

K-Setlist 0.5

Quick guide – QA bank management for Kurzweil keyboards

1 What the tool does

K-Setlist manages the Quick Access (QA) banks of a Kurzweil keyboard on a tablet or a computer, lists its Programs and Multis, sends Program Changes to the instrument over MIDI, and can read the instrument's objects back over MIDI without exporting a backup first. It is built for a tablet held in portrait and operated with a finger: every target is at least 44 px, nothing depends on hovering or right-clicking, and moves are done with the arrow buttons rather than by dragging.

It runs entirely in the browser. There is no account, no server and no upload - your objects stay in the browser's own database on the device.

Not in this version: setlists and songs (that is the next step), editing a loaded backup in place - the export always writes a fresh file holding QA banks only - and drag & drop reordering.

2 Running it

2.1 Browser

K-Setlist needs Web MIDI for everything that talks to the instrument, and that means Chrome or Edge - on Windows, macOS, Linux and Android. Safari has no Web MIDI on any platform, and on iOS and iPadOS every browser is Safari underneath, so installing Chrome there changes nothing.

On iOS the tool is still useful: build the banks and find the sounds here, and let the send buttons report the channel, Bank MSB/LSB and Program Change to be typed into a setlist app that can send them.

2.2 The three forms it comes in

Form	What it is	MIDI
Single file K-Setlist 0.5.html	One self-contained HTML file. Copy it anywhere, open it, done.	Only if it is served over https or http://localhost
PWA folder K-Setlist 0.5 pwa/	The same app plus a service worker and a manifest. Upload the folder to a web server, open it over https, and it installs to the home screen and starts without a network.	Yes
Sources in src/	index.html, app.css, app.js loaded separately - for development.	Only when served

⚠ MIDI needs a real address. A file opened from the file manager runs under file:// or content://, and a browser only stores a permission against a website - so the request is refused without even asking. http://localhost counts as secure; a plain http:// address on another machine in the wifi does not, and Web MIDI is missing there entirely.

The browser's database is tied to the address. The local file and the hosted copy are two different addresses, so they share nothing: keyboards, objects and banks set up in one are not in the other.

3 The screen

Where	What is there
Header	The version, Undo and Redo, the ACTIVE keyboard's name, the MIDI button and File. The keyboard button is where you read which instrument you are working on - tap it to switch or edit.
Tabs	PROGRAMS, MULTIS, QA BANKS. Switching keyboard switches all three.
Status bar	The bottom line. It reports what just happened, and for every object you tap it shows the MIDI numbers: ch 1 · MSB 0 · LSB 4 · PC 16.

4 Setting up a keyboard

Everything hangs off a keyboard, so this comes first. Tap the keyboard button in the header, then New keyboard.

Field	What to set
Name	Whatever you call it on stage.
Model	Every Kurzweil is offered, plus "Other". The model decides the file format, the bank ID limit and whether the QA tab is available at all.
MIDI output	Which MIDI Out of the interface this instrument hangs on. Leave it on Default for a single keyboard; with a two-port interface give each keyboard its own socket.
MIDI input	Only needed to read objects off the instrument (chapter 8). Sending never uses it.
MIDI channel	Set this to the instrument's Global ▶ Local Keyboard Channel.
Bank Select	Ctl 0/32, Ctl 0 or Ctl 32 - the same values the instrument's own Bank Select parameter has (Global ▶ MIDI). Set both to the same thing.
Program / Multi switch	How the app puts the instrument into the other mode: the Kurzweil CC 110-119 macro (default), custom MIDI bytes, or none.
Pause after the mode switch	120 ms by default. Raise it if the first tap only switches the mode and a second one is needed.

In the keyboard list, tap a row to make that keyboard active, use Edit to change it and the arrow buttons to put the list in the order you want.

Two keyboards may not share a MIDI channel on the same output - both would answer the same Program Change. On different outputs they may, which is what a two-port interface is for.

⚠ The Local Keyboard Channel is the panel channel, not an ordinary receive channel. A Program Change sent to it changes whichever channel the instrument is currently showing, which is what you want live - and a Multi can only be selected on it. Get this wrong and the symptom misleads: the mode switches to Multi and nothing else happens.

5 Getting the objects in

There are three sources, and they can be mixed.

Source	How	What you get
Factory objects	Always there, no action needed	The shipped object set of 16 models, so slot names resolve and the tool is useful with nothing loaded at all.
A backup file	File ▶ Open backup file...	IDs and names of Programs, Multis and QA banks, plus category and subcategory for Programs. The file is never modified, and the sample audio behind the object database is skipped - a 2.8 GB Forte backup costs 15 MB.
The instrument itself	File ▶ Read from the keyboard...	The same, over MIDI, with no export needed. See chapter 8.

A user object shadows the factory object of the same type and ID, which is what the instrument itself does. The Src column shows which is which: F on blue for factory, U for your own.

6 Programs and Multis

- Search by name or ID, filter Programs by category and subcategory.
- The source select switches between User + Factory, User only and Factory only.
- Tap a column heading to sort by it.
- The button on the right of a row sends that object to the active keyboard.
- Copy table puts the current list on the clipboard as TSV, ready for a spreadsheet.

7 QA banks

A QA bank holds ten slots, each pointing at a Program or a Multi, and the instrument recalls them with the numeric keypad. The tab is split: the banks on the left, the slots of the selected bank on the right. Hide folds the bank list away when the slots need the width.

7.1 Working with banks

Button	What it does
New	An empty bank on the next free ID.
Duplicate	A copy on the next free ID - and the way to take a factory bank into your own set, since factory banks themselves are never edited.
Rename / ID	Name (up to 16 characters, the Kurzweil field length) and bank ID. If the target ID is taken the two banks SWAP, rather than renumbering a whole chain.
↑ ↓	Move the selected banks in the list.
Delete	Removes the bank from the working set here.
Copy table	The bank list as TSV on the clipboard.

Button	What it does
All	Selects every bank of your own - the factory ones stay out, they are reference material. Pressing it again clears the selection.
Combine	Turns tapping into adding, so several banks can be selected at once. Their slots then form one continuous list and a move crosses bank borders.

Deleting a bank here does not delete it on the instrument, and loading the export cannot either: the export writes QA banks only, so it adds banks and overwrites ones of the same ID, but never removes one. Delete a bank on the keyboard itself, under GLOBAL ▸ OBJECTS ▸ QA BANKS (FAVORITES).

Factory banks are reference: they can be looked at and duplicated, never edited, and they are never exported. A bank of your own hides the factory bank on its ID without replacing it - delete yours and the factory one is back.

7.2 Working with slots

Button	What it does
Replace	Picks objects and writes them from the selected slot on, overwriting what is there.
Insert	Picks objects and pushes what is there down.
Cut	Takes the selected slots out and closes the gap - everything below moves up and the bank is padded back to ten slots.
Clear	Empties the selected slots and leaves the gap.
↑ ↓	Move the selected slots.
All / None	Take or clear the whole slot list.
Multi-select	Turns tapping into adding, the way Combine does for banks.

Replace and Insert open a picker with its own Programs / Multis tabs, a search box and a category filter. The order you tap the objects in is the order they are written in.

Cut and Clear carry the same red outline as Delete, because all three take content away.

7.3 The flags in the Src column

Flag	Means
P / M	The object type - Program (teal) or Multi (violet).
F / U	Where it came from - factory (blue) or your own.
Amber bar on the left	This row differs from what was last read in. It is a state, not a property, which is why no flag may take that colour.

8 Changes, Undo and the way back

The amber bar on a bank or slot row means it differs from what was last read in - from a file or from the instrument. It does not mean "unsaved": the working set lives in the database and survives closing the browser.

The mark also survives an export, on purpose - writing a file cannot know whether you then loaded it on the instrument. Reading in again is what sets a new starting point, which makes export → load → read back a way of checking that it arrived.

Way back	Reach	Survives a reload
Undo / Redo in the header	The last 30 edits, banks and slots alike. Loading a file, switching model or resetting clears the history.	No
File ▶ Changed QA banks, the undo button beside the count	The whole working set back to the state it was read in - changed banks exactly as they were, and banks made since then removed, because that is what "the state it was read in" means. It is one undo step.	Yes

Coming back the next day therefore shows the amber marks with an empty Undo history - the working set is kept, the history is not. The File sheet is the way out of that: it says how many banks have changed since the last read and carries the undo button beside the count.

9 Sending Program Changes

1. Tap the MIDI button in the header and press Connect. The browser may ask for permission.
2. Pick the default MIDI output. The sheet then lists which keyboard goes to which port.
3. Tap the send button at the right of any Program, Multi or QA slot row.

One tap puts that object on the active keyboard, preceded by the Program/Multi mode switch when the target needs one. The status bar shows the numbers that went out, whether or not anything could be sent - without an output the button stays live, is drawn with a dashed outline, and says why nothing went out.

The mode switch is only sent when the type actually changes, so a run of Multis costs two messages and no pause. If you change the mode at the front panel the app cannot know: tap the same send button again, or let 30 seconds pass, after which the next send carries the switch anyway.

⚠ If an object lands on the wrong sound, the Bank Select scheme is the first thing to check - the Program Change still arrives, on id mod 128, so the failure is quiet. If in doubt use Ctl 32: the bank of any Kurzweil object fits in the LSB, and at least one model (the Forte) goes wrong as soon as an MSB is sent at all. Always test with an object above 127 - below that the bank is 0 and every setting looks right.

10 Reading from the instrument

File ▶ Read from the keyboard... asks the instrument for its own objects, so no backup has to be exported and loaded at all. Nothing is lost this way: a QA bank comes back as the same 128 bytes a file holds, and a Program carries its category just as a file does. Nothing on the instrument is written to.

This needs two things the rest of the app does not: **SysEx**, which the browser treats as a permission of its own, and a **MIDI input** set on the keyboard - reading is a conversation, the instrument has to answer.

10.1 What to read

All three object types are ticked by default; untick what you do not need. The sheet estimates how long it will take, and a Stop button appears while it runs.

Categories	What it does	When
Only for what is new here (default)	A program that still has the same ID and the same name is the same program, so its category is already here. A first read pays for all of them, a repeat for almost none.	Normally
For every program	Fetches every category. About 45 ms per program - a PC4 with 2269 programs took 2 min 45 s this way, against 1 min with none.	When a category may have been changed at the instrument
None, names only	Names and IDs alone - the fastest check of "does what I have here still match the keyboard".	For a quick comparison

△ The trade with the default is that a category changed at the instrument on an otherwise untouched program goes unnoticed - nothing about it looks different from here.

10.2 What comes back

The result is always a comparison first: what the instrument has that is not here, what is named differently, which banks differ and in what. Taking it over is a second, separate press, so the usual question - does what I have here still match the keyboard - never costs you the categories a file gave you.

Banks are held against the merged list, factory included, because that is what the instrument reports; only the banks it calls the user's own are taken over. On take-over an object that kept both its ID and its name keeps its category.

11 Writing the banks to the instrument

File ► Export QA banks... writes a Kurzweil object file, named the way the target model needs it.

Setting	Choices
Which banks	Changed since the last read (the default - after a session of editing those are the banks worth carrying over), all of my banks, or the current selection. Only your own banks are ever written: a factory bank belongs to the instrument, which already has it.
Bank IDs in the file	Keep the IDs shown, or renumber consecutively from a starting ID.
File name	Free text. Before the PC4 the name is forced to 8.3 in upper case, which the sheet shows as you type. The extension must be upper case or the keyboard silently fails to offer every load option.

On the instrument, load the file with Fill from or Merge. A user QA bank shadows the factory bank of the same number without replacing it.

To be sure it arrived: export, load it on the instrument, then read back. The amber marks are cleared by the read, not by the export - so an empty set of marks after reading back is the confirmation.

12 Supported models

Group	Models
-------	--------

Group	Models
With QA bank editing, long file names	K2700, K2061/88, PC4, PC4 SE, SP8
With QA bank editing, 8.3 upper-case file names	Forte, Forte SE, PC3A, PC3K, PC3, PC3LE
Programs and Multis only	SP6, Artis, Artis SE (Favorites buttons, no banks) and SP4, SP5 (neither). The QA tab is off for them, with the reason on the tab.
Not offered	K2600, K2500, K2000 - they write the native PRAM/SROM container, which this parser does not read. A keyboard already set to one keeps its setting.
Name and MIDI channel only	"Other", for anything the list does not name.

13 If something does not work

Symptom	Cause and cure
Connect is refused without even asking	The page is not a website. Serve it over https or http://localhost and open it by that address.
MIDI was refused for this site	Tap the icon left of the address, Site settings, set MIDI to Allow, reload, then press Connect again.
No MIDI outputs are listed	Connect the interface, then reopen the MIDI sheet.
A named output sends nothing	It is not plugged in. A keyboard that names its own output does not fall back to the default - that would be another instrument.
The wrong sound comes up	The Bank Select scheme does not match the instrument's own parameter. See chapter 9.
The mode only switches on the second tap	Raise the pause after the mode switch in the keyboard editor - 90 ms was not enough on a Forte, 120 ms settled it.
A Multi does not switch	The MIDI channel is not the instrument's Local Keyboard Channel.
"The instrument did not answer"	Check that MIDI in and out both go to it and that it is not in a mode that ignores SysEx. If it stays silent, switch it off and on: an interrupted transfer can leave it waiting.
Amber marks with an empty Undo history	Normal after a reload - the working set is kept, the history is not. File ► Changed QA banks is the way back - the undo button beside the count.

14 Numbers worth knowing

What	Value
Slots per QA bank	10
QA bank IDs	0-99 on the OS4 instruments, which refuse anything outside that range. The PC3 generation inherited Quick Access without the limit.
Bank name length	16 printable characters

What	Value
Undo history	30 edits, not carried across a page reload
Object number	bank $\times$ 128 + Program Change, 0-based - the PC value IS the object number, never +1
Factory sets shipped	16 models