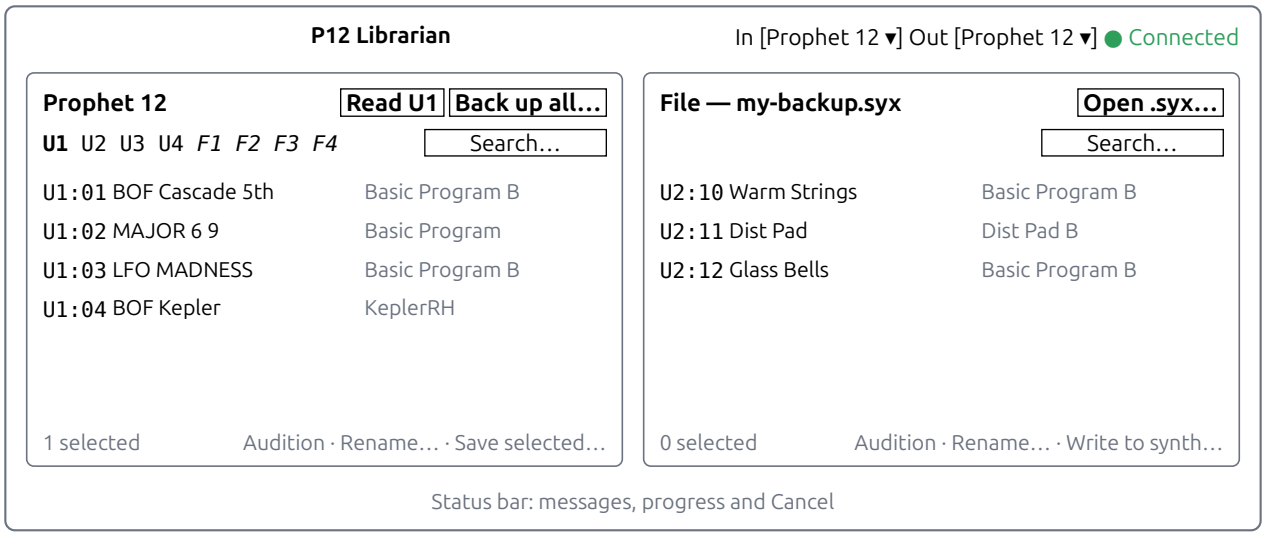


Figure 1: The main window: the synth on the left, an opened file on the right.

- **Header:** MIDI ports and connection status.
- **Prophet 12 pane (left):** the programs on the synth, one bank at a time. Tabs U1–U4 are the User banks; F1–F4 (in italics) are the read-only Factory banks. The small number on a tab is how many of its 99 slots have been read. Slots not read yet show "—".
- **File pane (right):** a list of patches, usually an opened `.syx` file. You can also build and edit lists here without a synth; see Section 5.
- **Status bar:** what just happened, or the progress of a running operation.

Drag the gap between the panes to make one wider; double-click it for equal widths. The app remembers the width.

Every button has a tooltip. If a button is greyed out, its tooltip says why.

2.5 Slots

A slot is written as bank and program number, as on the synth's panel: `U1:05` is User bank 1, program 5; `F3:99` is Factory bank 3, program 99. The User banks U1–U4 can be written; the Factory banks F1–F4 are read-only on the synth, and the app never writes to them.

Each program has two layers, A and B. The name in the **Name** column is Layer A's, the one the synth shows; the **Layer B** column shows the second layer's name (Section 7).

2.6 Selecting

Click	Select one patch.
`Ctrl` + click	Add or remove a patch.
`Shift` + click	Select a range.
Double-click	Audition the patch (Section 5.4).
Right-click	Menu with everything you can do with the selection.

2.7 Sorting

Click a column heading (★, **Slot**, **Name**, **Layer B** or **Compare**) to sort the list by it, again for the other way round, and a third time for the list's own order; ▲ or ▼ shows which. Sorting only changes how the list is shown: **Save** still saves a file in its own order. Selected patches follow the order shown when you write them to consecutive slots (by dragging or with **Write to synth…**) or use **Save selected…**. While a list is sorted, patches can't be dropped onto it, except onto a Layer B cell, because the slots they would fill aren't in order on screen.

3 Backing up

> **Important**
> Back up all User banks **before updating the synth's OS**. OS updates have been reported to change or erase user programs. Keep a copy of the backup somewhere other than the computer.

3.1 Reading programs

- **Read U1** (the name follows the open tab) reads the 99 programs of that bank, in about 3 seconds.
- **Read all User banks** reads all 396 User programs, in about 10 seconds.
- To read a Factory bank, open its tab and click **Read F1** (and so on).

A progress bar shows each program as it arrives. **Cancel** stops reading; programs read so far are kept.

The app doesn't notice changes made on the synth itself: the Prophet 12 sends nothing when you save with **Write**. After saving an edit on the synth, select the slot and press R, or right-click → **Re-read from synth**, to read just that slot again. The status line says which slots had changed.

3.2 Backing up everything

Back up all… (Ctrl+B, also in the **File** menu) asks where to save, reads all four User banks and saves them to one `.syx` file.

> **Example: a backup before an OS update**
> 1. Press Ctrl+B.
> 2. Keep the suggested name, e.g. `p12-backup-20261008-210500.syx`, or type your own, and click **Save**.
> 3. Wait about 10 seconds. The status bar says *"Saved all 396 User programs to …"*.
> 4. Copy the file to a USB stick or cloud storage.

3.3 Saving a selection

Select patches in either pane, then:

- **Save selected…** saves them into **one** `.syx` file;
- **Save as separate files…** saves **one file per patch** into a folder you choose, named after slot and name, e.g. `U1-05 BOF Cascade 5th.syx`.

The files are ordinary SysEx: other librarians and SysEx tools can read them, and each one remembers which slot it came from.

3.4 Bank sheets

File → Export bank sheet… saves a list of patch names, to print or to open in a spreadsheet:

- **Prophet 12 banks**: tick the banks to include. Only banks that have been read completely can be chosen.
- **The list in the File pane**: every patch in the list as it is shown, including changes not saved yet.

As a **PDF** it is A4, one bank per page in two columns: slot, name, Layer B, and a ★ for favourites. As **CSV** it has one row per patch with slot, name, Layer B, favourite and tags, for LibreOffice Calc, Excel or similar.

> **Example: a printed sheet of the User banks**
> 1. **Read all User banks**.

2. **File → Export bank sheet…**, keep **U1**–**U4** and **PDF, to print**, and click **Export…**.
3. Choose where to save it, then print the four pages.

4 Writing patches to the synth

4.1 Opening a file

Open .syx… (Ctrl+O) loads a `.syx` file into the File pane; **File → Open recent** lists the last eight files you opened or saved there. Files from this app, from the synth's own **Dump** functions and from other tools work. SysEx in the file that isn't a Prophet 12 program (for example the play lists in a **Dump All User Banks** file) is skipped and counted in the status bar.

4.2 Writing

There are two ways to write patches from the File pane:

- **Drag** them onto a slot in the Prophet 12 pane. Several patches fill consecutive slots from where you drop them; the slots that will be filled are outlined while you drag.
- Select them and click **Write to synth…**, then choose either the patches' **original locations** or **consecutive slots starting at** a slot you pick.

Either way, a confirmation lists every slot, the patch going there, and what it replaces. Nothing is written until you click **Write**.

Example: restoring one patch from a backup

1. **Open .syx…** and choose your backup.
2. Type part of the name in the File pane's search box, e.g. `strings`.
3. Select the patch and click **Write to synth…**.
4. Leave **Original locations** selected and check that the **Replaces** column shows what you expect.
5. Click **Write**. The status bar confirms that the patch was written and verified.

5 Editing patch lists

The File pane is also an editor for `.syx` files. It works without a synth, and nothing in it touches the synth until you write patches there.

- **File → New list** (Ctrl+N) starts an empty list.
- **Add patches** by dragging them from the Prophet 12 pane into the File pane, or right-click → **Add to the File list**. Drop them onto a patch to insert them before it, or below the last patch to add them at the end. **File → Add .syx file to the list…** adds a whole file's patches at the end.
- **Reorder** by dragging patches within the File pane. Hold Ctrl to copy them instead.
- **Remove** the selected patches with Delete, or right-click → **Remove from list**.
- **Renumber…** (right-click) gives the list, or the selected patches, consecutive User slots in list order, continuing into the next bank, for example to plan a bank before writing it with **Original locations**.
- **Rename…** renames patches in the list.
- **Save** (Ctrl+S) saves the list to its file, in the order shown; **File → Save list as…** (Ctrl+Shift+S) saves it under a new name. A restore point is never overwritten: saving one asks for a new name.

A * after the file name means the list has changes that aren't saved. Opening another file, starting a new list or closing the app then asks first. While a search is active, patches can't be dropped into the list.

5.1 Comparing the synth with a file

Click **Compare** above the File pane to add a Compare column to both panes. Patches are matched by **sound**, so a copy that was only renamed still counts as the same. Each patch shows one of:

- **same**: the same sound in the same slot (also when only the name differs, which is shown);
- **differs**: a different sound in that slot, and where the file's sound is, if anywhere;
- **elsewhere**: the sound is there, but in another slot;
- **not on the synth** / **not in the file**: the sound isn't on the other side at all;
- **slot not read**: that slot hasn't been read from the synth yet.

The File pane's toolbar sums it up, e.g. "92 same · 5 differ · 2 not on the synth". Read all banks first for a full comparison; until then, missing patches show as "not in the banks read".

Example: what changed since the last backup?

1. **Read all User banks**.
2. **Open .syx…** and choose the backup.
3. Click **Compare** and search `is:differs` in either pane.

5.2 Duplicates

A ×2 (or ×3, …) after a name means the same sound is in that pane more than once. Its tooltip says where. Search `is:dup` to list them all.

5.3 Rearranging patches on the synth

Drag a patch in the Prophet 12 pane onto another slot, and the app asks how to rearrange them:

- **Swap**: the two patches trade places. `U1:04` goes to `U4:50` and `U4:50` goes to `U1:04`. Several selected patches swap with the same number of consecutive slots from where you drop them, pair by pair; the two groups must not overlap.
- **Move and shift**: the patch goes to the slot you drop it on, and the patches in between move up or down one place to close the gap, as in a list. Several selected patches are gathered at the drop slot in slot order. The four User banks count as one long row of slots, so a patch moved from `U1:99` to `U2:01` pushes `U2:01` back into `U1:99`. Nothing is lost: the patches only change places.

The dialog lists what goes where and how long it takes. The app remembers your choice for the next drop. Either way it reads the slots that change, keeps them as a restore point, writes the patches and checks the result. **Undo** (Ctrl+Z) puts everything back. A long move rewrites many slots: moving a patch from `U1:01` to `U4:99` rewrites all 396, which takes a couple of minutes.

To drop in another bank, hold the dragged patch over the bank's tab for a moment; the tab opens.

To **copy** instead, hold Ctrl while dropping: the patch is written to the new slot and stays where it was. Patches from the Factory banks are always copied, since those banks can't be changed.

Example: moving a pad to the top of a bank

1. Read bank U1.
2. Drag `U1:37` onto `U1:01`, choose **Move and shift** and click **Move**.
3. The pad is now at `U1:01`, and the patches that were at `U1:01`–`U1:36` are one place further down, at `U1:02`–`U1:37`.

5.4 Audition

Double-click a patch, press Space, or select one and click **Audition**, to load it into the synth's **edit buffer**. The app then plays middle C (C3) for one second so you hear the patch straight away, and you can keep playing it from the keyboard. Nothing is stored: selecting a program on the synth loads its stored version again.

To step through a bank by ear, turn on **Audition each patch as you move with the arrow keys** in **File → Preferences…**: every ↑ or ↓ then loads and plays the next patch.

File → Preferences… (Ctrl+,) also sets the note, its length (0.25–4 s) and the MIDI channel, or turns the note off. The channel is normally **automatic**: the app asks the synth which channel it receives on, and uses channel 1 when it is set to All. The note stops early when you audition another patch, or when the app reads from or writes to the synth.

> **Important**
> Auditioning replaces the sound currently being edited on the synth. Save any unsaved edits on the synth first. To keep an auditioned sound, press **Write** on the synth and choose a slot.

6 Renaming

6.1 Rename

Select one or more patches and click **Rename…**, press F2, or right-click and choose **Rename…**.

Each patch gets an editable name field. Names can be up to 20 characters: letters (upper and lower case), digits, spaces and most punctuation. The synth can't show `* / : < > ? \ | ~`, so those are refused and the field turns red.

- In the **Prophet 12** pane, renaming stores the new name on the synth. Each slot is **read again first** and only its 20 name characters change, so edits you've made on the synth since reading it are kept. Factory patches can't be renamed; copy them to a User slot first.
- In the **File** pane, renaming changes the list in the app only. Save the patches or write them to the synth to keep the new names.

6.2 Cleanify names

Names typed on the synth are often in capitals. Select patches, right-click and choose **Cleanify names…**. The app suggests tidier names and shows them in the same dialog, where you can change any of them before confirming. Names that wouldn't change aren't listed.

Before	Suggested	Why
`BOF COOL PAD`	`Bof Cool Pad`	Each word in capitals gets one capital letter.
`DISTPAD`	`DistPad`	Run-together sound words are split.
`WARMSTRINGS`	`WarmStrings`	Plurals are recognised.
`LFO MADNESS`	`LFO Madness`	Known abbreviations (FM, LFO, PWM, EP…) stay in capitals.
`SUBBASS-DIST`	`SubBass-Dist`	Punctuation and digits are kept.
`Kepler`	`Kepler`	Words with lowercase letters are left alone.
`BOFJUPITER`	`Bofjupiter`	Unknown words can't be split; edit the suggestion.

Splitting uses a built-in list of about 100 sound words (pad, lead, bass, string, dist, warm…). Anything else is just given a capital letter, so check the suggestions before confirming.

7 Layer B

Each Prophet 12 program has two layers, A and B. Layer B is heard when **Split A | B** (B plays above the split point) or **Stack A + B** (both play together) is on; otherwise **Edit Layer B** on the synth plays it.

To put any patch into Layer B of a program, drag it onto the **Layer B** column of that program's row in the Prophet 12 pane. Only the Layer B cell is outlined while you hover over it. The dragged patch's Layer A sound and name become the program's Layer B; Layer A stays exactly as it is on the synth.

This is the synth's **Layers → Copy A -> B**, but with the sound taken from any patch. Dropping a patch onto its **own** Layer B does exactly what Copy A -> B does.

Example: a layered pad

1. Read bank `U1` and open a file with a string patch in the File pane.
2. Drag the string patch onto the **Layer B** cell of your pad, e.g. `U1:20`.
3. Confirm. The pad keeps its sound in Layer A and gets the strings in Layer B.
4. On the synth, select `U1:20` and press **Stack A + B** to hear both at once.

> **Note**
> Dropping a patch on the **rest** of a row writes the whole patch to that slot instead. Watch which part of the row is outlined before you let go.

8 Initializing slots

Right-click one or more User slots and choose **Initialize…** to replace them with the synth's Basic Patch, named **Init Patch**: a clean starting point for a new sound.

The first time, the app needs a copy of the Basic Patch, and the dialog explains how:

1. On the synth, press **Global** and go to **23 Basic Patch**.
2. Press **Write Now** (soft key 1). This loads the Basic Patch into the edit buffer, replacing any unsaved edits; nothing is stored.
3. Click **Capture template** in the app.

After that the template is remembered, and **Initialize…** goes straight to the confirmation. **Recapture template…** in the same dialog reads it again.

9 Searching

Type in the search box above either list, or press `Ctrl`+`F` to jump to the box of the pane you used last. `Esc` clears it.

- Upper and lower case don't matter. Several words must **all** match, in any order: `bof pad` finds "BOF COOL PAD".
- The slot counts too: `u2 pad` finds pads in bank U2.

- In the Prophet 12 pane the search covers **every bank that has been read**, not just the open tab. Each tab shows its number of matches; click a tab to clear the search and open that bank. Use **Read all User banks** first to search everything.
- `#pad` finds patches tagged Pad (see the next section). Tags and words can be combined: `#pad warm`.
- The ★ button next to the search box shows only favourites; it works together with the search.
- `is:dup` shows patches that are in the pane more than once (Section 5.2). `is:same`, `is:differs` and `is:missing` filter by how the synth and the File pane compare (Section 5.1).
- Matching letters are highlighted.

> **Note**
> Patches hidden by a search are never acted on, even if they were selected before. While searching, patches can't be dropped onto the Prophet 12 list, because the slots they would fill might be hidden. Dropping onto a Layer B cell still works.

10 Tags and favourites

Tags and favourites help you find patches again. They are kept by the app on your computer; nothing is stored on the synth or changed in your `.syx` files.

- **Favourites:** click the ☆ next to a patch, or select patches and press F. Press it again to remove the mark. The ★ button next to the search box shows only favourites.
- **Tags:** select one or more patches and press T, or right-click → **Tags…**. Type a tag and press Enter, or click a suggestion; Enter in the empty field saves. Spaces in a tag become hyphens, so `warm pad` is stored as `warm-pad`. With several patches selected, tags that only some of them have are shown dashed with a count; **+** gives all of them the tag.
- Tags appear after the name. Click one to show all patches with that tag, or search for it with `#tag`.

Tags and favourites belong to the **sound**, not to the slot or the name. They stay with a patch when it's swapped, moved, written to another slot, renamed, given a new Layer B, saved to a file or restored from a backup, and they show up on the same patch in any `.syx` file. Patches that sound exactly the same share them, so tagging one unedited "Basic Program" tags them all.

Editing a patch on the synth changes its sound. When the app reads that slot again (R, or reading the bank), it sees that the slot changed since it last read it and gives the new version the old one's tags and favourite mark; the old version keeps them too, for your backups. This works for slots read since the app was started; a change made while the app was closed gives an untagged patch.

11 Safety and backups

Every operation that changes the synth (writing, renaming, initializing and Layer B copies) works the same way:

1. The slots about to change are **read from the synth** and saved to a backup file.
2. The new programs are written, one at a time, with a short pause after each so the synth can store it.
3. Every slot is **read back** and compared with what was sent. A mismatch is reported, with the backup's location.

Backups go to `~/Documents/Prophet 12 Backups/` (on Windows: `Documents\Prophet 12 Backups`) and are named after what was done and when, e.g. `before-rename-20261008-210232.syx`. The app never deletes them unless you ask it to.

The Factory banks are never written: the app refuses before anything is sent.

11.1 Undo and restore points

Right after a write, swap, move, rename, initialize or Layer B copy, the status bar shows an **Undo** button (Ctrl+Z). It writes the previous contents back from the backup just made, with the same backup and verification, so the undo can itself be undone.

File → Restore points… lists every backup, newest first, e.g. *"10 Oct 2026, 20:32 — Before rename · U1:03–U1:07"*. Selecting one shows the slots it holds, the names they had then, and what is in those slots on the synth now. From there you can:

- **Restore** all its slots to the synth;
- **Open in File pane**, to restore only some of the patches by writing them as usual;
- **Show in folder** or **Delete** it (click **Delete** twice).

> **Example: undoing a rename from yesterday**
>
> 1. **File → Restore points…** and select the one marked *Before rename* at the right time.
> 2. Check the **Saved** and **On the synth now** columns.
> 3. Click **Restore**.

12 Settings

12.1 Zoom, font and theme

Ctrl++ or Ctrl + wheel up	Zoom in
Ctrl+- or Ctrl + wheel down	Zoom out
Ctrl+0	Actual size

The **View** menu has the same commands.

View → Appearance… (Ctrl+Shift+A) shows every theme as a small preview of the app and every font with a sample line. Click one to use it straight away. The same choices are in **View → Theme** and **View → Font**.

- **Themes:** System (light or dark, following the desktop), Dracula, Catppuccin Mocha, Catppuccin Latte (light), Tokyo Night, Gruvbox Dark, One Dark Pro, Rosé Pine, Kanagawa, Everforest Dark and Solarized Dark.
- **Fonts:** Fira Code, JetBrains Mono and Hack (Nerd Font versions, included with the app; an installed copy is used when present), or the system font.

Zoom (80–250%), font and theme are remembered.

12.2 Where the app keeps things

	Linux	Windows
Backups	`~/Documents/Prophet 12 Backups/`	`Documents\Prophet 12 Backups\`
Settings	`~/.config/app.p12librarian/settings.json`	`%APPDATA%\app.p12librarian\settings.json`
Tags and favourites	`~/.config/app.p12librarian/tags.json`	`%APPDATA%\app.p12librarian\tags.json`

	Linux	Windows
Init template	`~/.local/share/app.p12librarian/init-template.syx`	`%APPDATA%\app.p12librarian\init-template.syx`

13 Keyboard shortcuts

Keys	Action
Ctrl + N	New, empty list in the File pane
Ctrl + O	Open a `.syx` file
Ctrl + S	Save the File pane's list
Ctrl + Shift + S	Save the list under a new name
Ctrl + B	Back up all User banks
Ctrl + F	Search in the pane used last
F2	Rename the selected patches
F	Mark or unmark the selected patches as favourites
T	Tags for the selected patches
R	Prophet 12 pane: read the selected slots again
Space	Audition the selected patch
↑ ↓	Move through the list; with Shift, extend the selection
Home End	First or last patch
Page Up Page Down	Move a screenful
← →	Switch between the Prophet 12 pane and the File pane
Ctrl + Page Up Ctrl + Page Down	Previous or next bank
Ctrl + A	Select all patches in the pane
Enter	File pane: write the selected patches to the synth (asks first)
Delete	Prophet 12 pane: initialize the selected slots (asks first). File pane: remove them from the list
↓ in a search box	Jump to the first result
Ctrl + Z	Undo the last change on the synth
Ctrl + drag	Copy patches instead of moving them
Ctrl + ,	Preferences (audition note)
Ctrl + Shift + A	Appearance (theme and font)
F1	Open this guide
Ctrl + + Ctrl + - Ctrl + 0	Zoom in, out, actual size

Keys	Action
Esc	Close a dialog or menu; clear a search; clear the selection
Ctrl+Q	Quit

14 Troubleshooting

"The ports opened, but no Prophet 12 answered" Check **MIDI SysEx Enable** and **MIDI SysEx Cable** in the synth's Global menu (Section 2.2), and that the selected ports are the synth's.

Timeouts or failed verification Some MIDI interfaces corrupt long SysEx messages. Owners have reported problems with Native Instruments Maschine MIDI ports, Digidesign Mbox and CME U2MIDI interfaces. Connect the synth directly by USB.

The app doesn't find the synth after plugging it in Click ↻ next to the port lists.

Connecting fails on Windows although the synth is listed Another program is using the Prophet 12's MIDI port; Windows allows only one at a time. Close DAWs, editors and other MIDI software, then click **Connect**.

A patch sounds different after auditioning Auditioning replaced the edit buffer. Select the program on the synth again to reload it.

Layer B can't be heard Turn on **Split A | B** or **Stack A + B** on the synth, or press **Edit Layer B** to play Layer B alone.

15 Command-line tool

The Linux version also includes `p12`, a command-line version of the main functions, which is useful for scripts and scheduled backups. Run `p12 --help` for every option. It isn't part of the Windows version yet.

```
p12 ports                                    # list MIDI ports
p12 identify                                 # check the synth answers
p12 backup                                   # all User banks to one file
p12 backup -s U1,U2:10-20 -o mine.syx        # a selection
p12 backup -s user --split -o patches/       # one file per program
p12 edit-buffer -o current.syx               # the sound being edited
p12 list mine.syx                            # list a file (no synth needed)
p12 send mine.syx --select U1:05 --audition  # play without storing
p12 send mine.syx --select U1:05 --to U4:99 --verify
```

`send` backs up the slots it will overwrite, shows what will be replaced and asks before writing.

16 Technical notes

For the curious, and for anyone building on this work.

- **Messages** (manual, Appendix D): request a program with `F0 01 2A 05 <bank> <program> F7`; the reply and writes use `F0 01 2A 02 <bank> <program> <1171 bytes> F7`. The edit buffer uses types `06` (request) and `03` (dump). Banks 0–3 are U1–U4 and 4–7 are F1–F4; programs are 0–98.

- **Packing:** each program is 1024 bytes, sent as 1171 MIDI bytes: every 7 bytes travel as 8, with the top bits collected in a leading byte.
- **Layout** (found on the tested synth; not in the manual): Layer A is bytes 0–511 and Layer B bytes 512–1023. The name is at bytes 402–421 (Layer A) and 914–933 (Layer B), 20 characters padded with spaces. Byte positions are **not** the NRPN parameter numbers.
- **Copy A -> B** on the synth copies all 512 bytes of the layer. Bytes 23 and 293 of each layer hold the same value in both layers of every program examined, so they are kept when a layer is copied from another program.
- **Global Parameters dump**: the manual describes 50 nibbles, but the synth (OS 1.4.2.3) sends one byte per parameter, 26 bytes in all. The MIDI channel is the third byte (0 = All, 1–16).
- The **Device Inquiry** reply gives the OS version without its last part: OS 1.4.2.3 replies 1.4.2, which the header shows after connecting. The Global screen shows the full version.

17 About



P12 Librarian is made by **Qutrit** and **Claude** (Opus 5.5), an AI model made by Anthropic. Contact: qutritlogic@proton.me. Website, with new versions: qutrit.se/p12-librarian.

The **Prophet 12** was designed by **Dave Smith Instruments (DSI)**, founded by Dave Smith, and released in 2013. The company is now called Sequential: https://sequential.com/product/prophet-12-keyboard/.

This project is not affiliated with Sequential or Dave Smith Instruments. Prophet 12 is a trademark of its owner.

P12 Librarian is free and open source under the **MIT License**: you may use, change and share it, including for professional work. The source code is on GitHub: https://github.com/QutritLogic/P12-Librarian. The full licence, and the licences of the libraries, fonts and themes it includes, are under **Help → Licences**.

18 Disclaimer

1. Use at your own risk

P12 Librarian (the "Software") reads and writes data on a hardware synthesizer over MIDI System Exclusive (SysEx). Writing, initializing, renaming or copying programs replaces the data stored in the affected memory locations of the synthesizer. Defects in the Software, the computer, the operating system, USB or MIDI drivers, cables or interfaces, the synthesizer's firmware, interruptions of power or communication, or incorrect use can cause programs, banks, settings or other data on the synthesizer or the computer to be overwritten, corrupted or permanently lost, and may leave the instrument in a state that requires a factory reset, a firmware reinstallation or service. You use the Software entirely at your own risk.

2. No warranty

THE SOFTWARE IS PROVIDED "AS IS" AND "AS AVAILABLE", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, COMPATIBILITY AND NONINFRINGEMENT. The author does not warrant that the Software is free of errors, that it works with any particular synthesizer, operating system version or MIDI interface, or that its safety features (including automatic backups, read-back verification and confirmation prompts) will work, prevent data loss, or allow lost data to be recovered.

3. Your responsibility

Before using the Software, you are responsible for making and checking your own complete backups of all data on your synthesizer by means that do not depend on the Software. You alone decide whether and how to use the Software and are solely responsible for the results.

4. Limitation of liability

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL THE AUTHOR, COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO LOSS OR CORRUPTION OF DATA, PROGRAMS, SOUNDS OR SETTINGS, DAMAGE TO OR MALFUNCTION OF EQUIPMENT, COSTS OF REPAIR, RECOVERY OR REPLACEMENT, LOSS OF USE, LOSS OF PROFITS OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE), ARISING IN ANY WAY OUT OF OR IN CONNECTION WITH THE SOFTWARE OR ITS USE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5. Mandatory law and severability

Some jurisdictions do not allow certain warranties to be excluded or certain liability to be limited. In those jurisdictions the exclusions and limitations above apply to the fullest extent permitted by law, and nothing in this disclaimer limits liability that cannot lawfully be limited. If any provision of this disclaimer is held invalid or unenforceable, the remaining provisions remain in full force and effect.

6. Free of charge, non-commercial

The Software is provided free of charge, outside of any commercial activity, under the MIT License that comes with it (LICENSE). This disclaimer applies in addition to that licence. Copies built or distributed by others are not provided by the author.

7. No affiliation

The Software is an independent project. It is not made, endorsed, supported or approved by Sequential LLC or Dave Smith Instruments. Prophet 12 is a trademark of its owner. Do not contact the manufacturer for support with the Software.

8. Acceptance

By installing, running or otherwise using the Software you agree to this disclaimer. If you do not agree, do not install or use the Software.