Konfyt

Release 1.2.2

Gideon van der Kolf

Jan 21, 2023

CONTENTS

Konfyt - Digital Keyboard Workstation for Linux

CHAPTER ONE

INTRODUCTION

1.1 Background

Konfyt is a digital keyboard workstation for Linux. It allows you to instantly switch between patches for live keyboard playing. Each patch can contain multiple layers of soundfont (SF2 or SFZ) instruments, as well as audio input ports from, and MIDI output ports to other apps or hardware. Konfyt also has a searchable library which scans your filesystem for instruments and lets you easily browse and search soundfont contents.

Konfyt originally started out in 2014 as a library for soundfonts for quick viewing of contents since the existing methods was frustrating. From there it got SFZ support and eventually grew into a fully fledged live keyboard playing tool (“workstation”).

1.2 Concept

Konfyt is a stand-alone application that uses JACK for audio and MIDI input and output. Konfyt hosts SF2 and SFZ instruments and does audio and MIDI routing. Additional tasks such as effects and synth plugins must be done outside Konfyt by other software. Konfyt is designed to easily integrate into the Linux pro audio world. It allows you to use all your other JACK aware software while filling the gap of routing audio and MIDI in a very user friendly way with instant patch switching and other useful features geared towards live keyboard playing.

1.3 Features

- Scan filesystem for soundfonts and SFZs and organize it into a tree view for easy navigation.
- List soundfont contents (programs) by simply selecting a soundfont in the library.
- Search library for SFZs and soundfonts, including soundfont programs.
- Uninterrupted patch switching - when holding notes or sustain while switching patches, sound keeps on playing until release.
- Audio input ports, which can be inserted into patches as layers, allowing audio to be received from other applications or from system input.
- MIDI output ports, which can be inserted into patches as layers, allowing MIDI output to other applications or to hardware.
- Audio output busses, allowing the output of SFZ, soundfont and audio-in layers to be sent to different destinations.
- Port connections to other JACK clients are automatically persisted.
- Per layer MIDI filter including key zone, transpose, velocity filtering and MIDI CC filtering.

- Assign MIDI triggers to actions such as switching patches and modifying layer volume, mute and solo.
- MIDI events (including Sysex messages) can automatically be sent to an output port when a patch is activated - useful to change programs and modes on an external synthesizer.
- Per project external application launcher. Add commands and arguments for all external applications associated with your project to a list to easily launch them. Application files can be saved to the project directory and referenced relative to the directory, allowing for everything contained in a single directory and remaining in tact when the directory is moved.
- Filesystem browser to quickly load SFZs and soundfonts from directories not in the library.
- Live mode in which some actions can be controlled with the computer keyboard.
- Panic button which, when activated, mutes all sounds, stops all MIDI and emits MIDI all-notes-off messages as well as pitchbend zero and sustain pedal zero.
- Scalable GUI which fits comfortably on a 1366x768 laptop display.

CHAPTER

TWO

USAGE

2.1 Settings

It is a good idea to fill in all the options in the settings screen.

2.1.1 File Manager

The file manager entry is optional and is provided in case your default file manager doesn't play along well. You can test it by right-clicking on one of the library items and clicking *Open in File Manager*. If you receive an error message from your file manager, specify your preferred file manager in settings (e.g. either dolphin, nautilus, thunar, pcmanfm, etc.).

2.1.2 Projects Directory

This is the default directory for projects. This directory is scanned at startup and all the projects are listed in the projects Open menu for easy access. When saving a project you can seamlessly save it in this directory by accepting the proposed path (or choose No to select a different path).

2.1.3 Scanning

The *Apply and Rescan* button clears the database and scans the specified directories and subdirectories for SFZs, GIGs, soundfonts (sf2) and patches. This may take a while, especially with big soundfonts as each soundfont is loaded with Fluidsynth in order to extract its program information.

The *Apply and Quick Scan* button leaves the soundfonts already in the database intact and only scans and loads new files not yet in the database.

2.2 Library

In order to fill the library with instruments (patches, SFZs and sf2 soundfonts), a scan must first be performed from the settings screen.

The library organizes sounds using subdirectories. Thus, for easy navigation it is useful to sort your sounds into subfolders, for example using categories like Piano, Strings, Organs, Pads, etc.

When selecting a soundfont, the list of programs appear below the library tree. By double-clicking on a soundfont program or a SFZ or patch in the tree, the sound is added as a layer to the current patch.

When searching, file names and program names within soundfonts are searched.

2.3 Projects

A Konfyt project contains the following:

- Patches
- Ports (MIDI and audio input and output ports) with corresponding connections
- Audio input ports and output buses
- List of external applications
- Triggers - actions that are triggered by MIDI events
- List of "Other JACK Connections" - pairs of ports not belonging to Konfyt that will be persistently kept connected

Each project is saved in its own directory. A project directory contains the project file (an XML file with a name ending with ".konfytproject") and a "patches" folder containing the patches.

2.4 Ports and Buses

Konfyt uses ports and buses to receive and send audio and MIDI. A bus is actually just an audio output port. Each port and bus directly relates to a JACK port and can be connected to other JACK ports (i.e. other applications or hardware input/output).

Any port connections specified within Konfyt are persistently maintained, even after other applications restart, hardware is reconnected or when disconnects are attempted from outside Konfyt.

2.4.1 MIDI Input Ports

A MIDI input port is how Konfyt receives MIDI input from a hardware MIDI controller (e.g. keyboard) or other applications. The MIDI input is routed to one or more instrument layers inside patches. MIDI input ports are set up in the *Ports and Buses* screen.

A MIDI input port also has its own MIDI filter, which filters all MIDI before it reaches any patches/layers. When a MIDI input port is selected on the *Ports and Buses* screen, a button appears below the connections list to access the MIDI filter settings.

2.4.2 Audio Bus (Output Port)

A bus is an audio output port which relates to 2 JACK audio ports (left and right). Audio buses output audio to other applications or to the system output. Buses are set up in the *Ports and Buses* screen. Each instrument layer outputs to a specific bus. Although a new project contains a default *Master Bus*, this is merely a name and there is nothing special about it.

An audio bus can be set to ignore (bypass) the master output volume. When an audio bus is selected on the *Ports and Buses* screen, a checkbox appears below the connections list to enable this. This is ideal for when you want to send some audio to another application for applying effects, and loop it back into Konfyt (return) for further routing. By enabling this option on the "send" bus, the audio level isn't reduced by the master volume setting, preventing it from being reduced multiple times (each time the signal leaves Konfyt).

2.4.3 MIDI Output Ports

A MIDI output port sends MIDI out from Konfyt to another application or hardware. To use a MIDI output port, it is added as a layer in a patch. Thus, MIDI will only be sent to the port when a patch is active with that contains a layer corresponding to the port. As a layer, it has its own MIDI filter, output destination port and output MIDI channel specific to that layer (i.e. it can be different for each patch).

2.4.4 Audio Input Ports

An audio input port receives audio from other applications or hardware and routes it to a bus. To use an audio input port, it is added as a layer in a patch. Thus, audio will only be routed from an audio input port when a patch is active that contains a layer corresponding to that port. As a layer, it has its own gain slider and destination bus specific to that layer (i.e. it can be different for each patch).

2.5 MIDI Triggers

Konfyt allows MIDI triggers (similar to “MIDI learn” in other applications) to be set up in order to perform an action when the specified MIDI message is received. MIDI triggers are set up in the *Triggers* screen. Received MIDI messages are shown on the right and can be assigned to actions on the left by using the *Assign* button or by selecting the event and double-clicking on the action (or vice versa).

When bank select messages (CC0 and CC32) are directly followed by a program change message, the messages are grouped together as one bank and program message.

MIDI message values are interpreted corresponding to the action it is assigned. For actions such as gain, the absolute MIDI value is used. For other actions such as patch switching or toggling a setting, the action is only triggered if the MIDI message value is larger than zero. This eliminates double triggering when a button or note sends both 127 and zero when pressed.

MIDI events from all MIDI input ports are used for triggers.

Triggers that relate to patch layers will only affect the layers of the currently active patch.

In addition to actions/triggers, the *Triggers* screen contains additional options that are saved in the project:

- *Program Change messages with no bank select switch patches*: When this is enabled, if a program change MIDI message is received on its own (without any preceding bank select messages), the patch will be changed based on the message program number.
- *Slider MIDI Pickup Range* specifies the range in which sliders will jump to the value of received MIDI messages. A maximum range of 127 means that a slider will jump to any received MIDI message value. Lower ranges mean the slider will only jump to (“pickup”) the value when the difference between the slider value and the received MIDI message value is in the range.

2.6 Patch/layers View

The main patch/layers view shows the currently active patch. Normally, audio and MIDI will only play from the currently active (currently visible) patch. Switching to another patch will deactivate the patch and activate the other patch. When switching away from a patch, audio output ports are faded out so a sudden audio drop isn’t heard. For instrument layers with MIDI input and audio output, their sound will continue to be heard if notes or a sustain pedal are being held down. When the notes or sustain pedal are lifted, the appropriate messages are passed to the inactive patch.

A patch can be set as *always active* from the *Patch Menu*. This will keep the patch active even when you switch to other patches.

Below the layers section is a *Note* section in which you can place text that will be saved with the patch.

MIDI output port layers have a *MIDI Send List*, which is a list of MIDI messages which will be sent out on the layer whenever the patch is activated. This is useful for switching programs on hardware when activating a patch.

2.7 Other JACK Connections

In the *Other Jack Connections* screen, JACK audio and MIDI ports not belonging to Konfyt can be connected to each other. Just like with Konfyt ports, these connections are persistently maintianed by Konfyt, even when another app tries to disconnect it or after other apps restart or hardware is reconnected.

2.8 External Applications

The *External Applications* list is a simple list that allows you to store commands in a project for easily running external applications. It can be used as a simple crude way to restore your session of applications every time you open the project.

The string `$PROJ_DIR$` will be replaced by the current project directory. This can be used to make a project more self-contained by storing files used by external applications in the Konfyt project directory and using the string in arguments for the commands.

For instance, you may use Carla to host plugins:

> For synth plugins, MIDI output port layers send MIDI data from Konfyt to Carla and the synth audio is sent back to Konfyt using audio input port layers. For effect plugins, a Konfyt bus sends audio to the plugin.
>
> The external application command `carla /path/to/my_carla_project.carxp` is added to the *External Applications* list in order to quickly launch Carla next time with the correct project. However, another option is to save the Carla project file in your Konfyt project directory. Then, go to the *Filesystem* tab on the left-hand side of the Konfyt window, click on the *Project Directory* button, right-click on the Carla project file and select *Add Path to External App Box (Relative to Project)*. This adds the path of the Carla project file to the *External Applications* text box using the string `$PROJ_DIR$` to indicate that the file is relative to the Konfyt project directory. Just add "carla " in front of the path so that it becomes `carla "$PROJ_DIR$/my_carla_project.carxp"` and click *Add* to add it to the *External Applications* list.

CHAPTER

THREE

SOUND ENGINES

Konfyt fundamentally uses JACK (the JACK Audio Connection Kit) for MIDI and audio.

Fluidsynth is used to load, inspect and play sf2 soundfont files.

Konfyt initially used only Carla for handling SFZ instruments. At the time, Carla still used Linuxsampler as a backend for playing SFZs. However, the Linuxsampler support became troublesome and (among other reasons) Carla moved to SFZero for handling SFZ. Finding the SFZ support of SFZero lacking, I implemented Linuxsampler support using the LSCP (Linuxsampler Control Protocol) library.

Currently Konfyt supports both Carla and Linuxsampler backends for SFZ. Linuxsampler is used by default if no command-line arguments are specified.

3.1 Linuxsampler

LSCP is used to communicate with Linuxsampler. At startup, if LSCP fails to initialise, it is assumed that Linuxsampler is not running and a Linuxsampler process is started. Then LSCP initialisation is retried.

Linuxsampler audio and MIDI JACK engines are created. The JACK client name is derived from Konfyt's JACK client name. If Linuxsampler engines with the same name already exists, it is assumed to be from a previous Konfyt instance that is no longer running and the engines and related SFZ instruments are removed before new ones are created for the current session.

Multiple instances of Konfyt can coexist since JACK requires that each client have a unique name.

When a SFZ layer is added in Konfyt, the instrument is loaded in Linuxsampler via LSCP. A MIDI port and two audio ports are created for the instrument in Linuxsampler. Konfyt also creates MIDI and audio ports specifically for the SFZ layer and automatically connects these to the Linuxsampler ports in order to transfer MIDI and audio between Konfyt and Linuxsampler.

3.2 Carla

If the user wants to use Carla for SFZ support, the appropriate command-line argument must be passed to Konfyt at startup. Use `--help` for more details.

The Carla backend runs inside Konfyt but MIDI and audio are exchanged similar to with Linuxsampler, using JACK ports.

CHAPTER

FOUR

USE CASES

4.1 The first steps

Perform these steps to set up Konfyt and get going:

1. Specify paths in Settings and let your instrument directories be scanned to fill the library.
2. In the *Ports and Busses* screen, assign a client to the MIDI input port. Access to ALSA hardware can be gained by using the command `a2jmidid -e -u` It is also useful to add this command to the *External Applications* list so it can easily be run next time.
3. Test MIDI input by watching the MIDI receive indicator at the top right or by selecting *Show MIDI Messages* at the console and watching the console.
4. In the *Ports and Busses* screen, connect the audio bus(es) to system output or JACK clients.
5. Add a SFZ or soundfont layer to the patch and test your audio.
6. Remember to enter a project name and save often!

4.2 Example with ZynAddSubFX and Ardour

Konfyt doesn't attempt to do everything itself but works in harmony with other JACK aware applications. This example shows how Konfyt can be used in combination with a stand-alone synth application and a plugin host.

1. Create a new Konfyt project, connect the MIDI input port, verify that you receive MIDI and save the project.

   Take note of the project path. If you forgot, you can quickly find it by clicking the *Project Directory* button in the *Filesystem* tab on the left-hand side of the Konfyt window.
2. Launch ZynAddSubFX and select or create a desired preset.
3. Save your ZynAddSubFX session (File->Save All Parameters…) to your Konfyt project directory as "zyn_lead.xmz" for example.
4. In Konfyt, navigate to the saved file in the *Filesystem* tab, right-click on it and select *Add Path To External App Box (Relative to Project)*.

   The *External Applications* text box should now contain something like "$PROJ_DIR$/zyn_lead.xmz".
5. Add "zynaddsubfx -l " to the beginning of the text. The -l argument tells Zyn you want to load the file.
6. Now, in the event that you might use another instance of ZynAddSubFX, you want the JACK port names of this instance to be uniquely identifiable and remain the same across sessions. To ensure this, add " -N zyn_lead" at the end of the command. This makes ZynAddSubFX append the "zyn_lead" string to its port names.

   The command should now look like this: `zynaddsubfx -l "$PROJ_DIR$/zyn_lead.xmz" -N zyn_lead`

7. Click *Add* to add it to the list.
8. Close ZynAddSubFX. Then double-click on the newly added entry in the *External Applications* list to test it. ZynAddSubFX should start up with the appropriate settings loaded from the file.
9. In the *Ports and Busses* screen, add a new MIDI output port, name it “zyn lead send” and connect it to the ZynAddSubFX midi_input port.
10. Add an audio input port, name it “zyn lead return” and connect it to the ZynAddSubFX out_1 and out_2 ports.
11. Create a new patch and add the newly created MIDI output port and audio input port as layers. When you play notes, MIDI is now sent to Zyn and the audio is returned to Konfyt. The audio is routed to the output bus specified in the audio input port layer.
12. Now, we might want to apply some effects like reverb and EQ to the Zyn audio. Open a plugin host, for example Ardour. Create a new project, saving it to the Konfyt project directory and adding it as an external application command like before. (Note that the -l and -N arguments that were used above are specific to Zyn and not applicable to Ardour.)
13. In Ardour, create an audio bus and add reverb and/or other effects plugins to it.
14. In Konfyt, in the *Ports and Busses* screen, add a bus, naming it “Ardour Effects” for example and selecting the Ardour audio bus/track as its clients to connect to.
15. In the patch, on the audio input port layer, click the *Bus* button on the right hand side and select the newly created bus. Now, the audio from Zyn will be routed to Ardour for some cool effects.

Remember to save your Konfyt project regularly.

When done, close Konfyt, Ardour and Zyn, re-open Konfyt, launch Ardour and Zyn from the *External Applications* list and test that everything still works. If everything was set up correctly, the JACK port connections should automatically be connected and everything should work as before.

4.3 More ideas

Some applications that might augment your workflow with Konfyt are:

- Carla, Ardour and other plugin hosts.
- mididings, QMidiRoute for more advanced MIDI filtering and routing.
- a2jmidid with the `-e` argument to get access to ALSA hardware and software MIDI ports and the `-u` argument to remove ALSA port numbers from port names.

CHAPTER

FIVE

KNOWN ISSUES

5.1 MIDI Triggers not working when screen locked

2021-08-23, Linux Mint 20.2, Xfce, Light Locker, Qt 5.12.8

When the screen is locked, MIDI triggers do not work and the actions are only performed once the screen is unlocked.

This is due to a Qt bug causing the GUI and timers to stop while the screen is locked.

A workaround is to set the following environment variable before running Konfyt:
`export QT_XCB_GL_INTEGRATION=none`

Discussions on this Qt bug can be found here:

- https://bugreports.qt.io/browse/QTBUG-47179?attachmentViewMode=list
- https://bugs.freedesktop.org/show_bug.cgi?id=91448
- https://gitlab.freedesktop.org/xorg/xserver/-/issues/206
- https://gitlab.freedesktop.org/xorg/xserver/-/issues/497

5.2 No sound, JACK complains that it can't allocate memory, crash (pre v1.2.2)

If the following is printed when running Konfyt from a console:
`Cannot lock down 82280346 byte memory area (Cannot allocate memory)`
along with potentially other memory or realtime related messages, your system may not be configured correctly for realtime audio work.

Two probable culprits are that your user may not be in the `audio` group, and that your `ulimits` file may not be set up correctly.

The rtcqs script from https://codeberg.org/rtcqs/rtcqs can be used to check whether your system is set up properly. Follow the guidelines in the script (especially related to the above mentioned) to make the required changes.

CHAPTER SIX

WHAT'S NEW

6.1 [1.2.2] - January 2023

Added

- Update MIDI filter immediately while editing it in the GUI editor.

Fixes

- Fix library item double-click working intermittently on some systems.
- Destination audio port null check to prevent crash when JACK/system is not configured correctly. (When JACK can't allocate memory due to user not belonging to audio group or ulimits not being set up correctly. Note that in this case the audio system is not functioning correctly in any case.)

Changes

- About dialog is now integrated into the main window.
- MIDI filter editor size constraints and scrollability removed for hopefully better layout across different systems.

6.2 [1.2.1] - October 2022

Added

- Checkbox in filesystem view to show hidden files.
- Add dropdown menu to filesystem view path text box with common paths and remove home and current project buttons.
- Warn when running an external app with a project directory replacement tag in the command if the project has not been saved yet.

Fixes

- Remove dot at end of project folder names when saved to default project location.
- Use the correct project directory when running an external app that was added before the project was saved.

Changes

- Insert new patches after currently selected patch when adding new or copying a patch.
- Show a menu when the remove patch button is clicked to serve as a confirmation.
- Print slightly less alarming message in console when MIDI map presets file or settings file does not exist.

- When running an app from the external app list, print the expanded name if it contains a $PROJ_DIR$ and print a message when an external app stops.

6.3 [1.2.0] - July 2022

Added

- Patch list drag and drop for re-ordering patches.
- Detect Linuxsampler process crash and automatically restart.
- Soundfont scanning for library is now done in a separate process so Fluidsynth crashes due to malformed soundfonts don't affect Konfyt.
- Pitchbend up and down ranges in MIDI filter.
- Add CC block-list to MIDI filter.
- Note velocity mapping with graph and presets in MIDI filter.
- Escape key returns to main patch view.
- Library context menu to remove patch from library.
- When a patch is selected in the library, show info and layers below library.
- Add liblscp version to about text.

Fixes

- Apply patch list number/note show/hide setting when loading a project.
- Change volume when master volume slider is moved without using the mouse.
- Only relay noteoff messages to GUI (e.g. in console or received MIDI events lists) if it actually resulted in a noteoff message being sent.
- MIDI send list editor now only shows received MIDI events for the corresponding layer.

Changes

- When console/main window not available any more (e.g. during program exit), send print output to stdout.
- CC allow-list in MIDI filter is now space-separated text.
- Entire top toolbar area becomes red to emphasise when Panic is activated.
- Patches in library view don't show file extension any more.
- Some code housekeeping and refactoring.

Removed

- MIDI filter note velocity limits - replaced by velocity mapping.

6.4 [1.1.7] - January 2022

Fixes

- Set QT_XCB_GL_INTEGRATION=none environment variable at startup to prevent event loop from stopping when screen is locked on some systems which caused some functionality like MIDI triggers and port reconnections to stop working.
- Fix .gig file loading.

6.5 [1.1.6] - October 2021

Added

- Filesystem view now supports previewing of instruments like the library.
- Global transpose keyboard shortcuts added to live mode.

Fixes

- Pitchbend now works if Carla backend is used to play SFZ files.
- Fix SF2 loading from command-line arguments.
- Fix global volume up action getting stuck due to rounding errors.

Changes

- Some GUI tweaks in the patch layer view and library/filesystem preview.
- Misc code refactoring, cleanup and fixes.

6.6 [1.1.5] - July 2021

Added

- Ignore global transpose option in layer MIDI filter
- Ignore global volume option for buses (checkbox on Ports and Buses screen)

6.7 [1.1.4] - April 2021

Added

- Basic MIDI pickup for layer and master gain sliders. Range setting can be set in the Triggers page.

6.8 [1.1.3] - January 2021

Added

- Per-layer audio output indicator
- Two Konfyt instances can now run in parallel, nicely sharing Linuxsampler
- Orphaned Linuxsampler channels and devices (due to previous app instance crash) are now cleaned up at startup
- MIDI output port layer output (right-hand side) button now has link to port connections
- Xrun messages show time since last xrun
- Patch layers can now be reordered

Removed

- Removed multi-project tabs. They were rarely (never?) used, buggy and resulted in unnecessary maintenance. Multiple app instances can be used instead.

Fixes

- Fix crash when loading sfz file if Linuxsampler is not installed
- Fix copying a patch pointing to original patch layers
- Fix always-active patch volume not changed by global volume when other patch is active
- SFZ engine startup delay handled better, fixes loading sfz files through command-line when Linuxsampler is not running yet
- Fix CC value not loaded correctly for MIDI Send Events

Changed

- Patch list behaviour and interaction improved
- Code cleanup and organisation, largely concerning the database
- Window title is now Patch - Project [Jack Client Name]
- Sustain indication in GUI has threshold of 64
- Bank/program messages received now handled per port and channel
- MIDI filter screen last received messages only shown from associated port/layer

6.9 [1.1.2] - October 2020

Added

- Per-layer MIDI indicators (activity, sustain and pitchbend non-zero).
- Global sustain and pitchbend non-zero indicators.

Changed

- Fairly big code refactor under the hood, should not be noticeable to users.
- Fix bug of empty MIDI input ports added after removal of non-last port while console MIDI output is enabled.
- Small GUI tweaks.

6.10 [1.1.1] - October 2020

Added

- Preview mode options: MIDI input port and channel, output bus.
- MIDI input port and bus menus now have a link to the connections page.
- Save-as finally implemented.
- Compatible with Carla 2.2.0 (API change)

Changed

- Default global and layer gain (volume) is now 1.0.
- SF3 soundfonts are now detected and loaded.
- Current setting is marked in MIDI channel/port/bus menus
- Small GUI tweaks.

6.11 [1.1.0] - December 2019

- Each patch can be set to send MIDI events to other apps/hardware when activated and sysex messages are also now supported. MIDI send events can be saved for reuse in other projects.
- Fluidsynth 2 is also now supported.
- Config files are now saved to standard XDG directories.
- Carla support is now optional.

6.12 [1.1.0-beta] - October 2019

- MIDI send events: each MIDI output layer can have MIDI send events associated with it which are sent when the patch is activated. MIDI send events can also be saved for future use in other projects.
- MIDI note names octave numbers changed to the IPN standard to match most other software numbering.
- Always-active option for patches: a patch can be set to always be active, even if it is not the current patch.
- GUI uses system specified font size and resizes properly based on font size.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- search