



# **Mixxx User Manual**

*Vydání 2.3.0*

**The Mixxx Development Team**

10.01.2022

---

## Contents

---

---

## Introduction to Mixxx

---

Mixxx enables you to perform live DJ mixes with your digital music collection. It helps you rock the party with DJ controllers, vinyl turntables, or even just your keyboard. It is developed by an international team of volunteers who want to bring the joy of DJing to everyone. The project is non-profit, open-source and community driven. Together, we aim to build the best DJ software ever created. We hope you will *join us!*

### 1.1 How Mixxx Works

Mixxx is different from typical music player applications because it plays multiple audio files at the same time and has many features to manipulate the playback of the audio files. This lets you creatively mix different tracks together on the fly.

You can use Mixxx to preview the next track in headphones before the audience hears it, also known as headphone cueing. This helps you choose a track that is appropriate for the present moment and that will mix well into the currently playing track(s). To use headphone cueing, you need at least 2 separate audio outputs, typically provided by a USB audio interface (also known as a “sound card”, although most of them are not shaped like cards anymore).

The *shortcut* for headphone cueing in Deck1 is T while for Deck 2 the shortcut is Y.

Mixxx can be used in two different ways for headphone cueing:

#### **Internal Mixing**

Mixxx plays multiple audio files on the computer at the same time, mixes them together, and sends the mixed signal to one audio interface output. A separate signal is sent to another audio interface output for headphone cueing. Often a DJ controller is used with internal mixing to provide easier control over Mixxx than a mouse and keyboard.

#### **External Mixing**

In this kind of setup, Mixxx plays multiple audio files on the computer at the same time and sends each track to a separate audio interface output. The audio interface’s outputs are plugged into an external hardware mixer. The hardware mixer performs the actual mixing of the audio signals. On the hardware mixer, there are separate outputs for the mixed signal to play to the audience and for the DJ to use headphone cueing.

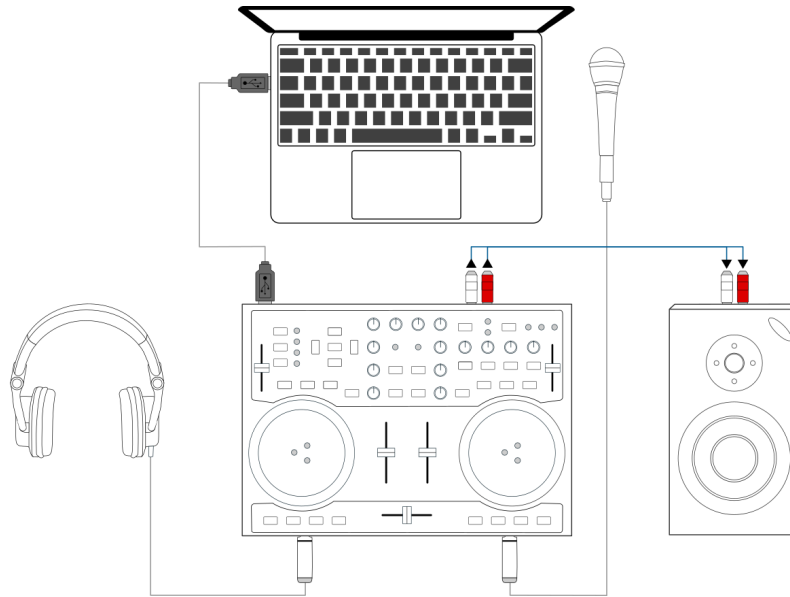


Figure1: Using Mixxx together with a DJ controller for internal mixing. The DJ controller has an integrated audio interface that provides two separate stereo outputs.

External mixing is typically used with turntables and *special vinyl records* to manipulate digital music files as if the music was on the vinyl records.

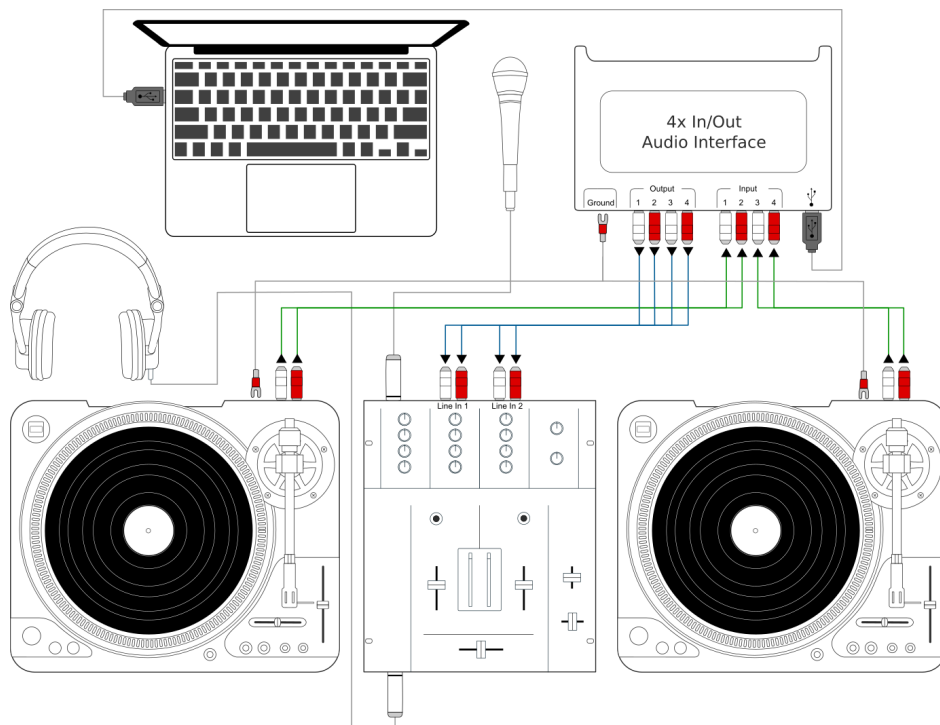


Figure2: Using Mixxx together with a USB audio interface, external mixer, and timecode vinyl for external mixing

### 2.1 Installing Mixxx

After downloading Mixxx from our Website, you need to install it on your computer.

**Windows** Double-click the Mixxx installation file and go through the installation wizard.

**macOS** Double-click the Mixxx .dmg file and drag the Mixxx icon to your Applications folder.

**GNU/Linux** No steps are necessary after you follow the instructions on the download page.

### 2.2 Upgrading Mixxx

You may safely install Mixxx on top of any previous version. You don't need to uninstall the old version.

Your settings and library will not be removed or reset, but they may be upgraded when you launch the new version of Mixxx. Note that we do not guarantee that new configurations will work on older versions of Mixxx, therefore you may want to make a backup of your existing settings just in case.

**Windows**

Mixxx settings are in `%LOCALAPPDATA%\Mixxx`, which is usually something like `c:\Users\\AppData\Local\Mixxx`. If you have trouble finding this folder, see [these instructions](#).

**macOS** Mixxx settings are in `/Users/<username>/Library/Containers/org.mixxx.mixxx/Data/Library/Application Support/Mixxx`

**GNU/Linux** Mixxx settings are in `/home/<username>/.mixxx`

Copy this directory to a backup drive or somewhere else on your system to make a backup of your settings.

**Viz také:**

For detailed descriptions of the settings files, go to [The Mixxx Settings Directory](#).

## 2.3 Opening Mixxx



Once you've installed Mixxx, start by opening Mixxx and importing your music to the Mixxx library.

**Windows** Double-click the Mixxx icon on the Desktop. Alternatively, browse your Windows start menu and click the Mixxx icon, or perform a search for `Mixxx.exe`.

**macOS** Double-click the Mixxx icon in the Applications folder. Alternatively, drag the Mixxx icon to the dock and double-click it there or search for `mixxx` in Spotlight.

**GNU/Linux** Click the Mixxx icon in the applications menu or launcher of your desktop environment or perform a search for `mixxx`. Alternatively type `mixxx` into the terminal, then hit Return. If your *audio interface* is not accessible from Mixxx because it is used by other applications via PulseAudio, you may close and restart Mixxx using `pasuspender mixxx`. See *Launching Mixxx from the Command Line* for an overview of Mixxx's command line options.

## 2.4 Importing your audio files

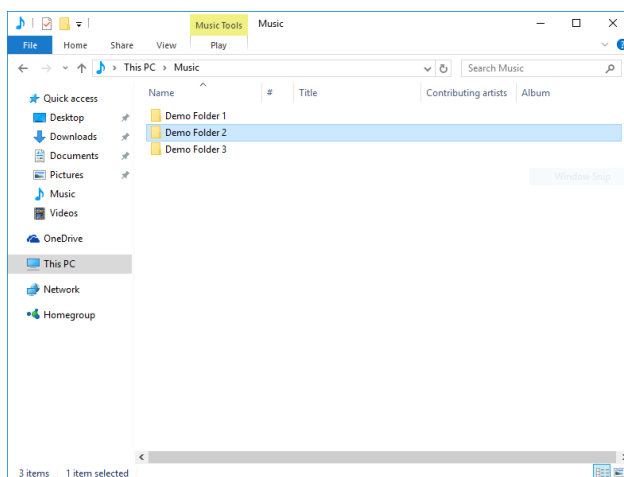


Figure1: Mixxx running on Windows 10 - Choose music library directory dialog

The first time you run Mixxx, you are asked to select a directory where your music is stored. By default, the dialog points to a location where music files are typically held, but you can select any location on your computer.

Click *Select Folder* and Mixxx will scan your music library. Depending on the size of your library this could take some time. All the supported music files Mixxx detects will be listed in the *Tracks - View and edit your whole collection*.

If you want to refresh your library (for example because you added or moved files), you can do this with *Library ▶ Rescan Library* in the menu. If you want to rescan at every launch, select *Preferences ▶ Library ▶ Rescan on startup*.

**Varování:** On Windows 7 and 8 the import dialog points to your Windows “Music“ Library, a special-purpose virtual folder. You can **not** use these virtual folders. Select a regular folder instead, usually “My Music“, like pictured above.

## 2.5 Analyze your library



Mixxx automatically analyzes tracks the first time you load them in a deck. This allows you to run *beatgrid*, *key*, and *ReplayGain* detection on tracks in advance. It also prepares the waveform overviews, detects file corruption, and calculates the *BPM* values.

### Viz také:

For more information, go to *Rytmus*.

### 2.5.1 Why you should analyze your audio files

If the tracks have never been played with Mixxx before, the library columns *BPM*, *Duration* and *Key* will be empty or unreliable (because the values are just read from the file's tags if present and might be wrong). Pre-generating beatgrids gives you the chance to double-check the beat markers to make sure they are correct. Errors in beatgrids can cause difficulty when syncing tracks. Furthermore, track analysis takes considerable CPU power and might cause skips in the audio — things you surely don't need while performing.

### 2.5.2 How to analyze your audio files

Once you have *imported your music library*, press *OK* on the Preferences window.

- Go to the *Analyze* view on the left side panel of the library.
- To analyze a few specific tracks, hold down the `Ctrl` button, and select the tracks that you want to analyze or you can `shift`-select a range of tracks. Once they've been highlighted, click the *Analyze* button on the top right of the library section.
- To analyze the newly added tracks, toggle the *New* label and then click *Analyze*.
- To analyze the entire library, toggle the *All* label and then click *Analyze*.
- The *New* and *All* toggle buttons are located at the top left of the analysis pane - just above the *Preview* and *Cover Art* column titles of the library.

While analyzing, the progress in percentage and total queue length are shown.

## 2.6 Configuring sound input and output

Before you can start using Mixxx, you need to configure Mixxx to use your sound hardware in *Preferences* ▶ *Sound Hardware*. You do not need to configure all inputs and outputs.

The *DJ Hardware* chapter explains different types of DJ hardware and how to set them up with the input and output options. For Mixxx to perform the best it can on your system, configure the other *Zvuk* preferences in addition to the inputs and outputs.



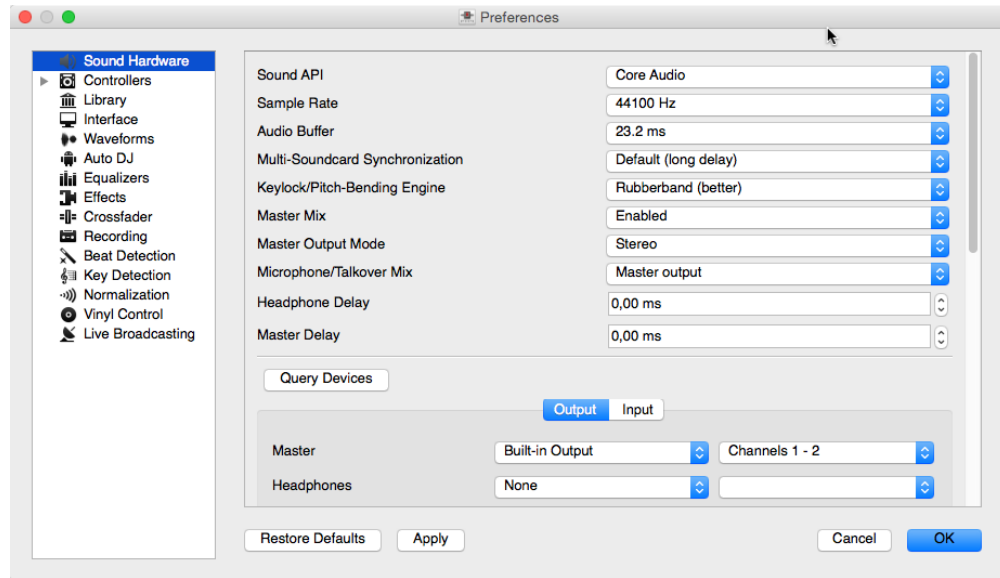


Figure2: Mixxx Sound Hardware Preferences

## 2.6.1 Output Options

### Internal mixing:

Use these outputs if you are not using an *external hardware mixer*. If you are using a controller with a built-in audio interface, use these outputs.

- *Master*: all decks, samplers, microphones, and auxiliary inputs mixed together. This output is sent to the speakers for the audience.
- *Headphones*: all decks, samplers, microphones, and auxiliary inputs assigned to *pre-fader listening*
- *Booth*: same as Master output, but has a separate gain control

### External mixing:

- *Decks 1-4*: the individual unmixed decks to send to an external mixer
- *Bus Left/Center/Right*: all decks, samplers, and auxiliary inputs assigned to each side of Mixxx's *crossfader*

## 2.6.2 Input Options

### Internal mixing:

Use these inputs if you are not using an external hardware mixer. If you are using a controller with a built-in audio interface, use these inputs.

- *Microphone 1-4*: live microphone or musical instrument inputs
- *Auxiliary 1-4*: other sound sources

### External mixing:

- *Record/Broadcast*: When this is configured, Mixxx will record and broadcast from this audio interface input instead of the internal master mix. If you have a microphone plugged into the mixer, do not configure the Microphone 1-4 inputs.

### Either internal or external mixing:

- *Vinyl Control 1-4*: timecode input from turntables or CDJs for manipulating decks 1-4

**Viz také:**

*Using Microphones*

## The Mixxx User Interface

Welcome to the Mixxx interface. This is where the magic happens. You are going to want to get very familiar with this interface because it is the main way to use Mixxx. In this chapter, we present the default interface of Mixxx and describe its elements, knobs and faders.



Figure1: The Mixxx interface - Latenight skin

This is the LateNight skin, PaleMoon color theme. It is the default skin supplied with Mixxx 2.3. There are a variety of

other skins included with Mixxx. You should explore them all to figure out which one you prefer. This section will focus on LateNight only, though all Mixxx skins provide most of the controls described here.

## 3.1 Skin Settings

On top, and on the right side of the user interface are the section expansion buttons and options.



Figure2: The GUI section quick toggles buttons in the top tool bar

If you click on either *WAVEFORMS*, *4 DECKS*, *MIXER*, *EFFECTS*, *SAMPLERS* or *MICS & AUX* then you will enable control sections for interacting with:

- *Waveform Displays*
- *Decks*
- *Mixer*
- *Effects*
- *Samplers*
- *Microphones & Auxiliary Inputs*

You can also use the specific *Application shortcuts*.

Click the *SETTINGS* button or gear icon (depending on the skin) to open the skin settings menu. This provides common basic controls as well as some skin-specific options.

## 3.2 Decks

The deck section comes in three sizes: Full, Compact and Mini. You can switch between any of the three sizes by clicking on the *Settings* button or the gear icon. Under the *Decks* section, click on „hide mixer to select“. This will automatically select the Compact deck (underlined and bolded). If you wish to change to a different deck size, you can choose FULL or Mini.

The FULL deck (which is similar in all skins) will show the *Vinyl Control Mode and Cueing controls* and *Key controls* at the top of the deck section and then the *Hotcue Controls, Intro/Outro Markers, Loop Controls* at the bottom.

The Compact deck has a reduced set of controls for when you don't need all of them on screen, for example when they're also accessible from your controller. Reverse is removed from the *Playback/Cue Controls*, and the *Hotcue Controls, Intro/Outro Markers, Loop Controls* and rate adjust buttons are not shown.

The mini deck is of a much smaller size and shows only the waveform and the *Track Information Display*. You can also use minimal decks with the maximized library.

---

**Rada:** Hiding the mixer will by default switch to the compact decks.

---

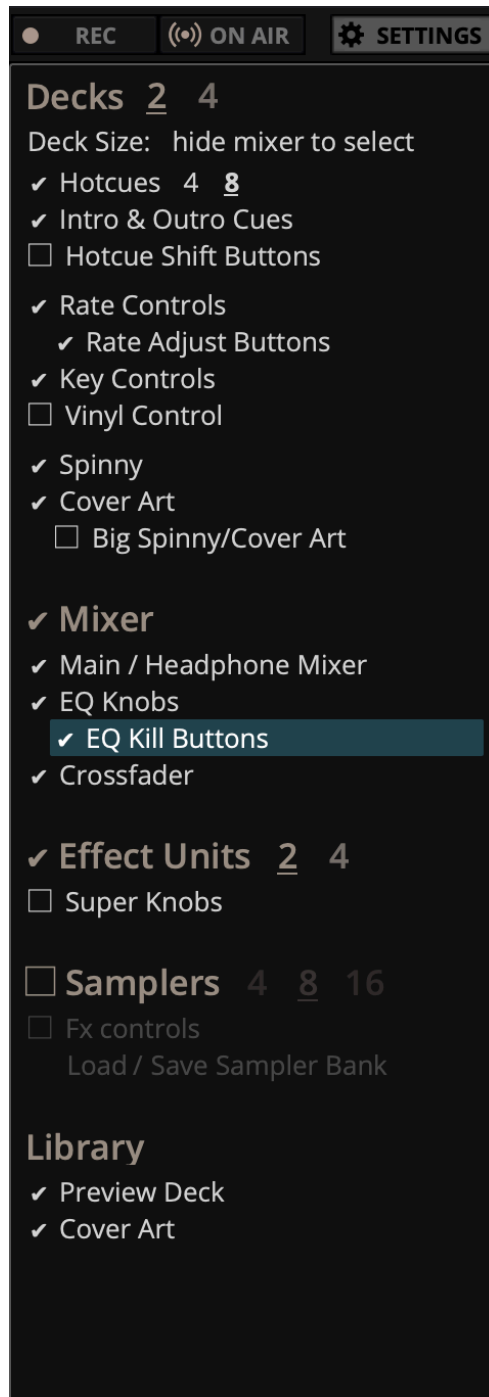


Figure3: The Skin Settings menu on the right

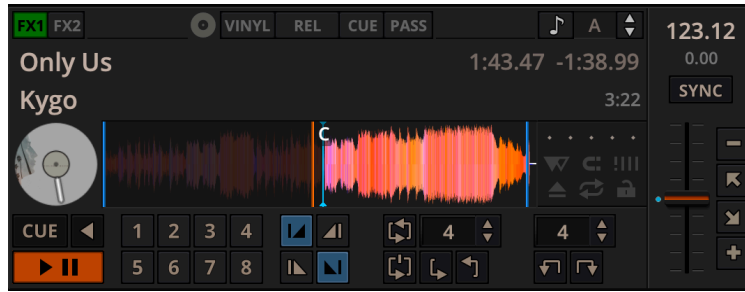


Figure4: A deck with a track loaded

The deck section allows you to control everything relating to a virtual turntable *deck*. We are going to break this down into sections.

### 3.2.1 Track Information Display



Figure5: The track information section of the deck

**Track Title** The title of the track that was loaded into a deck is displayed on top. This is the same as the title listed under the *Title* column heading in the *Mixxx library*. This information is initially loaded from the track's *metadata*. If the metadata is missing, the title and artist are guessed from the filename.

**Track Artist** The artist of the track is listed below. It is the same as listed under the *Artist* column heading in the *Mixxx library*.

**Time Elapsed/Remaining/Both** By default it displays the total elapsed time in the track up to the millisecond. Clicking on the display switches to the *Time Remaining* view, which indicates how much time is left before the track reaches the end. Clicking the display again, both the elapsed and remaining time are shown simultaneously in the format *elapsed time / remaining time*. You can change the default in *Preferences > Decks > Deck options*.

**BPM (Tempo)** The number at the top right is the effective *BPM* of the track. This is the detected *BPM* of the track, adjusted for the playback rate of the track. For example, if the track is normally 100 BPM, and it is playing at +5%, then the effective BPM will read 105 BPM.

---

**Rada:** You can use a keyboard shortcut and tap with the beat to set the BPM to the speed you are tapping, go to *Using a Keyboard* for more information.

---

#### Key

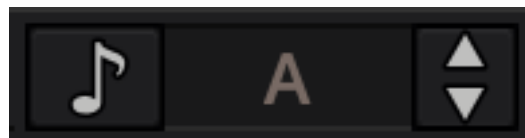


Figure6: The Key options.

The key displays the current musical key of the loaded track after pitch shifting. The key is displayed below the BPM (Tempo). This is the same as the key that is shown under the *Key* column heading in the *Mixxx library*. Clicking the key will expand it to reveal more options:

- **Pitch Control:** This changes the track pitch independent of the tempo. Left-clicking the - button decreases the pitch by one semitone. Right-clicking it decreases the pitch by 10 cents. Left-clicking the + button increases the pitch by one semitone. Right-clicking it increases the pitch by 10 cents.
- **Sync and Reset Key:** Left-clicking the *MATCH* button sets the key to a pitch that allows harmonic transition from the other track. It requires a detected key on both involved decks. Right-clicking it resets the key to the original track key.

**Cover Art** The cover art of the track loaded into the deck is displayed on the left side of the track. This is the same as the cover art that is shown under the *Cover Art* column heading in the *Mixxx library*. In Deere skin you can choose to enable or disable the cover art by clicking the gear icon in the upper right corner, then toggling the *Decks* ▶ *Cover Art* option. Clicking on the cover art will expand it into a full blown image. The cover art is also shown in the Vinyl Widget next to the *parallel waveforms*.

### 3.2.2 Waveform Displays

The waveform section of a deck allows you to visualize the audio changes that occur throughout the track, you can basically “see” your music.

By default the waveform displays are aligned parallel to each other in order to make *beatmatching* easier, as it is possible to beatmatch visually by aligning the beats that appear in each waveform.



Figure7: Parallel waveforms in the default Mixxx skin (LateNight, PaleMoon theme)

Depending on the skin, Mixxx displays either parallel waveforms (default) or separate waveforms. LateNight and Tango skins only have resizable parallel waveforms while Shade skin provides only separate waveforms. In Deere skin you can select your preferred appearance by clicking the gear icon in the upper right corner, then toggling the *Skin Settings* ▶ *Parallel Waveforms* option.

**Varování:** If you have a slower computer and notice performance issues with Mixxx, try lowering the frame rate or changing the waveform type in *Preferences* ▶ *Waveforms*.

**Waveform summary** The big waveform summary shows the waveform envelope of the track near the current playback position and is updated in real time. The mouse can be used on the waveform to pause, scratch, spin-back or throw the tracks. Right-clicking on the waveforms allows you to drag with the mouse to make temporary pitch adjustments.

---

**Rada:** You can select from different types of displays for the waveform, which differ primarily in the level of detail shown in the waveform, in *Preferences* ▶ *Waveforms* ▶ *Waveform type*.

---

**Waveform overview** The smaller, zoomed-out version of the waveform shows the various markers within the track as well as the waveform envelope of the entire track. This is useful because they allow DJs to see breakdowns far in advance. As visual aid, the part of the track that has already been played is darkened.

Clicking somewhere on the waveform allows you to jump to an arbitrary position in the track. Like with the Waveform summary, you can select from different types of displays for the overview in *Preferences* ▶ *Waveforms* ▶ *Overview type*.

**Vinyl Widget** The line on the vinyl widget rotates if the track moves. It is similar to the position marker found on scratch records. Use the mouse on the vinyl widget to pause, scratch, spin-back or throw tracks - just like a real record. When performing *Loop rolls* or right-clicking on the vinyl during playback, a “ghost” marker hints where the playback will continue. The Vinyl Widget is hidden by default and can be enabled in the *Deck Options Button Grid*.

If *vinyl control* is enabled, it can optionally display the time-coded vinyl signal quality. Activate the option in *Preferences* ▶ *Vinyl Control* ▶ *Show Signal Quality in Skin*.

**Waveform Zoom** Using the mouse-wheel everywhere in the waveform summary will zoom the waveform in or out. You can choose whether to synchronize the zoom level between the decks in *Preferences* ▶ *Waveforms*.

**Waveform Marker** While mixing, various additional markers can appear on the waveforms:

- **Position marker:** The static vertical line in the center of the waveform summary indicates the playback point of the deck. The waveform overview includes a vertical line to show the current position within the track.
- **Beat marker:** The regular white lines on the waveform summary indicate the locations of beats in the audio, also called the *beatgrid*.
- **Cue marker:** Indicates the position of the *cue point*.
- **Hotcue marker:** Indicate the position and number of a *hotcue* point if set.
- **Loop-in/Out marker:** Indicate the beginning and the end of a loop.
- **Loop overlay:** Is drawn between the Loop-in/Out markers and changes color depending on whether a loop is activated or not.
- **Track ending notification:** If the waveform overview flashes red, only 30 seconds are left before the track reaches the end.

### 3.2.3 Deck Options Button Grid



Figure8: The Options Button Grid of the deck

The six buttons at the bottom right below the waveform allow you to configure the deck. Starting from the top row (left to right) and the bottom row (left to right), the buttons are as follows:

**Slip Mode Toggle** When active, the playback continues muted in the background during a loop, reverse, scratch, etc. Once disabled, the audible playback will resume where the track would have been.



**Quantize Mode Toggle** If enabled, all cues, hotcues, loops, and beatloops will be automatically *quantized* so that they begin on a beat.

**Beat-grid Adjust Button** Clicking this button adjusts the track beat-grid so that the current position is marked as a beat. This is useful if Mixxx was able to accurately detect the track's *BPM* but failed to align the beat markers on the beats. For more information, go to the chapter *Rytmus*.

**Eject Track Button** Clicking this button ejects the track from the deck. Alternatively you can use a keyboard shortcut, go to the chapter *Using a Keyboard* for more information.

**Repeat Mode Toggle** If enabled, the repeat mode will jump back to the beginning and continue playing when the track finishes.

**Keylock Toggle** *Key lock* locks the track's pitch so that adjustments to its tempo via the rate slider do not affect its pitch. This is useful if you would like to speed up or slow down a track and avoid the "chipmunk" effect that speeding up vocals has.

### 3.2.4 Playback/Cue Controls



Figure9: The playback and cueing controls of the deck

**Play/Pause Button** Clicking the Play/Pause button starts and pauses the playback.

**Viz také:**

The actual behavior of the Play/Pause button depends on the selected *Cue mode*.

**Cue Button** If the button is pressed while the track is playing, the play position jumps to the *Cue point* on the track and the track is paused. Once the track is paused at the Cue point, you can preview it by holding the Cue Button. If you have not moved the Cue point yet, the default point is at the track's beginning.

**Viz také:**

The actual behavior of the Cue button depends on the selected *Cue mode*.

**Reverse button** Press and hold this button to temporarily play the track backwards during regular playback. Right-click and hold for „Censor“ mode. This reverses the track while the playback continues muted in the background. On release playback is resumed where the track would have been if it had not been reversed.

#### Setting Cue points

Every track has a Cue point, by default on its beginning. Unlike with *Hotcues*, you can't delete, but only move Cue points. The Cue point is moved according to the selected Cue mode, as explained below.

## Auto Cue

Mixxx automatically seeks to the saved Cue point on track load. If none exists, seeks to the beginning of the track.

Uncheck *Preferences* ▶ *Interface* ▶ *Auto Cue* ▶ *Jump to main cue point on track load* to always seek to the beginning of the track.

## Using Cue Modes

Mixxx supports various Cue modes that adapt to fit other devices in your setup or your personal preferences. The actual behavior of the Play, Pause, and Cue buttons depends on the selected Cue mode. To change the Cue mode, go to *Preferences* ▶ *Decks* ▶ *Cue mode*.

### Mixxx mode (default)

- **While paused at Cue point (Cue button lit):**
  - Hold down the Cue button to play the track from the *Cue point*, as long as the button is pressed. After releasing the Cue button, the track jumps back to the Cue point and pauses again.
  - Press Play/Pause before releasing the Cue button to continue playback normally.
- **While paused not at Cue point (Cue button flashing):**
  - Tap the Cue button to store the current position as new *Cue point*.
- **While playing:**
  - Tap the Cue button to jump to the *Cue point* and pause playback.

### Mixxx mode (no blinking)

- Same as the default Mixxx mode, but without blinking indicators.

### Pioneer mode

- Same as Mixxx mode, but the buttons are more flashy matching a Pioneer CDJ player.
  - The Play/Pause button flashes once per second when a track is paused.
  - The Cue button flashes twice a second when the next push will move the *Cue point*.

### Denon mode

- **While paused at Cue point (Cue button lit) - same as Mixxx mode:**
  - Hold down the Cue button to play the track from the *Cue point*, as long as the button is pressed. After releasing the Cue button, the track jumps back to the *Cue point* and pauses again.
  - Press Play/Pause before releasing the Cue button to continue playback normally.
- **While paused not at Cue point (Play button flashing):**
  - Tap the Cue button to move the track back to *Cue point*.
  - Tap the Play/Pause button to move the *Cue point* to the current position and start playback.
- **While playing - same as Mixxx mode:**
  - Tap the Cue button to jump to the *Cue point* and pause playback.

### Numark mode

- Same as Denon mode, but without a flashing Play/Pause button.
- This mode is useful if you want to implement a custom skin with 3-button decks, featuring a Cue button, a dedicated Pause button, and a Play/Stutter button.

### CUP (Cue + Play) mode

- Press the Cue button to jump to the *Cue point* and pause, continuing playback only when either the Cue button is released or Play/Pause is pressed.
- If the Cue button is pressed while paused, the *Cue point* is first moved to the current position.

---

**Rada:** Use the *Hotcue Controls* to place more reference points on a track.

---

#### Viz také:

You can also use keyboard shortcuts for Cueing. Go to *Using a Keyboard* for more information.

## 3.2.5 Intro/Outro Markers



Figure10: The Intro/Outro End and Start Markers

You can use the Intro/Outro Markers with Auto DJ to mix in and out of points but unlike hotcues, they do not mark only one point. The intro and outro are sections and each section is defined by two points. See *Auto DJ - Automate your mix*.

**Intro Start/End Markers** The analyzer places the intro start point at the first sound of a track and you can decide where to place the intro end point. Left-clicking on the marker when it is set jumps to the marker. When it is not set, clicking sets the marker to the current play position. Right-clicking on the marker when it is set clears the marker.

**Outro Start/End Markers** The analyzer places the outro end point at the last sound of a track and you can decide where to place the outro start point. Left-clicking on the marker when it is set jumps to the marker. When it is not set, clicking sets the marker to the current play position. Right-clicking on the marker when it is set clears the marker.

---

**Poznámka:** If *quantization* is enabled, left-clicking on any of the markers snaps to the nearest beat.

---

## 3.2.6 Transport Controls



Figure11: The transport controls of the deck

**Beatjump/ Loop Move Size** You can type, use the arrow keys or scroll vertically in the input box to select the number of beats to jump or move, with the Beatjump Forward/ Backward buttons.

**Beatjump Backward button** Clicking this while the loop is inactive will make the track jump backward by the set number of beats (left click) or by one beat (right click). Clicking this while the loop is active will move the loop backward by the set number of beats (left click) or by one beat (right click).

**Beatjump Forward button** Clicking this while the loop is inactive will make the track jump forward by the set number of beats (left click) or by one beat (right click). Clicking this while the loop is active will move the loop forward by the set number of beats (left click) or by one beat (right click).

### 3.2.7 Vinyl Control Mode and Cueing controls

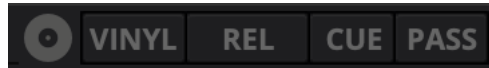


Figure12: The Vinyl Control Mode and Cueing controls of a deck

The control above the waveforms relate to the *vinyl control* feature in Mixxx and is **hidden** in the default *Mixxx user interface*. Click the gear icon in the upper right corner, then toggle *Skin Settings* ▶ *Vinyl Control Options*, or use the specific *Application shortcuts* to show or hide the section.

**Viz také:**

For more information on how to use Vinyl control in your setup, go to the chapter *Ovládání vinylovou gramodeskou*.

#### Vinyl button

This button enables *vinyl control*. When enabled in Mixxx, the track on the Mixxx deck is controlled by the external vinyl control.

#### Pass button

This button enables passthrough. When enabled, audio from an external vinyl can be played on the Mixxx deck.

#### Abs/Rel/Const button

- **Absolute mode:** The track position equals needle position and speed.
- **Relative mode:** The track speed equals needle speed regardless of needle position.
- **Constant mode:** The track speed equals last known-steady speed regardless of needle input.

#### Cue/Hot button

This button determines how *cue points* are treated in vinyl control relative mode:

- **Off:** Cue points are ignored.
- **Cue:** If the needle is dropped after the cue point, the track will seek to that cue point.
- **Hot:** The track will seek to the nearest previous *hotcue* point.

### 3.2.8 Sync and Rate Controls

The rate controls allow you to change the rate at which tracks are played. This is very useful for *beatmatching* two tracks together in a mix. You can control rate changes also from your computer's keyboard, see the chapter *Using a Keyboard* for more information.

#### SYNC button

- **Left-Click:** Changes the *rate* of the track so that the *BPM* and *phase* of the track matches the other deck.
- **Right-Click:** Only changes the *rate* of the track to match the other deck but does not adjust the *phase*.

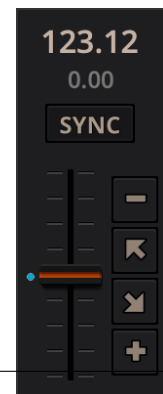


Figure 13: Rate controls

Decks and samplers now pick which sync target to sync to on the fly. Decks can't sync to samplers and samplers can only sync to decks. The sync target for samplers is:

- The first deck (in numerical order) that is playing a track with a detected beatgrid, and has a rate different than zero.
- The first deck (in numerical order) that has a track loaded with a detected beatgrid, no matter if it is playing or stopped.

So basically, if you sync a sampler and both deck 1 and deck 2 are playing a track with a beatgrid then deck 1 will win since numerically it is first. The sync targets for decks are chosen with the new master sync feature, see *Master Sync* for more information.

**Pitch/Rate slider** The slider allows you to change the speed of the song, by default up to 10% from the tracks original tempos. The speed will increase as you move the slider up, opposite to the behavior found on DJ turntables and *CDJ*. Right-clicking on the slider will reset the tempo to its original value.

**Pitch Rate Display** The percent that the track's rate is sped up or slowed down is noted here. Is the Pitch/Rate slider positioned at the center, the pitch rate display is at +0.0%, which indicates no change.

**Temporary Pitch/Rate buttons (Nudge)** Pushing the diagonal arrow buttons is like nudging the metal edge of a turntable, or the outer edge of a CD player. It will give the track a push or pull forwards or backwards. If the buttons are released the previous tempo is restored. The buttons can act as either a fixed *pitch bend* or a *ramping pitch bend*.

**Pitch/Rate buttons** Clicking the - button will set the pitch/rate lower while the + button will set the pitch/rate higher. This is like moving the pitch slider up or down by a fixed amount. Unlike using the temporary pitch/rate buttons, the change is permanent so these buttons don't need to be held down.

**Viz také:**

To customize the amount by which the buttons alter the pitch of the track, the slider range as well as the direction, go to *Preferences* ▶ *Interface* ▶ *Adjustment buttons* ▶ *Permanent / Temporary*.

**Rada:** If the tempo of a track changes, you'll notice that the tone changes based on the pitch used (e.g. playing at faster pitch gives a chipmunk sound). You can enable the *Keylock* feature to maintain a constant tone.

### 3.2.9 Loop Controls



Figure14: The beatloop and looping controls of the deck

In this section of the Mixxx interface you can control (beat-)loops and set the loop points of a track. You can also shift loops using the *beatjump / loop move controls* shown at the right side of the looping controls.

**Beatloop size** The input box allows you to set the size of the loop in beats to set with the Beatloop button. Changing this resizes the loop if the loop already matches the size. If a loop is set, a loop overlay will be drawn on the *waveforms*.

**Beatloop** Left-clicking the Beatloop button starts a loop over the set number of beats. If quantize is enabled, it snaps to the nearest beat. This works for manually placed loops as well as automatic loops set by the beatlooping buttons.

Depending on the current status of the loop, the loop overlay on the waveforms changes color. Right-clicking the Beatloop button temporarily enables a rolling loop over the set number of beats. Playback will resume where the track would have been if it had not entered the loop.

**Double loop** Clicking on the arrow pointing up will double the current loop's length up to 64 bars. The length of the loop overlay in the waveform will increase accordingly.

**Halve loop** Clicking on the arrow pointing down will halve the current loop's length down to 1/32 bars. The length of the loop overlay in the waveform will decrease accordingly.

**Loop-In** This button allows you to manually set the start point of a loop. A loop-in marker is placed on the waveform indicating the position. If clicked when a loop was already set, it moves the start point of a loop to a new position.

**Loop-Out** This button allows you to manually set the end point of a loop. A loop-out marker is placed on the waveform indicating the position. If clicked when a loop was already set, it moves the end point of a loop to a new position.

**Reloop** Left-clicking this button toggles the current loop on or off. If the loop is ahead of the current position, looping will start when the loop is reached. This works only if the Loop-in and Loop-Out marker are set. Right-clicking this button enables the loop, jumps to Loop-in marker and stops playback.

---

**Rada:** If you are playing inside a loop and want to move the end point beyond its current position in the track, click on the *Beatloop* button first and when the play position reaches the desired position, click on the Loop-Out button.

---

#### Viz také:

If the *Quantize* mode is enabled, the loops will automatically snap to the nearest beat. This is disabled by default, click on the *Quantize Mode Toggle* to enable it.

### 3.2.10 Hotcue Controls

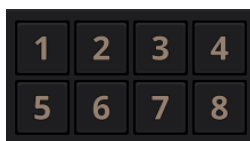


Figure 15: The hotcue controls of the deck

To jump in between different parts of a track, you can use these numbered buttons. You can also use keyboard shortcuts, go to *Using a Keyboard* for more information.

#### Setting Hotcues

Clicking on a numbered button will set a *hotcue* at the current play position on the track. A marker with the corresponding number will appear in the waveform and the button will light up to indicate that the hotcue is set.

## Playing Hotcues

- **While playing:** Tap a hotcue button to cause the track to instantly jump to the location of the hotcue and continue playing. If you are playing inside a loop and tap a hotcue whose position is outside of the loop, then the track still instantly jumps to the hotcue but the loop will be deactivated.
- **While stopped:** Tap a hotcue button to cause the track to instantly jump to the location of the hotcue and start playing as long as the button is pressed. Press the *Play* keyboard shortcut while the hotcue button is pressed to continue playback, then release the hotcue button.

## Highlighting Hotcues

It's possible to add labels or colors to your hotcues to make them easier to distinguish and make it possible to get their „meaning“ at first glance.

Labels and colors can be assigned from the Hotcue context menu, which can be opened by right-clicking either the hotcue button or the hotcue marker in the waveform overview.

---

**Rada:** Mixxx allows *customizing the color palette* and *bulk replacing hotcue colors*.

---

## Deleting Hotcues

To delete a hotcue, right-click on the hotcue button or the waveform marker in the overview waveform to open the context menu, then click the trash icon.

---

**Poznámka:** Mixxx supports up to 36 hotcues per deck. By default, only some of them are visible in the user interface. You can customize your *keyboard* or *controller* mappings to use all of them.

---

### Viz také:

Just as with the loops (see above), if the *quantize* mode is enabled, the hotcues will automatically snap to the nearest beat. This is disabled by default, click on the *Quantize Mode Toggle* to enable it.

## 3.3 Mixer

The mixer section of the *Mixxx user interface* allows you to control how the different decks and samplers are mixed together.

### 3.3.1 Channel Faders and Level Meters

**Level meters** The left meter shows the volume level for the left deck, while the right meter shows the volume level for the right deck. The double meter at the center shows the current volume of the main output's left and right channel separately. These should stay at the top of the green region, with the loudest parts of the music (transients) briefly going into the yellow region. They should never be in the red region. Refer to *Setting your levels properly (gain staging)* for more details.

**Line faders** The two large faders on either side of the level meters are the deck volume faders, also known as Channel- or Line-faders. Adjusting these controls the volume of each deck.



Figure16: The mixer section

---

**Rada:** Some DJ's prefer to use the line faders over the crossfader for fading between tracks. Try it, you may like it.

---

### 3.3.2 Equalizers and Gain Knobs

**Gain Knob** The gain knob (orange) above the volume fader allows you to adjust the gain applied to the deck. Use this to compensate for the differences in recording levels between tracks. In general, you should adjust this knob so that the track's *level meter* stays around the top of the green region with the loudest parts of the track (the transients) briefly going into the yellow region. **Do NOT turn the Gain Knob up so much that the level meter is in the red region. At this point the track is clipping, which sounds bad and could damage equipment.**

---

**Poznámka:** By default, Mixxx automatically applies an additional ReplayGain so tracks have approximately equal loudness at unity gain. Your tracks must be *analyzed* to take advantage of this feature. When an unanalyzed track is loaded, Mixxx calculates its ReplayGain value, but will not apply a newly calculated ReplayGain value to a track after it has already started playing (to avoid a sudden change in the gain of a playing track).

---

#### Viz také:

For an explanation of why you should set your gains this way, see [Setting your levels properly \(gain staging\)](#).

#### EQ Knobs

The low, mid, and high EQ knobs allow you to change the filters of the audio. This allows you to selectively reduce or boost certain frequency ranges of the audio.





Figure17: EQ Controls

**Varování:** Be careful not to raise the EQs so much that the signal clips, indicated by a *level meter* being in the red region. See *Setting your levels properly (gain staging)* for a more detailed explanation.

**Kill Switches** The small boxes next to each EQ knob are called *kills*. Hold these buttons to fully remove that frequency range. Short click on the buttons for latching. When in Latch mode, click again to restore the frequency range. If the Kill switches do not work as expected, check the high/low shelf EQ settings in the preferences.

**Quick Effect Super knob** This knob lets you control the Meta knob of the deck's Quick Effect. „Filter“ is the default. Right-click this knob to center it. This is equivalent to „no audible effect“ for the built-in filter effects only (Filter, Moog Ladder 4 Filter). The button next to the knob toggles the effect. It provides the same latching capabilities like the EQ Kill switches.

**Viz také:**

You can customize the EQ settings in *Preferences* ▶ *Equalizer*.

### 3.3.3 PFL/Headphone Button



Figure 18: The headphone buttons of both decks in the mixer

The headphone button is also known as the *pre-fader listen* or *PFL* button and allows you to use headphone cueing. That is, it allows you to hear the track you will play next in your headphones before your audience hears it. This helps you plan when to start the next track and manipulate it to make it blend better with the currently playing track (for example, by synchronizing the beats, setting a loop, or using EQs). Pressing the PFL button plays the track on the headphone output at full volume regardless of the position of the crossfader or the deck's line fader. To play a track in your headphones but not have the audience hear it, press the PFL button and move the crossfader all the way to the opposite side of the deck or turn the deck's volume fader all the way down.

---

**Poznámka:** Headphone cueing is only available if you have configured a Headphone Output in *Preferences* ▶ *Sound Hardware*.

---



---

**Poznámka:** If you use an external hardware mixer, plug your headphones into that rather than your sound card. Use the headphone buttons on the mixer rather than the buttons in Mixxx.

---

### 3.3.4 Prolínač



Figure19: The crossfader section of the mixer

The *crossfader* determines the actual volume of each deck when moving the slider from left to right. If both decks are playing and the crossfader is in its default center position, then you will hear both decks. Right-clicking on the crossfader will reset the slider to its default position.

To move the crossfader left or right, you can use the *keyboard shortcut* G and H. To apply small crossfade left and small crossfade right, the *shortcuts* are Shift + G and Shift + H respectively.

You can reverse the configuration of the crossfader, so that the right deck is on the left end of the crossfader and vice versa. This is also known as *Hamster Style*. To adjust the crossfader to your style of mixing, go to *Preferences* ▶ *Crossfader*.

The *crossfader* orientation determines to which side of the crossfader the deck is assigned. It can be set to the left or right side, or to the center where it is unaffected by the crossfader knob.

---

**Rada:** Using the *AutoDJ* feature in Mixxx, you can automate the crossfade between the decks.

---

### 3.3.5 Headphone and Main Mix Controls



Figure20: The headphone and main mix knobs

**Headphone Mix knob** Allows you to control how much of the Main output you hear in the *Headphones* output. It works like a crossfader between the stereo Main and stereo *PFL* signal. If the knob is set to the left, you only hear the *PFL* signal, which is the usual position for prelistening tracks.

This can be used to practice DJing with only one stereo output and a set of either speakers or headphones.

Additionally, this can be used to mix in headphones when you cannot hear the speakers connected to the *main* output and there are no booth monitor speakers. Some DJs prefer to use this knob to mix in headphones all the

time, but many find it to be a difficult way to play. If there is only one set of speakers and no booth monitors, you can avoid the need to mix in headphones by setting up the speakers behind you so both you and your audience can hear them.

---

**Poznámka:** Don't forget to activate the *PFL* button on the deck you want to listen to in your headphones.

---

### Head knob (Headphone gain)

This knob controls the gain applied to the headphone output signal that Mixxx sends to your sound card. Generally, this knob should not be adjusted. To adjust the volume in your headphones, adjust your sound card's gain on the headphone output. Most external sound cards have a gain knob specifically for the headphone jack. If you use an onboard sound card for your headphones, adjust the gain control in your *OS* mixer program. If your sound card only has one gain knob that controls both the Main output and the headphone output, then adjust the *Head* knob in Mixxx to control the volume of your headphones without affecting your Main output.

---

**Poznámka:** Not all controls may be present in all skins anymore. However, these controls can be mapped to keyboards and MIDI controllers. See *Mixxx Controls* for detailed information.

---

**Headphone Split Cue button** If activated, two mono signals are sent to the headphone output instead of one stereo signal. The main signal plays in the right channel, while the cueing signal plays in the left channel. You can still adjust the *Head Mix* knob to control the mix of the Main and cue signals in the left channel.

**Balance Knob** This knob allows you to adjust the *balance* (left/right orientation) of the Main output.

**Main Knob** The Main Gain Knob controls the gain applied to the Main output signal that Mixxx sends to your sound card, Internet broadcasting servers, and recorded mixes. Generally, this knob should not be adjusted. To change the output volume, a gain control should be adjusted as close to the speakers as possible for the best signal-to-noise ratio throughout your signal chain. Ideally, the volume would only be adjusted by adjusting the gain on the power amplifier. On active speakers, including computer speakers, studio monitors, and PA systems that do not require a separate amplifier, this means adjusting the gain on the speakers (which may be labeled „Volume“).

As a DJ, you may not have access to the power amplifier. In that case, adjust the volume by adjusting the gain on the equipment closest to the power amplifier that you have access to. Typically, this is a DJ mixer. If you are not going through a mixer, adjust the gain of your sound card. If your sound card does not have any controls on it (for example, for sound cards built into computers), adjust the sound card settings with your *OS* mixer program. The Main Gain Knob in Mixxx should be a last resort for adjusting volume.

**Varování:** In no case should any part of the signal chain be clipping, indicated by a level meter going into its red region or an LED labeled „clipping“ turning on.

### Viz také:

For an explanation of why you should set your gains this way, see *Setting your levels properly (gain staging)*.

## 3.4 Samplers



Figure21: Sample decks



Figure22: Sample decks expanded

Samplers are miniature decks. They allow you to play short samples and jingles but also additional tracks in your mix. They come with a small overview waveform and a limited number of controls. All controls work just like on the regular decks, see [Decks](#).

The Sampler section can be **hidden** in the *Mixxx user interface*. Click the *SAMPLERS* button at the top of the user interface, or use the specific *Application shortcuts* to show or hide the section.

### 3.4.1 Zobrazení průběhové křivky

**Waveform overview** The waveform shows the various markers within the track as well as the waveform envelope of the entire track. Clicking somewhere on the waveform allows you to jump to an arbitrary position in the track.

**Waveform Marker** While mixing, some additional markers can appear on the waveforms:

- **Position marker:** A vertical line indicates the playback point of the sample deck.
- **Hotcue marker:** Indicate the position and number of a *hotcue* point if set.

### 3.4.2 Track Information Display

**Track Artist/Title** The artist and title of the track that was loaded into a sampler deck is displayed here.

**BPM (Tempo)** The number at the top right is the effective *BPM* of the track. Tap the BPM to set the BPM to the speed you are tapping.

### 3.4.3 Sample Deck Options Button Grid

The buttons grid next to the waveform overview allow you to configure the sampler deck. Starting from the top row (left to right) and the bottom row (left to right), the buttons are as follows:

**Repeat Mode Toggle** If enabled, the repeat mode will jump back to the beginning and continue playing when the track finishes.

**Eject Track Button** Clicking this button ejects the track from the deck.

**Keylock Toggle** *Keylock* locks the track's pitch so that adjustments to its tempo via the rate slider do not affect its pitch.

**Crossfader Orientation Toggle** The *crossfader* orientation determines to which side of the crossfader the sampler is assigned. It can be set to the left or right side, or to the center where it is unaffected by the crossfader knob.

### 3.4.4 Sample Deck Controls

**Play/Pause button** Clicking the button always starts playback from the *cue point*. If playback is already enabled the play position will jump back to the *cue point*. Right-clicking on the button will stop playback and seek the play position to the beginning of the sample.

**Sync Button** Syncs the Sampler deck to a regular deck, as described in *Sync and Rate Controls*.

- **Left-Click:** Changes the *rate* of the track so that the *BPM* and *phase* of the track matches the sync target.
- **Right-Click:** Only changes the *rate* of the track to match the sync target but does not adjust the *phase*.

**Hotcue Controls** To jump in between different parts of a sample, you can use these numbered hotcue buttons.

**PFL/Headphone Button** If active, the *pre-fader listen or PFL* button allows you to listen to the track in your headphones.

**Gain Knob** Allows you to adjust the gain applied to the track.

**Level Meter** Shows the level of the signal and can be adjusted with the Gain knob.

**Pitch/Rate slider** The slider allows you to change the speed of the sample. Right-clicking on the slider will reset the tempo to its original value.

## 3.5 Effects



Figure23: Collapsed effect unit

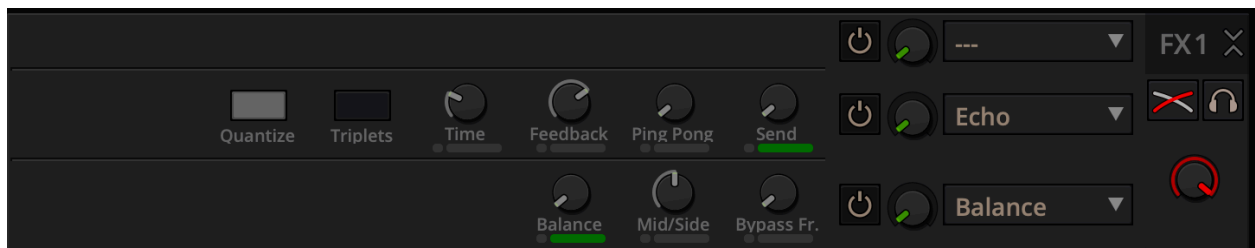


Figure24: Expanded effect unit

Mixxx comes with some built in sound effects that can be chained together in effect units. Effect units can be freely assigned to different decks as well as other inputs. The buttons for routing decks to effect units are in the *Decks*.

**Viz také:**

The *Efekty* chapter has a more detailed explanation of the effects system.

## 3.6 Microphones & Auxiliary Inputs



Figure25: The Microphone section

The microphone section is **hidden** in the default *Mixxx user interface*. Click the *MICS & AUX* button at the top of your user interface, or use the specific *Application shortcuts* to show or hide the section.

**Viz také:**

The *Using Microphones* chapter has detailed instructions for setting up microphones with various hardware.

### 3.6.1 Microphone Controls

**Talk Button** Hold this button and talk to mix the microphone input into the Mixxx Main output. Short click on the button for latching. This is handy for talking for an extended period, for example when *streaming* a radio show. When in Latch mode, click again to mute the microphone input.

**Mix Orientation Toggle** This control determines the microphone input's mix orientation. Either to the left side of crossfader, to the right side or to the center (default). Clicking cycles through all the options.

**Microphone Volume Meter** This displays the microphone volume input signal strength.

**Microphone Gain Knob** Use this knob to adjust the gain of the microphone output. Try to keep the volume at a reasonable level to avoid signal clipping.

## 3.7 Preview Deck

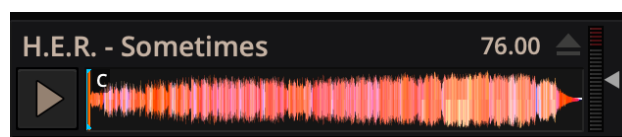


Figure26: The Preview Deck with a track loaded

The Preview Deck is a special deck that allows you to pre-listen to tracks in the headphones before using them in your mix. Pre-listening to a track does not change its *Played* state as well as the play counter and is not logged in the *History*. Press CTRL + 4 (Windows/Linux) or CMD + 4 (Mac) to display the Preview Deck.

The features in detail:

- **Track Artist/Title:** The artist and title of the track is displayed here. This is the same as that listed under the *Track* and *Title* column heading in the *Mixxx library*. This information is initially loaded from the track's *metadata*.
- **Eject Track button:** Clicking this button ejects the track from the deck.
- **Play/Pause button:** Clicking the button starts and pauses the playback. Right-clicking on the button will stop playback and seek the play position to the beginning of the tracks.

- **Waveform overview:** Shows the various markers (Cues, Hotcues) within the track as well as the waveform envelope of the entire track. Clicking somewhere on the waveform allows you to jump to an arbitrary position in the track.
- **Gain slider:** Move the slider to adjust the gain of the track.
- **Volume Meter:** Shows the current volume of the track. If it's too loud and distorted, a peak indicator flashes red.

**Viz také:**

For more information, go to the chapter *Previewing Tracks*.



## 4.1 Overview of the Library features

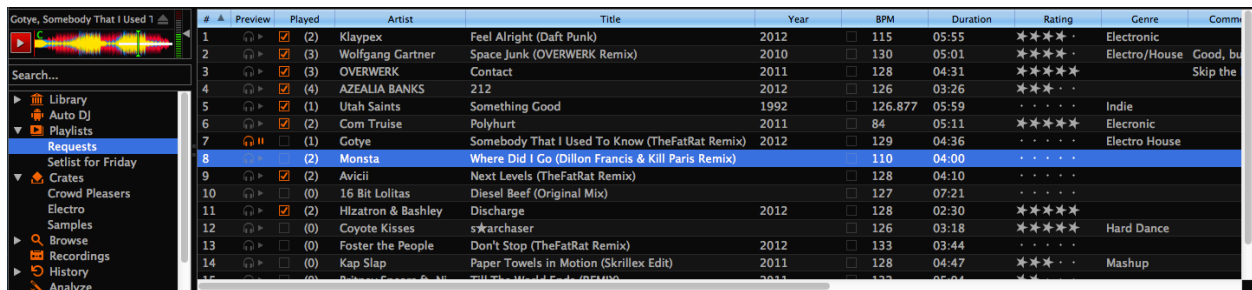


Figure1: The Mixxx Library

The library manages all of your music files. This is where you can find the tracks you want to play and load them into a *deck* or *sampler*; see *Loading Tracks*. Mixxx imports your music library automatically when it is run for the first time, see *Changing music directories* for more information.

The sidebar on the left contains different collections of music. The track list view on the right displays the tracks in those collections.

### Sidebar:

- **Search:** Search for tracks in your Mixxx library.
- **Tracks:** View and edit your whole collection.
- **Auto DJ:** Automatically load and crossfade tracks for unattended mixing.
- **Playlists:** Arrange tracks in a set order.
- **Crates:** Organize tracks in (unordered) collections.
- **Computer:** Navigate, view, and load tracks from folders on your hard disk and connected devices.

- **Recordings:** Record your mix and view previous recordings.
- **History:** Browse lists of tracks played in past mixing sessions.
- **Analyze:** Generate Mixxx-specific data for tracks before a gig.
- **External Libraries:** Access your existing iTunes, Traktor, Rhythmbox, and Banshee libraries.

#### Track List:

- **Sort:** Display and sort track collections by different criteria.
- **Load:** Drag tracks you want to play to the waveform display.
- **Edit:** Rate tracks and edit track properties.

## 4.2 Tracks - View and edit your whole collection



A sortable list of all the tracks in your music library.

**Customizing the view** Move columns by clicking a column header and dragging it to another position. Right-click on a column header to show or hide individual columns. Adjust the column width to fit the contents of the rows by double-clicking on the separator between two column headers.

**Sorting Tracks** Tracks are automatically sorted by the active column. Click on the active column header to reverse the sort order. Click on another column header to change the active column.

- **Multi-Column sorting**

You can sort multiple columns by clicking *up to three* column headers you wish to sort your tracks by. This makes searching for tracks in a large list more efficient. Clicking twice on a column header will reverse the sort order.

Example:

1. Click on the *BPM* column header -> list is sorted by BPM
2. Click on the *Year* column header -> list is sorted by year. If year is equal, it is still sorted by the last criteria, e.g. BPM.
3. Click on the *Genre* column header -> list is sorted by genre. If genre is equal, it is still sorted by year. If genre and year is equal, then it is sorted by BPM.

- **Random sorting**

Click on the *Preview* column header to randomly sort tracks.

- **Sort keys by circle of fifths**

Click on the *Key* column header. The order starts with *C* at the top, then its relative minor (*a*), then adds 1 sharp (*G*), then *G*'s relative minor (*e*), and so adding sharps/removing flats around the circle of fifths. For *Lancelot* notation, it starts with *G#m* at the top, which is *1A* in that notation, and proceeds alphanumerically (minor keys appear before their relative major keys in this case).

**Rating tracks** Make sure the *Rating* column is not hidden. Rate tracks by hovering over the rating field and clicking the stars.

**Track Inline editing** Mixxx reads *metadata* from the tracks to fill the columns of the library. Double-click on a field, edit the data, and hit `Enter` when you are done. Go to the chapter [edit metadata](#) for detailed information.

**Loading tracks** To load a track into a *deck*, you can either drag it to the waveform display, use the context menu or use *keyboard shortcuts*. Go to the chapter [Loading Tracks](#) for detailed information.

**Importing tracks** Mixxx imports your music library automatically when it is run for the first time. Go to the chapter *Changing music directories* for detailed information.

**Previewing Tracks** To pre-listen to tracks in your headphones without loading them to a regular deck, click the  icon in the *Preview* column. Go to the chapter *Previewing Tracks* for detailed information.

**Cover/Album Art** Mixxx can display cover art it finds for a track in the library. It will search for available cover art in the following places in order and will use the first one it finds:

1. The first image saved in the track's ID3v2/Vorbis *tags*
2. An image file in the track folder (if there is only one image file there.)
3. `%track-file-base%.jpg` in the track directory for `%track-file-base%.mp3`
4. `%album%.jpg`
5. `cover.jpg`
6. `front.jpg`
7. `album.jpg`
8. `folder.jpg`

Mixxx supports the following image types: jpg, jpeg, png, gif, bmp

**Rescan Library** If you want to manually refresh your library without exiting (for example because you added or moved files) you can do this with *Library* ▶ *Rescan Library* in the menu at the top of the application window. You can prompt an automatic rescan when Mixxx is started under *Preferences* ▶ *Library* ▶ *Rescan on startup*.

Rescanning the library will add new files to the library and mark tracks as missing if the corresponding file has been deleted. It tries to detect and relocate missing tracks if files have been renamed or moved into another directory.

Automatically refreshing the metadata of tracks when files have been modified by an external application is not supported, yet. In this case you need to select the affected tracks and (re-)import their metadata from file tags.

## 4.2.1 Track list context menu

Right-clicking on selected tracks in the track list reveals the context menu:

- **Add to Auto DJ**

Adds the content of the selection to either the *bottom* or *top* of the *Auto DJ* playlist for automatic mixing. The *replace* option deletes the Auto DJ queue first, and then fills it with the selected tracks.

- **Load to**

Loads a selected file to one of the available *Decks*, *Samplers* or to the *Preview Deck*. Alternatively simply drag it to the *Waveform Displays*. The number of available decks and samplers might differ, depending on the selected skin. Note that you can't load multiple files at once.

- **Add to Playlist**

Add selected tracks to an existing playlist, or drag the selection to a playlist crate in the sidebar. Choose *Create new playlist* to put them into a new playlist.

- **Crates**

The checkbox shows in which existing crates the selection is in. Changing the crates selection does not collapse the menu, which allows much easier categorization of tracks without going through the menu from scratch. Choose *Create new Crate* to put them into a new crate.

- **Select Color**

Assign a color to all selected tracks to indicate mood, energy etc. Select the black button at the top left to remove the color.



- **Metadata**

- **Import from File Tags:** If the track's *metadata* changes, (e.g. if you used iTunes to edit it,) this option lets you save the new values for the selected tracks to the Mixxx library. See [edit metadata](#).
- **Import from MusicBrainz:** Look up *metadata* online by searching the *MusicBrainz* database and apply the search results to your tracks. See [edit metadata](#).
- **Export to File Tags:** Saves *metadata* to files. Mixxx may wait to modify files until they are not loaded to any decks or samplers. If you do not see changed metadata in other programs immediately, eject the track from all decks and samplers or shutdown Mixxx.
- **Cover Art**
  - \* **Choose New Cover:** Select an image from the file browser as the cover.
  - \* **Clear Cover:** Delete any cover information saved for this track.
  - \* **Reload from track/folder:** Reload the cover from the tracks' ID3v2 *tags* or a picture in the track folder if the tags do not supply any.

- **Clear**

- **BPM and Beatgrid:** Removes the *BPM* and *beatgrid* data from the selected tracks in the Mixxx library. After doing this, we recommend [analyzing](#) the tracks again.
- **Play Count:** Marks selected tracks as not played in the current session and sets their play counter to zero. The icon in the *Played* column changes.
- **Hotcues:** Clears all saved *hotcues* and *hotcue labels*.
- **ReplayGain:** Clears all saved *ReplayGain* values.
- **Waveform:** Clears the saved waveform data from the selected tracks. This is useful if a file is overwritten by another file of the same name, or was edited externally (e.g. in a waveform editor). To clear the waveform cache for all tracks in the library, select *Preferences* ▶ *Waveforms* ▶ *Clear cached Waveforms* .
- **All:** Clears all of the data above from the Mixxx library database at once.

- **Change BPM**

- Allows to change the *BPM* to 50%, 66%, 75%, 133,33%, 150%, or 200% of the *BPM* set by Mixxx when [analyzing](#) the tracks. If many of the detected *BPM* values are incorrect, you may try other settings in the [Analyzer Settings](#) and re-run the analysis.
- **Lock/Unlock BPM:** Locks/Unlocks the *BPM* of selected tracks so you can't edit them in the track properties. The  icon next to the track's *BPM* in the library row is a toggle. Clicking it will set the status to „locked“, and the icon changes to .

- **Hide from Library**

Temporarily hides selected tracks from the track list. Hidden tracks are listed in the *Hidden Tracks* menu item which is explained below.

- **Open in File Browser**

Brings up the selected files in your file manager.

- **Properties**

Similar to inline editing explained above, the properties dialog allows you to view and edit metadata such as title, artist, album, and view the full file name and path. Note that you can not edit multiple files at once.

---

**Poznámka:** Most of the context menu items are available in file lists of other views like Auto DJ, Playlists, and Crates as well.

---

## 4.2.2 Chybějící skladby

The Missing Tracks view is accessible by expanding the Library tree item in the sidebar. Any tracks that were previously loaded into your library but were later not found on your drive will appear here. Mixxx does not automatically delete records of missing tracks so that extra metadata Mixxx might have (such as hot cues and beat grids) will not be lost in case the file is replaced.

The features in detail:

- **Select All button:** Selects all tracks in the current view.
- **Purge button:** Purges the selected tracks from the Mixxx library, but does not remove them from your computer. This will delete all *metadata* Mixxx has for a track.

## 4.2.3 Skryté skladby

The Hidden Tracks view is accessible by expanding the Library tree item in the sidebar. Any tracks that were previously loaded into your library but were later set to be hidden from the library will appear here. Use the *Hide from Library* context menu item in the *tracks* view to temporarily hide selected tracks.

The features in detail:

- **Select All button:** Selects all tracks in the current view.
- **Purge Button:** Purges the selected tracks from the Mixxx library, but does not remove them from your computer. This will delete all *metadata* Mixxx has for a track.
- **Unhide Button:** Removes the selected tracks from the *Hidden Tracks* view and makes them available in the regular track lists again. The tracks will re-appear in every playlist or crate they were in before being hidden.

**Viz také:**

To learn how to permanently remove tracks from the Mixxx library using the *Hidden Tracks* view, see [permanently remove tracks](#).

## 4.3 Loading Tracks

Tracks can be loaded into a deck in several ways:

- Right-click the *library track table*: Right-clicking on a track in the table will present the options *Load in Deck 1* and *Load in Deck 2*, among others. Making either selection will load a track into a deck.
- By *Using a Keyboard* to load the selected track in the library track table. Use keyboard  $\uparrow$  and  $\downarrow$  arrow button to select a track,  $\text{Shift} + \leftarrow$  to load into Deck 1 and  $\text{Shift} + \rightarrow$  to load into Deck 2.
- Drag-and-drop from library track table: Dragging-and-dropping a track from the track table onto a waveform display will load the track into that deck.

- Drag-and-drop from deck to deck: Once you've loaded a track to deck, sampler, or preview deck, click on the *track title*, *track artist* or cover art and drag it to another deck or sampler. Note that you can also drop tracks onto Playlists and Crates in the sidebar, as well as into track tables of active Playlists or Crates.
- Drag-and-drop from external file browser: Dragging-and-dropping a track from an external file browser directly onto a waveform display in Mixxx will load that track. This function also works with some other applications. For example, on macOS, dragging-and-dropping a track from iTunes onto one of Mixxx's waveform displays will load it into that deck.

## 4.4 Finding Tracks (Search)

The search function filters the currently displayed list (e.g. a playlist, a crate, or even the whole library) for tracks that match your search query.

- Activate the search input field by clicking in it or pressing CTRL + F (Windows/Linux) or CMD + F (Mac).
- Type your search term(s). Mixxx filters the tracks as you type and shows only those matching the search term(s). Search terms can include an artist's name, a song title, BPM, etc.
- To clear the search string hit ESC or click the clear button right next to the input field.
- Hit TAB to cycle between the search and the list of results in the library. Use the ↑ and ↓ keys to scroll in the list of results.

---

**Poznámka:** If the search input field has focus, the Mixxx keyboard shortcuts are disabled, see *Using a Keyboard*.

---

### 4.4.1 Using search operators

Search operators allow you to form more complex search queries. They allow you to limit certain search terms to particular properties of your tracks.

Mixxx supports the following filters:

- **Text filtering:** album\_artist, album, artist, comment, composer, crate, genre, grouping, location, title

#### Examples

```
artist: "com truisse"
album: Danger
genre: Trance
title: foo
composer: foo
comment: foo
genre: hip-hop -genre: gangsta
```

---

**Poznámka:** It doesn't matter if you have space between the colon and the argument or not. Quotes must be used for multi-word text arguments.

---

- **Numeric filtering:** bitrate, bpm, played, rating, track, year

#### Examples

```
bpm:140
bpm: >140
year: <2010
bpm: >=140
rating: <=4
bpm: 140-150
played: >10
```

---

**Poznámka:** You can put a space after the colon but currently there must be no space between the operator and the number.

---

- **Special filtering:** key, duration, added, dateadded, datetime\_added, date\_added
  - Supports fuzzy matching of key searches. The following example lists tracks with harmonically compatible keys to C# minor.

```
~key:c#m
```

You can combine operators but there's no way to do an "OR" search right now. The following example lists all tracks by "Danger" over 3 minutes long that are rated 4 or 5.

```
artist: Danger duration:>3m rating:>=4
```

- Negative search filters. Use the – prefix as negation operator. The following example would find "hip-hop" from any year but 1990.

```
genre:hip-hop -year:1990
```

- Search for empty fields. The following example lists all tracks without a genre. It works for all text fields, including crates.

```
genre:""
```

### Examples



```
duration:2m10
duration:<2:10
duration:>1m35s
duration:>62
```

---

**Poznámka:** You can put a space after the colon but currently there must be no space between the operator and the number.

---

## 4.5 Previewing Tracks

To pre-listen to a track, activate the *Preview* column in a library view. Clicking the  icon in the library's *Preview* column loads the selected track into a special *Preview Deck* that will only output sound in the *headphones* channel. Click the  icon to stop the playback.

Alternatively, select a track from the track list of the Mixxx library, drag the track to the waveform view of the *Preview Deck* and click the *Play* button next to the waveform.

To display the *Preview Deck*, press CTRL + 4 (Windows/Linux) or CMD + 4 (Mac).

---

**Rada:** Click on the *Preview* column header to randomly sort tracks.

---

## 4.6 Edit metadata of audio files

Mixxx reads relevant *metadata* from the tracks and displays them in the library columns. You are free to edit most metadata, and Mixxx offers a number of different ways to do so. Note that some information can not be edited, such as bitrate, size, length, type, filename, and location.

---

**Poznámka:** Mixxx does not support editing the metadata of many tracks at a time (bulk editing).

---

**Varování:** Mixxx won't touch your audio files by default. Changes to a track's metadata will be saved to the Mixxx library, but **not** to the track itself.

If you wish to write modified track metadata from the library into file tags, check the box in *Preferences* ▶ *Library* ▶ *Track Metadata Synchronisation*.

### 4.6.1 Manual Edit

**Track Inline editing:** To enable inline editing in the *library* go to *Preferences* ▶ *Library* ▶ *Miscellaneous* and check the box „Edit metadata after clicking selected track“.

Select any track in the *library* and click on the field you want to edit. If the field is editable, it will become an editable text box. Enter a value and hit *Enter* when you are done.

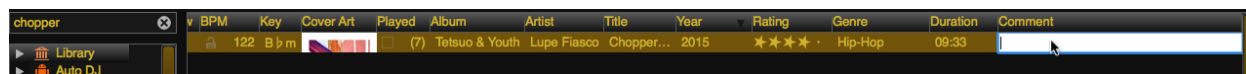


Figure2: Mixxx library - Inline editing

**Properties editor:** To open the editor right-click on a **single track** in the library or any track label in decks or samplers, then select *Properties* in the track menu. You can also double-click any track label in the decks or samplers to open the editor directly.

Add or change values in the editable fields, and save your changes as explained below.

- **OK:** Accept the changes and close the editor.
- **Apply:** Accept the changes and leave the editor open.



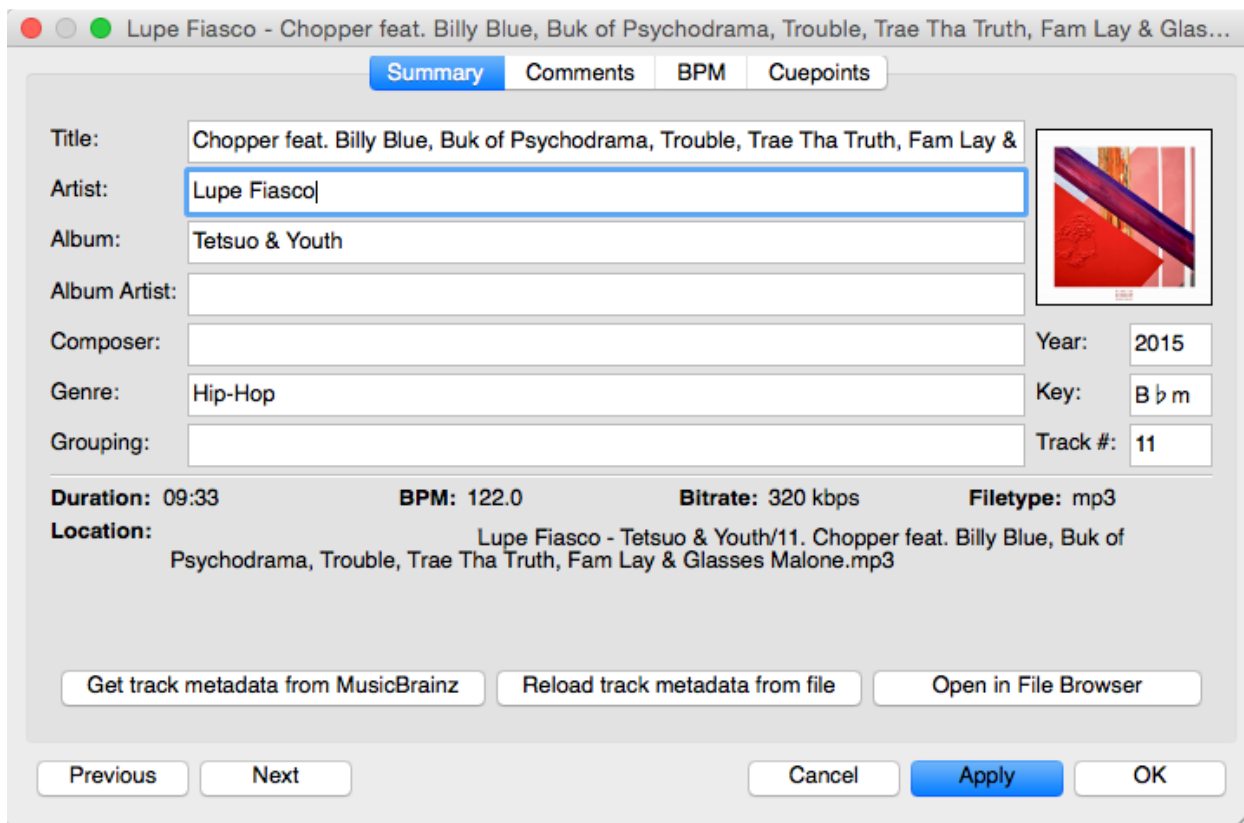


Figure3: Mixxx library - Properties editor

- **Cancel:** Discard the changes and close the editor.
- **Previous/Next:** Load the previous or next track in the current library view.
- **Import Metadata from File:** Prompts Mixxx to re-read the metadata of the selected track (such as when you have modified it externally with another application.)
- **Import Metadata from MusicBrainz:** Look up metadata on-line by searching the *MusicBrainz* database. See below.

## 4.6.2 Using the MusicBrainz online database

MusicBrainz is an *open-source* music encyclopedia that collects music *metadata* and makes it available to the public.

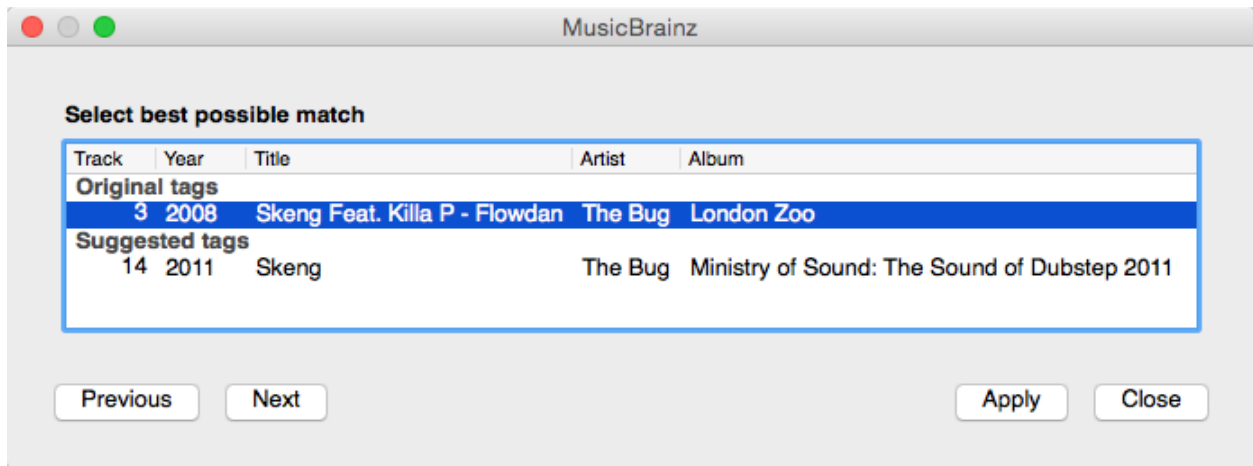


Figure4: Mixxx library - MusicBrainz Wizard

The MusicBrainz wizard in Mixxx allows searching the MusicBrainz database and applying the results to your tracks.

Click on a **single track** in the library and select *Get Metadata from MusicBrainz*. Mixxx fetches track data from the MusicBrainz database and displays the search results.

Select the best match from the search results by clicking on it in the list.

- **Apply:** Apply the selected MusicBrainz metadata to the track.
- **Close:** Close the wizard.
- **Previous/Next:** Load the previous or next track in the current library view and perform a MusicBrainz lookup on it as well.

### Fetching track metadata from MusicBrainz can fail:

- if Mixxx is unable to find the specified track in the MusicBrainz database,
- if Mixxx cannot reach the MusicBrainz servers (if they are down),
- or if you are not connected to the Internet.

---

**Rada:** The MusicBrainz service has been designed for identifying full audio files. If a track is less than 2 minutes long, identifying the file will likely fail. Identifying a layered mix-track or mash-up may produce false positives in the result list.

---

### 4.6.3 Using third-party software

If you have edited file metadata in third-party software you need to re-import the modified metadata into Mixxx manually by selecting the affected tracks, open the context menu, and then choose *Metadata* ▶ *Import From File Tags*.

---

**Rada:** Rescanning the library with *Library* ▶ *Rescan Library* in the menu on top of the application window will only detect newly added or removed tracks but no modifications of existing tracks. This is a known limitation.

---

#### Popular software to edit metadata of audio files include:

- [Mp3tag](#) (Windows)
- [Kid3](#) (Mac, Windows, Linux)
- [Picard](#) (Mac, Windows, Linux)

## 4.7 Auto DJ - Automate your mix



The Auto DJ queue is a special playlist that contains extra controls for automatic mixing. This is useful when taking a break from live mixing or for using Mixxx as media player.

The Auto DJ features in detail:

- **Shuffle button:** Randomizes the order of tracks in the Auto DJ playlist.
- **Add Random button:** Adds a random track from track sources (crates) to the Auto DJ queue. If no track sources are configured, the track is added from the library instead.
- **Skip track button:** Skips the next track in the Auto DJ playlist.
- **Fade now button:** Triggers the transition to the next track.
- **Transition time spin-box:** Determines the duration of the transition. A negative value will add a pause between tracks.
- **Selection Info label:** Displays the duration and number of selected tracks. Press `Ctrl + A` to get the total duration of the AutoDJ queue.
- **Enable Auto DJ button:** Toggles the Auto DJ mode on or off.

The *Skip track*, *Add Random* and *Fade now* buttons are only accessible if the Auto DJ mode is enabled. The Search field in the upper left corner is disabled in Auto DJ. By default, Auto DJ removes tracks after playing them but if you want it to play the same tracks over and over again, you can activate the *Auto DJ Requeue* option in *Preferences* ▶ *Auto DJ* ▶ *Re-queue tracks after playback*.

#### Auto DJ Crates

It is possible to add random tracks to the bottom of the Auto DJ playlist. The tracks are chosen from a set of crates that you have set as a source for Auto DJ, or from the standard library. Mixxx will normally try to select tracks that haven't yet been played. You can set a minimum number of tracks that are always available for selection no matter when they were last played in *Preferences* ▶ *Auto DJ* ▶ *Minimum available tracks in Track Source*.

#### Viz také:

For more information, go to the chapter [Auto DJ](#).

## 4.8 Playlists - Arranging tracks in a set order



Playlists are ordered lists of tracks that allow you to plan your DJ sets. Some DJs construct playlists before they perform live; others prefer to build them on-the-fly.

Playlists are not directly loadable into decks as Mixxx is primarily intended for live, attended performance use. However, you can right-click a playlist to queue it to *Auto DJ*, so in a sense you can „play“ it.


- **Create a playlist:** Right-click on the *Playlists* sidebar item and select *Create New Playlist*. Name the playlist and click *OK*. The total number of tracks and total duration are displayed next to the name.
- **Import playlists:** Right-click on the *Playlists* sidebar item and select *Import Playlist*. Select one or more playlists and click *OK*. Mixxx creates an import of these existing external playlists as new Mixxx playlists.
- **Add Tracks:** Add tracks to a previously created playlist by dragging-and-dropping a selection of tracks from the library or playlists onto the name of a playlist in the sidebar. Alternatively, use the right-click context menu in the library's *track list*.

---

**Rada:** If you have a track selected in the library, all of the visible playlist and crates that it is in are shown in bold.

---

Right-click on an existing playlist name to access the different features in the context menu:

- **Add to Auto DJ:** Adds the content of the playlist to the *Auto DJ* queue for automatic mixing.
- **Rename:** To rename a playlist, just enter a new playlist name and click *OK*.
- **Duplicate:** Sometimes you want to build a playlist based on an existing one. Select the playlist you would like to duplicate, choose *Duplicate Playlist*, name the new playlist and click *OK*.
- **Remove:** Removes an unlocked playlist. Tracks that were in the playlist remain in the library for later use.
- **Lock:**  This icon indicates a locked playlist. If a playlist is locked, you cannot add tracks, rename or delete the playlist. Choose *Unlock* from the context menu to unlock the playlist.
- **Analyze entire playlist:** Forces the analysis of the tracks in the playlist in the *Analyze* view.
- **Import playlist:** Import tracks from external playlists in various file formats to the selected playlist. For more information, go to *Using libraries from other software*.
- **Export playlist:** Export a playlist to various file formats such as *m3u*, *m3u8*, *pls*, *txt*, or *csv*. This allows you to use the data in other applications.
- **Export Track files:** This copies all of the tracks in a playlist into a folder such as an external USB stick. Only files are copied, not Mixxx-specific metadata like waveforms or cover art that is not embedded in the file.

## 4.9 Crates - Organizing tracks into collections




Crates are unordered collections of tracks, just like DJ cases holding your favorite records.

- **Create a crate:** Right-click on the *Crates* sidebar item and select *Create New Crate*. Name the crate and click *OK*. The total number of tracks and total duration are displayed next to the name.
- **Import crates:** Right-click on the *Crates* sidebar item and select *Import Crate*. Select one or more playlists and click *OK*. Mixxx creates an import of these existing external playlists as new Mixxx crates.

- **Add Tracks:** Add tracks to a previously created crate by dragging-and-dropping a selection of tracks from the library or playlists onto the name of a crate in the sidebar. Alternatively use the context menu in the library's *track list*.

Right-click on an existing crate's name to access the different features in the context menu:

- **Rename:** To rename a crate, enter the new crate name and click *OK*.
- **Duplicate:** Just like playlists you can duplicate an existing crate. Select the crate you would like to duplicate, choose *Duplicate Crate*, name the new crate and click *OK*.
- **Remove:** Removes an unlocked crate. Tracks in the crate are still available in the library for later use.
- **Lock:**  This icon indicates a locked crate. If a crate is locked, you cannot add tracks, rename or delete the crate. Choose *Unlock* from the context menu to unlock the crate.
- **Auto DJ Track Source:** Use this crate as a source for random tracks in Auto DJ, see *Auto DJ crates*.
- **Analyze entire crate:** Forces the analysis of the crate in the *Analyze* view.
- **Import crate:** Import tracks from external playlists in various file formats to the selected crate.
- **Export crate:** Export a crate to various file formats, such as *m3u*, *m3u8*, *pls*, *txt*, or *csv*. This allows you to use the data in other applications.
- **Export Track files:** This copies all of the tracks in a crate into a folder such as an external USB stick. Only files are copied, not Mixxx-specific metadata like waveforms or cover art that is not embedded in the file.

### 4.9.1 Crates vs. Playlists

Crates are unordered collections of tracks. Unlike playlists, they cannot contain duplicate entries and do not support arranging tracks within them.


Often DJs group tracks by genre, keep lists of favorite tracks, and/or prepare lists of tracks they want to play at a particular party. In these cases they rarely care about the order since they will choose it at the party based on the dance floor and mood. They certainly don't want duplicates in these lists. Crates serve these purposes. You can think of them like labels in GMail or Web 2.0 tags for your music.

If however you want to plan out a specific set and practice the transitions, you will want an ordered list of tracks. Playlists serve this purpose.

## 4.10 Computer - Loading tracks from anywhere on your computer



This mode works like a file manager and allows you to load tracks from anywhere on your computer and connected devices, that are not necessarily already in your Mixxx library.

Click the *Computer* sidebar item to navigate the computer and find your music. Depending on your *operating system*, your music will usually be found in the "My Music" or "Music" folder. Drag the files you want to import to the  *Tracks* icon or directly to the *Waveform Displays*.

---

**Poznámka:** Currently you can only drag files (not folders) to the Mixxx library.

---

Right-click on a folder and choose *Add to Library* to add the folder as an additional music directory. Mixxx will watch this directory and its subdirectories for new tracks.

**Viz také:**

For more information, go to [Changing music directories](#).

### 4.10.1 Quick Links - Bookmark your favorite folders

Using the *Quick Links* sub-menu you can bookmark folders for direct access. Click the *Browse* sidebar item and navigate to the folder you would like to bookmark. Right-click and choose *Add to Quick Links*. The folder is now pinned below the *Quick Links*. To un-pin that folder, right-click and choose *Remove from Quick Links*.

## 4.11 Nahrávání



In this section of the library you can start and stop recording your session as well as view previous recordings and the dates they were made.

**Viz také:**

For more information, go to [Recording Your Mix](#).

## 4.12 History - Keep track of your sessions




The history section automatically keeps a list of tracks you play in your DJ sets. This is handy for remembering what worked in your DJ sets, posting set-lists, or reporting your plays to licensing organizations. Every time you start Mixxx, a new history section is created. If you don't play any tracks during the session, it will be discarded.



This icon indicates the current session.

Click on the *History* icon in the sidebar to switch to the *History* view, then right-click on a session's name/date to access the different features:

- **Add to Auto DJ:** Adds the content of the session to the *Auto DJ* queue for automatic mixing.
- **Rename:** Rename a session. The default is the calendar date (YYYY-MM-DD).
- **Remove:** Remove a previous (unlocked) session. (You cannot remove the current session.)
- **Lock:** Protect a previous session against accidental merge or deletion. An  icon indicates a locked session.
- **Create new history playlist:** Split off the current history session and add a new session without having to restart Mixxx. The current history must contain at least one track for this option to be available.
- **Join with previous:** Append the selected history session to the end of the previous one.
- **Export playlist:** Export a session in various file formats. This allows you to use the data in other applications.

## 4.13 Analyze - Prepare your tracks

This section allows you to analyze your tracks in advance of loading them into a deck. Analyzing tracks requires considerable CPU power and may cause skips in the audio while performing, so it helps to have your tracks analyzed before you play. See *Analyze your library* for details.

## 4.14 Using libraries from other software

Supported libraries:

-  iTunes (Windows, macOS)
-  Traktor (Windows, macOS)
-  Rhythmbox (GNU/Linux)
-  Banshee (Windows, macOS, GNU/Linux)
-  Serato (Windows, macOS)
-  Rekordbox (Windows, macOS)

The external library views allow you to use music libraries you have created in these third-party applications. You can access music as well as playlists. If available, Mixxx automatically loads the external libraries from their default locations on your hard drive.

---

**Poznámka:** Playing a track from an external library will add it to your Mixxx library.

---

**Viz také:**

External libraries can be disabled under *Preferences* ▶ *Library*.

### 4.14.1 Using the iTunes/Traktor/Rhythmbox/Bashee/Rekordbox library

Right-click on the iTunes icon in the Library tree and select *Choose Library* to load the iTunes Music Library.xml from a different location. Select *Use Default Library* to reset.

- Right-click on a iTunes/Traktor/Rhythmbox/Banshee playlist and choose *Import Playlist* to import it into a new Mixxx playlist.
- If you have an iTunes configuration file (\*.xml) from a Windows or Mac partition mounted in Linux, you can load it and use your iTunes tracks and playlists as well.

### 4.14.2 Using the Rekordbox library

To read Rekordbox databases for CDJ/XDJ players on USB or SD devices, the database has to be exported to a USB or SD device with a FAT or HFS file system using the Rekordbox Export mode. Databases that have been moved to an external device via *Preferences > Advanced > Database management* in Rekordbox are not supported.

1. Plug your USB or SD drive in while Mixxx is running.
2. Click on the *Rekordbox* item in the library sidebar to scan for Rekordbox prepared USB drives. If no Rekordbox devices show up, you can click on *Check for attached Rekordbox devices (refresh)*.
3. Click on the desired removable device. All your folders and playlists will be revealed for you to play from.

### How Rekordbox hotcues, memory cues, and loops are imported

The following data is read:

- Folders
- Playlists
- Beatgrids
- Hot cues
- Memory cues
- Loops

Mixxx currently has one main cue point, one loop, and thirty eight hotcues. As such:

- All Rekordbox hotcues are mapped to Mixxx hotcues as expected, the first chronological Rekordbox memory cue is mapped to the Mixxx main cue.
- The first chronological Rekordbox loop is also mapped to the Mixxx loop.
- All loops (including the first loop), and all subsequent Rekordbox memory cues are appended as Mixxx hotcues following the previously imported Rekordbox hotcues.
- The hotcue and memory cue colors you assigned in Rekordbox are also imported, assisting in distinguishing between them.

Although additional imported Rekordbox loops only appear in Mixxx 2.3 as hotcues, their loop information is stored in the Mixxx library and will become usable in Mixxx 2.4.

### 4.14.3 Using the Serato library

Mixxx supports reading your Serato library and crates both from your local hard drive as well as portable USB drives. Smart crates are currently not supported.

Serato stores information such as hot cues, track color and beatgrid in the file tags. Regardless of whether a track is loaded into deck from the library or directly from the file browser, Mixxx will import that information automatically if present.

Saved loops are imported as well, but since Mixxx is lacking support for multiple saved loops, only the first loop is usable in Mixxx. All other saved loops can be used like regular hotcues for now.

Waveforms, Gain values and „Flips“ are not imported from Serato.

---

**Poznámka:** Due to differences in the metadata format, importing Serato information from Ogg Vorbis files is currently not supported.

---



To import Serato libraries from external USB drives:

1. Copy music library from Serato into external USB drive
2. Plug in USB drive while Mixxx is running
3. Click on *Serato* in the library. All USB drives containing a Serato library will be detected automatically.
4. For each removable device, the library and all crates from Serato will be accessible.

If you have already added a track containing Serato Metadata to your Mixxx library in a Mixxx version that didn't support reading that data, you can reimport it from the file's tags via the track context menu.

**Varování:** This action will clear your existing cuepoints in Mixxx if the file contains Serato Metadata

To reimport metadata from files:

- Right click on the track to show the context menu
- Select *Metadata* ▶ *Import From File Tags*.

**Rada:** Decoders may slightly detect different track start and end times for files from different sources, this could possibly be because of the presence of countless encoders and decoders for MP3 and M4A/AAC. This problem may cause your cues to be shifted up by a few milliseconds. Mixxx might not always be able to mitigate this problem, but it's possible to shift all cues for a track at once as a workaround.

## 4.15 Removing tracks from the library

Removing tracks from the Mixxx library will **not** physically delete them from your drive. However it does delete extra metadata Mixxx might have (such as hotcues and the beatgrids), and removes links to playlists or crates.

1. Click the *Library* item in the sidebar.
2. Find and select the tracks you want to remove, perform a right-click on them and select *Hide from Library* from the context menu.
3. Expand the *Library* item in the sidebar and click on the *Hidden tracks* sub-item. All tracks that were set to be hidden from the library will appear here.
4. Select the tracks you want to remove, or use the *Select All* button.
5. To confirm you want to permanently remove these tracks from the library, click *Purge*.

**Rada:** If you want to hide or purge only tracks from a specific folder you may use the `location: search operator`, or `sort the table` by „Location“.

**Rada:** If you later decide to add some of the deleted tracks back, import them to the Mixxx library again, see *Changing music directories*.

## 4.16 Compatible file formats

Mixxx supports a variety of audio file formats:

Beztrátová	Ztrátová
<i>FLAC</i> (flac)	<i>MP3</i> (mp3)
<i>Wave</i> (wav)	<i>Ogg Vorbis</i> (ogg)
<i>Aiff</i> (aiff, aif)	<i>AAC</i> (aac, m4a, mp4)
	<i>Opus</i> (opus)

Lossless formats preserve every detail of the original recording and are recommended for the best sound quality. The *FLAC* format compresses the files to take less space on your computer without sacrificing audio quality. *WAV* and *AIFF* files are uncompressed and take up much more space.

Lossy formats take up less storage space on your computer than lossless formats at the expense of reducing sound quality. Assuming an average track length of about 4 minutes, a 1 TB hard drive or solid state drive can store about 42000 *FLAC* files or about 115000 *MP3* files at 320 kbps.

There is no benefit to converting lossy file to lossless formats. The detail lost when making the lossy file cannot be recovered.

*AAC* (*M4A*) is supported on Windows Vista and macOS 10.8 onwards. The [Platform Update Supplement](#) is required for Windows Vista.

On Linux, *AAC* playback is disabled by default due to licensing restrictions. To enable the playback of *AAC* files, [build Mixxx from source with m4a/AAC files support](#).

**Varování:** *DRM* protected files, such as *m4p* files purchased in the iTunes Store in 2009 or earlier, are not supported.

**Varování:** *ALAC* files use the same container *MP4* container format and file extensions (*m4a*, *mp4*) as *AAC* files, but Mixxx cannot play *ALAC* files. It is recommended to convert *ALAC* files to *FLAC* for use in Mixxx. This will not reduce the sound quality of the files.

## 4.17 Music from audio CDs

Mixxx can not play music directly from audio CDs. Use a CD ripping program to copy the audio from the CD to files on your computer. It is recommended to save the ripped files in the *FLAC* format. Many different programs are available to rip CDs, but not every program is capable of saving them to *FLAC*.

# CHAPTER 5

---

## DJ Hardware

---

Although Mixxx can be used with just a laptop or desktop computer, fully taking advantage of Mixxx's features requires specialized DJ hardware. Depending on your budget and application area, your setup and requirements may vary. This chapter provides general background information about various types of DJ hardware.

To make DJing with Mixxx a fun experience, at least the following pieces are needed:

- speakers to play the music to the audience,
- an *audio interface* with 4 mono/2 stereo output channels (or 2 mono output channels + a *DJ splitter cable*),
- *headphones* for prelistening tracks while another track is playing on the speakers, and
- audio cables and adapters to connect the equipment to your computer.

Additionally, you might consider getting the following equipment:

- a *controller*,
- *turntables*, an audio interface with phono inputs and *timecode vinyl*,
- a laptop/controller stand,
- a surge protector to protect the equipment from voltage spikes,
- cases for laptop, controller and headphones, and
- custom fader and knob caps to customize your gear.

### **Viz také:**

Instructions how to use DJ Controllers that work out-of-the-box with Mixxx can be found in the *Hardware Manuals* section of this manual. The [Mixxx DJ Hardware Guide](#) on the Mixxx Wiki also lists additional devices, including information about their prices, features, and compatibility with Mixxx.

## 5.1 Řadiče

DJ controllers are devices with knobs, faders, buttons, and jog wheels to control DJ software such as Mixxx. Controllers allow quick access to different controls while providing tactile and visual feedback indicating the state of each control, such as the position of a knob or whether a switch is on or off. This allows you to focus on manipulating the music without needing to look at your computer screen all the time. Controllers also allow using two hands to manipulate two different controls at the same time, which is required for many mixing techniques.

DJ controllers typically do not do any actual audio processing. Instead, they send signals (typically *MIDI* or *HID* over a USB cable) to the computer to instruct DJ software how to manipulate the audio. Many DJ controllers include an *audio interface* with 2 separate stereo outputs built into the device. This allows the DJ to transport and setup only one piece of hardware in addition to a laptop. Some devices can be used as both a controller and *hardware mixer*.

Mixxx can work with any *MIDI* or *HID controller* as long as there is a mapping file to tell Mixxx how to understand the controller's signals. Mixxx comes bundled with a number of MIDI and HID mappings, which are listed in the *Hardware Manuals* section. For controllers that Mixxx does not yet support, you can [search the forum](#) to see if anyone has started a mapping. You can also start one yourself using the information in the [Controller Mapping Documentation](#) on the wiki.

### Viz také:

[Using MIDI/HID Controllers](#) describes how to configure Mixxx to use controllers.

## 5.2 Audio Interfaces

An audio interface (also known as a „sound card“, although few of them are shaped like cards anymore) is a device that allows a computer to send output to and receive input from audio equipment. Whether using *internal or external mixing*, it is recommended to use a single audio interface with at least 4 independent output channels (2 separate stereo pairs).

The headphone jack on most laptops is not a second audio output. Rather, plugging headphones into the jack simply redirects the laptop's single stereo output from its speakers to your headphones. A *splitter cable* can be used to separate the stereo output of a headphone jack into two separate mono outputs for headphone cueing, but it is recommended to use a sound card with at least four mono outputs (for two stereo pairs). Such sound cards tend to be higher quality than those built into laptops and allow your mix to be enjoyed in stereo by your audience.

Unlike some proprietary DJ systems, Mixxx can use any audio interface and any *MIDI* or *HID* controller that your *OS* has drivers to use. If your controller has an integrated audio interface, you may choose to use a different audio interface for higher quality audio. Mixxx can also use multiple audio interfaces simultaneously.

### 5.2.1 Audio Interface Considerations

This section provides background information to help you choose an audio interface to use with Mixxx.

#### Bit Depth and Sample Rate

Most music is published with a bit depth of 16 bits at a sample rate of 44.1 kHz because this is all that is needed to store all the detail of music in digital form.

Bit depth determines the possible dynamic range of the signal. 16 bits is more than enough for playing back music. While 24 bits is helpful for recording, [it is useless for playback](#).

Half the sample rate determines the maximum frequency that can be represented by the signal. Humans generally can't hear frequencies above 20 kHz, so a sampling rate of 44.1 kHz, representing a maximum frequency of 22.05 kHz, is fine for playback. Higher sample rates like 88.2 kHz and 96 kHz can be helpful to reduce aliasing distortion when recording, but have no benefit for playback and make your computer work harder.

For a more thorough and technical explanation of why 16 bits at 44.1 kHz is all that is needed for playback, read [24/192 Music Downloads Are Very Silly Indeed](#).

## Specifications

When considering specifications, higher dynamic range, higher signal-to-noise ratio (SNR), higher maximum output level, lower THD+N (Total Harmonic Distortion + Noise; look for a more negative dB value or smaller percentage), and lower crosstalk (more negative dB value) are better. Cheap audio interfaces tend to not have these specifications published.

## Connector and Cable Types

If you are unfamiliar with professional audio equipment, read Digital DJ Tips' [Essential Guide to Audio Cables for DJs](#) to understand the different kinds of connectors on audio interfaces. It is better to use an audio interface with balanced outputs, especially if you will run long cables directly into an amplifier or active speakers without going through a *hardware mixer*. Balanced signals reject interference and are less susceptible to ground loop hum issues (which can be a problem when plugging unbalanced gear into separate power sources).

However, most venues have DJs plug into *hardware DJ mixers*, which typically only have RCA inputs (RCA cables cannot be balanced). Most home/computer speakers and amplifiers have RCA and/or 1/8" TRS stereo inputs. Most live sound mixers have balanced 1/4" TRS mono inputs. If you need to interconnect balanced and unbalanced gear, refer to this [guide from Presonus \(deleted, via archive.org\)](#) and this [guide from Rane \(deleted, via archive.org\)](#).

## Number of Channels

Audio interfaces sometimes have multiple connectors for a single channel, resulting in more connectors than channels. So, not every connector can send or receive an independent signal. For example, some audio interfaces made for DJing have 4 output channels with 4 mono output connectors and 1 stereo headphone connector. This does not mean that the audio interface can send out 6 different signals at the same time; rather, the signal on 2 of the mono outputs and the stereo headphone output would be the same. Also, many controllers have separate master and booth outputs with independent volume controls, but they both play the same signal.

## Vinyl Control and Phono Preamplifiers

Turntables output low voltage (phono level) signals that need to be amplified to line level before most audio equipment can work with them. So, if you want to use *Ovládání vinylovou gramodeskou*, sometimes referred to as a Digital Vinyl System (DVS), it is best to have phono preamplifiers (one for each deck) somewhere between your turntable and sound card to boost the turntable's phono level signal to line level. Mixxx can amplify phono level signals in software, but it is better to do it in hardware. The phono preamp can be in the turntable, in the audio interface, or a stand alone device. Most audio interfaces do not have phono preamps; these are generally found on audio interfaces specifically made for controlling DJ software with timecode vinyl. *Mixers* with audio interfaces have phono preamps on their deck inputs, but not necessarily on every deck input. Many higher-end all-in-one controllers also include audio interfaces with phono preamps.

## Driver Support

Unlike some proprietary DJ programs, Mixxx works with any audio interface that is compatible with your operating system — including for timecode vinyl (DVS) use.

USB Audio Class compliant audio interfaces should work out of the box on all operating systems supported by Mixxx. This is the case for the integrated audio interfaces in most DJ controllers. Sound cards that aren't USB Audio Class compliant need a driver for each OS.

---

**Poznámka:** On Windows, you should install the manufacturer's drivers even for USB class compliant audio interfaces to be able to use the recommended ASIO sound API.

---

---

**Rada:** If you are considering buying a controller or audio interface, the easiest way to tell if it is USB class compliant before you buy it is to search for macOS drivers for the device on the manufacturer's website. If it is advertised as compatible with macOS or iOS but there are no drivers to download for macOS, the device is USB class compliant.

---

## 5.3 Mixers

Mixers are devices that combine audio signals. DJ mixers are different from live and studio mixers because they have multiple stereo channels with phono preamplifiers for connecting *Turntables*. It is conventional to use a DJ mixer with *Ovládání vinylovou gramodeskou*, but vinyl control can be used without a hardware mixer.

Using Mixxx with a DJ mixer requires an audio interface with at least 4 mono outputs (2 stereo pairs) to send Mixxx's decks to the mixer's stereo channels. Some DJ mixers have a USB *audio interface* built into them. This lets Mixxx send unmixed audio files directly to the mixer without needing a separate stand-alone audio interface.

Often DJs who use DJ software with internal mixing send their master output to a hardware mixer. This can be helpful to send the mixed signal to both a main speaker output for the audience and booth speakers for the DJ with separate gain controls for each output. It also facilitates smooth transitions between DJs.

However, using an external mixer with internal mixing is not necessary and reduces the sound quality. Each piece of equipment an audio signal passes through reduces the sound quality, so avoiding unnecessary equipment in the signal path can provide better sound quality. Many DJ controllers provide separate master and booth outputs with independent volume controls. Alternatively, a sound card with at least 6 output channels can be used with Mixxx's *Booth* output.

Many people confuse „analog mixers“ and „hardware mixers“, but these are not the same. Many hardware mixers process audio digitally with specialized signal processors as opposed to a general purpose CPU like those in laptop and desktop computers.

### Viz také:

*Recording And Broadcasting With An External Hardware Mixer*

## 5.4 Turntables

Turntables are mechanical devices that play music recordings cut into vinyl phonograph records. Before software like Mixxx was available, the art of DJing originated with turntables and *DJ mixers*. Special vinyl records can be used with turntables to control the playback of digital files in Mixxx as if the digital file was pressed onto the vinyl record.

### Viz také:

*Ovládání vinylovou gramodeskou*

## 5.5 CDJs

CDJs are devices that play digital audio files and have controls from manipulating the playback of the audio. Originally CDJs played audio CDs, but new devices typically use USB storage drives or SD cards. Some CDJs can be plugged into a computer with a USB cable to function as a *controller* for Mixxx.

## 5.6 Microphones

Microphones convert changes in sound pressure in air to changes in voltage (an analog audio signal). Mixxx can work with any microphone that can be plugged into your *audio interface*.

### Viz také:

*Using Microphones*

## 5.7 Headphones

Being able to listen to what you will play next in headphones before your audience hears on the main speakers or your broadcast stream is an essential part of DJing. DJ headphones are generally better quality than typical consumer headphones. Importantly, they are designed to have good isolation, meaning they block outside sound. This allows the DJ to focus on the sound in the headphones better without having to turn up the volume to drown out sound from the monitors, PA system, and chatter. Also, DJ headphones have hinges or flexible headbands that allow the DJ to take off one headphone cup to hear the monitor or PA output better in one ear and compare it to the headphone output in the other ear.

When shopping for headphones, consider how well they isolate sound and their durability. The most frequent places that headphones break are the cable attachment point and the headband. Look for headphones with detachable cables, metal headbands, and other individually replaceable parts. Avoid Bluetooth headphones because these may be unreliable and will always reduce sound quality. Active noise canceling headphones also tend to have worse sound quality than other headphones of a comparable price. DJ headphones with a closed design block out a lot of sound simply by being closed.

Whether to get on-ear or over-ear headphones is a matter of personal preference. Everyone's head is different, so headphones that are comfortable for a friend or a reviewer may not be comfortable for you.

In-ear monitors (IEMs) isolate sound better than headphones but are more expensive. They usually cost more than double the price of good DJ headphones with comparable quality. They may be a sound investment for musicians who play often in very loud environments to preserve their hearing.

## 5.8 Splitter Cables

DJ splitter cables are the cheapest way to get two separate sound outputs from your computer. These plug into the onboard audio interface built into computer motherboards and split the stereo signal into two separate mono signals. However, onboard audio interfaces are not good quality, and you lose the stereo effect of hearing different sounds arranged in space.

Devices marketed as „headphone splitter“ instead of DJ splitters duplicate one stereo signal in two jacks. These cannot be used for headphone cueing. Also, generic stereo-to-mono splitter cables or adapters typically have two mono jack outputs. Plugging headphones or stereo speakers into a generic stereo-to-mono splitter will only play sound on one side of the headphones or speakers.



---

## Example Setups

---

This chapter presents some example hardware setups with instructions for how to configure Mixxx to work with them.

**Viz také:**

- *DJ Hardware*: background information about different types of equipment.
- *Configuring sound input and output*: details about each available input and output option.
- *Using Microphones*: details for configuring Mixxx to work with microphones.
- *DJ Hardware Guide* lists specific devices with information about their prices, features, and compatibility with Mixxx.

## 6.1 Computer Only

The built-in audio interface on most computers comes with a single stereo line-out and a microphone input. This can be set up with Mixxx in two ways:

### 6.1.1 Practicing Mixing / Broadcasting

This setup can be used with a single audio output for speakers or headphones to practice DJing without investing in any equipment other than a computer and a set of speakers or headphones. With this configuration, use the *Pre/Main Knob* to switch back and forth between the Master mix and *PFL* signal. This setup cannot be used to play for a live audience because the Master mix is not sent to an audio interface output. However, it can be used by radio DJs who only need to send the Master mix to a broadcast audience.

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab
3. From the *Headphones* drop-down menu, select your built-in audio interface, then *Channels 1-2*. Although this is labeled *Headphones*, you can also plug speakers into the output jack of your computer.
4. Depending on your audio interface, you can specify a *microphone*

- a. Use your built-in microphone or connect a microphone to your computer
  - b. Check that your microphone is detected by your *OS*
  - c. Open the *Input* tab
  - d. From the *Microphone* drop-down menu, select the input your microphone is connected to
5. Click *Apply* to save the changes.

## 6.1.2 Playing for an Audience

This setup allows for playing to an audience without headphone *cueing*.

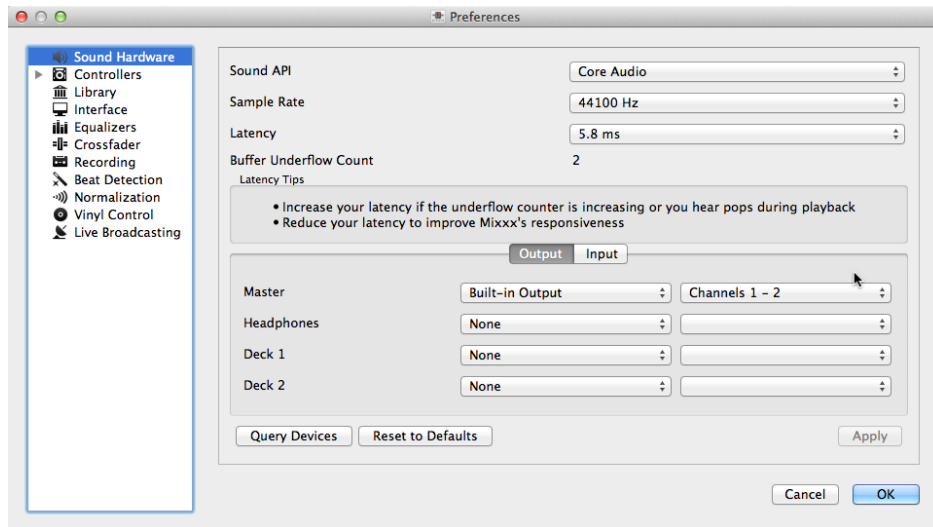


Figure 1: Using Mixxx with your built-in sound card

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab
3. From the *Master* drop-down menu, select your built-in audio interface, then *Channels 1-2*
4. Depending on your audio interface, you can specify a *microphone*
  - a. Use your built-in microphone or connect a microphone to your computer
  - b. Check that your microphone is detected by your *OS*
  - c. Open the *Input* tab
  - d. From the *Microphone* drop-down menu, select the input your microphone is connected to
5. Click *Apply* to save the changes.

## 6.2 Laptop and Splitter Cable

The cheapest way to DJ and *cue* with headphones uses a stereo-to-mono DJ splitter cable (also known as a “Y cable”) plugged into the headphone jack of a laptop or other computer. This cable divides the stereo output from a single jack into two separate mono signals.

This setup allows you to start DJing without having to invest in expensive equipment. However, because the signals are in mono, you and your audience will not be able to enjoy music producers’ positioning of sound in space (although many club PA systems are wired in mono anyway). Furthermore, audio interfaces built into computers are usually low quality and often pick up interference from other components of the computer.

### Using a built-in audio interface and a stereo-to-mono splitter

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab
3. Check that your headphones are plugged into the side of the cable with a headphones symbol and that the speakers are plugged into the side of the cable with a speaker symbol.
4. From the *Master* drop-down menu, select your built-in audio interface, then *Channel 1*
5. From the *Headphones* drop-down menu, select your built-in audio interface, then *Channel 2*
6. Click *Apply* to save the changes.

**Varování:** Most splitter cables, particularly those marketed as headphone splitter cables, output the same stereo signal to two headphone jacks and will not work for this DJ setup. Also, if you use a generic stereo-to-mono splitter adapter, you may only be able to hear out of one side of your speakers and headphones. DJ splitter cables allow you to hear the same mono signal on both sides of your headphones and speakers.

## 6.3 Laptop and External USB Audio Interface

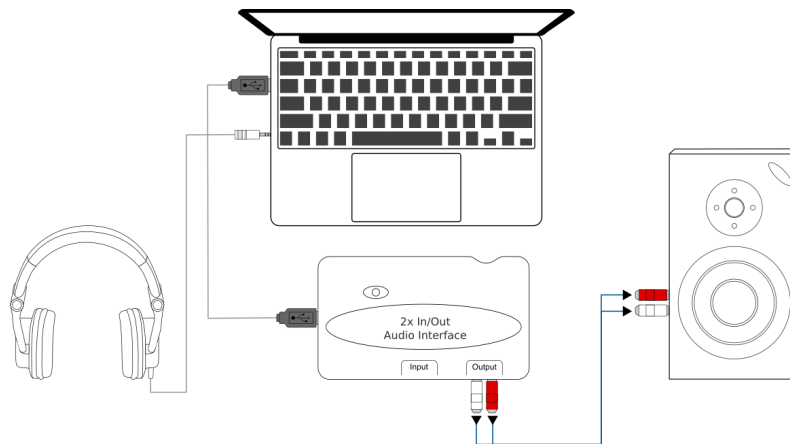


Figure2: Using Mixxx together with an external audio interface

### Using two audio interfaces (built-in + external)

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab

3. From the *Master* drop-down menu, select the external audio interface, then *Channels 1-2*
4. From the *Headphones* drop-down menu, select the built-in audio interface, then *Channels 1-2*

#### Using an external multi-channel audio interface

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab
3. From the *Master* drop-down menu, select the external audio interface, then *Channels 1-2*
4. From the *Headphones* drop-down menu, select the external audio interface, then *Channels 3-4*
5. Click *Apply* to save the changes.

## 6.4 Laptop and MIDI or HID Controller

### Viz také:

The *Řadiče* section of the *DJ Hardware* chapter has background information about controllers.

### 6.4.1 Controllers with an integrated audio interface

Many DJ controllers include an integrated *multi-channel* audio interface. A setup with this kind of controller may look like the diagram below:

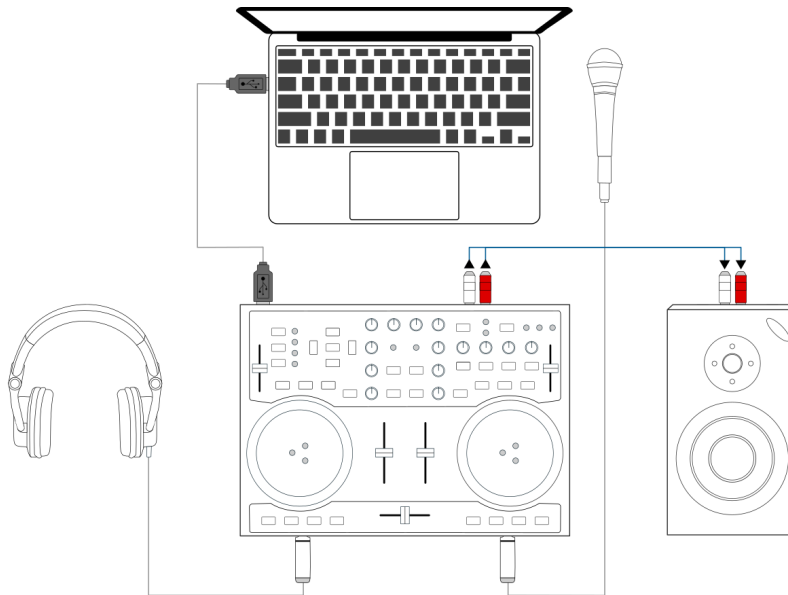


Figure3: Using Mixxx together with a DJ controller and integrated audio interface

#### Using a controller with an integrated multi-channel audio interface

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab
3. From the *Master* drop-down menu, select your controller's audio interface, then *Channels 1-2*
4. From the *Headphones* drop-down menu, select your controller's audio interface, then *Channels 3-4*

5. Click *Apply* to save the changes.

## 6.4.2 Controllers without an integrated audio interface

Controllers without an integrated audio interface can be used together with a USB audio interface as depicted in the diagram below:

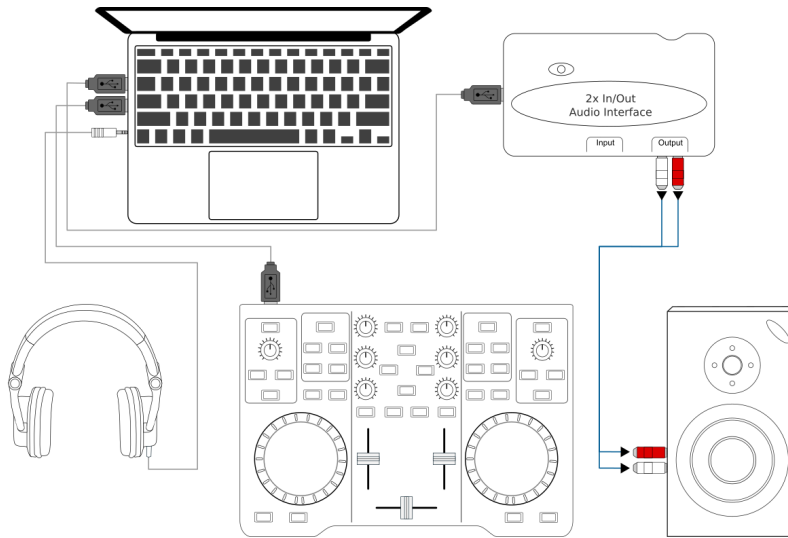


Figure4: Using Mixxx together with a DJ controller and external audio interface

### Using a controller without an integrated audio interface

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Output* tab
3. From the *Master* drop-down menu, select the external audio interface, then *Channels 1-2*
4. From the *Headphones* drop-down menu, select the built-in audio interface, then *Channels 1-2*
5. Click *Apply* to save the changes.

## 6.5 Laptop, External Hardware Mixer and Vinyl Control

This setup allows DJs to use the techniques of DJing with vinyl record turntables combined with the portability and flexibility of a laptop computer. Instead of carrying crates of records or CDs, DJs can carry their entire music collection on their laptop. In addition to a laptop and headphones, this setup requires an audio interface with at least two pairs of stereo inputs and outputs as well as a pair of *timecode* records.

### Viz také:

Go to the chapter *Ovládání vinylovou gramodeskou* for detailed information.

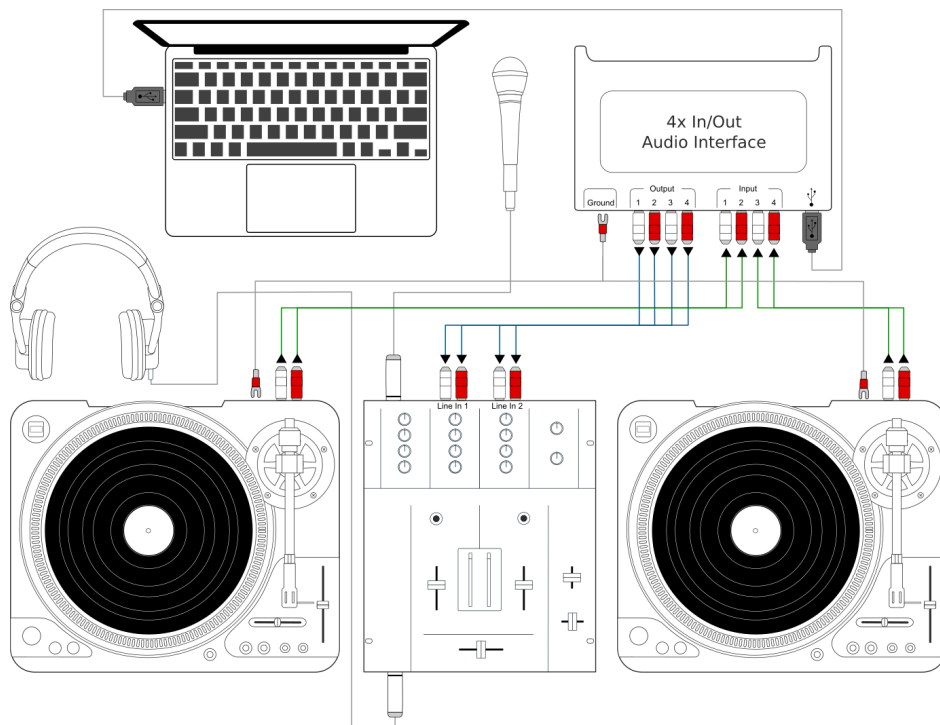


Figure5: Using Mixxx together with turntables and external mixer

Mixxx can be controlled with a keyboard, a mouse, *MIDI/HID* controllers, time-code records/CDs, or a combination of these. The choice usually depends on your budget or style of DJing.

### 7.1 Using a Mouse and Touchpad

Use your mouse to interact with the controls in the *Mixxx User Interface* on your computer screen. You can perform actions by pointing and clicking with your mouse.

- **Single Left Click:** Performing a left mouse click on buttons triggers the default action for that control. Moreover if you click on a value in the track list of the Mixxx *library*, you can change the value. For example the year of a track.
- **Single Right Click:** On most buttons, performing a right click triggers a different action than a left click. You can return any knob or fader to its default value by right-clicking that control.
- **Mouse Drag:** Click on a knob or fader, hold the left mouse button and move your cursor to change the value of the control. Use mouse drag to drop tracks from the file list to the decks, crates, playlists, etc.
- **Mouse Wheel:** Just like Mouse Drag, you can change knob and fader values by placing the cursor over the control and moving the mouse wheel.

---

**Rada:** If you hover with the mouse over a control (e.g. the crossfader) in the Mixxx user interface, the tooltip lists the left- and right-click action respectively.

---

---

**Rada:** On a Macbook or a computer with a touchpad that lacks mouse buttons, a two-finger click gesture is often treated as a right click.

---

**Viz také:**

Using the Mouse drag/wheel on the waveforms you can adjust zoom and playback rate of the tracks. For more information,

go to *Waveform Displays*.

## 7.2 Using a Keyboard

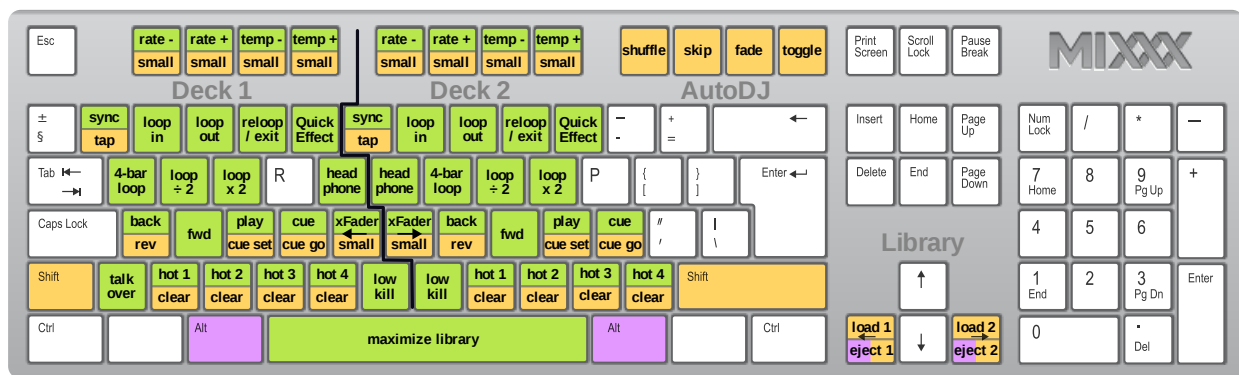


Figure 1: Mixxx Keyboard shortcuts (for en-us keyboard layout)  
[Download the image](#)

Controlling Mixxx with a keyboard is handy. Unlike mouse control, the keyboard allows you to manage things simultaneously. For example, you can start playing a track on deck 1 whilst stopping deck 2 at the same time.

The default mapping for English keyboards is depicted in the figure above. It is divided into a left-hand side for deck 1 and right-hand side for deck 2. Please note that you can also access the functions through Mixxx's interface.

For some user groups, like those using *controllers* or *vinyl control*, it might be useful to enable/disable the keyboard mappings at runtime. You can do so by clicking *Options* ▶ *Enable Keyboard Shortcuts*.

---

**Rada:** If you hover with the mouse over a control (e.g. the crossfader) in the Mixxx user interface, the tooltip lists the keyboard shortcuts of the control among a description of the control.

---

### Viz také:

For a list of default shortcuts, go to *Keyboard mapping table*.

### 7.2.1 Customizing the keyboard mapping

Mixxx allows you to customize the keyboard control. For more information, go to:

- *Making a Custom Keyboard Mapping*



## 7.3 Using MIDI/HID Controllers

Mixxx supports MIDI and HID *controllers* which are external hardware devices used to control audio applications. Many DJs prefer the hands-on feel of a controller with Mixxx because it can feel similar to using a real mixer and turntables or *CDJ*.

### 7.3.1 Loading a controller mapping

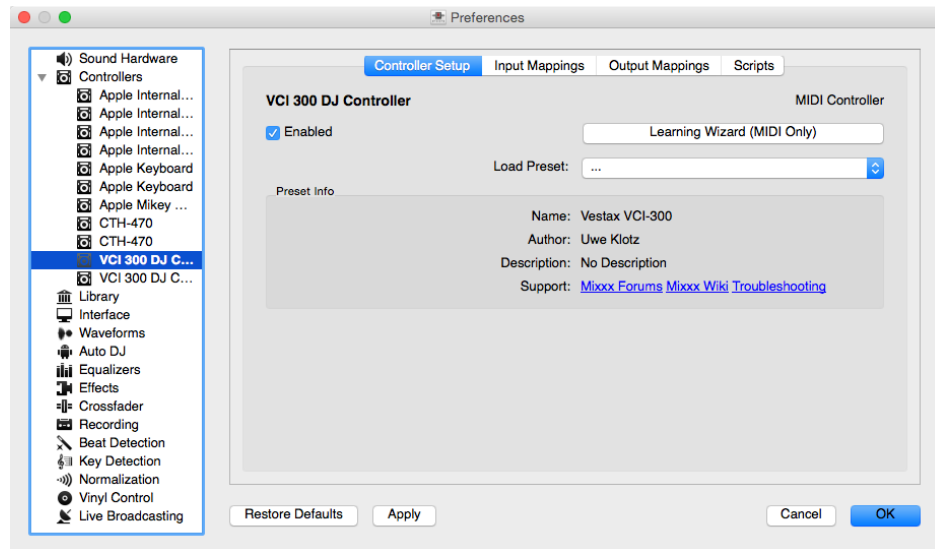


Figure2: Mixxx Preferences - Loading a controller mapping

Without loading the correct mapping, your controller does not work with Mixxx.

1. Connect your controller(s) to your computer
2. Start Mixxx
3. Go to *Preferences* ▶ *Controllers*
4. Select your device from the list of available devices on the left, and the right pane will change
5. Select the mapping for your controller from the *Load Mapping* drop-down menu
6. Make sure that the *Enabled* checkbox is ticked
7. Click *Apply* and Mixxx can now be controlled by your controller(s).
8. Repeat step 4-7 for any of the controllers you want to use

---

**Rada:** If you do not find a connected device in the list, make sure to install any necessary drivers for that device.

---

## 7.3.2 Supported controllers

Mixxx can use any *MIDI/HID* controller that is recognized by your *OS* (some may require drivers), as long as there is a MIDI/HID mapping file to tell Mixxx how to understand it. Mixxx comes bundled with a number of mappings for various devices. There are two levels of controller mappings:

- **Mixxx Certified Mappings:** These mappings are verified by the Mixxx Development Team.
- **Community Supported Mappings:** These mappings are provided and have been verified as working by the Mixxx community, but the Mixxx Development Team is unable to verify their quality because we don't have the devices ourselves. They might have bugs or rough edges.

If you run into issues with any of these mappings, please file a *bug report* on our [Bug Tracker](#) or tell us about it on our mailing list, forums, or *IRC* channel. Device support varies for each supported *OS*, so make sure to consult the documentation of the device.

---

**Rada:** Additional mappings are available in the [Controller mapping forum](#).

---

### Viz také:

Before purchasing a controller to use with Mixxx, consult our [Hardware Compatibility wiki page](#). It contains the most up-to-date information about which controllers work with Mixxx and the details of each.

## 7.3.3 Installing a mapping from the forum

To use a controller mapping that did not come bundled with Mixxx, place the controller mapping in your User Mappings folder. This is the folder called „controllers“ inside your User Preferences folder. To open your User Mappings folder, go to *Preferences* ▶ *Controllers* and click the *Open User Mapping Folder* button. Alternatively, navigate to the folder in your *OS* file browser:

- GNU/Linux: `/home/<username>/.mixxx/controllers`
- macOS: `/Users/<username>/Library/Containers/org.mixxx.mixxx/Data/Library/Application Support/Mixxx/controllers`
- Windows Vista and newer: `C:\Users\<username>\AppData\Mixxx\controllers`
- Windows XP: `C:\Documents and Settings\<username>\AppData\controllers`

On Windows, the `AppData` folder is hidden, so if you have not already, you will need to set Windows Explorer to show hidden files and folders (how to do this varies with each version of Windows).

If the mapping was posted as a `.zip` file, first extract the `.zip` file by right clicking on it in your file browser and selecting the option for extracting an archive or compressed folder. Copy the `.xml` file into your User Mapping folder. If there is a `.js` file, also copy that to your User Mapping folder. Restart Mixxx and *load the mapping*.

## 7.3.4 Map your own controller

There is no mapping available for your controller or you want to change an existing mapping? You can map your controller by using the Controller Wizard or take full control with the MIDI Scripting support in Mixxx.

### Viz také:

Go to [Adding support for your MIDI/HID Controller](#) for detailed information.

## 7.4 Using Timecode Vinyl Records and CDs

*Vinyl control* allows a user to manipulate the playback of a track in Mixxx using a turntable or *CDJ* as an interface. In effect, it simulates the sound and feel of having your digital music collection on vinyl. Many DJs prefer the tactile feel of vinyl, and vinyl control allows that feel to be preserved while retaining the benefits of using digital audio.

**Viz také:**

Go to *Ovládání vinylovou gramodeskou* for detailed information.

---

## Ovládání vinylovou gramodeskou

---

Vinyl control allows a user to manipulate the playback of a track in Mixxx using a real turntable as a controller. In effect, it simulates the sound and feel of having your digital music collection on vinyl. Many DJs prefer the tactile feel of vinyl, and vinyl control allows that feel to be preserved while retaining the benefits of using digital audio.

### 8.1 How does it work?

Vinyl control uses special *timecode* records or CD's which are placed on real turntables or *CDJ*. The audio output of the turntables is plugged into a audio interface connected to a computer on which Mixxx is running. When a record is played on one of the attached turntables, Mixxx reads the timecode from the record and uses the information to manipulate whatever track is loaded.

### 8.2 What do I need to use it?

It is possible to use Mixxx's vinyl control with a variety of setups. Unlike some proprietary DJ software, Mixxx can use several types of timecode media and any audio interface that your *operating system* has drivers to use. Optionally, any *MIDI/HID controller* can be used together with timecode control. The basic setups for timecode control are:

#### Setup 1: Vinyl DJ

2-4 timecode vinyls, 2-4 turntables with phono pre-amplifiers (or line-level output), and 2-4 stereo sound inputs.

Your audio interface inputs must match the turntable outputs. Turntables output very quiet phono-level outputs that must be amplified for Mixxx to use. You should either find a audio interface that has phono pre-amplifiers built in (one for each deck), use standalone phono pre-amplifiers, or use turntables with built in phono pre-amplifiers.

You can try skipping the phono pre-amps if you use the software pre-amp in Mixxx on the Vinyl Control preferences pane. This may not work for everyone. Line-level signals are preferred and recommended.

For turntables a typical setup is depicted in the figure below. First, connect the RCA cables from the turntables to the inputs on your audio interface. Second, connect the outputs of your audio interface to the inputs on your mixer.



Figure1: Using Mixxx together with turntables and external mixer

**Poznámka:** Many modern turntables have a *Line/Phono* select switch. If yours does, make sure it matches the input or switch setting on your sound card.

## Setup 2: CDJ

2-4 timecode CDs, 2-4 CD decks, and 2-4 stereo sound inputs.

Setting up CDJs for use with timecode CDs is similar to setting up turntables. The figure below outlines a typical configuration. First, connect the RCA cables of the CDJs to inputs of your audio interface. Second, connect the inputs of your mixer to the outputs of your audio interface. Make sure the input mode of your audio interface is set to *Line*.

### Sound inputs

- It is recommended to use a DJ audio interface that has **multiple** stereo line inputs on it.
- Alternatively you can use multiple audio interfaces, each with a single stereo line in, but this is strongly discouraged.

Mixxx supports vinyl control input through a single audio interface with at least 4-channels of input (two stereo line-in jacks), or through multiple separate audio interfaces which each have 2-channels of input (a single stereo line-in jack). Vinyl control requires the use of stereo line-in jacks - Mono or microphone inputs will not work.

It is strongly recommended to use a audio interface that comes with native low latency drivers such as ASIO (Windows) or CoreAudio (Mac). Linux users depend on the quality of the corresponding open source driver.

**Poznámka:** For best scratch performance with vinyl control, we recommend using a system capable of *latencies* of

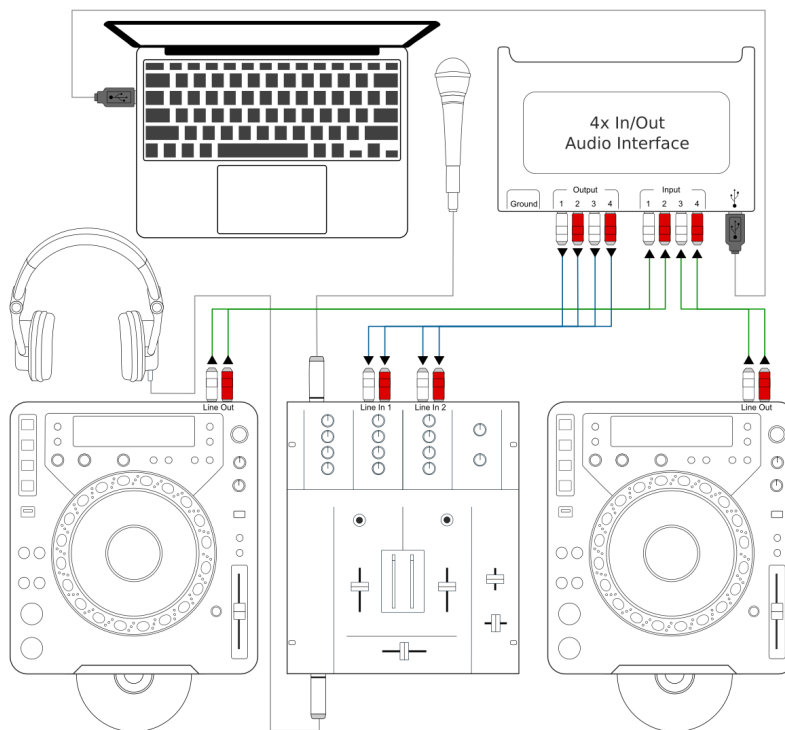


Figure2: Using Mixxx together with CDJs and external mixer

10ms. With higher latencies the scratch sound will start to become distorted.

**Poznámka:** An external hardware mixer is not required for using timecode control. You can use a *MIDI or HID controller* to mix in software instead.

### 8.2.1 Supported Timecode media

You can use any of the timecode media supported by Mixxx:

Typ	Responsiveness
Serato CV02/2.5 Vinyl	Very high
Serato Control CD 1.0	Very high
Traktor Scratch MK1 Vinyl	Very high
Traktor Scratch MK2 Vinyl	Not supported
MixVibes DVS V2 Vinyl	Very high

At the present time, Serato records are recommended if you are looking to buy vinyl. If you want to use CDs, you can download a free copy for personal use from [Serato](#).

## 8.3 Configuring Input and Output devices

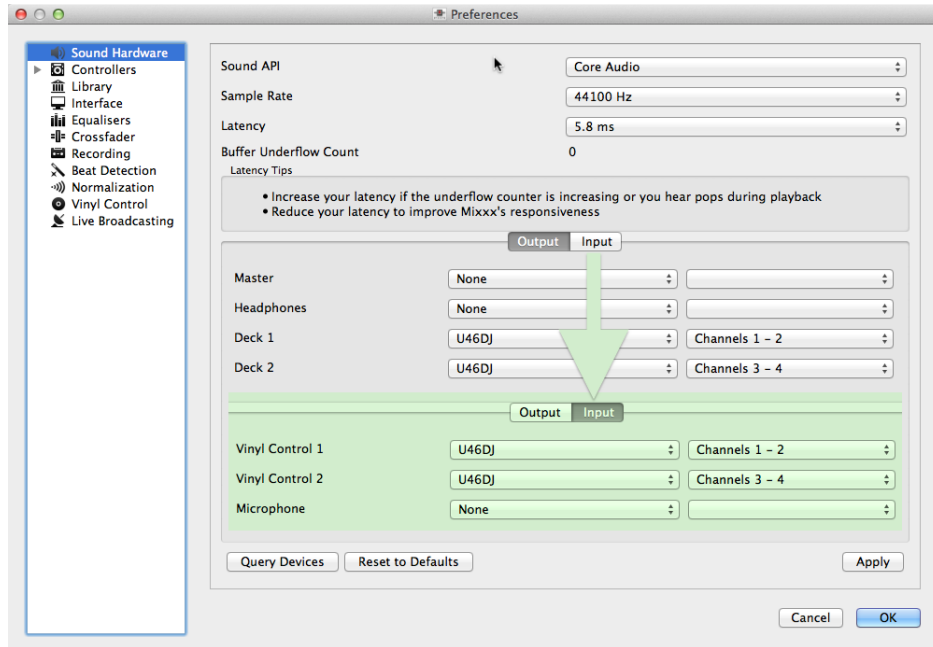


Figure3: Mixxx preferences - Setting up Input and Output devices for Vinyl Control

Mixxx can be controlled by up to 4 decks with either timecoded vinyl or timecoded CDs.

1. Open *Preferences* ▶ *Sound Hardware*
2. Select the *Input* tab
3. From the *Vinyl Control 1 - Vinyl Control 4* drop-down menus, select the audio interface(s) which your decks are plugged into.
4. In the channel selection drop-down menu, select the channel pairs corresponding to the plug on your audio interface that your deck(s) are plugged into, typically *Channels 1-2* and *Channels 3-4* etc.
5. Click *Apply* to save the changes.
6. Select the *Output* tab
7. Specify *Deck 1 - Deck 4* to route Mixxx's output directly to the external mixer
8. Click *Apply* to save the changes.

## 8.4 Configuring Vinyl Control

You can configure vinyl control in *Preferences* ▶ *Vinyl Control*.

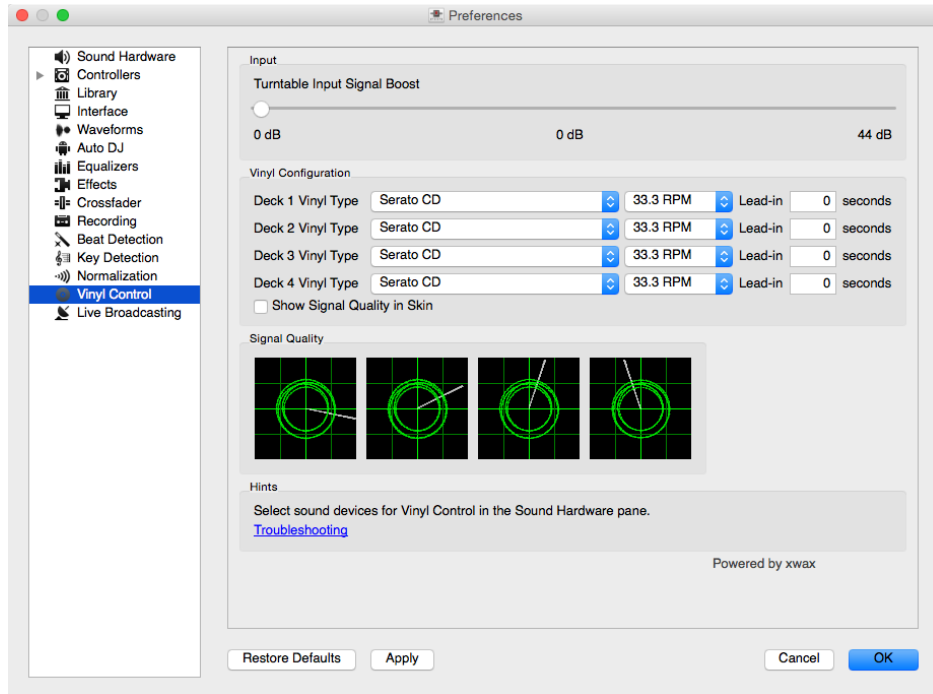


Figure4: Mixxx preferences - Setting up vinyl control

### 8.4.1 Zdůraznění signálu vstupu přehrávače (gramofonu)

Many turntables provide unamplified “phono level” output which must be boosted to a “line level” signal. Normally, a mixer or audio interface provides this amplification, but if you’re plugging a turntable’s phono output directly into a audio interface that doesn’t support phono input, Mixxx can do the amplification. The *Turntable Input Signal Boost* slider allows you to adjust the level of preamplification applied to your decks’ signal. However, it is always preferable to use a proper phono preamplifier if you have one.

### 8.4.2 Nastavení vinylové gramodesky

- **Vinyl Type:** Several types of timecoded media can be used to control Mixxx. Configure the drop-down menus to match what type of timecoded media you are using on your decks.
- **Deck Vinyl Tempo:** Change this if you prefer to scratch with your turntable or CDJ set at 45 RPM, default is 33 RPM.
- **Lead-in Time:** Allows you to set a dead-zone at the start of your vinyl or CD, where the time code will be ignored. This is useful in situations where the time code begins very close to the edge of a vinyl which can make back-cueing over the first beat in a track tricky to do without your turntable’s needle falling off the vinyl. Setting a lead-in time of 20 seconds or more helps alleviate this by giving you more room on the vinyl to cue with. It’s also useful when you’ve worn the grooves at the edge of a control record to the point that it no longer works reliably: you simply set the lead-in so that the start of the tracks begin in good groove area. You can keep doing this until you run out of groove area, thereby decreasing your record replacement frequency.
- **Show Signal Quality in Skins:** If enabled, a circular representation of the incoming timecode signal is displayed in real-time on the *Vinyl Widget* in the Mixxx Interface.

---

**Rada:** Although not recommended, you can use different kinds of timecode media per deck.

---



### 8.4.3 Režim ovládání

Mixxx has three control modes regardless of which control records you use.

- **Absolute Mode:** Provides Mixxx with both pitch and position information from the timecode, and allows you to seek by needle dropping on your vinyl or seeking on your CDJ.
- **Relative Mode:** Takes the position to be relative to your deck's starting point, and only controls the pitch in Mixxx. It is not possible to seek using your deck in relative mode, with one exception. If you drop the needle in the *Lead-in Time* area (see *Vinyl Configuration*), Mixxx will seek back to the beginning of the track. Once a track is playing in Relative Mode, Mixxx will not allow you to select Absolute mode. Otherwise, the track would suddenly jump to a new position. Make sure the record is stopped and then you're free to select Absolute mode.
- **Constant Mode:** This special mode is usually only activated when Mixxx needs to keep playing a track even if the vinyl control signal isn't present. Constant mode is automatically enabled when the needle reaches the end of a vinyl control record.

### 8.4.4 Cueing Modes

Mixxx has two optional cueing modes for vinyl control which **only** apply to **Relative Mode**. These modes are useful for quickly seeking to specific positions in a track without touching the laptop. Turntablists often need to find a break or a vocal sample, and cueing modes make these easy to find.

- **Cue:** In this mode, when you drop the needle on the record *after* the Cue point, Mixxx will seek to that cue point.
- **Hot Cue:** When the needle is dropped Mixxx will seek to the nearest hotcue point, working backwards. So if you have a hotcue at the one-minute mark, and another at the two-minute mark, you can drop the needle anywhere between one and two minutes and Mixxx will seek to the one-minute hotcue.

---

**Rada:** You can change the Control and Cueing Modes directly in the Mixxx user interface while mixing, go to *Vinyl Control Mode and Cueing controls*. Alternatively, use the *Application shortcuts*.

---

### 8.4.5 Kvalita signálu

A successful vinyl control setup hinges on good signal quality. Many factors can affect signal quality, but the most important one is ensuring the volume level of your timecode signal is moderate. A signal that is too loud or too quiet will cause adverse performance, often characterized by a loss of position data causing absolute mode to behave like relative mode.

Mixxx represents the quality of the timecode signal as a circular graphic coming from the vinyl control audio itself. The graphs correspond to your active input devices *Deck 1 - Deck 4*.

If your vinyl signal is working, you should see a round, green circle drawn with two lines. This vinyl “doughnut” should be as circular as possible, and the lines should be clear with a minimum amount of fuzz.

---

**Rada:** You can display the signal quality in the Mixxx user interface while mixing, see *Configuring Input and Output devices*.

---

## 8.5 Enabling Vinyl Control

For each deck, select *Options* ▶ *Vinyl Control* ▶ *Enable Vinyl Control*, or use the *Application shortcuts*.

### 8.5.1 Vinyl Control Status Display



Figure5: The vinyl control status, and signal quality display of the deck

A rectangle provides visual feedback with regards to vinyl control status of a deck.

- **Off:** Vinyl control disabled on the deck.
- **Green:** Vinyl control is enabled on the deck.
- **Yellow:** The needle has reached End-Of-Record.

## 8.6 Single Deck Control

If you only have one turntable, you can still use Mixxx to play your sets. In the *Preferences* ▶ *Sound Hardware* ▶ *Input* tab, use the same audio input for both *Vinyl Control* decks.

When mixing, you'll have one track playing automatically while the other is controlled by the turntable. Start by enabling vinyl control on the first deck, and start mixing. When you're ready to cue up the next record, just disable vinyl control on the first deck and enable it on the second deck. Playback will continue on the first deck without interruption. Cue up the second track and mix it in. When you're ready to cue the first deck again, just toggle the second deck off and the first deck on again.

Use the *Application shortcuts* to swap focus from one deck to the other. Alternatively, select *Vinyl* ▶ *Single Deck mode* in the *Controller Wizard*, to assign the control to a key on your controller.

## 8.7 End-Of-Record Mode

Sometimes the track you are playing is longer than the timecode area on your record. If the needle reaches the end of the vinyl timecode, Mixxx will enter a special *End-Of-Record Mode*. When this happens, the vinyl indicator (usually a green rectangle) will blink yellow as a warning. At the same time, Mixxx will automatically enable *Constant Mode* and will ignore the turntable's pitch. At this point the track will simply continue playing until the end. When you load a new track, *End-Of-Record Mode* will be automatically disabled and *Constant Mode* turned off.

When you're in *End-Of-Record Mode* and you want to still use the turntable for pitch control, simply lift the needle and place it anywhere in the main timecode area. It's OK, the track won't stop playing. As soon as Mixxx detects that the needle has settled, it will automatically switch to *Relative Mode*.

## 8.8 Control Track

Some vinyl control records have a special portion of grooves near the center of the record for selecting and loading tracks in the user interface. You can see these grooves because there is a gap between the main timecode area and the inner circle. Some control records have not been tested with Mixxx, and their control areas are not supported. Serato CV02 Vinyl and Traktor Vinyl MK1 are known to work. Serato CV02 CDs are known not to work.

Just drop the needle in the control area and let the record play forward or backward. The highlighted row in the library should move up or down. Simply lift the needle and drop it anywhere in the main timecode area to load the track.

## 8.9 Odstraňování závad


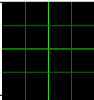

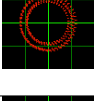
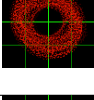

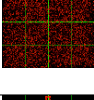
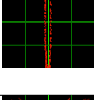
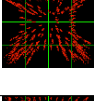
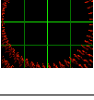
### Configuration

To complete your setup you must configure Mixxx properly. Make sure you have:

- specified *Deck 1 - Deck 4* in *Preferences* ▶ *Sound Hardware* to route Mixxx's output directly to the external mixer
- specified *Vinyl Control 1 - Vinyl Control 4* under *Preferences* ▶ *Sound Hardware* ▶ *Input*
- selected the correct control type under *Preferences* ▶ *Vinyl Control* ▶ *Vinyl type* in the preferences. See the list of *Supported Timecode media*
- enabled vinyl control via the menu under *Options* ▶ *Vinyl Control* ▶ *Enable Vinyl Control*.

### Signal Quality

The graph is displayed in *Preferences* ▶ *Vinyl Control*, see *Kvalita signálu*.

Signal	Meaning	Odstraňování závad
	Vinyl control input channels not correctly set up. Vinyl control will not work.	Open Preferences/ Sound Hardware and set up the devices you want to use, see <i>Configuring Input and Output devices</i>
	Vinyl control input channels set up correctly. Waiting for timecode input signal.	Start your turntable or CDJ with the selected timecode vinyl or CD.
	Nearly perfect green doughnut about half the width of the graph. Rotating radar sweeps in the direction your record on the turntable is spinning.	Everything is OK, nothing to do. Go spin some beats.
	Incorrect choice of vinyl control source. Doughnut looks good but has a solid red color.	Select the correct vinyl type in the preferences. E.g. maybe you selected side B of the Serato Vinyl but your record is on side A.
	Input signal is bad. As the vinyl signal gets worse, you'll see the color of the doughnut change from green over orange to red.	Check the wiring / grounding on your turntable / CDJ. You timecode record / CD might be worn out or the needle on your turntable is dirty.
	Wrong direction. The green doughnut looks good but the radar sweeps in the opposite direction of the record on your turntable / CDJ.	You probably have a wiring problem. Try reversing the left and right inputs on your turntable. You may also have to rewire the headshell.
	Input signal is no timecode signal. Mixxx receives a normal audio signal while it expects a timecode signal.	Start your turntable or CDJ with the selected timecode vinyl or CD. You may want to check if the inputs devices are set up correctly in the preferences.
	Left or right channel is missing. Mixxx receives only a mono signal but needs a stereo timecode signal to operate.	Check the wiring on your turntable / CDJ and also the headshell and needle on your turntable.
	Timecode signal too loud. Mixxx receives a signal which has too much amplification.	Lower the amplification, either on your external mixer, audio interface, or with the <i>Turntable Input Signal Boost</i> .
	Timecode signal too quiet. Mixxx receives a signal which has not enough amplification.	Boost the amplification, either on your external Mixer, audio interface, or with the <i>Turntable Input Signal Boost</i> .

---

### Using Microphones

---

Mixxx can work with any microphone that can be plugged into your *audio interface*. There are three ways to use microphones with Mixxx, each with their own pros and cons:

- *Software Monitoring*: Cheapest and simplest to set up, but you cannot hear yourself without a disorienting delay, referred to as „latency“.
- *Direct Monitoring*: Recommended for most users. This is the cheapest to set up with good sound quality and without the latency of software monitoring.
- *Hardware Mixers*: Most expensive option to set up with high sound quality. This does not have the latency of software monitoring.

**Varování:** USB microphones are not recommended. These devices have their own audio interface built in and can only be used with software monitoring. Some USB microphones have headphone jacks for direct monitoring, but the music from Mixxx cannot be heard in this headphone jack. Also, they can be difficult to configure at the same time as a different audio interface for music output.

**Viz také:**

The [Mixxx DJ Hardware Guide](#) lists specific audio interfaces with information about their prices, features, and suitability for use with microphones.

## 9.1 Software Monitoring

Software monitoring can be used with the built-in microphone on a laptop or with a headset plugged into the built-in audio interface on a computer. However, there is a delay between the time you make sound into the microphone and hear it in the Master output. Due to the nature of digital audio, it takes time for input from the audio interface to be available for Mixxx to process, time for Mixxx to process the audio, and more time to send the audio back out the audio interface. This time is referred to as „latency“.

Mixxx can be configured to run at low latencies by choosing a smaller *Vyrovňovací paměť zvuku*, but there is no way to completely eliminate latency. Although latency is measured in milliseconds, even a few milliseconds of latency can be disorienting and distracting to hear, whether you are using the microphone for spoken announcements, vocals, or playing a musical instrument.

To use software monitoring:

1. Open *Preferences* ▶ *Sound Hardware*.
2. Select the *Input* tab.
3. For *Microphone 1*, select the audio interface that your microphone is plugged into.
4. Click the *Apply* button.
5. Click the *OK* button.
6. Click the *Mics* button in the main Mixxx window to show the microphone controls.
7. Click the *Talk* button when you are using the microphone.

---

**Rada:** If you would prefer to not hear the microphone input, you can set the *Microphone Monitor Mode* option to *Direct monitor (recording and broadcasting only)*. This will still mix the microphone input with your recorded and broadcasted mixes, but it will not mix the microphone with the Master output. Don't forget to press the *Talk* button when using the microphone if you use this option.

---

## 9.2 Direct Monitoring

An audio interface with direct monitoring and a *Loopback Input* is the recommended way to use microphones with Mixxx for most users. Except for audio interfaces built into computers, most audio interfaces with microphone inputs support direct monitoring. However, not all audio interfaces that support direct monitoring also have a loopback input.

Direct monitoring routes the audio from the audio interface's inputs directly its outputs. This avoids the latency of sending it into the computer and back out with *Software Monitoring*. At the same time, the audio interface sends the input into the computer so Mixxx can record and broadcast it.

### 9.2.1 Loopback Input

Some audio interfaces that support *Direct Monitoring* have a loopback input feature. They add Mixxx's output to the microphone signal before sending it to the computer. This makes it easier to set up Mixxx with direct monitoring than configuring *Latency Compensation*.

To configure Mixxx with a loopback input:

1. Open *Preferences* ▶ *Sound Hardware*.
2. Click the *Input* tab.

3. Select the audio interface for the *Record/Broadcast* input. Do not configure anything for the *Microphone 1-4* inputs.
4. Click the *Apply* button.
5. Click the *OK* button.
6. *Activate direct monitoring and loopback on your audio interface.*
7. Adjust the microphone volume with the input gain knob on your audio interface.

The microphone controls in Mixxx will not affect your microphones because they are mixed by the audio interface and not sent directly to Mixxx.

## 9.2.2 Latency Compensation

When using *Direct Monitoring*, you will hear the microphone mixed with the music from Mixxx without any noticeable latency. However, it still takes time for Mixxx to receive the microphone signal and process it. Without compensating for this latency or using a *Loopback Input*, the microphone inputs will be out of time relative to the music in your recorded and broadcasted mixes.

Mixxx can compensate for this timing misalignment, but it is complicated to set up. If you do not want to record your microphone into your mix or broadcast, you can *activate direct monitoring on your audio interface* without configuring latency compensation.

### Measuring Round Trip Latency

To configure Mixxx to compensate for input latency while using direct monitoring, first you must measure the round trip latency of your setup. The round trip latency is different from the size of the *Vyrovnávací paměť zvuku* configured in Mixxx. It is the amount of time it takes for audio to make a complete trip from your audio interface's input, through your computer, and back out the audio interface.

Mixxx cannot calculate the round trip latency because it depends on details of your audio interface's hardware, your operating system, your audio interface's driver, and other factors in your computer's hardware. The round trip latency can only be found by measuring it. To do this, use a physical cable to connect the audio interface's output to its input. Then, use a third party program to measure the round trip latency. These programs are recommended on each OS:

- **GNU/Linux:** `jack_iodelay`
- **Windows:** RTL Utility
- **macOS:** Audacity

You must use the same sample rate and audio buffer size in the measurement program as you do in Mixxx for the measurement to be accurate. If you decide to change the sample rate or buffer size you use with Mixxx, you will need to remeasure your round trip latency to have your microphone inputs aligned in your recorded and broadcasted mixes.

**Varování:** Make sure direct monitoring is *not activated on your audio interface* while you are measuring the round trip latency or you will not get an accurate measurement.

## Configuring Latency Compensation

1. Before opening Mixxx, *measure the round trip latency*.
2. Open Mixxx.
3. Open *Preferences ▶ Sound Hardware*.
4. Click the *Input* tab.
5. Select the audio interface input(s) for the *Microphone 1-4* inputs. Select a single mono channel for each Microphone input unless you are using stereo microphones.
6. For the *Microphone Monitor Mode* option, select *Direct monitor (recording and broadcasting only)*.
7. Enter the measured round trip latency in milliseconds for the *Microphone Latency Compensation* option.
8. Click the *Apply* button.
9. Click the *OK* button.
10. *Activate direct monitoring on your audio interface*.
11. Click the *Mics* button in the main Mixxx window to show the microphone controls.
12. Click the *Talk* button when you are using the microphone.
13. Adjust the microphone volume with the input gain knob on your audio interface. Do not adjust the microphone gain in Mixxx. If you do, the relative volume of the mics and music will be different in your recorded and broadcasted mixes compared to what you hear out of your audio interface.

Mixxx will not record or broadcast your microphone if the *Talk* button is not active. However, you will still hear the microphone in your main output because the microphone is mixed by your audio interface, not Mixxx. You may leave the *Talk* button on to ensure you do not forget it, but this will record and broadcast background noise when you are not actively using the microphone. If your microphone has an on/off switch on it, you may leave the *Talk* button enabled in Mixxx and use the switch on the microphone to avoid adding background noise to your mix. Alternatively, you can adjust the input gain on your audio interface throughout your mix.

### 9.2.3 Activating Direct Monitoring And Loopback On The Audio Interface

General tips for activating direct monitoring and loopback are described below. However, every audio interface is different, so refer to the audio interface manual from the manufacturer for more details.

On stand-alone audio interfaces, there is often a knob on the device that controls direct monitoring. This knob crossfades the audio interface's output between the signal from the computer and the directly monitored inputs. If your audio interface has a knob like this, set the knob to the center. Alternatively, some audio interfaces have a switch to toggle direct monitoring instead of a knob.

To hear mono microphone inputs on both sides of the stereo output with direct monitoring, you typically need to toggle a switch on the device.

If the audio interface supports *Loopback Input*, that may be activated by a switch on the device.

Audio interfaces with lots of inputs and outputs often have a control panel program provided by the manufacturer that is installed automatically with the driver. This may be used instead of knobs and switches on the hardware to control direct monitoring, loopback, stereo/mono switches, and other audio interface features. On GNU/Linux, you might be able to access these controls with **alsamixer**.

Refer to the section below for details about *DJ Controllers With Microphone Inputs*.



## 9.3 Hardware Mixers

Mixxx can be used with a microphone plugged into an external hardware mixer. This does not have the problem with latency that happens with *Software Monitoring*. However, it is generally recommended to use an audio interface that supports *Direct Monitoring* and a *Loopback Input* instead of an external mixer. If you are using *Ovládání vinylovou gramodeskou* and a microphone, you may need an external mixer.

Mixxx can send each deck to separate stereo channels on an external mixer by using the *Deck 1-4* outputs. This requires an audio interface with at least 4 output channels (2 stereo pairs). Audio interfaces built into computers only have one stereo output and they do not have high sound quality. Thus, another audio interface is recommended. Audio interfaces with at least 4 output channels typically have microphone inputs and support *Direct Monitoring*, so there is no need for the external mixer.

However, audio interfaces with phono preamplifiers for *Ovládání vinylovou gramodeskou* do not have microphone inputs. If you want to use vinyl control with a microphone, it is recommended to plug the microphone into a DJ mixer.

**Varování:** Some mixers that are not designed for DJing have a built-in USB audio interface. However, the audio interfaces in these mixers typically send only 2 channels (one stereo pair) to the mixer, so they are not recommended.

### 9.3.1 Recording And Broadcasting With An External Hardware Mixer

To record or broadcast with an external hardware mixer, the output of the mixer needs to be connected to the input of an audio interface. Most DJ mixers have an extra output for this which may be labeled „record“, „session“, „auxiliary“, or „booth“. Some audio interfaces for vinyl control have enough input channels to receive the output of the mixer and timecode from two turntables. If yours does not, you may use the input of the audio interface built into your computer, but these do not have high sound quality.

Alternatively, some DJ mixers have a built-in USB audio interface. These have inputs with phono preamplifiers for vinyl control and usually can send the record output back to the computer without a separate audio interface. Many (but not all) of these mixers are digital mixers, so they can send signals back and forth to Mixxx without converting them to analog. This results in higher sound quality for your recorded and broadcasted mixes compared to using a separate audio interface with an external mixer.

To configure Mixxx for using a hardware mixer for recording and/or broadcasting:

1. Open *Preferences* ▶ *Sound Hardware*.
2. In the *Output* tab, select the audio interface for the *Deck 1-2* outputs (and *Deck 3-4* outputs if your audio interface and mixer support 4 decks).
3. Click the *Input* tab.
4. For the *Record/Broadcast* input, select the audio interface connected to the mixer's output. For mixers with a built-in USB audio interface, refer to the mixer manufacturer's manual to find which channels of the mixer's audio interface send the record output.
5. If you are using vinyl control, select the audio interface connected to the turntables for the *Vinyl Control 1-4* inputs.
6. Click the *Apply* button.
7. Click the *OK* button.

Do not configure anything for the *Microphone 1-4* inputs when using an external mixer. The microphone controls in Mixxx will not affect your microphones because the microphone is mixed by the external mixer.

DJ mixers typically do not supply phantom power required for condenser microphones. Stand-alone audio interfaces typically do supply phantom power.

## 9.4 DJ Controllers With Microphone Inputs

DJ controllers with microphone inputs typically mix the microphone input with the master output without sending it to the computer. This does not have the problem with latency that happens with *Software Monitoring*. However, many cheap DJ controllers do not digitize the signal to make it available to the computer. Refer to the controller's page on the [Mixxx wiki](#) or the manual from the controller manufacturer for details about your particular controller.

If the controller does not digitize the microphone input, you cannot use Mixxx's microphone input to record or broadcast with the microphone using Mixxx's *Microphone 1-4* inputs. It is still possible to record and broadcast the microphone by connecting the booth output of the controller to an audio interface input and configuring this for Mixxx's *Record/Broadcast* input. If the controller does not have a booth output, a Y splitter cable can be used on each side of the stereo main output to connect the main output to both the speakers and to another audio interface's input.

DJ controllers typically do not supply phantom power required for condenser microphones. Stand-alone audio interfaces typically do supply phantom power.

## 9.5 Microphone Ducking

Microphone ducking is a technique to improve the audibility of speech by reducing the volume of background music while a voice signal is recorded by the microphone and allows listeners to hear the voice more clearly and consistently. While using Mixxx with a microphone, you might want the music to automatically fade into the background each time you speak, or you might prefer lowering the volume of the music to a specific level. Microphone ducking helps you achieve this. The music will 'duck' out of the way of the microphone audio.



Figure1: The microphone ducking controls

The *Strength* knob adjusts the microphone talkover ducking strength and its behaviour depends on any of the Microphone talkover modes described below. Clicking on the *Auto* button will switch it to any of three modes:

- *Off*: When you speak through the microphone, the music volume does not reduce.
- *Auto*: When you speak through the microphone, the music volume will be reduced when Mixxx detects your voice. You can adjust the amount by which the volume is automatically reduced with the Strength knob.
- *Manual*: In this mode, the music will play at the volume set by the Strength knob whenever the microphone Talk button is active, regardless of the volume coming through the microphone input. When you speak through your microphone, set the volume to which the music should reduce by turning the Strength knob to a level that suits you. When you disable Talk mode, the music will return to the regular volume

---

**Rada:** To set the Ducking volume correctly, you can use manual mode to adjust the knob to your taste and then set to auto when happy.

---

---

## Live Broadcasting - Start your own Internet radio

---

Live Broadcasting in Mixxx allow you to stream your mix over the Internet to listeners around the world.

### Viz také:

The *Using Microphones* chapter has detailed instructions for including input from microphones in your broadcasted mixes.

## 10.1 Streaming Servers

**Remote streaming server** Mixxx allows you to feed your audio stream directly to *Shoutcast* and *Icecast* streaming servers. Depending on the number of listeners, streaming audio requires a significant amount of bandwidth. Streaming servers provide the required bandwidth and broadcast the stream to your listeners. A popular free streaming service is [Caster.fm](#). A review of several free and paid stream hosts is available at [broadcastingworld.net](#).

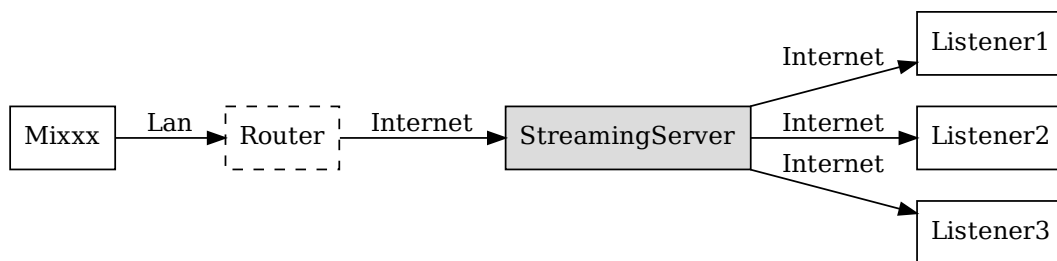


Figure 1: Mixxx as client-side streaming source broadcasting to an remote streaming server

**Local streaming server** For experienced users, it may be interesting to set up your own local streaming server. This turns your personal computer into a radio station and listeners connect directly to your server. Mixxx as a streaming

source does not need to run on the same computer as your streaming server. However, professional stations often run the streaming source on the same computer as the streaming server for stability and reliability reasons. Keep in mind that if you want to stream audio to a significant number of listeners, you'll need enough bandwidth. Read the [Shoutcast documentation](#) or [Icecast documentation](#) for server setup instructions.

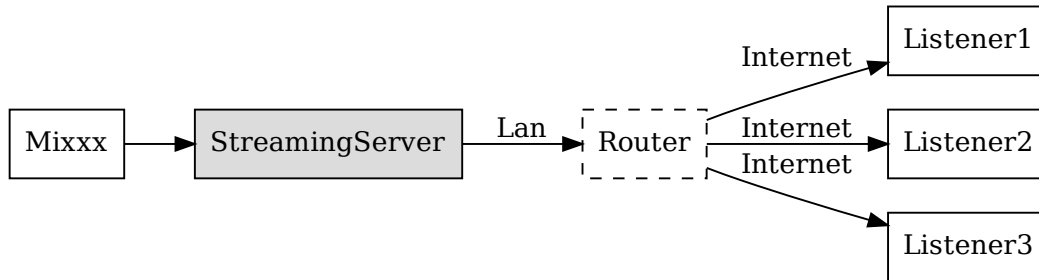


Figure2: Mixxx as client-side streaming source broadcasting to an local streaming server

## 10.2 Configuring Mixxx

Start by supplying Mixxx with all information needed to establish a connection to the streaming server:

- Open *Preferences* ▶ *Live Broadcasting*.
- Insert the settings following the descriptions in the *Mixxx preferences - Setting up live broadcasting*
- Click *OK*
- Go to *Options* ▶ *Enable Live Broadcasting* or use the *Application shortcuts* to start broadcasting.

### 10.2.1 Server Connection

- **Type:** Select the type of streaming server you want to connect with. *Shoutcast 1*, *Icecast 1*, and *Icecast 2* servers are supported.

Mixxx works with Shoutcast 2 using the Shoutcast 1 protocol if you provide a stream name in *Preferences* ▶ *Live Broadcasting* ▶ *Stream Settings*. If you don't provide a stream name, Shoutcast 2 rejects the connection (where Shoutcast 1 would accept this case).

- **Host:** You can enter the host as either a host name or an IP address.
- **Login:** As provided by your streaming server provider. Without this, you will not connect successfully to the server. The default login for *Icecast* is `source` while the default login for Shoutcast is `admin`.
- **Mount:** A mount point is a unique name identifying a particular stream. For *Shoutcast* it is not necessary to specify a mount point. The setting must not be blank if you are using *Icecast*. Try the default `/mount` or `/live`. If you haven't been given a specific mount point you can usually make one up. It always begins with a `/` (slash) followed by a text without any special characters in it.
- **Port:** As provided by your streaming server provider. Most servers use the default port 8000.

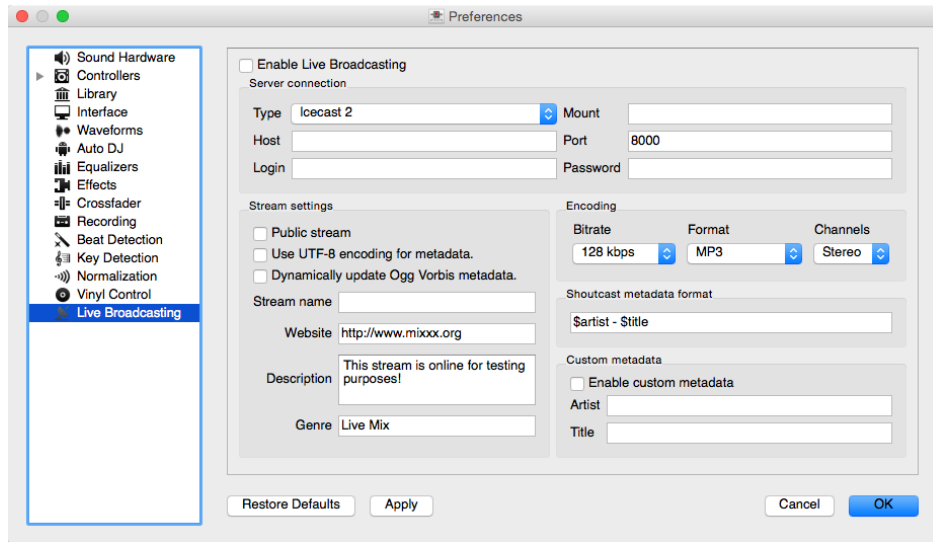


Figure3: Mixxx preferences - Setting up live broadcasting

- **Password:** As provided by your streaming server provider, unless you run your own radio server. It is required to establish the connection to the server and to start the broadcast.

**Varování:** Do not enter a *URL* as the host! `http://example.com:8000` does not work. Use `example.com` in the *Host* field and `8000` in the *Port* field instead.

## 10.2.2 Stream Settings

- **Public stream:** If enabled, this option adds your radio station to the Shoutcast/Icecast directory.
- **Enable UTF-8 metadata:** If enabled, this option fixes broken accented and foreign language symbols in *metadata*, assuming the streaming provider has configured the server to support UTF-8 metadata.
- **Dynamically update Ogg Vorbis metadata:** Due to flaws in some streaming clients, updating Ogg Vorbis metadata dynamically can cause listener glitches and disconnections. Check this box to update the metadata anyway. Some players that listeners can use have bugs that can cause audio glitches or disconnections when the Ogg Vorbis metadata is updated dynamically. If this is not a problem, you can enable this checkbox.
- **Stream name:** So, what's the name of your show?
- **Website:** The website you would like your listeners to visit.
- **Description:** Enter your DJ name and a short tagline.
- **Genre:** List the main genres you play. This attracts search hits on stream directories. Genre must not be blank.

## 10.2.3 Encoding

- **Bitrate:** Selecting a *bitrate* of 128 or 160 *kbps* is common and provides sufficient quality to your listeners. Higher bitrates will use a larger chunk in your Internet connection bandwidth to stream and for your listeners to receive the stream.
- **Format:** Mixxx supports streaming to Icecast servers either in *MP3* or *Ogg Vorbis* format, streaming to Shoutcast servers is supported in *MP3* format.

## 10.2.4 Metadata

### Shoutcast metadata format

This allows to set custom metadata formats for the Shoutcast title field. Previously only `artist - title` was allowed. For example if you were broadcasting as part of a station, you could add the station's name or the presenter's name in the title: `MyStation | $artist - $title`. Or if you were doing a live mix with several artists, you could have: `Live mix by MyName - currently playing: $artist`. Or even if you wanted a very unusual format: `Hey, I like $artist, here is $title by $artist`.

The changes **do not** affect the case for the combination of OGG/Icecast2.

### Custom metadata

By default, Mixxx broadcasts artist and title information of the files that you play to your listeners. You can disable this feature and use your own custom metadata.

- **Enable custom metadata:** Toggles custom metadata on and off.
- **Artist:** Insert your custom artist metadata here, your DJ name for example.
- **Title:** Insert your custom title metadata here.

## 10.2.5 Icecast vs. Shoutcast

Both essentially serve the same purpose. An Icecast server can stream either *MP3* or *Ogg Vorbis*. However, although Ogg is more efficient and effective (you get higher-fidelity sound than MP3 at lower data rates) not all players can play Ogg streams. As a result MP3 is probably a safe choice unless you know your listeners can hear an Ogg stream successfully.

## 10.2.6 Broadcast directories

Generally your streaming server host adds your radio station to the Shoutcast/Icecast directory, if you enable the *Public Stream* option in *Preferences* ▶ *Live Broadcasting* ▶ *Stream Settings*.

- **Shoutcast radio directory:** [www.shoutcast.com](http://www.shoutcast.com)
- **Icecast radio directory:** [dir.xiph.org](http://dir.xiph.org)

Often streaming hosts will run their own directories. Check your host's FAQ to find out. If you want to promote your streaming radio station even more, register at services like [streamfinder.com](http://streamfinder.com). An overview of different internet radio directories is available at [shoutcheap.com](http://shoutcheap.com)

## 10.3 Odstraňování závad

- If you have trouble connecting to your streaming server, check the configuration in the *Mixxx preferences - Setting up live broadcasting*.
- You may have selected the *Ogg Vorbis* format that is unsupported by Shoutcast servers.
- You may need to check your firewall settings. Both Icecast and Shoutcast use two ports. If you are streaming on a port (for example, port 8000) then you need to open up the next port (port 8001) as well.
- You may need to configure port forwarding if you are behind a router or your router will block requests to your streaming port (for example, port 8000)

Mixxx has a powerful sound effects system. Understanding how it works can open you up to a wide variety of mixing techniques.

## 11.1 Efektové jednotky

Mixxx has 4 effect units. By default, only 2 of them are shown. The other 2 effect units can be shown by toggling the setting in the skin settings menu.

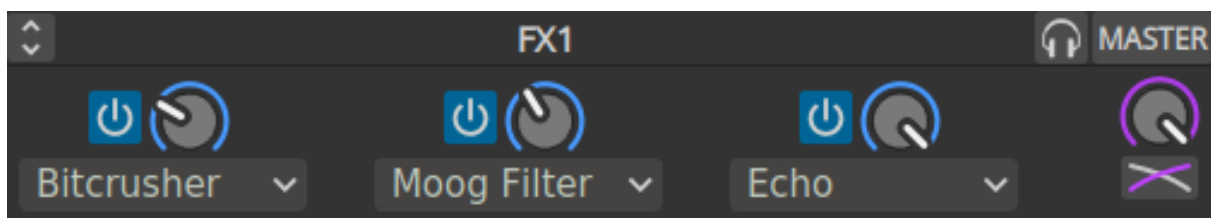


Figure1: Efektová jednotka

Each effect unit can load up to 3 effects in a chain. These effects are processed one after the other in series. You will hear different sounds depending on the order of effects within the chain. Each effect in the chain has its own button to toggle it on and off.

**Efektové jednotky mohou zpracovat zvuk z:**

- Přehrávače
- Mikrofony
- Pomocné vstupy (Aux)
- Hlavní míchání
- *PFL* (sluchátka) míchání



- Sběrnice levého/prostředního/pravého prolínače

The buttons to route effect units to decks, microphones, and auxiliary inputs are located in the deck, microphone, and auxiliary sections of the skin. The buttons for routing other inputs to effect units are in different locations depending on the skin.

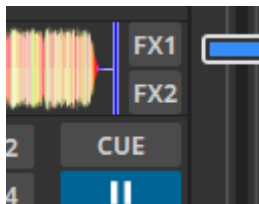


Figure2: The effect unit buttons located in the deck.

If a channel is assigned to multiple effect units, those effect units are chained together. This allows you to chain more than 3 effects at a time.

To route deck 1 to effect unit 1, click the *FX1* button on deck 1 to turn it on. To route deck 2 to effect unit 2, click the *FX2* button on deck 2, and so on. It's also possible to route a single deck to multiple effect units, or multiple decks to a single effect unit.

The mix knob adjusts the mix between the dry signal (input to the unit) and the wet signal (output of the last effect in the unit). It changes the level of all 3 effects in the unit together. Refer to the [Effect Unit Mix Modes](#) section below for details.

Effects are processed after the deck faders and crossfader. This allows effects like Echo and Reverb to continue outputting sound after their input has been cut off by lowering the fader. Unlike other DJ software and hardware, effects can also be heard in headphones even though they are processed after the faders for the master output. Refer to the [Effects In Headphones](#) section below for details.

Note that equalizer and filter effects create a phase shift. If the mix knob is anywhere in the middle between dry and wet and any of these effects are enabled in the unit, there will be comb filtering which is probably not desired. For more background information, read [Equalizers and Phase Shift](#).

## 11.2 Parametr efektu a regulátory

Every effect within an *Effect Unit* has its own set of parameters. By default, these are hidden. Instead, there is a single knob called a „metaknob“ for each effect which is linked to the different parameters. The metaknob allows you to easily control the effects in a unit without having to manipulate every parameter individually.

If you want more detailed control of effects, press the expansion button on the effect unit to reveal the effect parameters:

You can customize how the metaknob is linked to the parameters by clicking the buttons below the parameter knobs. The button under the parameter name controls the metaknob link mode:

- Inactive: parameter not linked
- Active: parameter moves with metaknob
- Left side active: parameter moves with left half of metaknob turn
- Right side active: parameter moves with right half of metaknob turn
- Left and right side active: parameter moves across range with half of metaknob turn and back with the other half

The button below the metaknob link button inverts the parameter's relationship to the metaknob.

For information about specific effects and their parameters, hover your mouse over them to show the tooltip. If you do not see tooltips, check that you have them enabled for the skin in *Preferences* ▶ *Interface*.

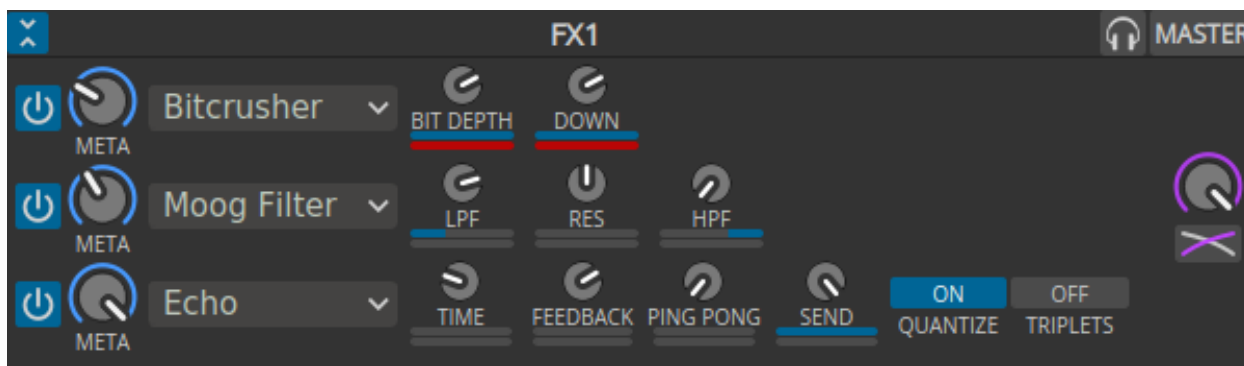


Figure3: An effect unit with parameters showing

Effects (except for Reverb) that have a length of time as a parameter are synchronized to the tempo of decks.

### 11.3 Effect Unit Mix Modes

The effect unit mix knob controls the level of all 3 effects in the unit together. It can be toggled between two different modes with the mix mode button adjacent to it.

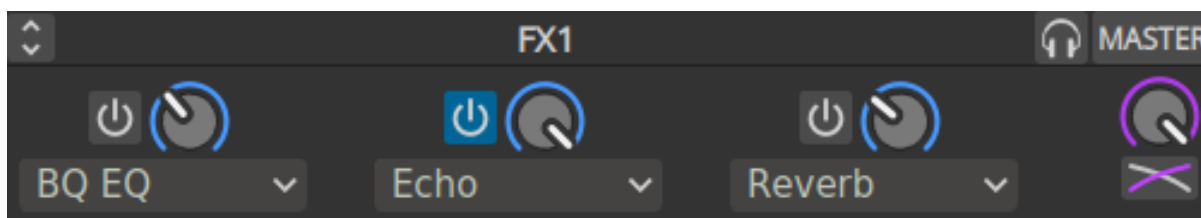


Figure4: An effect unit in Dry/Wet mode

In the default Dry/Wet mode, the mix knob crossfades between the dry signal (input to the first effect in the unit) and the wet signal (output of the last effect in the unit). When the knob is fully left, no sound will be heard from the effect unit. When the knob is fully right, only the output of the effect unit will be heard without any of the dry input signal.

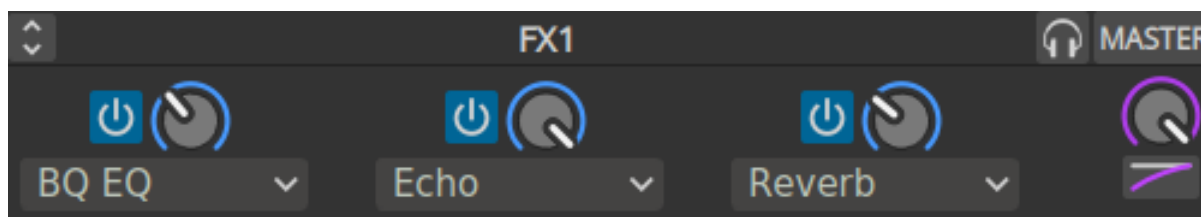


Figure5: An effect unit in Dry+Wet mode

In Dry+Wet mode, the dry signal is always at full volume. The mix knob controls how much of the wet signal is added to the dry signal. A useful example for Dry+Wet mode is loading a filter or equalizer effect before the Echo effect. With this setup, the filter or equalizer can remove bass frequencies from the echoed signal without removing them from the unechoed signal. If the effect unit is switched to Dry/Wet mode with the same effects in the same order, the bass frequencies will be taken out of the echoed signal, but they will also be removed from the unechoed signal.

## 11.4 Effects In Headphones

To preview how a track will sound with effects before you mix in the track, simply assign the deck to an effect unit and enable the headphone button for the deck. Note that this will increase CPU compared to assigning an effect unit to a deck without the headphone button enabled. This is because effects are processed in parallel for the headphone output (prefader) and master output (postfader).

Effects can also be previewed in headphones on decks that are playing to the audience, but this requires a few more steps:

1. Disable the effect unit for the deck
2. Enable the headphone button for the deck
3. Enable the headphone button for the effect unit
4. Enable effects and adjust their metaknobs and parameters for how you want the effects to sound.
5. When you are ready to mix in the effect, turn the mix knob fully left (dry)
6. Enable the effect unit for the deck
7. Turn the mix knob right so the effects are heard in the master output
8. Disable the headphone button for the effect unit so you do not forget to turn it off later

## 11.5 Controller Effects Mapping

This section describes how to use the standard Mixxx mapping for effects sections on controllers with the typical layout of 4 knobs (or 3 knobs + 1 encoder) and 4 buttons for effects. It allows you to switch between controlling all 3 effects of a chain or controlling each parameter of one effect.

By default, 3 knobs are used for controlling effect metaknobs and the buttons under them toggle each effect on/off. To temporarily toggle an effect on/off, press and hold an effect enable button. The 4th knob is used for the mix (dry/wet) knob of the whole chain. On controllers designed for Serato, the *Beats* encoder is used as the mix knob.

The button next to the mix knob is used for focusing effects. On controllers designed for Serato, this button is labeled *Tap*. While holding this button pressed, the enable buttons for each effect switch to choosing which effect is focused. When an effect is focused:

- the knobs switch to controlling the first 3 parameters of that effect instead of the metaknobs of each effect in the chain;
- the buttons switch to controlling the first 3 button parameters of the effect; and

The LED of the focus button indicates whether any effect is focused. When its LED is on, you can hold the button down to see which effect is focused without having to look at the computer screen. To unfocus the effect, you can short press the focus button. The focused effect is remembered, so when you short press the focus button again, the previously focused effect will be refocused (it is not remembered when you shut down Mixxx though).

Alternatively, if you want to keep all the parameters showing on screen, you can press and hold the focus button, then press the button of the focused effect to unfocus it. Showing all the parameters without having an effect focused is helpful for experimenting with different ways of linking parameters to the metaknob.

When the focus button is pressed with shift, it toggles the Effect Unit between controlling different *Effect Units* in Mixxx. Typically this is used to toggle between *Effect Units* 1 & 3 or 2 & 4, like deck toggle buttons.


To load different effects, hold shift and turn the knob for an effect.

To load chain presets 1-4, press the buttons with shift (new in Mixxx 2.4).

Mixxx was designed to be easy to learn for both novice and experienced DJs.

This part of the manual provides you with directions for using Mixxx's features to create your own mixes.

## 12.1 Setting Your Levels Properly (Gain Staging)



Setting your levels properly, also known as gain staging, is essential for getting the best sound quality out of the equipment you are using. At every link in your signal chain, from Mixxx's channel gains to the power amplifier, the level should be well above the noise floor, but lower than the maximum level before the signal clips. The *level meters* should mostly be around the top of their green region. The level meter pictured to the left shows where Mixxx's level meters should average at the loudest parts of tracks. The average level should not be in the yellow region. Use the yellow region to leave headroom, or available level above the average before the signal clips. The loudest parts of the music (the transients) should briefly go into the yellow region.

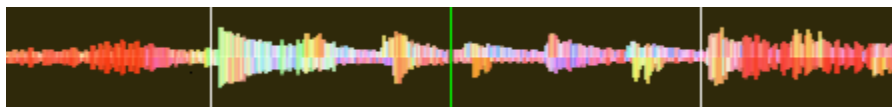


Figure1: A waveform at a good level. Note that the example waveforms in this section were made by adjusting the visual gain of the waveform display in Mixxx to illustrate the concepts. Adjusting the visual gain of the waveform display does not change the level of the audio.

**If a level meter is in its red region, the signal is clipping and the gain should be turned down.** Some equipment doesn't have a level meter and only has an LED that turns on when the signal clips. Clipping means that the peaks of the waveform are flattened because the equipment has reached the maximum level that it can amplify the signal to. This distorts sound in an unpleasant way and can damage equipment by driving it with more power than it is designed to handle. Increasing the gain past the point where clipping begins (further into the red on a meter) will distort the signal more. If you want to make the sound louder when every part of the signal chain is at its maximum without clipping, use more speakers or use speakers that are more sensitive and convert electrical energy into acoustic energy more efficiently.

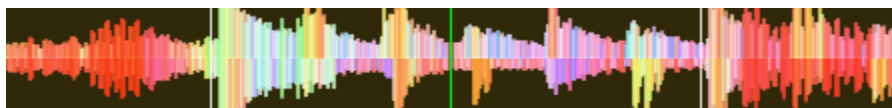


Figure2: A clipping waveform



On the other hand, the signal should not be too low. This is because every audio device generates a little noise at a level referred to as its noise floor. Additionally, analog signals pick up noise as they travel along wires. The signal measured by the meter on the left is relatively close to the noise floor. When a device is turned up, the noise floor does not go up; only the signal does. However, every time the signal is amplified by the gain of another piece of equipment, both the noise and the signal from previous devices in the signal chain are amplified. For example, if your sound card is turned down and you turn the gain up on your mixer to compensate, the signal-to-noise ratio (SNR) of the sound card output will be low and the mixer's gain will amplify the signal and the noise from the sound card, plus the noise picked up along the wire. The end result will have more noise than if the output of the sound card was turned up before the signal reached the mixer and the mixer's gain did not have to be turned up.



Figure3: A waveform that is too close to the noise floor

---

**Poznámka:** To adjust the output volume of a sound system while maintaining a high signal-to-noise ratio, the gain should be adjusted as close to the speakers as possible. Refer to the *Gain Knob* section for details.

---

## 12.2 Beatmatching and Mixing

*Beatmatching* is the process of adjusting the playback rate of a track so that it matches the tempo of another track. Beatmatching also involves adjusting the *phase* of the beats in a track so that they are aligned with the beats in the other track. Matching the *tempo* and aligning the beats are the two things a DJ must do to beatmatch.

Mixxx can match the tempo and align the beats for you but this requires an accurately detected BPM value and a correct beat grid for both tracks. To enable this feature, tap the *SYNC* button. To beatmatch manually, the tempo of the two tracks must be synchronized by adjusting the playback rate sliders. You can adjust the phase of the beats by right-clicking and dragging on either waveform display to temporarily speed up or slow down one of the tracks until the beats are aligned. The temporary pitch bend buttons can also be used to momentarily adjust the playback rate, allowing you to “shuffle” the beats in a track forwards or backwards, so they can be aligned with another track. See the chapter *Sync and Rate Controls*.

Two tracks are beatmatched once their tempos are matched and their beats are aligned. A “perfect” beatmatch is nearly impossible - there will always be a tiny difference in the playback rates. A keen DJ will keep his or her ears open and listen for the beats drifting out of alignment. This has a distinct “double bass kick” sound which is often preceded by the kick weakening in intensity as the two kicks drift out of phase. When this happens, the beats can be realigned by simply tapping one of the temporary pitch bend buttons a few times in the appropriate direction.

---

## 12.3 Master Sync

*Master Sync* is an intelligent assistant that allows you to leave the beatmatching to Mixxx so you can focus on track selection, effects manipulation, looping, 4 deck mixing, and other advanced DJing techniques.

To activate Master Sync on a deck, push and hold (or click and hold) the *SYNC* button. You'll know Master Sync is on because the sync button will remain lit. Changing the rate of any deck that has Sync lit will change the rates of all other decks that also have the Sync button lit. There is no need to set specific decks to be a master or followers. You can play, stop, eject, load, and queue any track with master sync and it won't interrupt the playback of the other decks. However, changing the rate of a stopped deck will change the rate of playing decks.

Master Sync will also notice if one of your tracks is double the BPM of another track and match them correctly. So if you want to mix a 140 BPM drum & bass track with a 70 bpm dubstep track, Master Sync will make sure they are lined up properly.

---

**Rada:** Usually, Master Sync will only make sure the rate sliders are set correctly. If you also want to make sure your beats are perfectly in sync, turn on the *QUANTIZE* button. When activated, Quantize will ensure that the beats are perfectly lined up as well.

---

## 12.4 Harmonic Mixing

Harmonic mixing is a technique to mix songs with matching melodies and harmonies. To learn more about harmonic mixing you might want to check out [the mixshare archive](#).

Mixxx has two features to help you with harmonic mixing. The first is a *key lock*. When it is active changing the speed of a track won't affect the key. To enable *key lock*, click the *key lock* button in the *Deck Options Button Grid*. The second is that Mixxx can automatically detect the key of a track and will display it in the library and the decks. The notation which is used to display a key can be changed in *Key Detection Preferences*.

## 12.5 Recording Your Mix

With the integrated recording feature you can record your mix as an audio file and listen to it later, distribute it as a *Podcast* or burn it to CD.

By default, Mixxx records the Master output. Alternatively, Mixxx can record the output of an external mixer if you have an audio interface configured for the *Record/Broadcast* input in *Preferences ▶ Sound Hardware*. Refer to *Recording And Broadcasting With An External Hardware Mixer* for details.

Mixxx can record your mix in various audio formats and quality settings. You can split your recordings, generate *cue files*, choose a custom recording directory and even set your own *metadata*. By default, Mixxx saves your recordings as lossless WAV files to a `Mixxx/Recordings` sub-folder in the Mixxx music directory. Before you start recording, we recommend that you adjust the settings in *Preferences ▶ Recording*.

If you click on the *Recordings* icon in the sidebar of the Mixxx library, the track table to the right displays the content of your recordings directory. New recordings are automatically saved to this directory as well as CUE files if you choose to create them in the preferences.

---

**Rada:** Recording your mixes and listening to them later to critique yourself is a great tool for improving your DJing skills. Sending your mixes to other DJs for feedback can also be helpful.

---

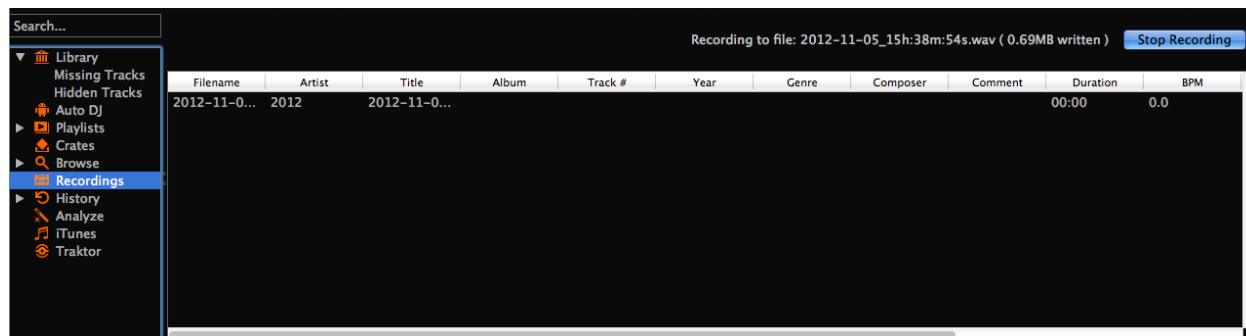


Figure4: Mixxx library - Recordings view

### 12.5.1 Record your mix to disk

Click on the recording icon on top of the application window to start, and stop the recording session. The duration of the running recording is displayed next to the icon.

---

**Poznámka:** The recording widget might not be available in some skins.

---

Alternatively:

- Click on the *Recordings* icon in the sidebar to switch to the *Recordings* view
- Click the *Start Recording* button or click *Options* ▶ *Record Mix* in the menu on top of the Mixxx application window.
- The display above the track table shows how much data has already been recorded, as well as the duration of the recording.
- Perform your mix
- Click the *Stop Recording* button to stop the recording when the mix has finished.

---

**Rada:** You can instantly play your recording as a track in Mixxx. Simply drag-and-drop the track to a deck.

---

## 12.6 Intro and Outro Cues

Intro/Outro cues are used to mix in and out of tracks and they can also be *used with Auto DJ* to mix in and out at those points. Unlike *hotcues*, they do not mark only one point. The intro and outro are sections and each section is defined by two points.

Mixxx detects where the first and last sounds are when analyzing a track making it easy to stop playing silent parts at the beginning and at the end of the track. The analyzer places the intro start point at the first sound and the outro end point at the last sound. The first and last sounds are determined by the first time the signal rises above -60 dBFS and the last time it goes below -60 dBFS.

You can decide where to place the intro end and the outro start points. Marking the whole intro and outro allows Mixxx to calculate how long each section is and show it on the overview waveform. This information can help you know when to press play on the next track. *Auto DJ can also use this information* to line up tracks and determine how long to crossfade.

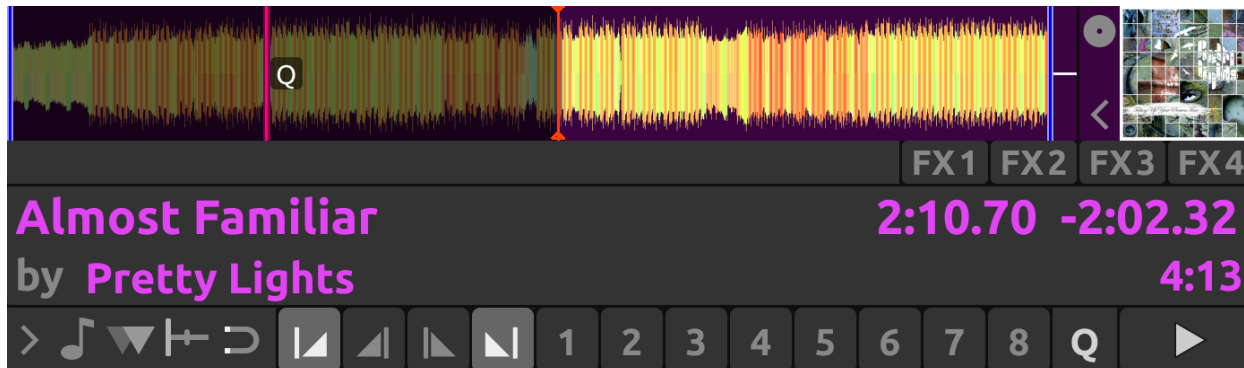


Figure5: Intro start and outro end cues placed at the first and last sound of a track respectively by the analyzer

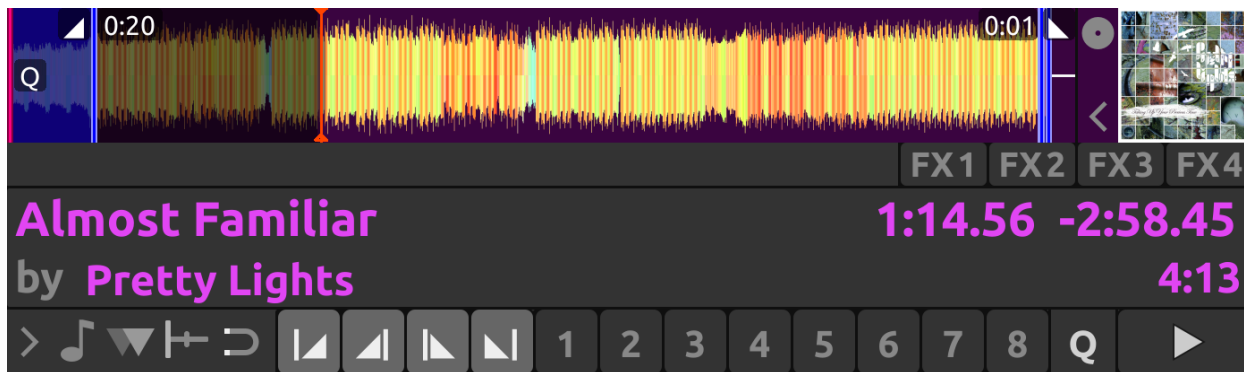


Figure6: Intro end and outro start markers placed on the track alongside the intro start and the outro end markers to make sections



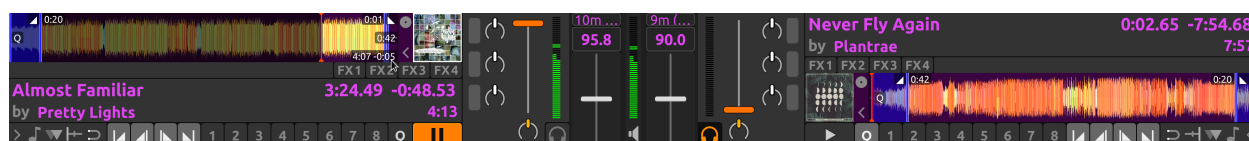
## 12.6.1 Mixing With Intro & Outro Cues

You can line up the ends of the intro and outro so the energy of the new song picks up right as the old track ends.



Figure7: Intro and outro end markers aligned showing on the waveform

If the intro and outro are short enough to fit on the waveform at the same time, you can simply watch the waveforms and press play when the markers align. Otherwise, you can watch the time remaining on the deck and hover your mouse cursor over the outro end line on the overview waveform. Mixxx will show you how much time is left until that point. When that time equals the length of the intro, press play to precisely line up the end of the intro. If you have changed the tempo of the track, the time remaining scales to show the real time remaining at the adjusted tempo.



Alternatively, you can line up the start of the intro & outro. In this case, watch the scrolling waveform when the outro start point is coming up on the old track. Then, press play on the new track when the old track reaches the outro start.

---

**Poznámka:** There are no rules on which method should be used. The method you decide to use is an artistic judgement based on the musical content of each track, what the vibe of the crowd is at that moment, and what you want to do with the mix.

---

If you do not have the intro end and outro start points marked when loading a track, you can find and mark these in headphones before you mix in the track. Then seek back to the intro start to get the track ready. If you don't have time to find the outro end point before mixing in the track, you can use the deck cloning feature to drag and drop to a new deck, then seek ahead on the new deck and listen in headphones to mark the outro start point. If you don't have time to do that either, you can right click on the overview waveform where you think the outro starts. Then look for how far that is from the end of the track to compare it to the length of the intro of the next track.

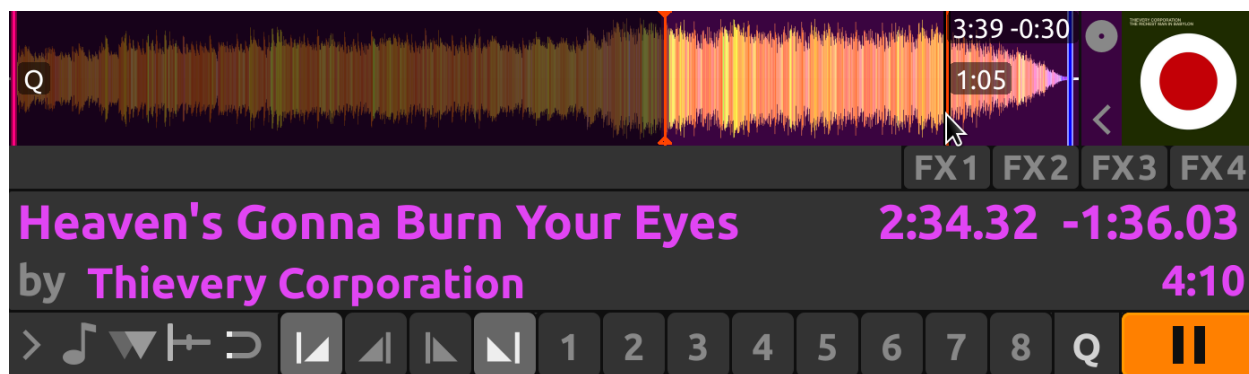


Figure8: Here the outro is approximately 30 seconds long. However, the analyzer placed the outro end point before the end of the track, so the outro is actually a little bit shorter than 30 seconds. Without marking the outro start point, you would have to do some math in your head to find exactly how long the outro is.

## 12.7 Auto DJ

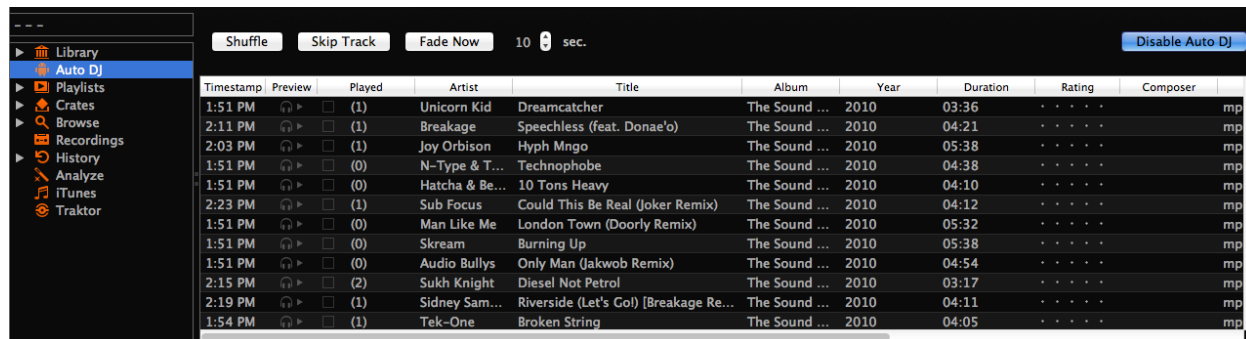


Figure9: Mixxx library - Auto DJ view

Auto DJ allows you to automatically load tracks from the Auto DJ playlist when the current track is nearly finished, and crossfade into it. See [Auto DJ - Automate your mix](#).

Auto DJ does not take into account the volume of each track, nor the frequency content, nor the rhythms, so it's not intended to be a replacement for a human DJ. However, it is good enough to give a human DJ a break without a major disruption to the mix.

### 12.7.1 Loading tracks into Auto DJ

To play tracks automatically, they must first be loaded into the Auto DJ playlist. The Auto DJ playlist is empty by default.

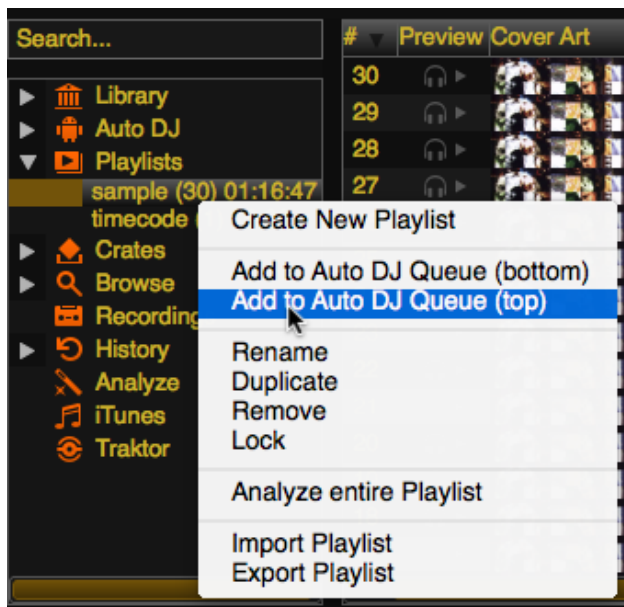


Figure10: Mixxx library - Adding a playlist to Auto DJ

There are several ways to load tracks into the Auto DJ playlist:

- Select single or multiple tracks from the library, a regular playlist or crate and drag them to the Auto DJ icon on the left.

- Select a regular playlist or crate, right-click with the mouse and select *Add to Auto DJ* from the mouse menu. This adds all tracks to Auto DJ.
- While in the Auto DJ view of the library, drag tracks from external file managers to the Auto DJ icon in the sidebar or to the Auto DJ track table on the right.

## 12.7.2 Automatically adding tracks from crates

Instead of inserting the contents of the crates directly into the Auto DJ playlist, *Auto DJ crates* are associated crates that become a source of random track selections.

- **Add a crate to the Auto DJ track sources:** Expand the *Auto DJ* item in the sidebar, and right-click on the *Crates* sub-item. Select one of the existing crates from the list. This makes the content of the crate available for random selection into the *Auto DJ* queue for automatic mixing. The name of the crate is now listed as sub-item below.

If you do not have any crates available for selection in the list, go and create a crate first and add some tracks, see *Crates - Organizing tracks into collections*.

- **Manually add tracks from track sources to Auto DJ:** Click the *Auto DJ* item in the sidebar and switch to the Auto DJ view of the library. Click the *Add random* button to add one track at the time from Auto DJ track sources to the bottom of the Auto DJ playlist.

---

**Poznámka:** If you did not set any crates as an Auto DJ track source, clicking the *Add Random* button will add a randomly selected track from the whole library instead.

---

- **Automatically add tracks from track sources to Auto DJ:** Mixxx can randomly add tracks to the Auto DJ playlist from track sources as soon as a specified minimum number of tracks remain. Select *Preferences* ▶ *Auto DJ* ▶ *Enable random track addition to queue* ▶ *On*. Set *Minimum allowed tracks before addition* to change the minimum number of tracks after which random tracks may be added.

---

**Poznámka:** If you did not set any crates as an Auto DJ track source, tracks are selected from the whole library instead.

---

- **Remove a crate from the Auto DJ track sources:** Expand the *Auto DJ* item in the sidebar, and right-click on one of the crates listed below the *Crates* sub-item. Select *Remove Crate as Track Source* to remove the crate from the Auto DJ track sources.

## 12.7.3 Playing tracks in Auto DJ

Now that you have loaded tracks into the Auto DJ playlist, you can activate Auto DJ as follows:

- Click on the *Auto DJ* icon in the sidebar to switch to the *Auto DJ* view of the library.
- Click the *Enable Auto DJ* button.
- The first tracks from your list are loaded into the decks and the playback starts.
- Mixxx will continue to automatically mix until the Auto DJ playlist is empty.
- Click the *Disable Auto DJ* button to stop the automatic mixing

---

**Rada:** You can add a track to the end of the Auto DJ playlist once it is played instead of removing it. Set *Preferences* ▶ *Auto DJ* ▶ *Re-queue tracks after playback* ▶ *On*.

---

## 12.7.4 Auto DJ Mix Modes

Auto DJ has several modes it can use to mix tracks:

- Full Intro + Outro
- Fade At Outro Start
- Full Track
- Skip Silence

The Full Intro + Outro and Fade At Outro Start modes both use the intro and outro cues but the Full Track and Skip Silence cues do not.

### Full Intro + Outro and Fade At Outro Start

The Full Intro + Outro and Fade At Outro start Auto DJ modes use the *intro & outro cues* to determine how long to crossfade and how to align the tracks in time. In both modes, Auto DJ compares the duration of the outro of the old track and the intro of the new track. Auto DJ picks the time of the shorter section as the crossfade time.

In both modes, if the outro is shorter than the intro, Auto DJ will align the start of the outro with the start of the intro.

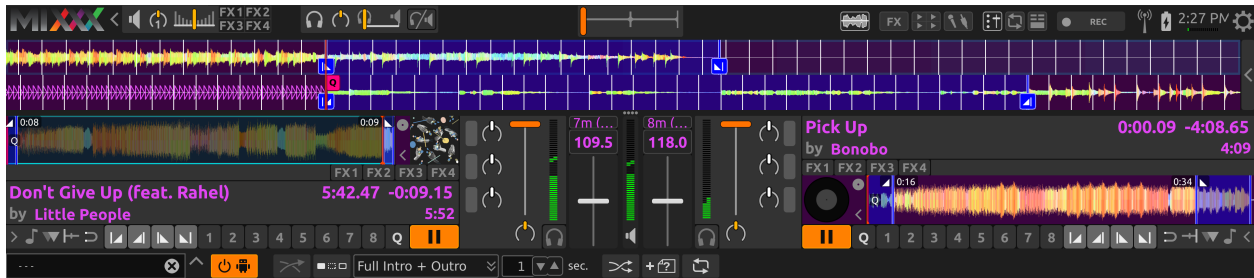


Figure11: Intro and outro start markers aligned by Auto DJ

The two modes work differently when the outro is longer than the intro.

**Full Intro + Outro Mode** The full length of both the intro and the outro are always played. When the outro is longer than the intro, Auto DJ aligns the end of the intro and outro by starting the next track during the outro of the previous track. This mode is the most likely to sound good with the widest variety of tracks. Therefore, it is the default mode.

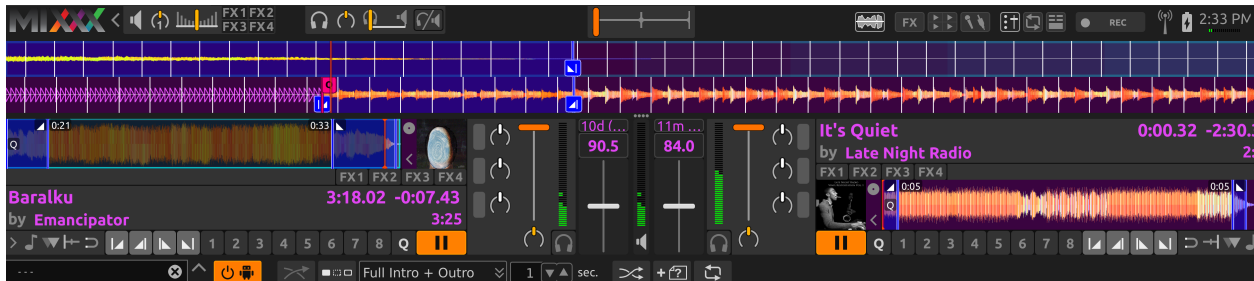


Figure12: Intro and outro start cues aligned by Auto DJ in Full Intro + Outro mode

**Fade At Outro Start** Auto DJ always aligns the start of the intro and outro. When the outro is longer than the intro, Auto DJ cuts off the end of the outro. This can be helpful if you want to prevent the energy of the mix from declining during a long outro. However, the transition may sound abrupt if the intro is short.

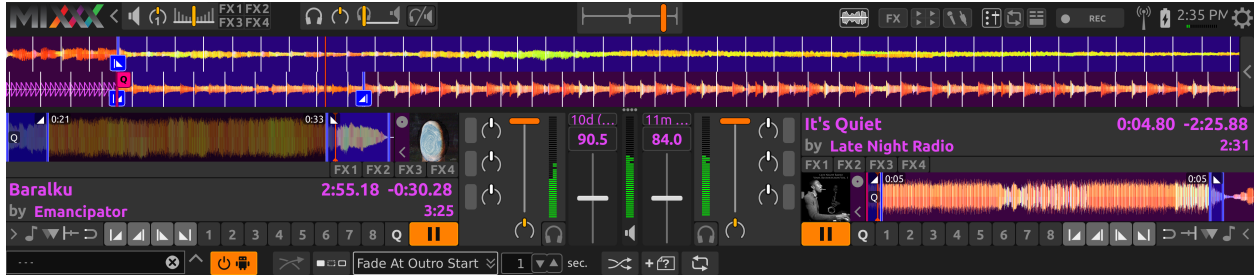


Figure13: Intro and outro start cues aligned by Auto DJ in Fade At Outro mode

### Full Track and Skip Silence

The Full Track and Skip Silence Auto DJ modes do not use the intro and outro cues. They crossfade tracks over a number of seconds that you specify in the box next to the mode selection menu. The Full Track mode plays the entire length of every file. The Skip Silence mode works the same way but automatically cuts out silence at the beginning and ends of tracks (defined as when the signal first goes above or below -60 dBFS).

**Rada:** If you still want to use Auto DJ like a normal music player application, use one of these modes with a transition time of 0 seconds.

Mixxx has many options to customize in *Options* ▶ *Preferences*.

## 13.1 Zvuk

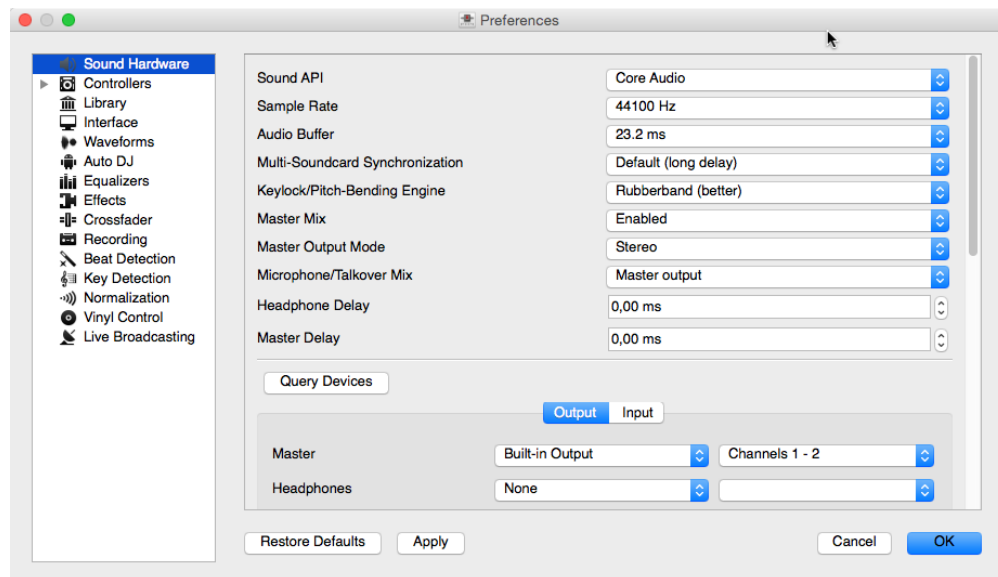


Figure 1: Mixxx Sound Hardware Preferences

To achieve the best performance with Mixxx it is essential to configure your *audio buffer*, *sample rate*, and *audio API*. These three factors largely determine Mixxx's responsiveness and reliability. The optimal settings will vary based on your computer and hardware quality.

For information about the different input and output options, refer to *Configuring sound input and output*. The *DJ Hardware*

chapter explains different types of DJ hardware and how to set them up with the input and output options. If you are using a microphone, refer to the *Using Microphones* chapter.

### 13.1.1 Vyrovnávací paměť zvuku

The audio buffer is the amount of audio in milliseconds that Mixxx processes at a time. This is a major factor that determines the latency between changing controls in Mixxx such as moving the crossfader or pressing the play button and hearing the audio change. For example, with an audio buffer of 23 ms, it will take approximately 23 milliseconds for Mixxx to stop the audio after you toggle the play button. The actual latency will be longer depending on a variety of factors.

The audio buffer setting determines how quickly your *operating system* expects Mixxx to react. A smaller audio buffer means Mixxx will be more responsive, but requires a faster CPU and quality audio interface. Setting your audio buffer too small may be too much for your computer and audio interface to handle. In this situation, Mixxx playback will be choppy and very clearly distorted as your system will not be able to keep up with how frequently Mixxx is trying to process audio. It is recommended to set your audio buffer as small as your system can handle reliably without glitches. Experiment with different audio buffer sizes to find what works for your system.

An audio buffer between 23-64 ms is acceptable if you are using Mixxx with a keyboard/mouse or a controller. An audio buffer below 10 ms is recommended when vinyl control is used because Mixxx will feel unresponsive otherwise.

The [Adjusting Audio Latency](#) page on the Mixxx Wiki has tips for different operating systems that may help you use a smaller audio buffer reliably.

**Varování:** Your system may glitch only occasionally if you have your audio buffer set a little too low. This will happen at unpredictable times and you may not notice if you are only doing brief tests of each audio buffer size. When you think you have found a good buffer size for your system, play with Mixxx for at least a half hour before performing to ensure no glitches happen.

### 13.1.2 Vzorkovací kmitočet

The sample rate setting in Mixxx controls how many samples (chunks of audio) per second are produced by Mixxx. This determines the maximum frequency in Mixxx's signal, which is half the sample rate. Humans can only hear up to 20 kHz, so there is generally no need to use more than a 44.1 kHz (44100 Hz) sample rate for playback. Most music is published with a 44100 Hz sample rate, and playing music in a different sample rate than the audio file slightly reduces sound quality.

**Varování:** A sample rate of 96 kHz gives your computer less than half the time to do the same processing. Increasing the sample rate will increase CPU usage and likely raise the minimum audio buffer size you can use reliably.

### 13.1.3 Zvukové API

The Sound *API* that Mixxx uses is the method by which Mixxx talks to your *operating system* in order to deliver audio to your audio interface. Your choice of Sound API can drastically affect Mixxx's performance on your computer. **Therefore it is important to take care to choose the best Sound API available to you.**

## Windows

The following Sound APIs are available on Windows:

- **ASIO**: Good
- **WDM-KS**: Good
- **WASAPI**: Acceptable
- **DirectSound**: Poor
- **MME**: Poor

It is best to use the ASIO Sound API that bypasses the sound processing of the Windows kernel. Using ASIO requires a driver for your audio interface from the manufacturer. Check the manufacturer's website to see if a driver download is available. Soundcards designed for musicians almost always have an ASIO driver available, but audio interfaces built into computers and very cheap audio interfaces typically do not have an ASIO driver. If there is no ASIO driver available for your audio interface, use the WDM-KS API. There is generally no advantage to using [ASIO4ALL](#), a wrapper around the WDM-KS API.

## GNU/Linux

The following Sound APIs are available on GNU/Linux:

- **ALSA**: Good
- **JACK**: Good
- **OSS**: Acceptable

ALSA is the simplest sound API to configure. Using ALSA will prevent any other programs from using the audio interface(s) that Mixxx is using.

JACK allows you to route audio between JACK-compatible applications in flexible ways and output sound from multiple programs at the same time. However, JACK can be complicated to set up. Unless you will be connecting Mixxx to other JACK-compatible applications, JACK offers no advantages over ALSA. To use JACK, start the JACK daemon *before* running Mixxx. Otherwise JACK will not appear as a Sound API in the preferences.

Most modern GNU/Linux distributions use PulseAudio by default. When launched from a GUI menu entry or icon, Mixxx suspends PulseAudio while it is running so that Mixxx can use ALSA directly. Like JACK, PulseAudio allows multiple programs to access one audio interface, but PulseAudio and JACK have opposite design goals. PulseAudio is designed to make ordinary computer usage such as watching videos online and listening to music easy whereas JACK is designed for demanding low latency audio programs like Mixxx. It can be difficult to setup JACK and PulseAudio to work well together. So, unless you already use JACK, it is easiest to let Mixxx suspend PulseAudio and use ALSA.

If the PulseAudio plugin for alsalibs is installed on GNU/Linux, you can choose the virtual device `pulse`. This allows Mixxx to share the default system audio interface with other media players. This only works if you start Mixxx without `pasuspender`, which you can do by running `mi.xxx` from a console rather than clicking the launcher icon in a menu or on your desktop. Since the sound stream is routed from ALSA to Pulse and back to ALSA, this adds an additional latency of  $\sim 2 \times$  the selected audio buffer size.

OSS is an Sound API that predates ALSA. Few modern audio interfaces have OSS drivers.



## macOS

CoreAudio is the only Sound API on macOS.

### 13.1.4 Other Sound Hardware options

- **Multi-Soundcard Synchronization:** Mixxx is able to use two or more *audio interface* at a time, each with its own clock. When multiple audio interfaces are in use, the Mixxx engine is driven by the Master audio interface. Here you can select the synchronization used for the other audio interfaces to avoid buffer overflows or underflows.
- **Keylock/Pitch-Bending Engine:** This allows you to select the engine used for independent tempo and pitch changes (e.g. *key lock*). Use *Soundtouch* on lower power machines (such as netbooks) or if you experience buffer underflows while using *key lock*.
- **Master Mix:** You may disable the master mix to reduce Mixxx's CPU usage if you do not use the Master output, recording or live broadcasting.
- **Master Output Mode:** In Mono mode, the left and right channel are combined into a mono signal which is passed to both channels of your master audio interface. This is useful for setups where the audience cannot hear your mix in stereo because of speaker placement or playing in a space with lots of reverberation.
- **Buffer Underflow Count:** Underflows (data is not available when needed) indicate that some of the subsystems in Mixxx can't keep up with real-time deadlines imposed by the current audio buffer size. This is useful to tune the latency settings. If the counter increases, then increase your audio buffer size, decrease the sample rate setting or change the sound API setting if available.

## 13.2 Knihovna

### 13.2.1 Changing music directories

You can manually add, relink, and remove Mixxx music directories in *Preferences ▶ Library*.

**Add a new music directory** Mixxx handles multiple music library folders. Click *Add* to browse to a directory where your music is stored. Mixxx will watch this directory and its subdirectories for new tracks.

If you add a directory that is already in your library, or you are currently *rescanning your library*, the operation is canceled.

Directories can also be added from the *Computer* sidebar item inside the library.

**Relink a existing music directory** If an existing music directory is moved or renamed, Mixxx doesn't know where to find the audio files in it. The tracks will still show in library but they can't be loaded on decks and the tracks will become missing after the library is rescanned.

To relink the music directory, go to *Preferences ▶ Library*, click *Relink* to select the music directory in its new location. This will re-establish the links to the audio files in the Mixxx library.

Click *Library ▶ Rescan Library*, this will update the tracks and cause them to show up again in the library, check the location column in the library and you should see that it now points to the new music directory.

The playlists, crates, history and all track data like cue points, rating, comments etc. are preserved after relinking the library, the tracks are also not re-analyzed as the cached analysis is reused. You can confirm this by checking the *Key* and *BPM* columns to see if there is a key and BPM for every track.

---

**Poznámka:** It is recommended to take a backup of your Mixxx configuration files before moving or renaming music directories used in Mixxx. The location of Mixxx configuration files vary based on the *operating system*:

- Windows: %USERPROFILE%\Local Settings\Application Data\Mixxx\
- macOS: Library/Application Support/Mixxx
- Linux: The configuration files are found in the `.mixxx` folder in your home directory.

---

**Remove a music directory** Click *Remove*, and Mixxx will no longer watch a directory and its subdirectories for new tracks, and asks what would you like to do with the tracks from these directories.

- Select *Hide Tracks* to hide all tracks from this directory and subdirectories.
- Select *Delete Track Metadata* to delete all metadata for these tracks from Mixxx permanently
- Select *Leave Tracks Unchanged* to leave the tracks unchanged in your library.

Hiding tracks saves their metadata in case you re-add them in the future.

Popisná data obsahují všechny podrobnosti o skladbě (umělec, název, počet přehrání atd.), stejně tak jako rytmické mřížky, rychlé značky a smyčky. Toto rozhodnutí ovlivní jen knihovnu Mixxxu. Žádné soubory na pevném disku nebudou změněny nebo smazány.

---

**Rada:** When changing music directories, you might want to run a library rescan afterwards. Select *Library* ▶ *Rescan Library* in the menu.

---

## 13.3 Colors

### 13.3.1 Selecting Palettes and Default Colors

**Track/Hotcue Palette** It is possible to assign colors to tracks and *hotcues* to highlight them. Mixxx offers a selection of different palettes that you can choose colors from.

You can also make your own palette by clicking the *Edit* button next to the palette name. This will open a the *palette editor*.

**Default Hotcue Color** It is possible to either select a single color from the current hotcue palette that will be assigned to all newly created hotcue colors, or select the *By hotcue number* option. The latter will automatically assign a color from the palette to new hotcues based on the hotcue number.

### 13.3.2 Editing Palettes

The palette editor allows you add, modify, reorder and remove colors in a color palette and save it under a new name. You can also delete custom palettes.

To modify a color, double-click it to open the color picker window.

When editing the hotcue palette, the palette editor also allows assigning a specific default color to a hotcue number. This setting is honored when *Hotcue default color* is set to *By hotcue number*.

That makes it possible to assign just a few distinct colors to your hotcues automatically, but still have more colors available in the palette in case you want to override the color manually.

For example, if you have a color palette consisting of 4 colors, and you configure the palette to assign the first color to hotcue 1 and the third color to hotcue 2, then hotcue 1 will use the first color and hotcue 2 will be assigned the third color. The colors will repeat when setting a hotcue with a number greater than those used in the palette. In the example above, hotcue 3 will use the first color and hotcue 4 the second color, and so on.

You can also assign the same color to more than one hotcue. For example by separating the hotcue numbers with a comma (“2, 4, 7”), expressing a range of numbers (“1 - 3”) or any combination of that (“6, 1 - 3, 7”).

If no color in the palette has a hotcue number assigned to it, new hotcues will use the color at the same index from the palette (i.e. hotcue 1 will be assigned the first color in the palette, hotcue 2 uses the second color in the palette, etc.).

### 13.3.3 Replacing Hotcue Colors

Sometimes it’s necessary to bulk replace colors of existing hotcues in your library. Instead of changing the color of every single hotcue manually, you can *Replace* to open the cue color replace dialog.

In addition to selecting the new color, you may set conditions that determine which hotcues the new color will be assigned to. If you tick both checkboxes, the new color will only applied if both conditions match.

## 13.4 Rytmus

Mixxx uses an ultra-precise BPM and beat detector. Manual adjustments are redundant in many cases because Mixxx knows where the beats are.

BPM and beat detection is a complex operation. Depending on your computer and the track’s bitrate and duration this may take some time. By default Mixxx analyzes the complete track. To accelerate beat detection on slower computers, a “Fast Analysis” option is available. If enabled, the BPM is computed by analyzing the first minute of the track. In most cases this does not affect the beat detection negatively because most of today’s dance music is written in a 4/4 signature with a fixed tempo.

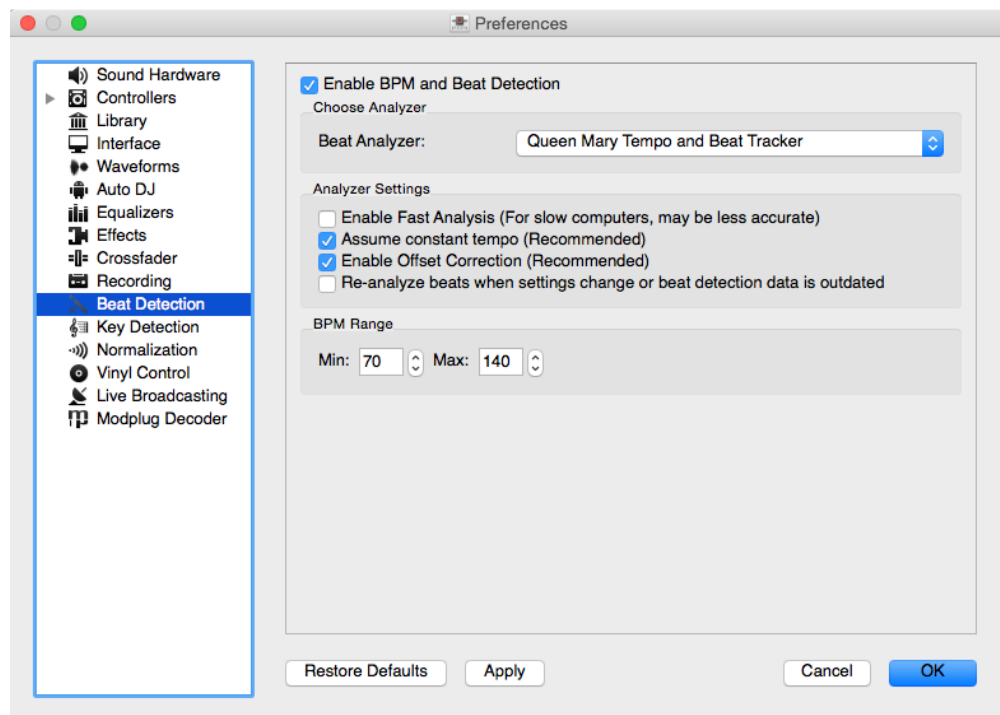


Figure2: Mixxx preferences - BPM settings

The table below summarizes the beat detection settings:

Option	Popis
Enable Fast Analysis	If enabled, BPM will be detected by only using the first minute of audio.
Předpokládat neměnné tempo	If enabled, Mixxx assumes that the distances between the beats are constant. If disabled, the raw beat grid obtained by the analyzer is presented. The latter is appropriate for tracks with variable tempo.
Enable Offset Correction	Prevents beat markers from being placed incorrectly.
Provést nový rozbor rytmu při změně nastavení, nebo když jsou data pro rozpoznávání rytmu zastaralá	If enabled, Mixxx over-writes old beat grids from prior versions. Moreover, it will re-analyze the BPM if your beat detection preferences change or BPM data from 3rd party programs are present.

### 13.4.1 Correcting Beat Grids

There may be situations where BPM and beat detection do not result in a proper beat grid.

Typically, the detected BPM is correct but the analyzer has failed to detect the location of the first beat. Consequently, the beat markers are shifted, i.e. the beat markers are a fixed distance from the true beat. To adjust the beat grid, cue the track before a real beat and click the *Beat-grid Adjust* button in the *Deck Options Button Grid*.

If the detected BPM is not accurate, the corresponding beat grid will also be inaccurate. A deviation of 0.02 BPM units from the correct BPM will cause beatgrid alignment issues on long tracks (e.g. a club mix). If this happens, your beatgrid may look aligned for the few minutes but you will notice a slight drift as the song goes on. Finding the correct BPM is easy in many cases - just follow the note below.

---

**Poznámka:** If the detected BPM value is not sufficiently accurate but very close to an integer value, try to set the BPM value manually to the integer.

---

## 13.5 Tónina

Mixxx comes with a high precision musical key detector to help you make smooth mixes by ensuring that your tracks are musically compatible.

### 13.5.1 Nastavení rozboru

Key detection is a complex operation. Depending on your computer and the track's bitrate and duration this may take some time. By default Mixxx analyzes the complete track. To accelerate key detection on slower computers, a "Fast Analysis" option is available. If enabled, the key is computed by analyzing the first minute of the track.

The table below summarizes the Key detection settings:

Option	Popis
Enable Fast Analysis	If enabled, the key will be detected by using only the first minute of audio.
Re-analyze key when settings change or Key detection data is outdated	If enabled, Mixxx will re-analyse tracks if you select a different key detection plugin or the key was generated by a program other than Mixxx.
Notace tóniny	Change the way keys are displayed in the library.

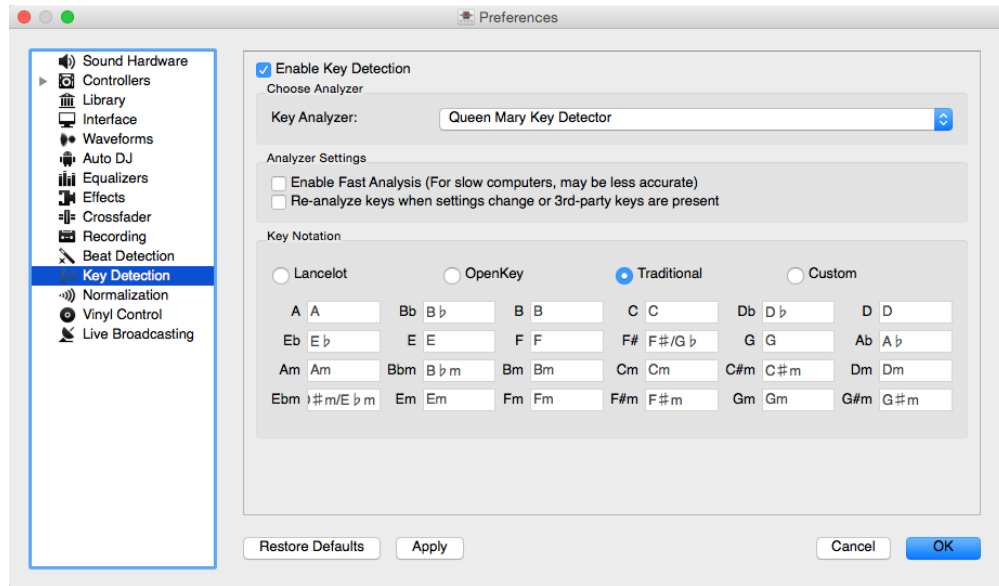


Figure3: Mixxx preferences - Key settings

## 13.6 Ekvalizéry

*Preferences* ▶ *Equalizer* allows you to setup the equalizers.

- **Equalizer Rack:** The Equalizer Rack is a special effect rack that is connected to the deck's equalizer and filter controls.
- In this section you can select the equalizers and quick effects that are used with the decks.
- **Equalizer Plugin:** Here you can select the effect that is used as the mixing EQ in each deck. By default only built-in equalizers are displayed. Unchecking *Only allow EQ knobs to control EQ specific effects* allows you to select any other effect.
- **Quick Effect:** Here you can select the effect that is controlled by the dedicated filter knob in each deck. By default only built-in filter effects are selected for all decks, but that can be changed as above.
- **High/Low Shelf EQ:** This slider sets the crossover frequencies of the mixing EQ. It controls which frequency range is affected by the low, mid, and high channel EQ knobs. By default the low knob controls the bass and sub bass range up to 246 Hz. The mid knob controls the mid range up to 2.5 kHz. The remaining treble range is controlled by the high knob.
- **Master EQ:** This section allows you to setup an EQ that affects the master output.

### 13.6.1 Mixing Equalizers

Mixxx offers three types of mixing equalizers with a full kill option. These equalizers are „isolators“, adapted from analog crossover networks. Each EQ is combination of a high shelf filter, a band pass filter, and a low shelf filter. Each EQ type has a unique sound, so try them out to find out which one you prefer.

The Bessel EQs with Lipshitz and Vanderkooy Mix (LV-Mix) do not alter the sound or take any processing time when their knobs are in the center position. They activate once you adjust an EQ knob.

The Linkwitz-Riley EQ on the other hand always applies a minimum, natural sounding phase shift to the sound. Their processing time does not change when you adjust the EQ knobs.

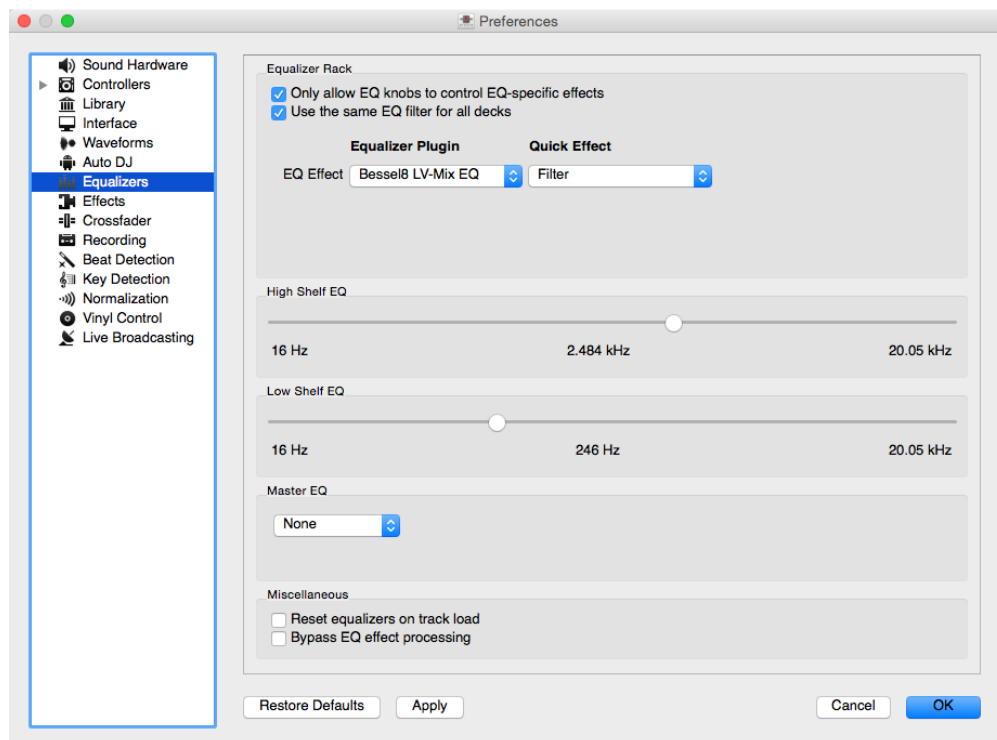


Figure4: Nastavení ekvalizéru

The following table compares some technical parameters:

Typ	cut	roll-off	phase shift	bit perfect	CPU usage
Bessel4 LV-Mix	soft	-24 db/Oct	linear	yes	low
Bessel8 LV-Mix	medium	-48 db/Oct	linear	yes	medium
Linkwitz-Riley	sharp	-48 db/Oct	minimum	no	high

- cut: the frequency response (curve form) at the cross over frequency.
- roll-off: The steepness of the EQ bands.
- linear phase: No phase distortion, all frequencies are processed with the same group delay.
- minimum phase: A natural phase distortion, the group delay changes by the frequency.
- bit perfect: Whether the EQ leaves the original samples untouched when the EQ is at unity.
- CPU usage: Processing time needed to calculate the EQ output.

### 14.1 Adding support for your MIDI/HID Controller

With several dozens of DJ controllers supported out-of-the-box, Mixxx gives you comprehensive hardware control for your DJ mixes, see *Using MIDI/HID Controllers*.

Support for additional devices can be added to Mixxx by creating a new mapping file. This file tells Mixxx how to translate, or map, *MIDI/HID* messages from a controller into commands that Mixxx understands.

You can download and share custom controller mappings in the [Mixxx User customizations forums](#).

For a list of controls that can be used in a controller mapping, see *Mixxx Controls*.

#### 14.1.1 Controller Wizard

By far, the easiest way to create a new MIDI mapping is by using the Controller Wizard.

1. Connect your controller(s) to your computer
2. Start Mixxx
3. Go to *Preferences* ▶ *Controllers*
4. Select your device from the list of available devices on the left, and the right pane will change
5. Activate the *Enabled* checkbox
6. Click on *Learning Wizard* to open the selection dialog
7. Click any control in the Mixxx *GUI*
8. Alternatively, click the *Choose Control* button and choose one from the selection list
9. Push *Learn* and then move a control or push a button on your controller to map it. You can also move a control without pushing the *Learn* button if you are learning many controls.
10. If you are learning a button, just push it once. If you are learning a knob or a slider, try to move it throughout its range.

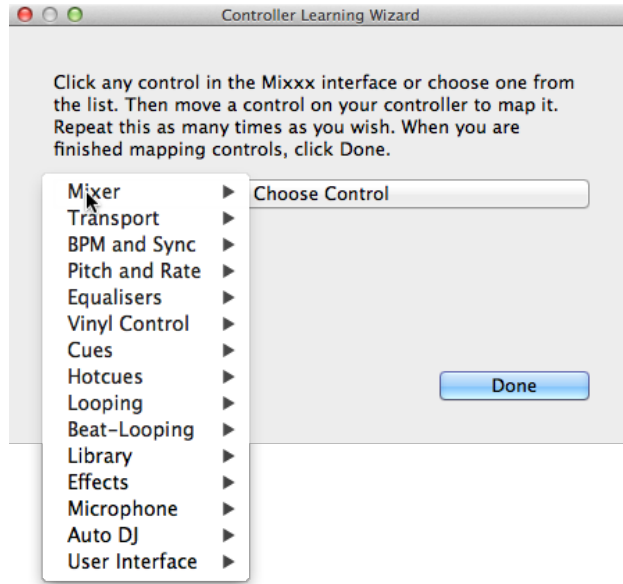


Figure1: Mixxx Controller Wizard - Mapping a control

11. After Mixxx detects the control, you may click *Learn Another* or you can click on another button in the Mixxx GUI to learn another control.
12. When you are finished mapping controls, click *Done*

There are also some advanced options in the Midi Wizard you may need to use:

- **Soft Takeover:** Use this option for knobs or sliders to avoid sudden jumps in when the knob in the GUI doesn't match the physical knob. If you select this option, you won't be able to perform super-fast motions as easily. (Recommended off)
- **Invert:** Use this option for controls that you want to work backwards from how they were detected.
- **Switch Mode:** Use this option on controllers that have controls that act like switches and emit one value on the first press, and a different value on the next press. (Think an actual hardware toggle switch, or a button that lights up on the first press and turns off on the second press).
- **Jog Wheel / Select Knob:** Use this for knobs that don't have a beginning or an end, but spin continuously.

The Controller wizard saves the new mapping to the following file paths:

- **Linux:** /home/<username>/.mixxx/controllers
- **macOS:** /Users/<username>/Library/Containers/org.mixxx.mixxx/Data/Library/Application Support/Mixxx/controllers
- **Windows:** %LOCALAPPDATA%\Mixxx\controllers

You can then modify the XML file it creates (or any of the ones that ship with Mixxx) if you'd like to fine-tune it or add more mappings. For more information, go to <https://github.com/mixxdj/mixxx/wiki/MIDI-Controller-Mapping-File-Format>.

The Controller Wizard works only for *MIDI* devices. Currently you can't map modifier (shift) keys and platter rotations. Use *MIDI Scripting* instead.



## 14.1.2 MIDI Scripting

In order to support the advanced features of many *MIDI/HID* controllers, Mixxx offers what we call MIDI Scripting.

It enables MIDI controls to be mapped to *QtScript* (aka Javascript/EMCAScript) functions stored in function library files, freeing Mixxx from a one-to-one MIDI mapping ideology. These user-created functions can then do anything desired with the MIDI event such as have a single controller button simultaneously affect two or more Mixxx properties (“controls”), adjust incoming control values to work better with Mixxx (scratching), display a complex LED sequence, or even send messages to text displays on the controller.

For more information, go to <https://github.com/mixxxdj/mixxx/wiki/Midi-Scripting> and [https://github.com/mixxxdj/mixxx/wiki/hid\\_mapping\\_format](https://github.com/mixxxdj/mixxx/wiki/hid_mapping_format).

## 14.2 Making a Custom Keyboard Mapping

The *default keyboard mappings* are defined in a text file which can be found at the following location:

- Linux: `/usr/local/share/mixxx/keyboard/en_US.kbd.cfg`
- macOS: `/Applications/Mixxx.app/Contents/Resources/keyboard/en_US.kbd.cfg`
- Windows: `<Mixxx installation directory>\keyboard\en_US.kbd.cfg`

Depending on your system’s language settings, Mixxx might use a different file as default, e.g. `de_DE.kbd.cfg` for German or `es_ES.kbd.cfg` for Spanish.

It is not recommended that you modify the system-wide keyboard mapping file because all your changes may be lost if you uninstall or upgrade Mixxx. Instead, copy the default mapping file to the following location:

- Linux: `~/ .mixxx/Custom.kbd.cfg`
- macOS: `~/Library/Containers/org.mixxx.mixxx/Data/Library/Application Support/Mixxx/Custom.kbd.cfg`
- Windows: `%LOCALAPPDATA%\Mixxx\Custom.kbd.cfg`

Then edit this file and save the changes. On the next startup, Mixxx will check if `Custom.kbd.cfg` is present and load that file instead of the default mapping file. This has the advantage that you can always revert back to the default mapping by deleting `Custom.kbd.cfg`.

For a list of controls that can be used in a keyboard mapping, see *Mixxx Controls*.

You can download and share custom keyboard mappings in the [Mixxx User customizations forums](#).

## 14.3 Additional Effects via external Mixer Mode

Mixxx comes with a set of *native effects*. Additionally, using the *external mixer mode* you can route each deck directly to 3rd party effect hosts.

The following examples are only intended to encourage experimentation, they are not definitive guidance.

### 14.3.1 Effects via AU Lab on macOS

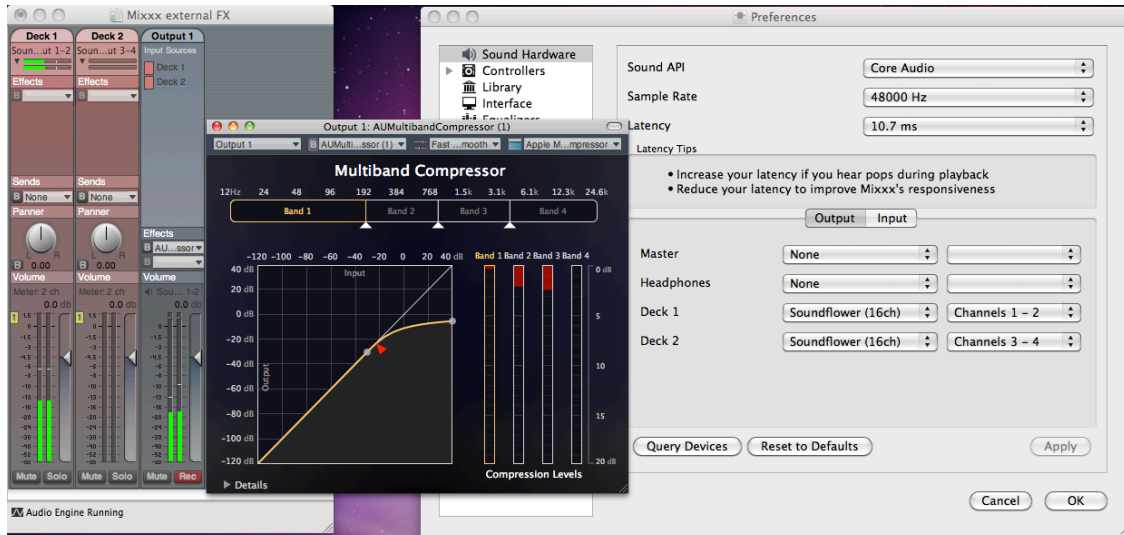


Figure2: The Au Lab routing for external effects on macOS

On macOS there is a simple and free way to give Mixxx access to the collection of AU/VST/MAS plugins that are installed on your system.

- Install the free [SoundflowerBed](#), a system extension for inter-application audio routing.
- Install the free digital audio mixing application [AU Lab](#).

**Rada:** macOS 10.15 (Catalina) introduces additional restrictions on running non-Apple software. Make sure to grant permission in *System Preferences* → *Security & Privacy* → *Microphone* → *AU Lab*.

Alternatively, try [Blackhole](#), a modern virtual audio driver that allows applications to pass audio to other applications with zero additional *latency*, and [Hosting AU](#), a free micro-sized *DAW* that hosts Audio Unit instruments and effect plugins.

#### In Mixxx

- Go to *Preferences* ▶ *Sound Hardware* ▶ *Output*
- Select for *Deck 1* the *Soundflower 16* device with *Channel 1-2*
- Select for *Deck 2* the *Soundflower 16* device with *Channel 3-4*
- Click the *Apply* button

#### In AU Lab

- Click on the + button to create a new configuration
- Add 2 stereo input tracks in the *Audio Input Tab*
- Add 2 stereo output tracks in the *Audio Output Tab*
- Click *OK*
- Change the audio input device to *Soundflower 16*
- Change the audio output device for example to *Built-in Output*
- Click *Create document*

- In the *Output 1* channel, select an effect from the drop-down menu, for example *Apple* ▶ *AUCompressor*

The effect should now react if you play a track in Mixxx.

### 14.3.2 Effects via JACK Rack on GNU/Linux

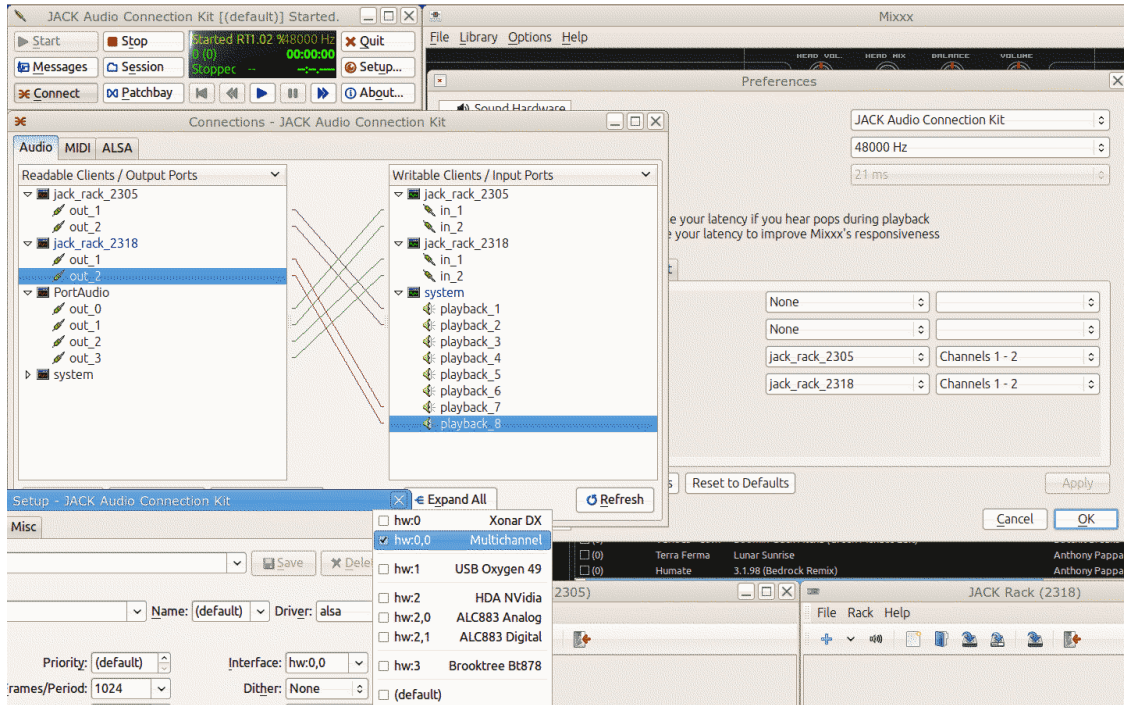


Figure3: The Jack routing for external effects on GNU/Linux

Use *Jack* to route each deck directly through *JACK Rack* effect racks, or for more control you can use *Ardour* (or another *DAW*) using sends for effects. This gives *Mixxx* access to the extensive collection of *LADSPA* plugins.

Make sure the correct multichannel audio interface has been selected in *Jack* (*Jack* settings visible bottom left). Note that *Mixxx* possibly labels its *Jack* ports as *Portaudio*.

## 14.4 Deleting Your Library

The library file is stored in the following places depending on your *operating system*:

**Windows** The *Mixxx* library is stored in the `%USERPROFILE%\Local Settings\Application Data\Mixxx\` folder. To delete your library on *Windows*, delete the `mixxxdb.sqlite` file in this folder.

**macOS** The *Mixxx* library is stored in the `Library/Application Support/Mixxx` folder in your home directory. To delete your library on *macOS* type the following command into a terminal:

```
rm ~/Library/Application\ Support/Mixxx/mixxxdb.sqlite
```

**GNU/Linux** The *Mixxx* library is stored in the `.mixxx` folder in your home directory. To delete your library on *GNU/Linux* type the following command into a terminal:

```
rm ~/.mixxx/mixxxdb.sqlite
```

**Varování:** Deleting your library will lose all of your *metadata*. This includes saved hotcues, loops, comments, ratings, and other library related metadata. Only delete your library if you are fine with losing these.

---

## Contributing to the Project

---

Mixxx is a community-driven project involving many DJs worldwide. Without the contributions from these DJs, Mixxx would not exist, and we're constantly looking for more contributors.

We want to make it as easy as possible for people to get involved in Mixxx development. First off, to be clear **you don't have to be a programmer to contribute**.

### 15.1 As a non-developer

- [Donate](#) to Mixxx using Paypal to help support and enhance development.
- [Make skins](#)
- [Make MIDI mappings](#) for your controller
- [Report bugs](#)
- Update our [Wiki](#) to make sure the information on it is up to date.
- Translate Mixxx using [Transifex](#)
- Answer questions on the [Troubleshooting Forum](#) and the [Launchpad Answers page](#)
- Help promote Mixxx: If you've got a blog, write an article about Mixxx. Blog about our new releases when they come out. Any exposure on the web and in print helps our project grow, and is much appreciated.
- Send us some photos of you using Mixxx at a gig!

## 15.2 As a software developer

- You don't have to know C++. There are developers who got into Mixxx development while learning C++ along the way.
- Join our mailing list: [mixxx-devel](#)
- Join our [Zulip chat](#).
- Tell us your ideas! Email us, talk on *Zulip*, file wishlist *bugs*, or post on the forums.
- Register an account on our development platform [Github](#) and fork our code. See [Using Git for Mixxx Development](#)
- Join the [Mixxx Development Team](#) on Launchpad.
- Get familiar with the code. Pick a bug off of the [Easy Bug List](#)
- Get help fixing the bug on *Zulip*, talk to us about the best way to do it.

### Contributing to Mixxx in 3 Easy Steps:

1. Pick a bug off of the [Easy Bug List](#)
2. Talk to us on *Zulip* for help on how to fix it.
3. Fix it! *Done!* Your name will now be on the [Mixxx contributor list](#).

## 16.1 Keyboard mapping table

Shortcuts let you work more efficiently. This table lists the default keyboard shortcuts for accessing Mixxx with an English keyboard layout. Depending on your language settings the defaults may be different.

<b>Mixer</b>		
Crossfade Left	G	
Crossfade Right	H	
Small Crossfade Left	Shift + G	
Small Crossfade Right	Shift + H	
Microphone Talkover	`	
	Deck 1	Deck 2
Activate/Disable Effects Unit	5	0
Headphone Cue (Pre-fader listen)	T	Y
Bass Kill	B	N

<b>Playback</b>	<b>Deck 1</b>	<b>Deck 2</b>
Load selected track to deck	Shift + ←	Shift + →
Unload track from deck	Alt + Shift + ←	Alt + Shift + →
Playback	D	L
Seek Backwards	A	J
Reverse Playback	Shift + A	Shift + J
Seek Forwards	S	K
Tempo Adjust Down	F1	F5
Tempo Adjust Up	F2	F6
Small Tempo Adjust Down	Shift + F1	Shift + F5
Small Tempo Adjust Up	Shift + F2	Shift + F6
Temporary Tempo Adjust Down	F3	F7
Temporary Tempo Adjust Up	F4	F8
Small Temporary Tempo Adjust Down	Shift + F3	Shift + F7
Small Temporary Tempo Adjust Up	Shift + F4	Shift + F8
Tempo (BPM) Sync	1	6
BPM Tap Tempo Adjust	Shift + !	Shift + ^

<b>Cue and Loop</b>	<b>Deck 1</b>	<b>Deck 2</b>
Set Cuepoint	Shift + D	Shift + L
Cue	F	;
Cue Go-to-and-stop	Shift + F	Shift + :
Activate 4 Beat Loop	Q	U
Halve Beat Loop Size	W	I
Double Beat Loop Size	E	O
Set Loop In Point	2	7
Set Loop Out Point (Activates Loop)	3	8
Toggle Loop On/Off	4	9
Set Hot cue 1	Z	M
Set Hot cue 2	X	,
Set Hot cue 3	C	.
Set Hot cue 4	V	/
Clear Hot cue 1	Shift + Z	Shift + M
Clear Hot cue 2	Shift + X	Shift + <
Clear Hot cue 3	Shift + C	Shift + >
Clear Hot cue 4	Shift + V	Shift + ?

<b>Vinyl Control</b>	<b>Deck 1</b>	<b>Deck 2</b>
Passthrough	Ctrl + J	Ctrl + K
Vinylcontrol Mode	Ctrl + Shift + Y	Ctrl + Shift + U
Vinylcontrol Cueing	Ctrl + Alt + Y	Ctrl + Alt + U
Single Deck Toggle / Swap Decks	Ctrl + T	



<b>Library</b>	
Maximize the library view	Space
Load selected track to deck	Shift + ← (Deck 1)   Shift + → (Deck 2)
Select all files in current list	Ctrl + A
Select group of files in current list	Click first item, hold Shift key, click last item
Focus search bar	Ctrl + F
Clear search input	Ctrl + Back
Exit search and leave focus	ESC

<b>Preview Deck</b>	
Load Selected Track And Play	Alt + Return
Start/Stop	Return
Seek Backwards	Alt + ←
Seek Forwards	Alt + →

<b>Auto DJ</b>	
Shuffle Auto DJ playlist	Shift + F9
Skip next track in Auto DJ	Shift + F10
Fade to next track in Auto DJ	Shift + F11
Toggle Auto DJ on/Off	Shift + F12

**Viz také:**

Mixxx allows you to customize the keyboard shortcuts. For more information, and to download a keyboard mapping image, go to [Using a Keyboard](#).

## 16.2 Application shortcuts

Use these shortcuts to access features available in the menu on top of the application window.

Function	Key
<b>File menu</b>	
Load Track to Deck 1	Ctrl + O
Load Track to Deck 2	Ctrl + Shift + O
<b>Library menu</b>	
Add new Playlist	Ctrl + N
Add new Crate	Ctrl + Shift + N
<b>View menu</b>	
Show Skin Settings Menu	Ctrl + 1
Show Microphone Section	Ctrl + 2
Show Vinyl Control Section	Ctrl + 3
Show Preview Deck	Ctrl + 4
Show Cover Art	Ctrl + 6
Maximize Library	Space
Full Screen (Windows & Linux)	F11
Full Screen (macOS)	Shift + Command + F
<b>Options menu</b>	

continues on next page

Table 1 – pokračujte na předchozí stránce

Function	Key
Enable Vinyl Control 1	Ctrl + T
Enable Vinyl Control 2	Ctrl + Y
Enable Vinyl Control 3	Ctrl + U
Enable Vinyl Control 4	Ctrl + I
Enable Live Broadcasting	Ctrl + L
Record Mix	Ctrl + R
Enable Keyboard Shortcuts	Ctrl + `
Open Preferences (Windows & Linux)	Ctrl + P
Open Preferences (macOS)	Ctrl + ,
Exit Mixxx	Ctrl + Q
<b>Developer menu</b>	
Reload skin	Ctrl + Shift + R
Developer Tools	Ctrl + Shift + T
Stats: Base Bucket	Ctrl + Shift + B
Stats: Experiment Bucket	Ctrl + Shift + E
Debugger Enabled	Ctrl + Shift + D

**Rada:** The `Ctrl` key on Windows & Linux is equivalent to the `Command` key on macOS. The `Alt` key on Windows & Linux is equivalent to the `Option` key on macOS. Application shortcuts are not customizable.

**Poznámka:** The Developer menu is available if Mixxx is started using the *Launching Mixxx from the Command Line*.

## 16.3 Launching Mixxx from the Command Line

To launch Mixxx using the command line:

**Windows** Locate the folder where Mixxx is installed and copy the file path. You can also obtain the file path by right clicking on the Mixxx icon and then choosing ‚Properties‘. Copy the path and paste it in terminal and add `\mixxx` at the end - everything should be inside quotes. Press Enter. The entire command will look like this

```
"C:\Program Files\Mixxx\mixxx"
```

**macOS** Similarly, for macOS, the command will be

```
/Applications/Mixxx.app/Contents/MacOS/mixxx
```

**GNU/Linux** If Mixxx is in your system’s search path (`$PATH`), you can just type

```
mixxx
```

### 16.3.1 Command line options

Mixxx is designed to be as user-friendly as possible. As such, its command line options are only useful for development or debugging, as they make these tasks easier. Command line options are case-sensitive.

To launch Mixxx with any of these command line options, simply append the option to the base command for your operating system. For example, to launch Mixxx in developer mode on Windows, type this in terminal

```
"C:\Program Files\Mixxx\mixxx" --developer
```

In addition to these options, it is possible to specify one or more music file(s) on the command line. These will be loaded at start-up. Each file you specify will be loaded into the next virtual deck. For a list of supported file types, go to *Compatible file formats*.

- resourcePath PATH** Top-level directory where Mixxx will look for its resource files such as MIDI mappings, overriding the default installation location.
- pluginPath PATH** Top-level directory where Mixxx will look for sound source plugins in addition to default locations.
- settingsPath PATH** Top-level directory where Mixxx will look for user settings files such as the library database and preferences configuration file.
- controllerDebug** Log all controller data Mixxx sends and receives as well as scripts it loads.
- developer** Enable developer mode. Includes extra logs, stats on performance and the Developer tools menu as well as tooltips and logs useful for skin developers.
- safeMode** Disable OpenGL widgets (scrolling waveforms, spinnies) to work around GPU driver bugs. If Mixxx is crashing on startup, try using this.
  - Automatically loads empty waveforms
  - Disables spinning vinyl widgets
  - Disables synchronization polling
  - Doesn't open controllers by default
- locale LOCALE** Use a custom locale for loading translations (e.g fr)
- f, --fullScreen** Start Mixxx in full-screen mode.
- logLevel LEVEL** Set the verbosity of command line logging.

Value	Meaning
critical	Critical/Fatal only
warning	Above + Warnings
info	Above + Informational messages
debug	Above + Debug/Developer messages
trace	Above + Profiling messages

- logFlushLevel LEVEL** Set the logging level at which the log buffer is flushed to `mixxx.log`. LEVEL is one of the values defined at `--logLevel` above.
- debugAssertBreak** Breaks (SIGINT) Mixxx if a `DEBUG_ASSERT` evaluates to false. A debugger can then be used to continue. This disables the `MIXXX_DEBUG_ASSERTIONS_FATAL` flag which can otherwise lead to a time consuming full rebuild.
- h, --help** Display this help message and exit

## 16.4 Developer tools

To start Mixxx in Developer mode from a custom resource directory with *MIDI* and *HID* logging enabled, type the following line into the terminal and hit return:

```
./mixxx --controllerDebug --developer --resourcePath res
```

### 16.4.1 Experiment modes for rapid development and testing

- Adds a static Experiment class with a tri-state mode flag that indicates whether the experiment mode is OFF, BASE, or EXPERIMENT.
- Adds *Developer* ▶ *Stats:Experiment Bucket* and *Developer* ▶ *Stats:Base Bucket*. Each one toggles between OFF and BASE/EXPERIMENT so you can choose exactly what time spans you would like to collect in your base and experiment buckets.
- Updates StatsManager to segment collected stats into a base and experiment bucket. This allows you to quickly measure the difference a code change has on relevant counters / timers within the same execution of Mixxx.

This is useful for quickly enabling and disabling a code change without a re-compile/re-run cycle to get an anecdotal sense of how it „feels“ as well as a quantified sense of how it differs in terms of stats Mixxx collects.

All stats collected via the usual Counter/Timer/ScopedTimer/etc. tools are segmented into a BASE STATS and EXPERIMENT STATS section printed to the log on exit.

## 16.5 Mixxx Controls

Nearly every knob, button, or fader you see in Mixxx’s interface is controllable via Mixxx’s „control“ system. The control system allows skins, *MIDI* controllers, *HID* controllers and keyboards to control Mixxx via a single interface.

A control is identified by a „group“ (which is used for grouping associated controls) and a „key“ (the name of the individual control).

For example, the volume fader for Deck 1 is identified by the group `[Channel1]` and key `volume`. Similarly, the volume fader for Sampler 1 is identified by the group `[Sampler1]` and key `volume`.

The group is used to collect all the controls that affect one component of Mixxx into one collection. Some groups have a high overlap of controls in common (e.g. samplers, decks, and the preview deck all share the same control keys).

In addition to controlling Mixxx, the control system can be used to inspect Mixxx’s state. For example, the sample rate of the track loaded in Deck 1 can be accessed via the `[Channel1],track_samplerate` control. You can read the `[Channel3],play` control to determine whether Deck 3 is playing.

The default value range is 0.0 to 1.0, unless otherwise noted. Binary means that it is either ‚ON‘ (non-zero) or ‚OFF‘ (zero).

---

#### **Rada:** Discovering Controls used in Skins

You can view the control connected to any part of a skin by running Mixxx with the `--developer` command line option and hovering your mouse cursor over part of the skin. If no tooltip appears, enable tooltips for the Library and Skin in *Options* ▶ *Preferences* ▶ *Interface*.

---



---

#### **Rada:** Changing any control from the GUI in Developer Mode

---

When running Mixxx in Developer Mode (with the `--developer` command line option), you can view and manually set the state of any control in Mixxx by going to *Developer* ▶ *Developer Tools*.

**Viz také:**

See `controlindex` for a full list.

### 16.5.1 ControlPotMeter controls

The following extensions add some features to `ControlPotMeter` controls (volume, *crossfader*, ...). Use in conjunction with `[ChannelN]`, `[SamplerN]`, `[Master]`, ... groups.

Control Suffix	Description, example
<code>_up</code>	Increases the value, e.g. <code>[ChannelN], rate_perm_up</code> sets the speed one step higher (4 % default)
<code>_down</code>	Decreases the value, sets the speed one step lower (4 % default)
<code>_up_small</code>	Increases the value by smaller step, sets the speed one small step higher (1 % default)
<code>_down_small</code>	Decreases the value by smaller step, sets the speed one small step lower (1 % default)
<code>_set_one</code>	Sets the value to 1.0, sets the channel volume to full
<code>_set_minus_one</code>	Sets the value to -1.0, sets the channel volume to zero
<code>_set_default</code>	Input: sets the control to its default, return to default waveform zoom level
<code>_set_default</code>	Output: set to 1.0 if the control is at its default, light up the pitch fader center indicator
<code>_set_zero</code>	Sets the value to 0.0, put the crossfader in the middle again
<code>_toggle</code>	Sets the value to 0.0 if the value was > 0.0, and to 1.0 if the value was 0.0, will cut off/on a track while you're playing
<code>_minus_toggle</code>	Sets the value to -1.0 if the value was > -1.0, and to 1.0 if the value was -1.0, can tilt the crossfader from left to right

These controls can be used in JavaScript files like this:

```
// This won't work:
engine.setValue(group, "pitch_up_small", 1.0);

// Use this instead:
script.triggerControl(group, "pitch_up_small", 50);
```

To use `*_toggle` the respective shortcut for scripts is:

```
script.toggleControl(group, "keylock_toggle", 100);
```

### 16.5.2 The `[Master]` group

The `[Master]` group generally corresponds to controls that affect the mixing engine. This will bear some similarity to what you will find on a DJ mixer (e.g. *crossfader* controls, headphone cueing controls, etc.).

**[Master]audio\_latency\_usage**

Reflects fraction of *latency*, given by the audio buffer size, spend for audio processing inside Mixxx. At value near 25 % there is a high risk of buffer underflows

This is a *ControlPotMeter control*.

**Range** 0 .. 25 %

**Feedback** latency meter

Nové ve verzi 2.0.0.

**[Master]audio\_latency\_overload**

Indicates a buffer under or over-flow. Resets after 500 ms

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Overload indicator

Nové ve verzi 2.0.0.

**[Master]audio\_latency\_overload\_count**

Counts buffer over and under-flows. Max one per 500 ms

**Range** 0 .. n

**Feedback** Counter in hardware preferences

Nové ve verzi 2.0.0.

**[Master]balance**

Adjusts the left/right channel balance on the Master output.

This is a *ControlPotMeter control*.

**Range** -1.0..1.0

**Feedback** Center Balance knob

**[Master]booth\_enabled**

Indicates whether a Booth output is configured in the *Sound Hardware Preferences*.

**Range** binary

**Feedback** Booth gain knob shown or hidden

Nové ve verzi 2.1.0.

**[Master]booth\_gain**

Adjusts the gain of the Booth output.

This is a *ControlPotMeter control*.

**Range** 0.0...1.0...5.0

**Feedback** Booth gain knob

Nové ve verzi 2.1.0.

**[Master]crossfader**

Adjusts the *crossfader* between players/decks (-1.0 is all the way left).

This is a *ControlPotMeter control*.

**Range** -1.0..1.0

**Feedback** Crossfader slider

**[Master]crossfader\_down**

Moves the *crossfader* left by 1/10th.

**Range** binary

**Feedback** Crossfader slider

**[Master]crossfader\_down\_small**

Moves the *crossfader* left by 1/100th.

**Range** binary

**Feedback** Crossfader slider

Nové ve verzi 1.10.0.

**[Master] crossfader\_up**

Moves the *crossfader* right by 1/10th.

**Range** binary

**Feedback** Crossfader slider

**[Master] crossfader\_up\_small**

Moves the *crossfader* right by 1/100th.

**Range** binary

**Feedback** Crossfader slider

Nové ve verzi 1.10.0.

**[Master] duckStrength**

Microphone ducking strength

This is a *ControlPotMeter control*.

**Range** 0.0..1.0

**Feedback** Strength knob

Nové ve verzi 2.0.0.

**[Master] enabled**

Indicator that the master mix is processed.

**Range** binary

**Feedback** None

Nové ve verzi 2.0.0.

**[Master] gain**

Adjusts the gain for the master output as well as recording and broadcasting signal.

This is a *ControlPotMeter control*.

**Range** 0.0..1.0..5.0

**Feedback** Master volume knob

Nové ve verzi 2.0.0.

**[Master] headEnabled**

Indicator that the headphone mix is processed.

**Range** binary

**Feedback** None

Nové ve verzi 2.0.0.

**[Master] headGain**

Adjusts the headphone output gain.

This is a *ControlPotMeter control*.

**Range** 0.0..1.0..5.0

**Feedback** Headphone volume knob

Nové ve verzi 2.0.0.

**[Master]headMix**

Adjusts the cue/main mix in the headphone output.

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Pre/Main knob

**[Master]headSplit**

Splits headphone stereo cueing into right (master mono) and left (*PFL* mono).

**Range** binary

**Feedback** Split Cue button

Nové ve verzi 2.0.0.

**[Master]latency**

*Latency* setting (sound buffer size) in milliseconds (default 64).

**Range** >=0 (absolute value)

**Feedback** Latency slider in the prefs

**[Master]maximize\_library**

Toggle maximized view of library.

**Range** binary

**Feedback** Toggle maximized view of library

Nové ve verzi 2.0.0.

**[Master]num\_decks**

The number of decks currently enabled.

**Range** integer

**Feedback** None

Nové ve verzi 1.9.0.

**[Master]num\_effectsavailable**

The number of available effects that can be selected in an effect slot.

**Range** integer, read-only

**Feedback** None

Nové ve verzi 2.1.0.

**[Master]num\_samplers**

The number of samplers currently enabled.

**Range** integer

**Feedback** None

Nové ve verzi 1.9.0.

**[Master]num\_preview\_decks**

The number of preview decks currently enabled.

**Range** integer



**Feedback** None

Nové ve verzi 1.11.0.

**[Master]PeakIndicator**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) (composite).

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (mono)

**[Master]PeakIndicatorL**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) for the left channel.

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (left)

**[Master]PeakIndicatorR**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) for the right channel.

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (right)

**[Master]samplerate**

The current output sample rate (default: 44100 Hz).

**Range** absolute value (in Hz)

**Feedback** None

**[Master]talkoverDucking**

Toggle microphone ducking mode (OFF, AUTO, MANUAL)

**Range** FIXME

**Feedback** Ducking mode button

Nové ve verzi 2.0.0.

**[Master]VuMeter**

Outputs the current instantaneous master volume (composite).

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Master meter (mono)

**[Master]VuMeterL**

Outputs the current instantaneous master volume for the left channel.

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Master meter L

**[Master]VuMeterR**

Outputs the current instantaneous master volume for the right channel.

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Master meter R

**[Master]headVolume**

Adjust headphone volume.

**Range** 0.0..1.0..5.0

**Feedback** Headphone Gain knob

Zastaralé od verze 2.0.0: Use `[Master],headGain` instead.

**[Master]volume**

Adjust master volume.

**Range** 0.0..1.0..5.0

**Feedback** Master Gain knob

Zastaralé od verze 2.0.0: Use `[Master],gain` instead.

### 16.5.3 Decks, Preview Decks and Samplers

Each deck in Mixxx corresponds to a `[ChannelN]` group. Whenever you see `[ChannelN]`, think „Deck N“. N can range from 1 to the number of active decks in Mixxx.

Preview decks and Sample decks („samplers“) in Mixxx are identical to regular decks, they simply have a different purpose (previewing tracks or playing samples, respectively). Any control listed above for `[ChannelN]` will work for a samplers and preview decks, just replace `[ChannelN]` with `[PreviewDeckN]` or `[SamplerN]`.

**Viz také:**

There are some *additional global controls for samplers*.

**[ChannelN]back**

**[PreviewDeckN]back**

**[SamplerN]back**

Fast rewind (REW)

**Range** binary

**Feedback** << button

**[ChannelN]beat\_active**

**[PreviewDeckN]beat\_active**

**[SamplerN]beat\_active**

Indicates whether the player is currently positioned within 50 milliseconds of a beat or not. This can be used to make controller LEDs blink on every beat.

**Range** binary

**Feedback** None

Nové ve verzi 1.10.0.

**[ChannelN]beat\_closest**

**[PreviewDeckN]beat\_closest**

**[SamplerN]beat\_closest**

Its value is set to the sample position of the closest beat of the active beat and is used for updating the beat LEDs.  
:range: -1, 0.0, real-valued :feedback: None

**[ChannelN]beat\_distance**

[PreviewDeckN]beat\_distance

[SamplerN]beat\_distance

Outputs the relative position of the play marker in the section between the the previous and next beat marker.  
:range: 0.0 - 1.0, real-valued :feedback: None

[ChannelN]beat\_jump

[PreviewDeckN]beat\_jump

[SamplerN]beat\_jump

Jump forward by *beatjump\_size* (positive) or backward by *beatjump\_size* (negative). If a loop is active, the loop is moved by *beatjump\_size*.

**Range** real number, -1, 0, 1

**Feedback** Player jumps forward or backward by X beats.

Nové ve verzi 2.0.0.

[ChannelN]beat\_jump\_size

[PreviewDeckN]beat\_jump\_size

[SamplerN]beat\_jump\_size

Set the number of beats to jump with *beatjump\_forward/beatjump\_backward*.

**Range** positive real number

**Feedback** Beatjump size spinbox

Nové ve verzi 2.1.0.

[ChannelN]beat\_jump\_forward

[PreviewDeckN]beat\_jump\_forward

[SamplerN]beat\_jump\_forward

Jump forward by *beatjump\_size*. If a loop is active, the loop is moved forward by X beats.

**Range** binary

**Feedback** Player jumps forward by *beatjump\_size*.

Nové ve verzi 2.1.0.

[ChannelN]beat\_jump\_backward

[PreviewDeckN]beat\_jump\_backward

[SamplerN]beat\_jump\_backward

Jump backward by *beatjump\_size*. If a loop is active, the loop is moved backward by X beats.

**Range** binary

**Feedback** Player jumps backward by *beatjump\_size*.

Nové ve verzi 2.1.0.

[ChannelN]beat\_jump\_X\_forward

[PreviewDeckN]beat\_jump\_X\_forward

[SamplerN]beat\_jump\_X\_forward

Jump forward by X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64. If a loop is active, the loop is moved forward by X betas.

**Range** binary

**Feedback** Player jumps forward by X beats.

Nové ve verzi 2.0.0.

[ChannelN]beat\_jump\_X\_backward

[PreviewDeckN]beat\_jump\_X\_backward

**[SamplerN]beatjump\_X\_backward**

Jump backward by X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64. If a loop is active, the loop is moved backward by X beats.

**Range** binary

**Feedback** Player jumps backward by X beats.

Nové ve verzi 2.0.0.

**[ChannelN]beatloop\_activate**

**[PreviewDeckN]beatloop\_activate**

**[SamplerN]beatloop\_activate**

Set a loop that is *beatloop\_size* beats long and enables the loop

**Range** binary

**Feedback** A loop is shown over *beatloop\_size* beats

Nové ve verzi 2.1.0.

**[ChannelN]beatloop\_X\_activate**

**[PreviewDeckN]beatloop\_X\_activate**

**[SamplerN]beatloop\_X\_activate**

Activates a loop over X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64

**Range** binary

**Feedback** A loop is shown over X beats.

Nové ve verzi 1.10.0.

**[ChannelN]beatloop\_size**

**[PreviewDeckN]beatloop\_size**

**[SamplerN]beatloop\_size**

Set the length of the loop in beats that will get set with *beatloop\_activate* and *beatlooproll\_activate*. Changing this will resize an existing loop if the length of the loop matches *beatloop\_size*.

**Range** positive real number

**Feedback** Beatloop size spinbox and possibly loop section on waveform

Nové ve verzi 2.1.0.

**[ChannelN]beatloop\_X\_toggle**

**[PreviewDeckN]beatloop\_X\_toggle**

**[SamplerN]beatloop\_X\_toggle**

Toggles a loop over X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64

**Range** binary

**Feedback** A loop is shown over X beats.

Nové ve verzi 1.10.0.

**[ChannelN]beatloop\_X\_enabled**

**[PreviewDeckN]beatloop\_X\_enabled**

**[SamplerN]beatloop\_X\_enabled**

1 if beatloop X is enabled, 0 if not.

**Range** binary

**Feedback** Beatloop X button in skin is lit.

Nové ve verzi 1.10.0.

[ChannelN]beatlooproll\_activate  
 [PreviewDeckN]beatlooproll\_activate  
 [SamplerN]beatlooproll\_activate

Activates a rolling loop over *beatloop\_size* beats. Once disabled, playback will resume where the track would have been if it had not entered the loop.

**Range** binary

**Feedback** A loop overlay is shown over *beatloop\_size* beats on waveform.

Nové ve verzi 2.1.0.

[ChannelN]beatlooproll\_X\_activate  
 [PreviewDeckN]beatlooproll\_X\_activate  
 [SamplerN]beatlooproll\_X\_activate

Activates a rolling loop over X beats. Once disabled, playback will resume where the track would have been if it had not entered the loop. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64

**Range** binary

**Feedback** Beatloop X button in skin is lit. A loop overlay is shown over X beats on waveform.

Nové ve verzi 1.11.0.

[ChannelN]beats\_adjust\_faster  
 [PreviewDeckN]beats\_adjust\_faster  
 [SamplerN]beats\_adjust\_faster

Adjust the average *BPM* up by +0.01

**Range** binary

**Feedback** The *beatgrid* lines move closer to each other.

Nové ve verzi 2.0.0.

[ChannelN]beats\_adjust\_slower  
 [PreviewDeckN]beats\_adjust\_slower  
 [SamplerN]beats\_adjust\_slower

Adjust the average *BPM* down by -0.01.

**Range** binary

**Feedback** The *beatgrid* lines move further apart from each other.

Nové ve verzi 2.0.0.

[ChannelN]beats\_translate\_curpos  
 [PreviewDeckN]beats\_translate\_curpos  
 [SamplerN]beats\_translate\_curpos

Adjust *beatgrid* so closest beat is aligned with the current playposition.

**Range** binary

**Feedback** The *beatgrid* moves to align with current playposition.

Nové ve verzi 1.10.0.

[ChannelN]beats\_translate\_match\_alignment  
 [PreviewDeckN]beats\_translate\_match\_alignment  
 [SamplerN]beats\_translate\_match\_alignment

Adjust *beatgrid* to match another playing deck.

**Range** binary

**Feedback** Instead of *syncing* the beatgrid to the current playposition, sync the beatgrid so the nearest beat lines up with the other track's nearest beat.

Nové ve verzi 2.0.0.

[ChannelN]beats\_translate\_earlier  
[PreviewDeckN]beats\_translate\_earlier  
[SamplerN]beats\_translate\_earlier

Move *beatgrid* to an earlier position.

**Range** binary

**Feedback** The beatgrid moves left by a small amount.

Nové ve verzi 2.0.0.

[ChannelN]beats\_translate\_later  
[PreviewDeckN]beats\_translate\_later  
[SamplerN]beats\_translate\_later

Move *beatgrid* to a later position.

**Range** binary

**Feedback** The beatgrid moves right by a small amount.

Nové ve verzi 2.0.0.

[ChannelN]beat\_sync  
[PreviewDeckN]beat\_sync  
[SamplerN]beat\_sync

*Syncs* the *tempo* and phase (depending on quantize) to that of the other track (if *BPM* is detected on both).

**range** binary

**feedback** The *Sync* button flashes and the *tempo* slider snaps to the appropriate value.

Změněno ve verzi 1.10.0.

[ChannelN]beat\_sync\_phase  
[PreviewDeckN]beat\_sync\_phase  
[SamplerN]beat\_sync\_phase

*Syncs* the *phase* to that of the other track (if *BPM* is detected on both).

**Range** binary

**Feedback** The *Sync* button flashes and the *tempo* slider snap to the appropriate value.

Nové ve verzi 1.10.0.

[ChannelN]beat\_sync\_tempo  
[PreviewDeckN]beat\_sync\_tempo  
[SamplerN]beat\_sync\_tempo

*Syncs* the *tempo* to that of the other track (if *BPM* is detected on both).

**Range** binary

**Feedback** The *Sync* button flashes and the *tempo* slider snaps to the appropriate value.

Nové ve verzi 1.10.0.

[ChannelN]bpm  
[PreviewDeckN]bpm  
[SamplerN]bpm

Reflects the perceived (rate-adjusted) *BPM* of the loaded file.

This is a *ControlPotMeter control*.

**Range** real-valued

**Feedback** *BPM* value display

Změněno ve verzi 1.10.0.

**[ChannelN] bpm\_tap**

**[PreviewDeckN] bpm\_tap**

**[SamplerN] bpm\_tap**

When tapped repeatedly, adjusts the *BPM* of the track on the deck (not the tempo slider!) to match the taps.

---

**Poznámka:** If you want to change the *rate* of the deck use `script.bpm.tapButton(deck)` in your controller mapping instead.

---

**Range** binary

**Feedback** *BPM* value display (playback speed doesn't change)

Nové ve verzi 1.9.2.

**[ChannelN] CloneFromDeck**

**[PreviewDeckN] CloneFromDeck**

**[SamplerN] CloneFromDeck**

Clone the given deck number, copying the play state, position, rate, and key. If 0 or a negative number is given, Mixxx will attempt to select the first playing deck as the source for the clone.

**Range** integer between 1 and `[Master], num_decks` (inclusive)

**Feedback** The channel will start playing at the rate and position of the source deck.

Nové ve verzi 2.3.0.

**[ChannelN] CloneFromSampler**

**[PreviewDeckN] CloneFromSampler**

**[SamplerN] CloneFromSampler**

Clone the given sampler number, copying the play state, position, rate, and key.

**Range** integer between 1 and `[Master], num_samplers` (inclusive)

**Feedback** The channel will start playing at the rate and position of the source deck.

Nové ve verzi 2.3.0.

**[ChannelN] cue\_cdj**

**[PreviewDeckN] cue\_cdj**

**[SamplerN] cue\_cdj**

Represents a *Cue* button that is always in *CDJ* mode.

**Range** binary

**Feedback** None

Nové ve verzi 1.10.0.

**[ChannelN] cue\_clear**

**[PreviewDeckN] cue\_clear**

**[SamplerN] cue\_clear**

Deletes the already set cue point and sets `[ChannelN], cue_point` to -1.

**Range** binary

**Feedback** None

[ChannelN] cue\_goto

[PreviewDeckN] cue\_goto

[SamplerN] cue\_goto

If the cue point is set, recalls the cue point.

**Range** binary

**Feedback** Player may change position

[ChannelN] cue\_default

[PreviewDeckN] cue\_default

[SamplerN] cue\_default

In *CDJ* mode, when playing, returns to the *cue point* and pauses. If stopped, sets a cue point at the current location.

If stopped and at a cue point, plays from that point until released (set to 0.)

**Range** binary

**Feedback** Cue button

[ChannelN] cue\_gotoandplay

[PreviewDeckN] cue\_gotoandplay

[SamplerN] cue\_gotoandplay

If the *cue point* is set, seeks the player to it and starts playback.

**Range** binary

**Feedback** Player may change position and start playing.

Nové ve verzi 1.11.0.

[ChannelN] cue\_gotoandstop

[PreviewDeckN] cue\_gotoandstop

[SamplerN] cue\_gotoandstop

If the *cue point* is set, seeks the player to it and stops.

**Range** binary

**Feedback** Player may change position.

Nové ve verzi 1.11.0.

[ChannelN] cue\_indicator

[PreviewDeckN] cue\_indicator

[SamplerN] cue\_indicator

Indicates the blinking pattern of the *CUE* button (i.e. 1.0 if the button is illuminated, 0.0 otherwise), depending on the chosen *cue mode*.

**Range** binary

**Feedback** Cue button

Nové ve verzi 2.0.0.

[ChannelN] cue\_mode

[PreviewDeckN] cue\_mode

[SamplerN] cue\_mode

Represents the currently chosen *cue mode*.

**Range**



Value	compatible hardware
0.0	Mixxx mode (default)
1.0	Pioneer mode
2.0	Denon mode
3.0	Numark mode
4.0	Mixxx mode (no blinking)
5.0	CUP (Cue + Play) mode

**Feedback** None

**[ChannelN] cue\_play**

**[PreviewDeckN] cue\_play**

**[SamplerN] cue\_play**

Go to *cue point* and play after release (CUP button behavior). If stopped, sets a cue point at the current location.

**Range** binary

**Feedback** None

Nové ve verzi 2.1.0.

**[ChannelN] cue\_point**

**[PreviewDeckN] cue\_point**

**[SamplerN] cue\_point**

The current position of the *cue point* in samples.

**Range** absolute value

**Feedback** Cue point marker

**[ChannelN] cue\_preview**

**[PreviewDeckN] cue\_preview**

**[SamplerN] cue\_preview**

Plays from the current *cue point*.

**Range** binary

**Feedback** Cue button lights and waveform moves

**[ChannelN] cue\_set**

**[PreviewDeckN] cue\_set**

**[SamplerN] cue\_set**

Sets a *cue point* at the current location.

**Range** binary

**Feedback** Cue mark appears on the waveform

**[ChannelN] cue\_simple**

**[PreviewDeckN] cue\_simple**

**[SamplerN] cue\_simple**

If the player is not playing, set the *cue point* at the current location otherwise seek to the cue point.

**Range** binary

**Feedback** Cue button

**[ChannelN] duration**

**[PreviewDeckN] duration**

**[SamplerN] duration**

Outputs the length of the current song in seconds

**Range** absolute value

**Feedback** None

[ChannelN] eject

[PreviewDeckN] eject

[SamplerN] eject

Eject currently loaded track

**Range** binary

**Feedback** Eject button is lit. Be sure to set back to 0 with scripts so the button does not stay lit.

Nové ve verzi 1.9.0.

[ChannelN] end

[PreviewDeckN] end

[SamplerN] end

Jump to end of track

**Range** binary

**Feedback** Track jumps to end

[ChannelN] end\_of\_track

[PreviewDeckN] end\_of\_track

[SamplerN] end\_of\_track

Switches to 1 if the play position is within the end range defined in *Preferences* ▶ *Waveforms* ▶ *End of track warning*.

**Range** binary, read-only

**Feedback** Waveform and Overview widgets show a flashing border

[ChannelN] file\_bpm

[PreviewDeckN] file\_bpm

[SamplerN] file\_bpm

The detected *BPM* of the loaded track.

**Range** positive value, read-only

**Feedback** None

[ChannelN] file\_key

[PreviewDeckN] file\_key

[SamplerN] file\_key

The detected key of the loaded track.

**Range** ?, read-only

**Feedback** None

Nové ve verzi 2.0.0.

[ChannelN] fwd

[PreviewDeckN] fwd

[SamplerN] fwd

Fast forward (FF)

**Range** binary

**Feedback** > button

[ChannelN] hotcue\_X\_activate

[PreviewDeckN] hotcue\_X\_activate

**[SamplerN]hotcue\_X\_activate**

If *hotcue* X is not set, this sets a hotcue at the current play position and saves it to hotcue slot X.

If hotcue X is set, the player seeks to hotcue X's position. Setting the control to 1 when the track is currently not playing (i.e. *play* is set to 0) will start hotcue previewing. After resetting the control to 0, playback will usually be stopped and the player will seek to the hotcue position. If *play* is set to 1 while previewing is active, the playback will continue and no seek occurs.

**Range** binary

**Feedback** Player may change position. Hotcue X marker may change on waveform.

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_X\_clear**

**[PreviewDeckN]hotcue\_X\_clear**

**[SamplerN]hotcue\_X\_clear**

If *hotcue* X is set, clears its hotcue status.

**Range** binary

**Feedback** Hotcue X marker changes on waveform.

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_X\_color**

**[PreviewDeckN]hotcue\_X\_color**

**[SamplerN]hotcue\_X\_color**

Color of *hotcue* X or -1 if the hotcue is not set.

**Range** 3-Byte *RGB* color code (or -1)

**Feedback** Color of Hotcue X button and waveform marker changes.

Nové ve verzi 2.3.0.

**[ChannelN]hotcue\_X\_enabled**

**[PreviewDeckN]hotcue\_X\_enabled**

**[SamplerN]hotcue\_X\_enabled**

Indicates if *hotcue* slot X is set. The value is 1 if the hotcue is set (position is not -1), 0 otherwise.

**Range** binary, read-only

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_X\_goto**

**[PreviewDeckN]hotcue\_X\_goto**

**[SamplerN]hotcue\_X\_goto**

If *hotcue* X is set, seeks the player to hotcue X's position.

**Range** binary

**Feedback** Player may change position.

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_X\_gotoandplay**

**[PreviewDeckN]hotcue\_X\_gotoandplay**

**[SamplerN]hotcue\_X\_gotoandplay**

If *hotcue* X is set, seeks the player to hotcue X's position and starts playback.

**Range** binary

**Feedback** Player may change position.

Nové ve verzi 1.11.0.

**[ChannelN]hotcue\_X\_gotoandstop**  
**[PreviewDeckN]hotcue\_X\_gotoandstop**  
**[SamplerN]hotcue\_X\_gotoandstop**

If *hotcue* X is set, seeks the player to hotcue X's position and stops.

**Range** binary

**Feedback** Player may change position.

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_X\_position**  
**[PreviewDeckN]hotcue\_X\_position**  
**[SamplerN]hotcue\_X\_position**

The position of *hotcue* X in samples, -1 if not set.

**Range** positive integer

**Feedback** Hotcue X marker changes on waveform.

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_X\_set**  
**[PreviewDeckN]hotcue\_X\_set**  
**[SamplerN]hotcue\_X\_set**

Set *hotcue* X to the current play position. If hotcue X was previously set, clears its hotcue status.

**Range** binary

**Feedback** Hotcue X marker changes on waveform.

Nové ve verzi 1.8.0.

**[ChannelN]hotcue\_focus**  
**[PreviewDeckN]hotcue\_focus**  
**[SamplerN]hotcue\_focus**

Contains the number of the most recently used *hotcue* (or -1 if no hotcue was used).

**Range** positive integer (or -1)

**Feedback** None

Nové ve verzi 2.3.0.

**[ChannelN]hotcue\_focus\_color\_prev**  
**[PreviewDeckN]hotcue\_focus\_color\_prev**  
**[SamplerN]hotcue\_focus\_color\_prev**

If there is a focused *hotcue*, sets its color to the previous color in the palette.

**Range** binary

**Feedback** Color of focused hotcue button and waveform marker changes.

Nové ve verzi 2.3.0.

**[ChannelN]hotcue\_focus\_color\_next**  
**[PreviewDeckN]hotcue\_focus\_color\_next**  
**[SamplerN]hotcue\_focus\_color\_next**

If there is a focused *hotcue*, sets its color to the next color in the palette.

**Range** binary

**Feedback** Color of focused hotcue button and waveform marker changes.

Nové ve verzi 2.3.0.

**[ChannelN]intro\_end\_activate**  
**[PreviewDeckN]intro\_end\_activate**  
**[SamplerN]intro\_end\_activate**

If the intro end cue is set, seeks the player to the intro end position. If the intro end is not set, sets the intro end to the current play position.

**Range** binary

**Feedback** Player may change position. Intro end marker may change on waveform.

Nové ve verzi 2.3.0.

**[ChannelN]intro\_end\_clear**  
**[PreviewDeckN]intro\_end\_clear**  
**[SamplerN]intro\_end\_clear**

If the intro end cue is set, clears its status.

**Range** binary

**Feedback** Intro end marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN]intro\_end\_enabled**  
**[PreviewDeckN]intro\_end\_enabled**  
**[SamplerN]intro\_end\_enabled**

1 if intro end cue is set, (position is not -1), 0 otherwise.

**Range** binary, read-only

**Feedback** Intro end button lights up.

Nové ve verzi 2.3.0.

**[ChannelN]intro\_end\_position**  
**[PreviewDeckN]intro\_end\_position**  
**[SamplerN]intro\_end\_position**

The position of the intro end in samples, -1 if not set.

**Range** positive integer

**Feedback** Intro end marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN]intro\_end\_set**  
**[PreviewDeckN]intro\_end\_set**  
**[SamplerN]intro\_end\_set**

Set intro end to the current play position. If intro end was previously set, it is moved to the new position.

**Range** binary

**Feedback** Intro end marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN]intro\_start\_activate**  
**[PreviewDeckN]intro\_start\_activate**  
**[SamplerN]intro\_start\_activate**

If the intro start cue is set, seeks the player to the intro start position. If the intro start is not set, sets the intro start to the current play position.

**Range** binary

**Feedback** Player may change position. Intro start marker may change on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] intro\_start\_clear**  
**[PreviewDeckN] intro\_start\_clear**  
**[SamplerN] intro\_start\_clear**

If the intro start cue is set, clears its status.

**Range** binary

**Feedback** Intro start marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] intro\_start\_enabled**  
**[PreviewDeckN] intro\_start\_enabled**  
**[SamplerN] intro\_start\_enabled**

1 if intro start cue is set, (position is not -1), 0 otherwise.

**Range** binary, read-only

**Feedback** Intro start button lights up.

Nové ve verzi 2.3.0.

**[ChannelN] intro\_start\_position**  
**[PreviewDeckN] intro\_start\_position**  
**[SamplerN] intro\_start\_position**

The position of the intro start in samples, -1 if not set.

**Range** positive integer

**Feedback** Intro start marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] intro\_start\_set**  
**[PreviewDeckN] intro\_start\_set**  
**[SamplerN] intro\_start\_set**

Set intro start to the current play position. If intro start was previously set, it is moved to the new position.

**Range** binary

**Feedback** Intro start marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] key**  
**[PreviewDeckN] key**  
**[SamplerN] key**

Current key of the track

**Range**

Value	OpenKey	Lancelot	Traditional
1	1d	8b	C
2	8d	3b	D $\flat$
3	3d	10b	D
4	10d	5b	E $\flat$
5	5d	12b	E
6	12d	7b	F
7	7d	2b	F $\sharp$ /G $\flat$
8	2d	9b	G
9	9d	4b	A $\flat$
10	4d	11b	A
11	11d	6b	B $\flat$
12	6d	1b	B
13	10m	5a	Cm
14	5m	12a	C $\sharp$ m
15	12m	7a	Dm
16	7m	2a	D $\sharp$ m/E $\flat$ m
17	2m	9a	Em
18	9m	4a	Fm
19	4m	11a	F $\sharp$ m
20	11m	6a	Gm
21	6m	1a	G $\sharp$ m
22	1m	8a	Am
23	8m	3a	B $\flat$ m
24	3m	10a	Bm

Nové ve verzi 2.0.0.

**[ChannelN] keylock**

**[PreviewDeckN] keylock**

**[SamplerN] keylock**

Enable key-lock for the specified deck (rate changes only affect tempo, not key)

**Range** binary

**Feedback** key-lock button activates

Nové ve verzi 1.9.0.

**[ChannelN] LoadSelectedTrack**

**[PreviewDeckN] LoadSelectedTrack**

**[SamplerN] LoadSelectedTrack**

Loads the currently highlighted track into the deck

**Range** binary

**Feedback** Track name & waveform change

**[ChannelN] LoadSelectedTrackAndPlay**

**[PreviewDeckN] LoadSelectedTrackAndPlay**

**[SamplerN] LoadSelectedTrackAndPlay**

Loads the currently highlighted track into the deck and starts playing

**Range** binary

**Feedback** Track name & waveform change & Play/pause button

Nové ve verzi 1.11.0.

[ChannelN] local\_bpm

[PreviewDeckN] local\_bpm

[SamplerN] local\_bpm

Reflects the average bpm around the current play position of the loaded file.

**Range** positive value

**Feedback** None

[ChannelN] loop\_double

[PreviewDeckN] loop\_double

[SamplerN] loop\_double

Doubles *beatloop\_size*. If *beatloop\_size* equals the size of the loop, the loop is resized.

**Range** binary

**Feedback** Beatloop size spinbox changes

Nové ve verzi 1.10.0.

Změněno ve verzi 2.1.0.

[ChannelN] loop\_enabled

[PreviewDeckN] loop\_enabled

[SamplerN] loop\_enabled

Indicates whether or not a loop is enabled.

**Range** binary, read-only

**Feedback** Loop in waveform is active.

Nové ve verzi 1.8.0.

[ChannelN] loop\_end\_position

[PreviewDeckN] loop\_end\_position

[SamplerN] loop\_end\_position

The player loop-out position in samples, -1 if not set.

**Range** positive integer

**Feedback** Loop-out marker shows on waveform.

Nové ve verzi 1.8.0.

[ChannelN] loop\_half

[PreviewDeckN] loop\_half

[SamplerN] loop\_half

Halves *beatloop\_size*. If *beatloop\_size* equals the size of the loop, the loop is resized.

**Range** binary

**Feedback** Beatloop size spinbox changes

Nové ve verzi 1.10.0.

Změněno ve verzi 2.1.0.

[ChannelN] loop\_in

[PreviewDeckN] loop\_in

[SamplerN] loop\_in

If loop is disabled, sets the player loop in position to the current play position. If loop is enabled, press and hold to move loop in position to the current play position. If quantize is enabled, *beatloop\_size* will be updated to reflect the new loop size.

**Range** binary



**Feedback** Loop-in marker changes on waveform.

Nové ve verzi 1.8.0.

Změněno ve verzi 2.1.0.

**[ChannelN] loop\_in\_goto**  
**[PreviewDeckN] loop\_in\_goto**  
**[SamplerN] loop\_in\_goto**

Seek to the loop in point.

**Range** binary

**Feedback** Waveform position jumps

Nové ve verzi 2.1.0.

**[ChannelN] loop\_out**  
**[PreviewDeckN] loop\_out**  
**[SamplerN] loop\_out**

If loop is disabled, sets the player loop out position to the current play position. If loop is enabled, press and hold to move loop out position to the current play position. If quantize is enabled, *beatloop\_size* will be updated to reflect the new loop size.

**Range** binary

**Feedback** Loop-out marker changes on waveform.

Nové ve verzi 1.8.0.

Změněno ve verzi 2.1.0.

**[ChannelN] loop\_out\_goto**  
**[PreviewDeckN] loop\_out\_goto**  
**[SamplerN] loop\_out\_goto**

Seek to the loop out point.

**Range** binary

**Feedback** Waveform position jumps

Nové ve verzi 2.1.0.

**[ChannelN] loop\_move**  
**[PreviewDeckN] loop\_move**  
**[SamplerN] loop\_move**

Move loop forward by X beats (positive) or backward by X beats (negative).

**Range** real number

**Feedback** Loop moves forward or backward by X beats.

Nové ve verzi 2.0.0.

**[ChannelN] loop\_move\_X\_forward**  
**[PreviewDeckN] loop\_move\_X\_forward**  
**[SamplerN] loop\_move\_X\_forward**

Moves the loop in and out points forward by X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64

**Range** binary

**Feedback** Loop moves forward by X beats.

Nové ve verzi 2.0.0.

[ChannelN] loop\_move\_X\_backward  
 [PreviewDeckN] loop\_move\_X\_backward  
 [SamplerN] loop\_move\_X\_backward

Loop moves by X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64

**Range** binary

**Feedback** Loop moves backward by X beats.

Nové ve verzi 2.0.0.

[ChannelN] loop\_scale  
 [PreviewDeckN] loop\_scale  
 [SamplerN] loop\_scale

Scale the loop length by the value scale is set to by moving the end marker. *beatloop\_size* is not updated to reflect the change.

**Range** 0.0 - infinity

**Feedback** Loop length is scaled by given amount on waveform.

Nové ve verzi 1.10.0.

[ChannelN] loop\_start\_position  
 [PreviewDeckN] loop\_start\_position  
 [SamplerN] loop\_start\_position

The player loop-in position in samples, -1 if not set.

**Range** positive integer

**Feedback** Loop-in marker changes on waveform.

Nové ve verzi 1.8.0.

[ChannelN] orientation  
 [PreviewDeckN] orientation  
 [SamplerN] orientation

Set channel orientation for the *crossfader*.

**Range**

Value	Meaning
0	Left side of crossfader
1	Center (not affected by crossfader)
2	Right side of crossfader

**Feedback** None

Nové ve verzi 1.9.0.

[ChannelN] orientation\_center  
 [PreviewDeckN] orientation\_center  
 [SamplerN] orientation\_center  
 [AuxiliaryN] orientation\_center

Assign channel to the center of the *crossfader*.

[ChannelN] orientation\_left  
 [PreviewDeckN] orientation\_left  
 [SamplerN] orientation\_left  
 [AuxiliaryN] orientation\_left

Assign channel to the left side of the *crossfader*.

[ChannelN]orientation\_right  
 [PreviewDeckN]orientation\_right  
 [SamplerN]orientation\_right  
 [AuxiliaryN]orientation\_right

Assign channel to the right side of the *crossfader*.

[ChannelN]outro\_end\_activate  
 [PreviewDeckN]outro\_end\_activate  
 [SamplerN]outro\_end\_activate

If the outro end cue is set, seeks the player to the outro end position. If the outro end is not set, sets the outro end to the current play position.

**Range** binary

**Feedback** Player may change position. Outro end marker may change on waveform.

Nové ve verzi 2.3.0.

[ChannelN]outro\_end\_clear  
 [PreviewDeckN]outro\_end\_clear  
 [SamplerN]outro\_end\_clear

If the outro end cue is set, clears its status.

**Range** binary

**Feedback** Outro end marker changes on waveform.

Nové ve verzi 2.3.0.

[ChannelN]outro\_end\_enabled  
 [PreviewDeckN]outro\_end\_enabled  
 [SamplerN]outro\_end\_enabled

1 if outro end cue is set, (position is not -1), 0 otherwise.

**Range** binary, read-only

**Feedback** Outro end button lights up.

Nové ve verzi 2.3.0.

[ChannelN]outro\_end\_position  
 [PreviewDeckN]outro\_end\_position  
 [SamplerN]outro\_end\_position

The position of the outro end in samples, -1 if not set.

**Range** positive integer

**Feedback** Outro end marker changes on waveform.

Nové ve verzi 2.3.0.

[ChannelN]outro\_end\_set  
 [PreviewDeckN]outro\_end\_set  
 [SamplerN]outro\_end\_set

Set outro end to the current play position. If outro end was previously set, it is moved to the new position.

**Range** binary

**Feedback** Outro end marker changes on waveform.

Nové ve verzi 2.3.0.

[ChannelN]outro\_start\_activate  
 [PreviewDeckN]outro\_start\_activate

**[SamplerN] outro\_start\_activate**

If the outro start cue is set, seeks the player to the outro start position. If the outro start is not set, sets the outro start to the current play position.

**Range** binary

**Feedback** Player may change position. Outro start marker may change on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] outro\_start\_clear**

**[PreviewDeckN] outro\_start\_clear**

**[SamplerN] outro\_start\_clear**

If the outro start cue is set, clears its status.

**Range** binary

**Feedback** Outro start marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] outro\_start\_enabled**

**[PreviewDeckN] outro\_start\_enabled**

**[SamplerN] outro\_start\_enabled**

1 if outro start cue is set, (position is not -1), 0 otherwise.

**Range** binary, read-only

**Feedback** Outro start button lights up.

Nové ve verzi 2.3.0.

**[ChannelN] outro\_start\_position**

**[PreviewDeckN] outro\_start\_position**

**[SamplerN] outro\_start\_position**

The position of the outro start in samples, -1 if not set.

**Range** positive integer

**Feedback** Outro start marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] outro\_start\_set**

**[PreviewDeckN] outro\_start\_set**

**[SamplerN] outro\_start\_set**

Set outro start to the current play position. If outro start was previously set, it is moved to the new position.

**Range** binary

**Feedback** Outro start marker changes on waveform.

Nové ve verzi 2.3.0.

**[ChannelN] passthrough**

**[PreviewDeckN] passthrough**

**[SamplerN] passthrough**

Connects the vinyl control input for vinyl control on that deck to the channel output. Allows to mix external media into DJ sets.

**Range** binary

**Feedback** Passthrough label in the *waveform overview* and passthrough button

Nové ve verzi 2.0.0.

**[ChannelN]PeakIndicator**

**[PreviewDeckN]PeakIndicator**

**[SamplerN]PeakIndicator**

Indicates when the signal is clipping (too loud for the hardware and is being distorted)

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light

**[ChannelN]PeakIndicatorL**

**[PreviewDeckN]PeakIndicatorL**

**[SamplerN]PeakIndicatorL**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) for the left channel

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (left)

Nové ve verzi 2.0.0.

**[ChannelN]PeakIndicatorR**

**[PreviewDeckN]PeakIndicatorR**

**[SamplerN]PeakIndicatorR**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) for the right channel

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (right)

Nové ve verzi 2.0.0.

**[ChannelN]pfl**

**[PreviewDeckN]pfl**

**[SamplerN]pfl**

Toggles *headphone cueing (PFL)*.

**Range** binary

**Feedback** Headphone button

**[ChannelN]pitch**

**[PreviewDeckN]pitch**

**[SamplerN]pitch**

The total adjustment to the track's pitch, including changes from the rate slider if keylock is off as well as *pitch\_adjust*.

---

**Poznámka:** Do not map this to knobs or sliders on controllers; map *pitch\_adjust* instead.

---

This is a *ControlPotMeter control*.

**Range** -6.0..6.0

**Feedback** Key display

Nové ve verzi 2.0.0.

**[ChannelN]pitch\_up**

**[PreviewDeckN]pitch\_up**

**[SamplerN]pitch\_up**

Changes the track pitch up one half step, independent of the tempo.

**Range** binary

**Feedback** Key display

Nové ve verzi 2.0.0.

**[ChannelN]pitch\_down**

**[PreviewDeckN]pitch\_down**

**[SamplerN]pitch\_down**

Changes the track pitch down one half step, independent of the tempo.

**Range** binary

**Feedback** Key display

Nové ve verzi 2.0.0.

**[ChannelN]pitch\_adjust**

**[PreviewDeckN]pitch\_adjust**

**[SamplerN]pitch\_adjust**

Adjust the pitch in addition to the *tempo* slider pitch.

This is a *ControlPotMeter control*.

**Range** -3.0..3.0

**Feedback** Key display

Nové ve verzi 2.0.0.

**[ChannelN]play**

**[PreviewDeckN]play**

**[SamplerN]play**

Toggles playing or pausing the track.

The value is set to 1 when the track is playing or when previewing from cue points and when the play command is adopted and track will be played after loading.

**Range** binary

**Feedback** Play/pause button

**[ChannelN]play\_indicator**

**[PreviewDeckN]play\_indicator**

**[SamplerN]play\_indicator**

Provides information to be bound with the a Play/Pause button e.g blinking when play is possible

**Range** binary, read-only

**Feedback** Play/pause button

Nové ve verzi 2.0.0.

**[ChannelN]play\_latched**

**[PreviewDeckN]play\_latched**

**[SamplerN]play\_latched**

This is set to 1 when the track is playing, but not when previewing (see *play*).

**Range** binary, read-only

**Feedback** Play/pause button

Nové ve verzi 2.3.0.

**[ChannelN]play\_stutter**  
**[PreviewDeckN]play\_stutter**  
**[SamplerN]play\_stutter**

A play button without pause. Pushing while playing, starts play at *cue point* again (Stutter).

**Range** binary

**Feedback** Play/Stutter button

Nové ve verzi 2.0.0.

**[ChannelN]playposition**  
**[PreviewDeckN]playposition**  
**[SamplerN]playposition**

Sets the absolute position in the track.

This is a *ControlPotMeter control*.

**Range** -0.14 to 1.14 (0 = beginning -> Midi 14, 1 = end -> Midi 114)

**Feedback** Waveform

**[ChannelN]pregain**  
**[PreviewDeckN]pregain**  
**[SamplerN]pregain**

Adjusts the pre-fader gain of the track (to avoid clipping)

This is a *ControlPotMeter control*.

**Range** 0.0..1.0..4.0

**Feedback** GAIN knob

**[ChannelN]quantize**  
**[PreviewDeckN]quantize**  
**[SamplerN]quantize**

Aligns Hot-cues and Loop In & Out to the next beat from the current position.

**Range** binary

**Feedback** Hot-cues or Loop In/Out markers

Nové ve verzi 1.10.0.

**[ChannelN]rate**  
**[PreviewDeckN]rate**  
**[SamplerN]rate**

Speed control

This is a *ControlPotMeter control*.

**Range** -1.0..1.0

**Feedback** Speed slider

**[ChannelN]rate\_dir**  
**[PreviewDeckN]rate\_dir**  
**[SamplerN]rate\_dir**

Indicates orientation of speed slider.

**Range** -1 or 1

**[ChannelN]rate\_perm\_down**

**[PreviewDeckN] rate\_perm\_down**  
**[SamplerN] rate\_perm\_down**  
 Sets the speed one step lower (4 % default) lower  
**Range** binary  
**Feedback** Perm down button & Speed slider

**[ChannelN] rate\_perm\_down\_small**  
**[PreviewDeckN] rate\_perm\_down\_small**  
**[SamplerN] rate\_perm\_down\_small**  
 Sets the speed one small step lower (1 % default)  
**Range** binary  
**Feedback** Perm down button & Speed slider

**[ChannelN] rate\_perm\_up**  
**[PreviewDeckN] rate\_perm\_up**  
**[SamplerN] rate\_perm\_up**  
 Sets the speed one step higher (4 % default)  
**Range** binary  
**Feedback** Perm up button & Speed slider

**[ChannelN] rate\_perm\_up\_small**  
**[PreviewDeckN] rate\_perm\_up\_small**  
**[SamplerN] rate\_perm\_up\_small**  
 Sets the speed one small step higher (1 % default)  
**Range** binary  
**Feedback** Perm up button & Speed slider

**[ChannelN] rate\_temp\_down**  
**[PreviewDeckN] rate\_temp\_down**  
**[SamplerN] rate\_temp\_down**  
 Holds the speed one step lower while active  
**Range** binary  
**Feedback** Temp down button & Speed slider

**[ChannelN] rate\_temp\_down\_small**  
**[PreviewDeckN] rate\_temp\_down\_small**  
**[SamplerN] rate\_temp\_down\_small**  
 Holds the speed one small step lower while active  
**Range** binary  
**Feedback** Temp down button & Speed slider

**[ChannelN] rate\_temp\_up**  
**[PreviewDeckN] rate\_temp\_up**  
**[SamplerN] rate\_temp\_up**  
 Holds the speed one step higher while active  
**Range** binary  
**Feedback** Temp up button & Speed slider

**[ChannelN] rate\_temp\_up\_small**  
**[PreviewDeckN] rate\_temp\_up\_small**



**[SamplerN] rate\_temp\_up\_small**

Holds the speed one small step higher while active

**Range** binary

**Feedback** Temp up button & Speed slider

**[ChannelN] rateRange**

**[PreviewDeckN] rateRange**

**[SamplerN] rateRange**

Sets the range of the Speed slider (0.08 = 8%)

This is a *ControlPotMeter control*.

**Range** 0.0..4.0

**Feedback** none, until you move the Speed slider

**[ChannelN] rateSearch**

**[PreviewDeckN] rateSearch**

**[SamplerN] rateSearch**

Seeks forward (positive values) or backward (negative values) at a speed determined by the value

This is a *ControlPotMeter control*.

**Range** -300..300

**Feedback** Deck seeks

**[ChannelN] rateEngine**

**[PreviewDeckN] rateEngine**

**[SamplerN] rateEngine**

Actual rate (used in visuals, not for control)

**[ChannelN] reloop\_andstop**

**[PreviewDeckN] reloop\_andstop**

**[SamplerN] reloop\_andstop**

Activate current loop, jump to its loop in point, and stop playback.

**Range** binary

**Feedback** Loop range in waveform activates or deactivates and play position moves to loop in point.

Nové ve verzi 2.1.0.

**[ChannelN] reloop\_toggle**

**[PreviewDeckN] reloop\_toggle**

**[SamplerN] reloop\_toggle**

Toggles the current loop on or off. If the loop is ahead of the current play position, the track will keep playing normally until it reaches the loop.

**Range** binary

**Feedback** Loop range in waveform activates or deactivates.

Nové ve verzi 2.1.0.

**[ChannelN] repeat**

**[PreviewDeckN] repeat**

**[SamplerN] repeat**

Enable repeat-mode for the specified deck

**Range** binary

**Feedback** when track finishes, song loops to beginning

Nové ve verzi 1.9.0.

**[ChannelN] reset\_key**  
**[PreviewDeckN] reset\_key**  
**[SamplerN] reset\_key**

Resets the key to the original track key.

**Range** binary

Nové ve verzi 2.0.0.

**[ChannelN] reverse**  
**[PreviewDeckN] reverse**  
**[SamplerN] reverse**

Toggles playing the track backwards

**Range** binary

**Feedback** REV button

**[ChannelN] reverseroll**  
**[PreviewDeckN] reverseroll**  
**[SamplerN] reverseroll**

Enables reverse and slip mode while held (Censor)

**Range** binary

**Feedback** REV button

Nové ve verzi 2.0.0.

**[ChannelN] scratch2**  
**[PreviewDeckN] scratch2**  
**[SamplerN] scratch2**

Affects absolute play speed & direction whether currently playing or not when *[ChannelN], scratch2\_enabled* is active. (multiplicative). Use JavaScript engine.*scratch* functions to manipulate in controller mappings.

**Range** -3.0..3.0

**Feedback** Waveform

Nové ve verzi 1.8.0.

**[ChannelN] scratch2\_enabled**  
**[PreviewDeckN] scratch2\_enabled**  
**[SamplerN] scratch2\_enabled**

Takes over play speed & direction for *[ChannelN], scratch2*.

**Range** binary

**Feedback** Waveform

Nové ve verzi 1.8.0.

**[ChannelN] slip\_enabled**  
**[PreviewDeckN] slip\_enabled**  
**[SamplerN] slip\_enabled**

Toggles slip mode. When active, the playback continues muted in the background during a loop, scratch etc. Once disabled, the audible playback will resume where the track would have been.

**Range** binary

**Feedback** Slip mode button

Nové ve verzi 1.11.0.

**[ChannelN] stars\_up**  
**[PreviewDeckN] stars\_up**  
**[SamplerN] stars\_up**

Increase the rating of the currently loaded track (if the skin has star widgets in the decks section).

**Range** binary

**Feedback** Star count is increased in the deck's star widget and in the library table.

Nové ve verzi 2.3.0.

**[ChannelN] stars\_down**  
**[PreviewDeckN] stars\_down**  
**[SamplerN] stars\_down**

Decrease the rating of the currently loaded track (if the skin has star widgets in the decks section).

**Range** binary

**Feedback** Star count is decreased in the deck's star widget and in the library table.

Nové ve verzi 2.3.0.

**[ChannelN] start**  
**[PreviewDeckN] start**  
**[SamplerN] start**

Jump to start of track

**Range** binary

**Feedback** Track jumps to start

**[ChannelN] start\_play**  
**[PreviewDeckN] start\_play**  
**[SamplerN] start\_play**

Start playback from the beginning of the deck.

**Range** binary

**Feedback** Deck plays from beginning

Nové ve verzi 1.10.0.

**[ChannelN] start\_stop**  
**[PreviewDeckN] start\_stop**  
**[SamplerN] start\_stop**

Seeks a player to the start and then stops it.

**Range** binary

**Feedback** Deck stops at the beginning.

Nové ve verzi 1.10.0.

**[ChannelN] stop**  
**[PreviewDeckN] stop**  
**[SamplerN] stop**

Stops a player.

**Range** binary

**Feedback** Pause Button. Deck pauses at the current position.

Nové ve verzi 1.10.0.

**[ChannelN] sync\_enabled**  
**[PreviewDeckN] sync\_enabled**  
**[SamplerN] sync\_enabled**

Syncs the *tempo* and *phase* (depending on quantize) to that of the other track (if *BPM* is detected on both). Click and hold for at least one second activates sync lock on that deck.

**Range** binary

**Feedback** If enabled, the *Sync* button stays lit and *tempo* slider snap to the appropriate value. Slider adjustments are linked on all decks that have *sync lock* enabled.

Nové ve verzi 2.0.0.

**[ChannelN] sync\_master**  
**[PreviewDeckN] sync\_master**  
**[SamplerN] sync\_master**

Sets deck as leader clock.

**Range** binary

**Feedback** If enabled, the *Sync* button stays lit and *tempo* slider snap to the appropriate value. Slider adjustments are linked on all decks that have *sync lock* enabled.

Nové ve verzi 2.0.0.

Změněno ve verzi 2.3.0: This button just enables *sync lock* mode (similar to *[ChannelN], sync\_enabled*), it does not actually guarantee the deck will be the sync leader. This will be fixed in a future version.

**[ChannelN] sync\_mode**  
**[PreviewDeckN] sync\_mode**  
**[SamplerN] sync\_mode**

Nové ve verzi 2.0.0.

**Range**

Value	Meaning
0	<i>Sync lock</i> disabled for that deck
1	Deck is sync follower
2	Deck is sync leader

**[ChannelN] sync\_key**  
**[PreviewDeckN] sync\_key**  
**[SamplerN] sync\_key**

**Feedback** Key value widget

Match musical key.

Nové ve verzi 2.0.0.

**[ChannelN] track\_color**  
**[PreviewDeckN] track\_color**  
**[SamplerN] track\_color**

Color of the currently loaded track or -1 if no track is loaded or the track has no color.

**Range** 3-Byte *RGB* color code (or -1)

**Feedback** Track color changes in the library view.

Nové ve verzi 2.3.0.

**[ChannelN] track\_loaded**

**[PreviewDeckN]track\_loaded**

**[SamplerN]track\_loaded**

Whether a track is loaded in the specified deck

**Range** binary, read-only

**Feedback** Waveform and track metadata shown in deck

Nové ve verzi 2.1.0.

**[ChannelN]track\_samplerate**

**[PreviewDeckN]track\_samplerate**

**[SamplerN]track\_samplerate**

Sample rate of the track loaded on the specified deck

**Range** absolute value, read-only

**Feedback** None

Nové ve verzi 1.9.0.

**[ChannelN]track\_samples**

**[PreviewDeckN]track\_samples**

**[SamplerN]track\_samples**

Number of sound samples in the track loaded on the specified deck

**Range** absolute value, read-only

**Feedback** None

**[ChannelN]volume**

**[PreviewDeckN]volume**

**[SamplerN]volume**

Adjusts the channel volume fader

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Deck volume fader

**[ChannelN]mute**

**[PreviewDeckN]mute**

**[SamplerN]mute**

Mutes the channel

**Range** binary

**Feedback** Mute button

Nové ve verzi 2.0.0.

**[ChannelN]vinylcontrol\_enabled**

**[PreviewDeckN]vinylcontrol\_enabled**

**[SamplerN]vinylcontrol\_enabled**

Toggles whether a deck is being controlled by digital vinyl.

**Range** binary

**Feedback** When enabled, a vinyl indication should appear onscreen indicating green for enabled.

Nové ve verzi 1.10.0.

**[ChannelN]vinylcontrol\_cueing**

**[PreviewDeckN]vinylcontrol\_cueing**

**[SamplerN]vinylcontrol\_cueing**

Determines how *cue points* are treated in vinyl control relative mode.

**Range**

Value	Meaning
0	Cue points ignored
1	One Cue - If needle is dropped after the <i>cue point</i> , track will seek to that cue point
2	Hot Cue - Track will seek to nearest previous <i>hotcue</i>

Nové ve verzi 1.10.0.

**[ChannelN]vinylcontrol\_mode**

**[PreviewDeckN]vinylcontrol\_mode**

**[SamplerN]vinylcontrol\_mode**

Determines how vinyl control interprets needle information.

**Range**

Value	Meaning
0	Absolute Mode (track position equals needle position and speed)
1	Relative Mode (track <i>tempo</i> equals needle speed regardless of needle position)
2	Constant Mode (track <i>tempo</i> equals last known-steady tempo regardless of needle input)

See *Control Mode* for details.

**Feedback** 3-way button indicates status

Nové ve verzi 1.10.0.

**[ChannelN]vinylcontrol\_status**

**[PreviewDeckN]vinylcontrol\_status**

**[SamplerN]vinylcontrol\_status**

Provides visual feedback with regards to vinyl control status.

**Range** 0.0-3.0, read-only

**Feedback** Off for control disabled, green for control enabled, blinking yellow for when the needle reaches the end of the record, and red for needle skip detected

Nové ve verzi 1.10.0.

**[ChannelN]visual\_bpm**

**[PreviewDeckN]visual\_bpm**

**[SamplerN]visual\_bpm**

*BPM* to display in the *GUI* (updated more slowly than the actual *BPM*).

**Range** ?

**Feedback** *BPM* value widget

Nové ve verzi 2.0.0.

**[ChannelN]visual\_key**

**[PreviewDeckN]visual\_key**

**[SamplerN]visual\_key**

Current musical key after pitch shifting to display in the *GUI* using the notation selected in the preferences

**Range** ?

**Feedback** Key value widget

Nové ve verzi 2.0.0.

**[ChannelN]visual\_key\_distance**  
**[PreviewDeckN]visual\_key\_distance**  
**[SamplerN]visual\_key\_distance**

The distance to the nearest key measured in cents

This is a *ControlPotMeter control*.

**Range** -0.5..0.5

**Feedback** Key value widget

Nové ve verzi 2.0.0.

**[ChannelN]VuMeter**  
**[PreviewDeckN]VuMeter**  
**[SamplerN]VuMeter**

Outputs the current instantaneous deck volume

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Deck VU meter

**[ChannelN]VuMeterL**  
**[PreviewDeckN]VuMeterL**  
**[SamplerN]VuMeterL**

Outputs the current instantaneous deck volume for the left channel

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Deck VU meter L

**[ChannelN]VuMeterR**  
**[PreviewDeckN]VuMeterR**  
**[SamplerN]VuMeterR**

Outputs the current instantaneous deck volume for the right channel

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Deck VU meter R

**[ChannelN]waveform\_zoom**  
**[PreviewDeckN]waveform\_zoom**  
**[SamplerN]waveform\_zoom**

Zooms the waveform to look ahead or back as needed.

**Range** 1.0 - 10.0

**Feedback** Waveform zoom buttons

Nové ve verzi 1.11.0.

**[ChannelN]waveform\_zoom\_up**  
**[PreviewDeckN]waveform\_zoom\_up**  
**[SamplerN]waveform\_zoom\_up**

Waveform Zoom Out

**Range ?**

**Feedback** Waveform zoom buttons

Nové ve verzi 1.11.0.

**[ChannelN] waveform\_zoom\_down**  
**[PreviewDeckN] waveform\_zoom\_down**  
**[SamplerN] waveform\_zoom\_down**

Waveform Zoom In

**Range ?**

**Feedback** Waveform zoom buttons

Nové ve verzi 1.11.0.

**[ChannelN] waveform\_zoom\_set\_default**  
**[PreviewDeckN] waveform\_zoom\_set\_default**  
**[SamplerN] waveform\_zoom\_set\_default**

Return to default waveform zoom level

**Range ?**

**Feedback** Waveform zoom buttons

Nové ve verzi 1.11.0.

**[ChannelN] wheel**  
**[PreviewDeckN] wheel**  
**[SamplerN] wheel**

Affects relative playback speed and direction persistently (additive offset & must manually be undone).

**Range** -3.0..3.0

**Feedback** Waveform

## Deprecated controls

These controls have been deprecated, new controller mappings should use the alternatives.

**[ChannelN] beatloop**  
**[PreviewDeckN] beatloop**  
**[SamplerN] beatloop**

Setup a loop over the set number of beats.

**Range** positive real number

**Feedback** A loop is shown over the set number of beats.

Zastaralé od verze 2.1.0: Use *[ChannelN], beatloop\_size* and *[ChannelN], beatloop\_toggle* instead.

**[ChannelN] reloop\_exit**  
**[PreviewDeckN] reloop\_exit**  
**[SamplerN] reloop\_exit**

Toggles the current loop on or off. If the loop is ahead of the current play position, the track will keep playing normally until it reaches the loop.

**Range** binary

**Feedback** Loop range in waveform activates or deactivates.

Zastaralé od verze 2.1.0: Use *[ChannelN], reloop\_toggle* instead.



**[ChannelN] jog**

**[PreviewDeckN] jog**

**[SamplerN] jog**

Affects relative playback speed and direction for short instances (additive & is automatically reset to 0).

**Range** -3.0..3.0

**Feedback** waveform

Zastaralé od verze ??: Use the JavaScript engine `.scratch` functions instead.

**[ChannelN] scratch**

**[PreviewDeckN] scratch**

**[SamplerN] scratch**

Affects playback speed and direction ([differently whether currently playing or not](<https://bugs.launchpad.net/mixxx/+bug/530281>)) (multiplicative).

**Range** -3.0..3.0

**Feedback** Waveform

Zastaralé od verze ??: Use the JavaScript engine `.scratch` functions instead.

**[ChannelN] filter**

**[PreviewDeckN] filter**

**[SamplerN] filter**

Toggles the filter effect.

**Range** binary

**Feedback** Filter button

Nové ve verzi 2.0.0.

Zastaralé od verze 2.0.0: Use `[QuickEffectRack1_[ChannelN]_Effect1], enabled` instead.

**[ChannelN] filterDepth**

**[PreviewDeckN] filterDepth**

**[SamplerN] filterDepth**

Adjusts the intensity of the filter effect.

**Range** default

**Feedback** Filter depth knob

Nové ve verzi 2.0.0.

Zastaralé od verze 2.0.0: Use `[QuickEffectRack1_[ChannelN]], super1` instead.

**[ChannelN] filterLow**

**[PreviewDeckN] filterLow**

**[SamplerN] filterLow**

Adjusts the gain of the low *EQ* filter.

**Range** 0.0..1.0..4.0

**Feedback** Low EQ knob

Zastaralé od verze 2.0.0: Use `[EqualizerRack1_[ChannelN]_Effect1], parameter1` instead.

**[ChannelN] filterLowKill**

**[PreviewDeckN] filterLowKill**

**[SamplerN] filterLowKill**

Holds the gain of the low *EQ* to -inf while active

**Range** binary

**Feedback** Low EQ *kill switch*

Zastaralé od verze 2.0.0: Use `[EqualizerRack1_[ChannelI]_Effect1], button_parameter1` instead.

**[ChannelN] filterMid**

**[PreviewDeckN] filterMid**

**[SamplerN] filterMid**

Adjusts the gain of the mid *EQ* filter..

**Range** 0.0..1.0..4.0

**Feedback** Mid EQ knob

Zastaralé od verze 2.0.0: Use `[EqualizerRack1_[ChannelI]_Effect1], parameter2` instead.

**[ChannelN] filterMidKill**

**[PreviewDeckN] filterMidKill**

**[SamplerN] filterMidKill**

Holds the gain of the mid *EQ* to -inf while active.

**Range** binary

**Feedback** Mid EQ *kill switch*

Zastaralé od verze 2.0.0: Use `[EqualizerRack1_[ChannelI]_Effect1], button_parameter2` instead.

**[ChannelN] filterHigh**

**[PreviewDeckN] filterHigh**

**[SamplerN] filterHigh**

Adjusts the gain of the high *EQ* filter.

**Range** 0.0..1.0..4.0

**Feedback** High EQ knob

Zastaralé od verze 2.0.0: Use `[EqualizerRack1_[ChannelI]_Effect1], parameter3` instead.

**[ChannelN] filterHighKill**

**[PreviewDeckN] filterHighKill**

**[SamplerN] filterHighKill**

Holds the gain of the high *EQ* to -inf while active.

**Range** binary

**Feedback** High EQ *kill switch*

Zastaralé od verze 2.0.0: Use `[EqualizerRack1_[ChannelI]_Effect1], button_parameter3` instead.

**[ChannelN] beatloop\_X**

**[PreviewDeckN] beatloop\_X**

**[SamplerN] beatloop\_X**

Setup a loop over X beats. A control exists for X = 0.03125, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64

**Range** toggle

**Feedback** A loop is shown over X beats.

Nové ve verzi 1.10.0.

Zastaralé od verze 2.0.0: Use `[ChannelN], beatloop_X_activate` instead.

## Global Sampler controls

These controls can be used to control all samplers.

### [Samplers] show\_samplers

**Range** binary

**Feedback** Shows Sampler bank(s)

### [Sampler] SaveSamplerBank

Save sampler configuration. Make currently loaded tracks in samplers instantly available at a later point.

**Range** binary

**Feedback** Opens file dialog. Configuration file can be named and saved.

### [Sampler] LoadSamplerBank

Load saved sampler configuration file and add tracks to the available samplers.

**Range** binary

**Feedback** Opens file dialog. Select configuration file.

## 16.5.4 Microphones and Auxiliary Channels

In contrast to *decks*, *preview decks* and *samplers*, microphones and auxiliary channels are input channels. You can map audio interface's inputs to mixxx's auxiliary input channels and connect external audio source to it (cellphone, mp3 player). Then you can use your *MIDI* controller to control its volume and some other parameters (orientation, gain, *volume*), apply effects and use the prelisten function.

---

**Poznámka:** Although the first auxiliary group is named [Auxiliary1], the group for the first microphone is just called [Microphone], not [Microphone1].

---

### [MicrophoneN] enabled

### [AuxiliaryN] enabled

1 if a channel input is enabled, 0 if not.

**Range** binary

**Feedback** Microphone is enabled.

Nové ve verzi 1.10.0.

Zastaralé od verze 2.0.0: Use [MicrophoneN], *input\_configured* instead.

### [MicrophoneN] input\_configured

### [AuxiliaryN] input\_configured

1 if there is input is configured for this channel, 0 if not.

**Range** binary, read-only

**Feedback** Configured channel in the sound preferences.

### [MicrophoneN] master

### [AuxiliaryN] master

Hold value at 1 to mix channel input into the master output. For [MicrophoneN] use [MicrophoneN], *talkover* instead. Note that [AuxiliaryN] also take [AuxiliaryN], *orientation* into account.

**Range** binary

**Feedback** Auxiliary: Play button Microphone: N/A

**[AuxiliaryN]orientation**

Set channel orientation for the *crossfader*.

**Range**

Value	Meaning
0	Left side of crossfader
1	Center (not affected by crossfader)
2	Right side of crossfader

**Feedback** None

Nové ve verzi 1.10.0.

**[MicrophoneN]orientation**

Nové ve verzi 1.10.0.

Zastaralé od verze 1.10.0: The control is not processed in the Mixer, which is also why there are no orientation controls for Microphones in the GUI.

**[MicrophoneN]PeakIndicator**

**[AuxiliaryN]PeakIndicator**

Indicates when the signal is clipping (too loud for the hardware and is being distorted)

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Microphone Clip light

Nové ve verzi 1.10.0.

**[MicrophoneN]PeakIndicatorL**

**[AuxiliaryN]PeakIndicatorL**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) for the left channel

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (left)

Nové ve verzi 2.0.0.

**[MicrophoneN]PeakIndicatorR**

**[AuxiliaryN]PeakIndicatorR**

Indicates when the signal is clipping (too loud for the hardware and is being distorted) for the right channel

This is a *ControlPotMeter control*.

**Range** binary

**Feedback** Clip light (right)

Nové ve verzi 2.0.0.

**[MicrophoneN]pfl**

**[AuxiliaryN]pfl**

Toggles *headphone cueing (PFL)*.

**Range** binary

**Feedback** Headphone button

**[MicrophoneN] talkover**

**[AuxiliaryN] talkover**

Hold value at 1 to mix channel input into the master output. For [AuxiliaryN] use *[AuxiliaryN], master* instead. Note that [AuxiliaryN] also take *[AuxiliaryN], orientation* into account.

**Range** binary

**Feedback** Microphone: Talk button Auxiliary: N/A

Nové ve verzi 1.10.0.

**[MicrophoneN] volume**

**[AuxiliaryN] volume**

Adjusts the channel volume fader

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Microphone volume fader changes

Nové ve verzi 1.10.0.

**[MicrophoneN] pregain**

**[AuxiliaryN] pregain**

Adjusts the gain of the input

This is a *ControlPotMeter control*.

**Range** 0.0..1.0..4.0

**Feedback** Microphone gain knob

**[MicrophoneN] mute**

**[AuxiliaryN] mute**

Mutes the channel

**Range** binary

**Feedback** Mute button

Nové ve verzi 2.0.0.

**[MicrophoneN] VuMeter**

**[AuxiliaryN] VuMeter**

Outputs the current instantaneous channel volume

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Microphone VU meter changes

Nové ve verzi 1.10.0.

**[MicrophoneN] VuMeterL**

**[AuxiliaryN] VuMeterL**

Outputs the current instantaneous deck volume for the left channel

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Deck VU meter L

Nové ve verzi 2.0.0.

**[MicrophoneN]VuMeterR**

**[AuxiliaryN]VuMeterR**

Outputs the current instantaneous deck volume for the right channel

This is a *ControlPotMeter control*.

**Range** default

**Feedback** Deck VU meter R

Nové ve verzi 2.0.0.

## 16.5.5 The [VinylControl] group

The [VinylControl] group can toggle the *vinyl control feature*.

**[VinylControl]Toggle**

Moves control by a vinyl control signal from one deck to another if using the single deck vinyl control (VC) feature.

**Range** binary

**Feedback** If VC isn't enabled on any decks, enable it on the first one we're receiving samples for. If VC is enabled on a single (exclusive) deck, and another deck is setup to receive samples, disable it on the former deck and enable it on the next eligible deck (ordered by deck number). If VC is enabled on multiple decks, don't do anything.

Nové ve verzi 1.10.0.

**[VinylControl]show\_vinylcontrol**

Toggle the vinyl control section in skins.

**Range** binary

**Feedback** Vinyl controls are shown

Nové ve verzi 1.10.0.

**[VinylControl]gain**

Allows to amplify the „phono“ level of attached turntables to „line“ level. This is equivalent to setting the *turntable boost* in *Options ▶ Preferences ▶ Vinyl Control*

**Range** binary

**Feedback** position of Boost slider in *Options ▶ Preferences ▶ Vinyl Control* (is not updated while viewing this Preferences page)

Nové ve verzi 1.10.0.

## 16.5.6 The [Recording] controls

The controls in the [Recording] group can be used to query and control the *recording of your mix*.

**[Recording]toggle\_recording**

Turns recording on or off.

**Range** binary

**Feedback** Recording icon

**[Recording]status**

Indicates whether Mixxx is currently recording.

**Range**

Value	Meaning
0	Recording Stopped
1	Initialize Recording
2	Recording Active

**Feedback** Recording icon

## 16.5.7 AutoDJ controls

The [AutoDJ] controls allow interacting with *AutoDJ*.

**[AutoDJ] enabled**

Turns Auto DJ on or off.

**Range** binary

**Feedback** AutoDJ button

Nové ve verzi 1.11.0.

**[AutoDJ] shuffle\_playlist**

Shuffles the content of the Auto DJ playlist.

**Range** binary

**Feedback** Order of tracks in the AutoDJ playlist changes.

Nové ve verzi 1.11.0.

**[AutoDJ] skip\_next**

Skips the next track in the Auto DJ playlist.

**Range** binary

**Feedback** Skipped track is removed from the AutoDJ playlist.

Nové ve verzi 1.11.0.

**[AutoDJ] fade\_now**

Triggers the transition to the next track.

**Range** binary

**Feedback** Crossfader slider moves to the other side.

Nové ve verzi 1.11.0.

## 16.5.8 The [Library] controls

The controls in the [Library] group can be used to navigate the *library*. Note that [Library], *MoveUp* and other Move and Scroll controls emulate keypresses and therefore require the Mixxx window to be focused.

**[Library] MoveUp**

Equivalent to pressing the Up key on the keyboard

**Range** Binary

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]MoveDown**

Equivalent to pressing the `Down` key on the keyboard

**Range** Binary

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]MoveVertical**

Move the specified number of locations up or down. Intended to be mapped to an encoder knob.

**Range** Relative (positive values move down, negative values move up)

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]ScrollUp**

Equivalent to pressing the `PageUp` key on the keyboard

**Range** Binary

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]ScrollDown**

Equivalent to pressing the `PageDown` key on the keyboard

**Range** Binary

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]ScrollVertical**

Scroll the specified number of pages up or down. Intended to be mapped to an encoder knob.

**Range** Relative (positive values move down, negative values move up)

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]MoveLeft**

Equivalent to pressing the `Left` key on the keyboard

**Range** Binary

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]MoveRight**

Equivalent to pressing the `Right` key on the keyboard

**Range** Binary

**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]MoveHorizontal**

Move the specified number of locations left or right. Intended to be mapped to an encoder knob.

**Range** Relative (positive values move right, negative values move left)



**Feedback** Currently selected item changes

Nové ve verzi 2.1.0.

**[Library]MoveFocusForward**

Equivalent to pressing the `Tab` key on the keyboard

**Range** Binary

**Feedback** Currently focused pane changes

Nové ve verzi 2.1.0.

**[Library]MoveFocusBackward**

Equivalent to pressing the `Shift + Tab` key on the keyboard

**Range** Binary

**Feedback** Currently focused pane changes

Nové ve verzi 2.1.0.

**[Library]MoveFocus**

Move focus the specified number of panes forward or backwards. Intended to be mapped to an encoder knob.

**Range** Relative (positive values move forward, negative values move backward)

**Feedback** Currently focused pane changes

Nové ve verzi 2.1.0.

**[Library]GoToItem**

Triggers different actions, depending on which interface element currently has keyboard focus:

<b>Search bar</b>	
text box	moves focus to tracks table
Clear button	clears search text
<b>Sidebar</b>	
collapsed node	expands the item (except Tracks and Auto DJ)
leaf node	moves focus to tracks table
<b>Tracks table</b>	Performs the action selected in <i>Preferences ▶ Library ▶ Track Double-Click Action</i> (default is „Load selected track“). Also see <i>Preferences ▶ Decks ▶ Playing track protection</i>
<b>Dialogs / popups</b>	presses <code>Enter</code> . Note: the <i>Move...</i> controls allow to move button focus.

**Range** Binary

**Feedback** Context dependent

Nové ve verzi 2.1.0.

**[Library]AutoDjAddBottom**

**[Playlist]AutoDjAddBottom**

Add selected track(s) to Auto DJ Queue (bottom).

**Range** Binary

**Feedback** Append track(s) to Auto DJ playlist

Nové ve verzi 2.0.0.

**[Library]AutoDjAddTop**

**[Playlist]AutoDjAddTop**

Add selected track(s) to Auto DJ Queue (top).

**Range** Binary

**Feedback** Prepend track(s) to Auto DJ playlist

Nové ve verzi 2.0.0.

**[Library]show\_coverart**

Toggle the Cover Art in Library

**Range** Binary

**[Library]font\_size\_increment**

Increase the size of the library font. If the row height is smaller than the font-size the larger of the two is used.

**Range** Binary

**Feedback** Library view

Nové ve verzi 2.0.0.

**[Library]font\_size\_decrement**

Decrease the size of the library font

Nové ve verzi 2.0.0.

**Range** Binary

**Feedback** Library view

**[Library]font\_size\_knob**

Increase or decrease the size of the library font

Nové ve verzi 2.0.0.

**Range** Relative

**Feedback** Library view

**[Library]sort\_column**

Indicates the sorting column the track table

**Range**

Value	Description	Library	Playlist	Crate	Browse
1	Artist	X	X	X	X
2	Title	X	X	X	X
3	Album	X	X	X	X
4	Albumartist	X	X	X	X
5	Year	X	X	X	X
6	Genre	X	X	X	X
7	Composer	X	X	X	X
8	Grouping	X	X	X	X
9	Tracknumber	X	X	X	X
10	Filetype	X	X	X	X
11	Native Location	X	X	X	X
12	Comment	X	X	X	X
13	Duration	X	X	X	X
14	Bitrate	X	X	X	X

continues on next page

Table 2 – pokračujte na předchozí stránce

Value	Description	Library	Playlist	Crate	Browse
15	BPM	X	X	X	X
16	ReplayGain	X	X	X	X
17	Datetime Added	X	X	X	X
18	Times Played	X	X	X	X
19	Rating	X	X	X	X
20	Key	X	X	X	X
21	Preview	X	X	X	X
22	Coverart	X	X	X	
23	Position		X		
24	Playlist ID		X		
25	Location		X		
26	Filename				X
27	File Modified Time				X
28	File Creation Time				X
29	Sample Rate				
30	Track Color	X	X	X	

**Feedback** Sorting indicator in the column headers of the track table

Nové ve verzi 2.3.0.

**[Library]sort\_column\_toggle**

Equivalent to clicking on column headers. A new value sets *[Library], sort\_column* to that value and *[Library], sort\_order* to 0, setting the same value again will toggle *[Library], sort\_order*.

**Range** Same as for *[Library], sort\_column* or value 0 for sorting according the current column with the cursor on it

**Feedback** Sorting indicator in the column headers of the track table

Nové ve verzi 2.3.0.

**[Library]sort\_order**

Indicate the sort order of the track tables.

**Range** Binary (0 for ascending, 1 for descending)

**Feedback** Sorting indicator in the column headers of the track table

Nové ve verzi 2.3.0.

**[Library]track\_color\_prev**

Set color of selected track to previous color in palette.

**Range** Binary

**Feedback** Track color changes in the library view.

Nové ve verzi 2.3.0.

**[Library]track\_color\_next**

Set color of selected track to next color in palette.

**Range** Binary

**Feedback** Track color changes in the library view.

Nové ve verzi 2.3.0.

## 16.5.9 The [Shoutcast] controls

### [Shoutcast] enabled

Shows if live Internet broadcasting is enabled.

**Range** ?

**Feedback** shoutcast only supports mp3 format as field

### [Shoutcast] status

This control displays whether broadcasting connection to Shoutcast server was successfully established.

**Range** binary

**Feedback** None

## 16.5.10 The [Playlist] controls

[Playlist] controls allow navigating the sidebar and tracks table directly without considering the currently focused widget. This is helpful when another application's window is focused. This group is going to be deprecated at some point, with its controls added to [Library] above.

### Viz také:

See bug #1772184 for the current status.

### [Playlist] SelectPlaylist

Scrolls the given number of items (view, playlist, crate, etc.) in the side pane (can be negative for reverse direction).

**Range** relative value

**Feedback** Library sidebar highlight

### [Playlist] SelectTrackKnob

Scrolls the given number of tracks in the track table (can be negative for reverse direction).

**Range** relative value

**Feedback** Library track table highlight

### [Playlist] LoadSelectedIntoFirstStopped

Loads the currently highlighted song into the first stopped deck

Zastaralé od verze 2.1.0: Use [Library], GoToItem instead.

### [Playlist] SelectNextPlaylist

Switches to the next view (Library, Queue, etc.)

Zastaralé od verze 2.1.0: Use [Library], MoveDown instead.

### [Playlist] SelectPrevPlaylist

**range** binary

**feedback** Library sidebar

Switches to the previous view (Library, Queue, etc.)

Zastaralé od verze 2.1.0: Use [Library], MoveUp instead.

### [Playlist] ToggleSelectedSidebarItem

Toggles (expands/collapses) the currently selected sidebar item.

Nové ve verzi 1.11.0.

Zastaralé od verze 2.1.0: Use `[Library]`, `GoToItem` instead.

**[Playlist] SelectNextTrack**

Scrolls to the next track in the track table.

Zastaralé od verze 2.1.0: Use `[Library]`, `MoveDown` instead.

**[Playlist] SelectPrevTrack**

Scrolls to the previous track in the track table.

Zastaralé od verze 2.1.0: Use `[Library]`, `MoveUp` instead.

### 16.5.11 The [Controls] controls

The `[Controls]` group contains controls that didn't fit in any other group.

**[Controls] touch\_shift**

Once enabled, all touch tab events are interpreted as right click. This control has been added to provide touchscreen compatibility in 2.0 and might be replaced by a general modifier solution in the future.

**Range** binary

**Feedback** All Widgets

Nové ve verzi 2.0.0.

**[Controls] AutoHotcueColors**

If enabled, colors will be assigned to newly created *hotcues* automatically.

**Range** binary

**Feedback** None

Nové ve verzi 2.3.0.

**[Controls] ShowDurationRemaining**

Represents the current state of the remaining time duration display of the loaded track.

**Range**

Value	Meaning
0	currently showing elapsed time, sets to remaining time
1	currently showing remaining time , sets to elapsed time
2	currently showing both (that means we are showing remaining, set to elapsed

**Feedback** None

### 16.5.12 The Effects Framework

In the list below,

- `EffectRack1` leaves room for future expansion to multiple `EffectRacks`.
- `N` ranges from 1 to `[EffectRack1]`, `num_effectunits`, inclusive.
- `M` ranges from 1 to `[EffectRack1_EffectUnitN]`, `num_effectslots`, inclusive. (For a given value of `N`)
- `K` ranges from 1 to `[EffectRack1_EffectUnitN_EffectM]`, `num_parameters`, inclusive. (For given values of `N` and `M`)

- I ranges from 1 to [Master], num\_decks, inclusive.
- J ranges from 1 to [Master], num\_samplers, inclusive.

Nové ve verzi 2.0.0.

## Linking Values

Effect parameters can be linked to the effect's metaknob. This linkage can be user-controlled by changing the link\_type and the link\_inverse control of the parameter. The default link type is loaded from the effect parameter's manifest's linkHint property.

Link Type	Integer Value	Interpretation
None	0	Not controlled by the metaknob
Linked	1	Controlled by the metaknob as it is
Linked Left	2	Controlled by the left side of the metaknob
Linked Right	3	Controlled by the right side of the metaknob
Linked Left Right	4	Controlled by both sides of the metaknob

Link Inverse	Integer Value	Interpretation
Normal	0	Linked in equal relation
Inverse	1	Linked in an inverse relation.

## EQs and Filters

Equalizers and filters are special effects units. The EQs are controlled by [EqualizerRack1\_[ChannelI]\_Effect1] and the filter knob is controlled by [QuickEffectRack1\_[ChannelI]], super1 and [QuickEffectRack1\_[ChannelI]\_Effect1], enabled. Users can choose between several options for the effects loaded in these racks in the Equalizers section of the Preferences window.

## Controls

**[EffectRack1] show**

Show the Effect Rack

**Range** binary

**[EffectRack1] num\_effectunits**

**[EqualizerRack1] num\_effectunits**

**[QuickEffectRack1] num\_effectunits**

The number of EffectUnits in this rack

**Range** integer, read-only

**[EffectRack1] clear**

**[EqualizerRack1] clear**

**[QuickEffectRack1] clear**

Clear the Effect Rack

**Range** binary

**[EffectRack1\_EffectUnitN] chain\_selector**

**[EqualizerRack1\_[ChannelI]] chain\_selector**

**[QuickEffectRack1\_[ChannelI]] chain\_selector**

Select EffectChain preset. > 0 goes one forward; < 0 goes one backward.

**Range** +1/-1

**[EffectRack1\_EffectUnitN] clear**

**[EqualizerRack1\_[ChannelI]] clear**

**[QuickEffectRack1\_[ChannelI]] clear**

Clear the currently loaded EffectChain in this EffectUnit.

**Range** binary

**[EffectRack1\_EffectUnitN] enabled**

**[EqualizerRack1\_[ChannelI]] enabled**

**[QuickEffectRack1\_[ChannelI]] enabled**

If true, the EffectChain in this EffectUnit will be processed. Meant to allow the user a quick toggle for the effect unit.

**Range** binary, default true

**[EffectRack1\_EffectUnitN] focused\_effect**

**[EqualizerRack1\_[ChannelI]] focused\_effect**

**[QuickEffectRack1\_[ChannelI]] focused\_effect**

0 indicates no effect is focused; > 0 indicates the index of the focused effect. Focusing an effect only does something if a controller mapping changes how it behaves when an effect is focused.

**Range** 0..num\_effects

**[EffectRack1\_EffectUnitN] group\_[ChannelI]\_enable**

**[EqualizerRack1\_[ChannelI]] group\_[ChannelI]\_enable**

**[QuickEffectRack1\_[ChannelI]] group\_[ChannelI]\_enable**

Whether or not this EffectChain applies to Deck I

**Range** binary

**[EffectRack1\_EffectUnitN] group\_[Headphone]\_enable**

Whether or not this EffectChain applies to the Headphone output

**Range** binary

**[EffectRack1\_EffectUnitN] group\_[Master]\_enable**

Whether or not this EffectChain applies to the Master output

**Range** binary

**[EffectRack1\_EffectUnitN] group\_[SamplerJ]\_enable**

Whether or not this EffectChain applies to Sampler J

**Range** binary

**[EffectRack1\_EffectUnitN] loaded**

**[EqualizerRack1\_[ChannelI]] loaded**

**[QuickEffectRack1\_[ChannelI]] loaded**

Whether an EffectChain is loaded into the EffectUnit

**Range** binary, read-only

**[EffectRack1\_EffectUnitN] mix**

**[EqualizerRack1\_[ChannelI]] mix**

**[QuickEffectRack1\_[ChannelI]] mix**

The dry/wet mixing ratio for this EffectChain with the EngineChannels it is mixed with

This is a *ControlPotMeter control*.

**Range** 0.0..1.0

[EffectRack1\_EffectUnitN]next\_chain  
 [EqualizerRack1\_[ChannelI]]next\_chain  
 [QuickEffectRack1\_[ChannelI]]next\_chain

Cycle to the next EffectChain preset after the currently loaded preset.

**Range** binary

[EffectRack1\_EffectUnitN]num\_effects  
 [EqualizerRack1\_[ChannelI]]num\_effects  
 [QuickEffectRack1\_[ChannelI]]num\_effects

The number of Effects that this EffectChain has

**Range** integer, read-only

[EffectRack1\_EffectUnitN]num\_effectslots  
 [EqualizerRack1\_[ChannelI]]num\_effectslots  
 [QuickEffectRack1\_[ChannelI]]num\_effectslots

The number of effect slots available in this EffectUnit.

**Range** integer, read-only

[EffectRack1\_EffectUnitN]prev\_chain  
 [EqualizerRack1\_[ChannelI]]prev\_chain  
 [QuickEffectRack1\_[ChannelI]]prev\_chain

Cycle to the previous EffectChain preset before the currently loaded preset.

**Range** binary

[EffectRack1\_EffectUnitN]show\_focus  
 [EqualizerRack1\_[ChannelI]]show\_focus  
 [QuickEffectRack1\_[ChannelI]]show\_focus

Whether to show focus buttons and draw a border around the focused effect in skins. This should not be manipulated by skins; it should only be changed by controller mappings.

**Range** binary

[EffectRack1\_EffectUnitN]show\_parameters  
 [EqualizerRack1\_[ChannelI]]show\_parameters  
 [QuickEffectRack1\_[ChannelI]]show\_parameters

**Range** binary

Whether to show all the parameters of each effect in skins or only show metaknobs.

[EffectRack1\_EffectUnitN]super1  
 [EqualizerRack1\_[ChannelI]]super1  
 [QuickEffectRack1\_[ChannelI]]super1

The EffectChain superknob. Moves the metaknobs for each effect in the chain.

This is a *ControlPotMeter control*.

**Range** 0.0..1.0

[EffectRack1\_EffectUnitN\_EffectM]clear  
 [EqualizerRack1\_[ChannelI]\_Effect1]clear  
 [QuickEffectRack1\_[ChannelI]\_Effect1]clear

Clear the currently loaded Effect in this Effect slot from the EffectUnit.

**Range** binary

[EffectRack1\_EffectUnitN\_EffectM]effect\_selector



[EqualizerRack1\_[ChannelI]\_Effect1]effect\_selector  
 [QuickEffectRack1\_[ChannelI]\_Effect1]effect\_selector  
 Select Effect – >0 goes one forward, <0 goes one backward.

**Range** +1/-1

[EffectRack1\_EffectUnitN\_EffectM]enabled  
 [EqualizerRack1\_[ChannelI]\_Effect1]enabled  
 [QuickEffectRack1\_[ChannelI]\_Effect1]enabled

If true, the effect in this slot will be processed. Meant to allow the user a quick toggle for this effect.

**Range** binary, default true

[EffectRack1\_EffectUnitN\_EffectM]loaded  
 [EqualizerRack1\_[ChannelI]\_Effect1]loaded  
 [QuickEffectRack1\_[ChannelI]\_Effect1]loaded

Whether an Effect is loaded into this EffectSlot

**Range** binary, read-only

[EffectRack1\_EffectUnitN\_EffectM]next\_effect  
 [EqualizerRack1\_[ChannelI]\_Effect1]next\_effect  
 [QuickEffectRack1\_[ChannelI]\_Effect1]next\_effect

Cycle to the next effect after the currently loaded effect.

**Range** binary

[EffectRack1\_EffectUnitN\_EffectM]num\_parameters  
 [EqualizerRack1\_[ChannelI]\_Effect1]num\_parameters  
 [QuickEffectRack1\_[ChannelI]\_Effect1]num\_parameters

The number of parameters the currently loaded effect has.

**Range** integer, read-only, 0 if no effect is loaded

[EffectRack1\_EffectUnitN\_EffectM]num\_parameterslots  
 [EqualizerRack1\_[ChannelI]\_Effect1]num\_parameterslots  
 [QuickEffectRack1\_[ChannelI]\_Effect1]num\_parameterslots

The number of parameter slots available.

**Range** integer, read-only

[EffectRack1\_EffectUnitN\_EffectM]num\_button\_parameters  
 [EqualizerRack1\_[ChannelI]\_Effect1]num\_button\_parameters  
 [QuickEffectRack1\_[ChannelI]\_Effect1]num\_button\_parameters

The number of button parameters the currently loaded effect has.

**Range** integer, read-only, 0 if no effect is loaded

[EffectRack1\_EffectUnitN\_EffectM]num\_button\_parameterslots  
 [EqualizerRack1\_[ChannelI]\_Effect1]num\_button\_parameterslots  
 [QuickEffectRack1\_[ChannelI]\_Effect1]num\_button\_parameterslots

The number of button parameter slots available.

**Range** integer, read-only

[EffectRack1\_EffectUnitN\_EffectM]meta  
 [EqualizerRack1\_[ChannelI]\_Effect1]meta  
 [QuickEffectRack1\_[ChannelI]\_Effect1]meta

Controls the parameters that are linked to the metaknob.

This is a *ControlPotMeter control*.

**Range** 0..1

[EffectRack1\_EffectUnitN\_EffectM]prev\_effect  
 [EqualizerRack1\_[ChannelI]\_Effect1]prev\_effect  
 [QuickEffectRack1\_[ChannelI]\_Effect1]prev\_effect

Cycle to the previous effect before the currently loaded effect.

**Range** binary

[EffectRack1\_EffectUnitN\_EffectM]parameterK  
 [EqualizerRack1\_[ChannelI]\_Effect1]parameterK  
 [QuickEffectRack1\_[ChannelI]\_Effect1]parameterK

The scaled value of the Kth parameter. See the [Parameter Values](#) section for more information.

This is a *ControlPotMeter control*.

**Range** double

[EffectRack1\_EffectUnitN\_EffectM]parameterK\_link\_inverse  
 [EqualizerRack1\_[ChannelI]\_Effect1]parameterK\_link\_inverse  
 [QuickEffectRack1\_[ChannelI]\_Effect1]parameterK\_link\_inverse

The link direction of the Kth parameter to the effect's metaknob.

**Range** bool

[EffectRack1\_EffectUnitN\_EffectM]parameterK\_link\_type  
 [EqualizerRack1\_[ChannelI]\_Effect1]parameterK\_link\_type  
 [QuickEffectRack1\_[ChannelI]\_Effect1]parameterK\_link\_type

The link type of the Kth parameter to the effects's metaknob.

**Range** enum

[EffectRack1\_EffectUnitN\_EffectM]parameterK\_loaded  
 [EqualizerRack1\_[ChannelI]\_Effect1]parameterK\_loaded  
 [QuickEffectRack1\_[ChannelI]\_Effect1]parameterK\_loaded

Whether or not the Kth parameter slot has an effect parameter loaded into it.

**Range** binary, read-only

[EffectRack1\_EffectUnitN\_EffectM]parameterK\_type  
 [EqualizerRack1\_[ChannelI]\_Effect1]parameterK\_type  
 [QuickEffectRack1\_[ChannelI]\_Effect1]parameterK\_type

The type of the Kth parameter value. See the [Parameter Value Types](#) table.

**Range** integer, read-only

[EffectRack1\_EffectUnitN\_EffectM]button\_parameterK  
 [EqualizerRack1\_[ChannelI]\_Effect1]button\_parameterK  
 [QuickEffectRack1\_[ChannelI]\_Effect1]button\_parameterK

The value of the Kth parameter. See the [Parameter Values](#) section for more information.

**Range** double

[EffectRack1\_EffectUnitN\_EffectM]button\_parameterK\_loaded  
 [EqualizerRack1\_[ChannelI]\_Effect1]button\_parameterK\_loaded  
 [QuickEffectRack1\_[ChannelI]\_Effect1]button\_parameterK\_loaded

Whether or not the Kth parameter slot has an effect parameter loaded into it.

**Range** binary, read-only

[EffectRack1\_EffectUnitN\_EffectM]button\_parameterK\_type  
 [EqualizerRack1\_[ChannelI]\_Effect1]button\_parameterK\_type  
 [QuickEffectRack1\_[ChannelI]\_Effect1]button\_parameterK\_type

The type of the Kth parameter value. See the [Parameter Value Types](#) table.

**Range** integer, read-only

### 16.5.13 Removed controls

These controls have been removed from Mixxx. Skins and controller mappings that attempt to use them will not work correctly.

**[ChannelN] flanger**

Toggles the flanger effect.

Zastaralé od verze 2.0.0: This control has been **removed** without a direct replacement. Use the *effects framework* instead.

**[ChannelN] Hercules1**

**[ChannelN] Hercules2**

**[ChannelN] Hercules3**

**[ChannelN] Hercules4**

Zastaralé od verze ??: This control has been **removed**.

**[ChannelN] NextTask**

Zastaralé od verze ??: This control has been **removed**.

**[ChannelN] NextTrack**

Zastaralé od verze ??: This control has been **removed**.

**[ChannelN] PrevTask**

Zastaralé od verze ??: This control has been **removed**.

**[ChannelN] PrevTrack**

Zastaralé od verze ??: This control has been **removed**.

**[ChannelN] transform**

Zastaralé od verze ??: This control has been **removed**.

**[Flanger] lfoDepth**

Adjusts the intensity of the flange effect

Zastaralé od verze 2.0.0: This control has been **removed** without a direct replacement. Use the *effects framework* instead.

**[Flanger] lfoDelay**

Adjusts the phase delay of the flange effect in microseconds

Zastaralé od verze 2.0.0: This control has been **removed** without a direct replacement. Use the *effects framework* instead.

**[Flanger] lfoPeriod**

Adjusts the wavelength of the flange effect in microseconds

Zastaralé od verze 2.0.0: This control has been **removed** without a direct replacement. Use the *effects framework* instead.

## 16.6 The Mixxx Settings Directory

The Mixxx settings directory contains all user data and settings of your Mixxx installation.

### 16.6.1 Location

#### Linux

`~/.mixxx/`

**Note:** make sure you have „Show hidden files“ enabled in your file manager in order to show all dot-files and directories.

#### Windows

Vista and up: `%LOCALAPPDATA%\Mixxx`

XP and below: `%USERPROFILE%\Local Settings\Application Data\Mixxx\`

Type either of those into the location bar of a Computer or Folder window.

#### macOS

Mixxx 2.3: `~/Library/Containers/org.mixxx.mixxx/Data/Library/Application Support/Mixxx`

Mixxx 2.2 and earlier: `~/Library/Application Support/Mixxx`

### 16.6.2 Content

**analysis** This contains all waveform analysis data. This is used to compose a track's scrolling waveform and track overview. If not existent, this data will be recreated each time a track is loaded into a Mixxx deck. Thus it does not belong to the essential data that needs to be backed up.

**broadcast\_profiles** All broadcast profiles you have configured.

**controllers** All controller mappings you stored. This can be downloaded and *self-built mappings*, as well as built-in mappings that you modified in *Preferences > Controllers > YourController* manually or with the MIDI Wizard.

**effects.xml** The current configuration of the 4 effect units, incl. the state of all controls.

**mixxx.cfg** Everything you configured in the Preferences, deck settings, skin settings, AutoDJ configuration, effect routing etc.

**mixxx.log[.NN]** Log files of the last few Mixxx sessions, with `mixxx.log` being the most recent one, followed by `mixxx.log.1` etc.

**mixxxdb.sqlite** The Mixxx library database. All track locations, all track metadata, saved cues, loops, colors, playlists, crates, ...

**samplers.xml** Stores tracks currently loaded to sample decks.

**sandbox.cfg** This is used under macOS to track which files Mixxx will have access to

**soundconfig.xml** Sound device configuration from Preferences > Sound Hardware

## 16.7 Additional Resources

Got questions? Need more information? Want to *contribute*? There are a variety of other resources where you can find out more:

- **Mixxx Website:** For general information and updates, as well as the latest news on Mixxx, go to: <https://mixxx.org>
- **Mixxx Wiki:** For up-to-date information about supported DJ hardware, documentation on making controller mappings, and other tips and tricks, go to: <https://github.com/mixxxdj/mixxx/wiki>
- **Keyboard Shortcuts:** This a list for all default Mixxx keyboard shortcuts. Please note that these shortcuts are for keyboards configured in English layout and default shortcuts may be different if your language setting is not English. *default keyboard mappings*
- **Mixxx Community Forums:** To search for answers, post your question or answer other DJ's questions, go to: <https://mixxx.org/forums>
- **Mixxx Zulip Chat:** Hang out with Mixxx developers and users on our Zulip chat. Stand-alone apps for Zulip can be configured to use the server `mixxx.zulipchat.com`.
- **Mixxx Source Code:** To download the source code for Mixxx, go to: <https://github.com/mixxxdj/mixxx>
- **Mixxx Bug Tracker:** To report a bug or request a feature, go to: <https://bugs.launchpad.net/mixxx>
- **Mixxx Translations:** To translate Mixxx and promote your mother tongue, go to: <https://www.transifex.com/mixxx-dj-software/public/>. Please read the [Translation FAQ](#) first.
- **Social Media:** Follow us on [Twitter](#), and [Facebook](#).

## 16.8 Changelog

### 16.8.1 2.3.2 (Unreleased)

- Playlist: Enable sorting by color #4352 lp:1945976
- Fix crash when using Doubling/Halving/etc. BPM from track's Properties window on tracks without BPM #4587 lp:1955853
- Fix writing metadata on Windows for files that have never been played #4586 lp:1955331
- Preserve file creation time when writing metadata on Windows #4586 lp:1955314
- Fix handling of file extension when importing and exporting sampler settings #4539
- Fix crash when using an empty directory as resource path using the `--resource-path` command line option #4575 lp:1934560

### Packaging

- Downloads of external dependencies are placed in `build/downloads`
- The sources for `libkeyfinder` are now expected in `build/downloads/libkeyfinder-2.2.5.zip` instead of `build/download/libkeyfinder/v2.2.5.zip`
- CMake: Adjust the download directory and name of external dependencies #4511
- Fix/Improve Appstream metainfo #4344 #4346 #4349

## 16.8.2 2.3.1 (2021-09-29)

- Added mapping for the Numark DJ2GO2 Touch controller #4108 #4287
- Added mapping for the Numark Mixtrack Pro FX controller #4160
- Updated mapping for Behringer DDM4000 mixer #4262
- Updated mapping for Denon MC7000 controller #4021
- Hercules Inpulse 300: Add better FX controls and other minor improvements #4246
- Denon MC7000: Improve slip mode and jog wheel handling #4021 #4324
- Disabled detection of keyboards and mice as HID controllers #4243
- Disabled detection of all HID controllers with Apple's vendor ID. Apple doesn't build actual controllers. #4260 #4273
- Add support for HiDPI scale factors of 125% and 175% (only with Qt 5.14+) lp1938102 #4161
- Fix unhandled exception when parsing corrupt Rekordbox PDB files lp1933853 #4040
- Fix Echo effect adding left channel samples to right channel #4141
- Fix bad phase seek when starting from preroll lp1930143 #4093
- Fix bad phase seek when a channel's audible status changes #4156
- Tango skin: Show crossfader assign buttons by default #4046
- Fix keyfinder library in arm64 builds #4047
- Fix wrong track being recorded in History lp1933991 #4041 #4059 #4107 #4296
- Fix support for relative paths in the skin system which caused missing images in third-party skins #4151
- Fix relocation of directories with special/reserved characters in path name #4146
- Update keyboard shortcuts sheet #4042
- Library: resize the Played checkbox and BPM lock with the library font #4050
- Don't allow Input focus on waveforms #4134
- Fix performance issue on AArch64 by enabling flush-to-zero for floating-point arithmetic #4144
- Fix custom key notation not restored correctly after restart #4136
- Traktor S3: Disable scratch when switching decks to prevent locked scratch issue #4073
- FFmpeg: Ignore inaudible samples before start of stream #4245
- Controller Preferences: Don't automatically enable checkbox if controller is disabled #4244 lp:1941042
- Tooltips: Always show tooltips in preferences #4198 lp:1840493
- Tooltips: Use item label for tooltips in library side bar and show ID when debugging. #4247
- Library sidebar: Also activate items on PageUp/Down events. #4237
- Fix handling of preview button cell events in developer mode. #4264 lp:1929141
- Auto DJ: Fix bug which could make an empty track stop Auto DJ. #4267 lp:1941743
- Fix Auto DJ skipping tracks randomly #4319 lp1941989
- Fix high CPU load due to extremely high internal sync clock values #4312 lp1943320
- Fix preference option for re-analyzing beatgrids imported from other software #4288

- Fix wrong base tag used for deployment and displayed in About dialog #4070

## Packaging

- It is no longer necessary to manually copy the udev rule file in packaging scripts. Now pkg-config is used to determine the udevdir used to install the rules file in the CMake install step when CMAKE\_INSTALL\_PREFIX is / or /usr. #4126
- Various build issues on FreeBSD are fixed #4122 #4123 #4124
- .desktop file has been renamed to org.mixxx.Mixxx.desktop according to freedesktop standards #4206
- Uses system provided hidapi library if version >= 0.10.1 #4215
- Please update PortAudio to 19.7 if you have not done so already. This is required for Mixxx to work with PipeWire via the JACK API for many devices.
- Install multiple sizes of rasterized icons #4204 #4315 #4254
- CMake: Fixed detection of SoundTouch pkgconfig file and version #4209
- Fix AppStream metainfo #4205 #4317

## 16.8.3 2.3.0 (2021-06-28)

### Hotcues

- Add hotcue colors and custom labels by right clicking hotcue buttons or right clicking hotcues on overview waveforms #2016 #2520 #2238 #2560 #2557 #2362
- Mouse hover cues on overview waveform to show time remaining until the cue #2238

### Hotcue & Track Colors

- Add configurable color per track #2470 #2539 #2545 #2630 lp:1100882
- Add customizable color palettes for hotcue and track colors #2530 #2589 #3749 #2902
- Add hotcue color find-and-replace tool #2547

### Importing From Other DJ Software

- Import cue points, track colors, and playlists from Serato file tags & database #2480 #2526 #2499 #2495 #2673 #3885
  - Note: Mixxx does not yet support multiple loops per track. We are [working on this for Mixxx 2.4](#). In Mixxx 2.3, if you import a track with multiple loops from Serato, Mixxx will use the first loop cue as the single loop Mixxx currently supports. The imported loops are still stored in Mixxx's database and are treated as hotcues in Mixxx 2.3. If you do not delete these hotcues, they will be usable as loops in Mixxx 2.4. Serato keeps loops and hotcues in separate lists, but Mixxx does not, so loops from Serato are imported starting as hotcue 9.
- Import cue points, track colors, and playlists from Rekordbox USB drives #2119 #2555 #2543 #2779
  - Note: The first Rekordbox memory cue is imported for the main cue button in Mixxx and the remaining Rekordbox memory cues are imported as Mixxx hotcues, starting with the next hotcue number after the last hotcue from Rekordbox.

- Note: Mixxx does not yet support multiple loops per track. Imported loops from Rekordbox are treated like imported loops from Serato, so refer to the note above for details.

## Intro & Outro Cues

- Add intro & outro range cues with automatic silence detection #1242
- Show duration of intro & outro ranges on overview waveform #2089
- Use intro & outro cues in AutoDJ transitions #2103

## Deck cloning

- Add deck cloning (also known as „instant doubles“ in other DJ software) by dragging and dropping between decks #1892 and samplers #3200
- Clone decks by double pressing the load button on a controller (with option to disable this) #2024 #2042

## Skins & GUI

- Aesthetically revamped LateNight skin #2298 #2342
- Right click overview waveform to show time remaining until that point #2238
- Show track info dialog when double clicking track labels in decks #2990
- Show track context menu when right clicking text in decks #2612 #2675 #2684 #2696
- Add laptop battery widget to skins #2283 #2277 #2250 #2228 #2221 #2163 #2160 #2147 #2281 #2319 #2287
- Show when passthrough mode is active on overview waveforms #2575 #2616
- Changed format of currently playing track in window title from „artist, title“ to „artist - title“ #2807
- Workaround Linux skin change crash #3144 lp:1885009
- Fix touch control lp:1895431
- Fix broken knob interaction on touchscreens #3512
- AutoDJ: Make „enable“ shortcut work after startup #3242
- Add rate range indicator #3693
- Allow menubar to be styled #3372 #3788
- Add Donate button to About dialog #3838 #3846
- Add Scrollable Skin Widget #3890
- Fix minor visual issues in Skins #3958 #3954 #3941 #3938 #3936 #3886 #3927 #3844 #3933 #3835 #3902 #3931



## Music Feature Analysis

- Multithreaded analysis for much faster batch analysis on multicore CPUs #1624 #2142 lp:1641153
- Fix bugs affecting key detection accuracy #2137 #2152 #2112 #2136
  - Note: Users who have not manually corrected keys are advised to clear all keys in their library by pressing Ctrl + A in the library, right clicking, going to Reset -> Key, then reanalyzing their library. This will freeze the GUI while Mixxx clears the keys; this is a known problem that we will not be able to fix for 2.3. Wait until it is finished and you will be able to reanalyze tracks for better key detection results.
- Remove VAMP plugin support and use Queen Mary DSP library directly. vamp-plugin-sdk and vamp-hostsdk are no longer required dependencies. #926
- Improvements BPM detection on non-const beatgrids #3626
- Fix const beatgrid placement #3965 #3973

## Music Library

- Add support for searching for empty fields (for example `crate: ""`) lp:1788086
- Improve synchronization of track metadata and file tags #2406
- Library Scanner: Improve hashing of directory contents #2497
- Rework of Cover Image Hashing lp:1607097 #2507 #2508
- MusicBrainz: Handle 301 status response #2510
- MusicBrainz: Add extended metadata support lp:1581256 #2522
- TagLib: Fix detection of empty or missing file tags lp:1865957 #2535
- Fix caching of duplicate tracks that reference the same file #3027
- Use 6 instead of only 4 compatible musical keys (major/minor) #3205
- Fix possible crash when trying to refocus the tracks table while another Mixxx window has focus #3201
- Don't create new tags in file when exporting metadata to it #3898
- Fix playlist files beginning with non-english characters not being loaded #3916
- Enable sorting in „Hidden Tracks“ and „Missing Tracks“ views #3828 lp:1828555 lp:1924616
- Fix track table being empty after start #3935 lp:1930546 lp:1924843

## Audio Codecs

- Add FFmpeg audio decoder, bringing support for ALAC files #1356
- Include LAME MP3 encoder with Mixxx now that the MP3 patent has expired lp:1294128 buildserver:#37 buildserver:9e8bcee
- Add Opus streaming and recording support. lp:1338413
- Remove support for SoundSource plugins because the code was not well-maintained and could lead to crashes lp:1792747
- Add HE-AAC encoding capabilities for recording and broadcasting #3615

## Audio Engine

- Fix loss of precision when dealing with floating-point sample positions while setting loop out position and seeking using vinyl control #3126 #3127
- Prevent moving a loop beyond track end #3117 lp:1799574
- Fix possible memory corruption using JACK on Linux #3160
- Fix changing of vinyl lead-in time lp:1915483 #3781
- Fix tempo change of non-const beatgrid track on audible deck when cueing another track #3772
- Fix crash when changing effect unit routing #3882 lp:1775497
- Make microphone ducking use strength knob the same way in automatic & manual mode #2750

## Controllers

- Improve workflow for configuring controller mappings and editing mappings #2569 #3278 #3667
- Improve error reporting from controller scripts #2588
- Make hotcue and track colors mappable on controllers #2030 #2541 #2665 #2520
- Add way to change library table sorting from controllers #2118
- Add support for velocity sensitive sampler buttons in Components JS library #2032
- Add logging when script ControlObject callback is disconnected successfully #2054
- Add controller mapping for Roland DJ-505 #2111
- Add controller mapping for Numark iDJ Live II #2818
- Add controller mapping for Hercules DJControl Inpulse 200 #2542
- Add controller mapping for Hercules DJControl Jogvision #2370
- Add controller mapping for Pioneer DDJ-200 #3185 #3193 #3742 #3793 #3949
- Add controller mapping for Pioneer DDJ-400 #3479
- Add controller mapping for ION Discover DJ Pro #2893
- Add controller mapping for Native Instrument Traktor Kontrol S3 #3031
- Add controller mapping for Behringer BCR2000 #3342 #3943
- Add controller mapping for Behringer DDM4000 #3542
- Update controller mapping for Allen & Heath Xone K2 to add intro/outro cues #2236
- Update controller mapping for Hercules P32 for more accurate headmix control #3537
- Update controller mapping for Native Instruments Traktor Kontrol S4MK2 to add auto-slip mode and pitch fader range #3331
- Fix Pioneer DDJ-SB2 controller mapping auto tempo going to infinity bug #2559 lp:1846403
- Fix Numark Mixtrack Pro 3 controller mapping inverted FX on/off control #3758
- Gracefully handle MIDI overflow #825

## Other

- Add CMake build system with `ccache` and `sccache` support for faster compilation times and remove SCons #2280 #3618
- Make Mixxx compile even though `QT_NO_OPENGL` or `QT_OPENGL_ES_2` is defined (fixes build on Raspberry Pi) [lp:1863440](#) #2504
- Fix ARM build issues #3602
- Fix missing manual in DEB package [lp:1889776](#) #2985
- Add macOS codesigning and notarization to fix startup warnings #3281
- Don't trash user configuration if an error occurs when writing #3192
- Enable CUE sheet recording by default #3374
- Fix crash when double clicking GLSL waveforms with right mouse button #3904
- Derive Mixxx version from `git describe` #3824 #3841 #3848
- Improve tapping the BPM of a deck #3790 [lp:1882776](#)
- And countless other small fixes and improvements (too many to list them all!)

### 16.8.4 2.2.4 (2020-06-27)

- Store default recording format after „Restore Defaults“ [lp:1857806](#) #2414
- Prevent infinite loop when decoding corrupt MP3 files #2417
- Add workaround for broken libshout versions #2040 #2438
- Speed up purging of tracks [lp:1845837](#) #2393
- Don't stop playback if vinyl passthrough input is configured and PASS button is pressed #2474
- Fix debug assertion for invalid crate names [lp:1861431](#) #2477
- Fix crashes when executing actions on tracks that already disappeared from the DB #2527
- AutoDJ: Skip next track when both deck are playing [lp:1399974](#) #2531
- Tweak scratch parameters for Mixtrack Platinum #2028
- Fix auto tempo going to infinity on Pioneer DDJ-SB2 #2559
- Fix `bpm.tapButton` logic and reject missed & double taps #2594
- Add controller mapping for Native Instruments Traktor Kontrol S2 MK3 #2348
- Add controller mapping for Soundless joyMIDI #2425
- Add controller mapping for Hercules DJControl Inpulse 300 #2465
- Add controller mapping for Denon MC7000 #2546
- Add controller mapping for Stanton DJC.4 #2607
- Fix broadcasting via broadcast/recording input [lp:1876222](#) #2743
- Only apply ducking gain in manual ducking mode when talkover is enabled [lp:1394968](#) [lp:1737113](#) [lp:1662536](#) #2759
- Ignore MIDI Clock Messages (0xF8) because they are not usable in Mixxx and inhibited the screensaver #2786

### 16.8.5 2.2.3 (2019-11-24)

- Don't make users reconfigure sound hardware when it has not changed #2253
- Fix MusicBrainz metadata lookup lp:1848887 #2328
- Fix high DPI scaling of cover art #2247
- Fix high DPI scaling of cue point labels on scrolling waveforms #2331
- Fix high DPI scaling of sliders in Tango skin #2318
- Fix sound dropping out during recording lp:1842679 #2265 #2305 #2308 #2309
- Fix rare crash on application shutdown #2293
- Workaround various rare bugs caused by database inconsistencies lp:1846971 #2321
- Improve handling of corrupt FLAC files #2315
- Don't immediately jump to loop start when loop\_out is pressed in quantized mode lp:1837077 #2269
- Preserve order of tracks when dragging and dropping from AutoDJ to playlist lp:1829601 #2237
- Explicitly use X11 Qt platform plugin instead of Wayland in .desktop launcher lp:1850729 #2340
- Pioneer DDJ-SX: fix delayed sending of MIDI messages with low audio buffer sizes #2326
- Enable modplug support on Linux by default lp:1840537 #2244 #2272
- Fix keyboard shortcut for View > Skin Preferences lp:1851993 #2358 #2372
- Re-loop Terminal Mix: Fix mapping of sampler buttons 5-8 lp:1846966 #2330

### 16.8.6 2.2.2 (2019-08-10)

- Fix battery widget with upower <= 0.99.7. #2221
- Fix BPM adjust in BpmControl. lp:1836480
- Disable track metadata export for .ogg files and TagLib 1.11.1. lp:1833190
- Fix interaction of hot cue buttons and looping. lp:1778246
- Fix detection of moved tracks. #2197
- Fix playlist import. #2200 lp:1687828
- Fix updating playlist labels. lp:1837315
- Fix potential segfault on exit. lp:1828360
- Fix parsing of invalid BPM values in MP3 files. lp:1832325
- Fix crash when removing rows from empty model. #2128
- Fix high DPI scaling of RGB overview waveforms. #2090
- Fix for OpenGL SL detection on macOS. lp:1828019
- Fix OpenGL ES detection. lp:1825461
- Fix FX1/2 buttons missing Mic unit in Deere (64 samplers). lp:1837716
- Tango64: Re-enable 64 samplers. #2223
- Numark DJ2Go re-enable note-off for deck A cue button. #2087

- Replace Flanger with QuickEffect in keyboard mapping. #2233

### 16.8.7 2.2.1 (2019-04-22)

- Include all fixes from Mixxx 2.1.7 and 2.1.8
- Fix high CPU usage on MAC due to preview column [lp:1812763](#)
- Fix HID controller output on Windows with common-hid-packet-parser.js
- Fix rendering slow down by not using QPainter in WSpinny [lp:1530720](#)
- Fix broken Mic mute button [lp:1782568](#)
- added quick effect enable button to the control picker menu
- Fix Cover Window close issue with empty cover arts
- Fix Numark Mixtrack 3 mapping. #2057

### 16.8.8 2.2.0 (2018-12-17)

#### General

- Update from Qt4 to Qt5.
- Use Qt5's automatic high DPI scaling (and remove the old scaling option from the preferences).
- Vectorize remaining raster graphics for better HiDPI support.

#### Effects

- Add mix mode switch (Dry/Wet vs Dry+Wet) for effect units.
- Add support for LV2 effects plugins (currently no way to show plugin GUIs).
- Add preference option for selecting which effects are shown in the list of available effects in the main window (all LV2 effects are hidden by default and must be explicitly enabled by users).

#### Skins

- Add 8 sampler and small sampler options to LateNight.
- Add key / BPM expansion indicators to Deere decks.
- Add skin settings menu to LateNight.

## Controllers

- Add controller mapping for Numark Mixtrack Platinum.
- Update controller mapping for Numark N4.
- Add spinback and break for Vestax VCI-400 mapping.

## Miscellaneous

- Add preference option to adjust the play position marker of scrolling waveforms.
- Add preference option to adjust opacity of beatgrid markers on scrolling waveforms.
- Support IRC/AIM/ICQ broadcast metadata.

### 16.8.9 2.1.8 (2019-04-07)

- Fix a rare chance for a corrupt track file while writing metadata in out of disk situations. [lp:1815305](#)
- Fix export of BPM track file metadata. [lp:1816490](#)
- Fix sending of broadcast metadata with TLS enabled libshout 2.4.1. [lp:1817395](#)
- Fix resdiscovering purged tracks in all cases. [lp:1821514](#)
- Fix dropping track from OSX Finder. [lp:1822424](#)

### 16.8.10 2.1.7 (2019-01-15)

- Fix syncing to doublespeed [lp:1808697](#)
- Fix issues when changing beats of a synced track [lp:1808698](#)
- Fix direction of pitch bend buttons when inverting rate slider [lp:1770745](#)
- Use first loaded deck if no playing deck is found [lp:1784185](#)
- Encode file names correctly on macOS [lp:1776949](#)

### 16.8.11 2.1.6 (2018-12-23)

- Fix crash when loading a Qt5 Soundsource / Vamp Plug-In. [lp:1774639](#)
- Validate effect parameter range. [lp:1795234](#)
- Fix crash using the bpm\_tap button without a track loaded. [lp:1801844](#)
- Fix possible crash after ejecting a track. [lp:1801874](#)
- Fix wrong bitrate reported for faulty mp3 files. [lp:1782912](#)
- Fix Echo effect syncing [lp:1793232](#)
- Fix iTunes context menu [lp:1799932](#)
- Fix loading the wrong track after delete search and scroll. [lp:1803148](#)
- Improve search bar timing. [lp:1635087](#)
- Fix quoted search sentence. [lp:1784141](#)

- Fix loading a track formerly not existing. [lp:1800395](#)
- Fix importing m3u files with blank lines. [lp:1806271](#)
- Fix position in sampler overview waveforms. [lp:1744170](#)
- Don't reset rate slider, syncing a track without a beatgrid. [lp:1783020](#)
- Clean up iTunes track context menu. [lp:1800335](#)
- Collapsed sampler are not analyzed on startup. [lp:1801126](#)
- search for decoration characters like „““. [lp:1802730](#)
- Fix cue button blinking after pressing eject on an empty deck. [lp:1808222](#)

### **16.8.12 2.1.5 (2018-10-28)**

- Code signing for Windows builds. [lp:1517823](#)
- Fix crash on exit when preferences is open. [lp:1793185](#)
- Fix crash when analyzing corrupt MP3s. [lp:1793387](#)
- Fix crash when importing metadata from MusicBrainz. [lp:1794993](#)
- Library search fixes when single quotes are used. [lp:1784090](#) [lp:1789728](#)
- Fix scrolling waveform on Windows with WDM-KS sound API. [lp:1729345](#)
- Fix right clicking on beatgrid alignment button in Tango and LateNight skins. [lp:1798237](#)
- Improve speed of importing iTunes library. [lp:1785545](#)
- Add 2 deck mapping for DJTechTools MIDI Fighter Twister.

### **16.8.13 2.1.4 (2018-08-29)**

Fix track selection not getting shown in the track table on Windows. There are no changes to the source code, but the Jenkins build configuration was changed to delete the Jenkins workspace before each build. [lp:1751482](#)

### **16.8.14 2.1.3 (2018-08-20)**

Fix a severe performance regression on Windows: [Mixxx 2.1.2](#) running much slower than 2.1.1

### **16.8.15 2.1.2 (2018-08-10)**

Yet another bugfix release of Mixxx 2.1. Here is a quick summary of what is new in [Mixxx 2.1.2](#):

- Allow maximum deck speed of 4x normal.
- Don't always quantize hotcues, a 2.1.1 regression. [lp:1777429](#)
- Fix artifacts using more than 32 samplers. [lp:1779559](#)
- store No EQ and Filter persistently. [lp:1780479](#)
- Pad unreadable samples with silence on cache miss. [lp:1777480](#)
- Fixing painting of preview column for Qt5 builds. [lp:1776555](#)

- LateNight: Fix play button right click. [lp:1781829](#)
- LateNight: Added missing sort up/down buttons.
- Fix sampler play button tooltips. [lp:1779468](#)
- Shade: remove superfluid margins and padding in sampler.xml. [lp:1773588](#)
- Deere: Fix background-color code.
- iTunes: Don't stop import in case of duplicated Playlists. [lp:1783493](#)

### **16.8.16 2.1.1 (2018-06-13)**

After two months it is time to do a bugfix release of Mixxx 2.1. Here is a quick summary of what is new in Mixxx 2.1.1:

- Require Soundtouch 2.0 to avoid segfault. [lp:1577042](#)
- Improved skins including library view fix. [lp:1773709](#) [lp:1772202](#) [lp:1763953](#)
- Fix crash when importing ID3v2 APIC frames. [lp:1774790](#)
- Synchronize execution of Vamp analyzers. [lp:1743256](#)
- DlgTrackInfo: Mismatching signal/slot connection.
- Detect M4A decoding errors on Windows. [lp:1766834](#)
- Fix spinback inertia effect.
- Fix decoding fixes and upgrade DB schema. [lp:1766042](#) [lp:1769717](#)
- Fix integration of external track libraries. [lp:1766360](#)
- Fix memory leak when loading cover art. [lp:1767068](#)
- Fix clearing of ReplayGain gain/ratio in file tags. [lp:1766094](#)
- Fix crash when removing a quick link. [lp:1510068](#)
- Fdlib: Thread-safe and reentrant generation of filters. [lp:1765210](#)
- Fix unresponsive scrolling through crates & playlists using encoder. [lp:1719474](#)
- Swap default values for temp/perm rate changes. [lp:1764254](#)

### **16.8.17 2.1.0 (2018-04-15)**

After two years of hard work, we are pleased to announce Mixxx 2.1. We have overhauled the effects system, redesigned the skins, added and improved lots of controller mappings, rewrote the audio file decoders twice, and of course fixed a bunch of bugs. Download it!

Here is a quick summary of what is new in Mixxx 2.1.0:

- Graphical interface scales for high resolution screens
- Overhauled Deere and LateNight skins
- New Tango skin
- Effects are synchronized to the tempo
- Effects are processed post-fader and post-crossfader and can be previewed in headphones
- One metaknob per effect with customizable parameter control for intuitive use of effect chains



- Nine new effects: Autopan, Biquad Equalizer, Biquad Full Kill Equalizer, Loudness Contour, Metronome, Parametric Equalizer, Phaser, Stereo Balance, Tremolo
- Loaded effects and their parameters are saved and restored when Mixxx restarts
- More transparent sounding equalizers (Biquad Equalizer and Biquad Full Kill Equalizer)
- Improved scratching sounds with jog wheels, vinyl control, and dragging waveforms with the mouse
- Simplified looping and beatjump controls
- Configurable rows of 8 samplers with up to 8 rows available for a total of 64 samplers
- Files loaded to samplers are reloaded when Mixxx restarts
- Improved volume normalization algorithm (EBU-R 128)
- Filter library table by crates
- Sort musical keys in library table by circle of fifths
- Write metadata tags back to audio files
- New JavaScript library for controller mapping
- Configure multiple Internet broadcasting stations and use multiple stations at the same time
- Broadcast and record microphones with direct monitoring and latency compensation
- Broadcast and record from an external mixer
- Booth output with independent gain knob for using sound cards with 6 output channels without an external mixer
- Prevent screensaver from starting while Mixxx is running
- CUP (Cue And Play) cue button mode
- Time remaining and time elapsed now take into account the tempo fader
- Clicking cover art now shows it full size in a separate window
- and of course, lots and lots of bug fixes.

Here are controllers with mappings that have been added or updated since the 2.0 release. Mappings marked with an asterisk (\*) have been updated for the new effects interface:

- American Audio VMS2
- American Audio VMS4
- Allen & Heath Xone K2/K1\*
- Behringer CMD Micro
- Behringer CMD MM1\*
- Behringer CMD Studio 4a
- Denon MC4000\*
- Denon MC6000 Mk2\*
- FaderFox DJ2
- Hercules DJ Console 4-Mx\*
- Hercules DJ Control MP3 LE / Glow
- Hercules DJ Control Compact
- Hercules P32\*

- Ion Discover DJ
- Korg Nanokontrol 2
- Korg KAOS DJ
- M-Audio Xponent
- Native Instruments Traktor Kontrol S4 Mk2\*
- Novation Launchpad Mk1 & Mk2
- Novation Twitch
- Numark Mixtrack Pro 3 & Numark Mixtrack 3\*
- Pioneer DDJ-SB2\*
- Pioneer DDJ-SX\*
- Reloop Beatmix 2
- Reloop Beatmix 4
- Reloop Digital Jockey 3 ME
- Reloop Terminal Mix 2
- Reloop Terminal Mix 4
- Vestax VCI-100 Mk2
- Vestax Typhoon

For users upgrading from older versions of Mixxx, we have a few important announcements. First, if you are using Windows, you will have to uninstall any old versions of Mixxx before you can install 2.1. How to uninstall Mixxx varies on different versions of Windows:

- Windows Vista, 7, and 8: [Start > Control Panel > Programs > Uninstall a Program](#)
- Windows 10: [Start > Control Panel > Programs > Programs And Features > look for Mixxx > Uninstall](#)

If you are upgrading from an older version of Mixxx and have MP3 files in your library, we have another important announcement. The good news is that we fixed a bug where the waveforms and audio playback of MP3 files were misaligned. The bad news is that we have no way of knowing which MP3 files were affected or how much the offset was. That means that waveforms, beatgrids, cues, and loops from older versions of Mixxx may be offset by an unknown amount for any MP3 file. Only MP3 files were affected by this bug; other audio file types are unaffected. You can either correct your beatgrids and cue points manually for each track, or you can clear this information for all MP3s and start fresh. Regardless, we recommend clearing the waveforms for all MP3 files. To clear these, type „location:mp3“ into the library search bar, press Control + A to select all tracks, right click, and select the information you want to clear from the menu.

In the works for Mixxx 2.2, we have a big redesign of the library GUI. Along with that will come saving & restoring search queries plus nested crates. We are also planning on adding support for saving and loading custom effect chain presets with the ability to import and export them to share online.

Want to help make Mixxx even more awesome? The biggest thing we need is more people. You do not need to be a programmer to help out. Giving feedback on the design of new features as they are being made is very valuable. Refer to the Testing page on the wiki for more information on how to get involved with that. Reporting bugs and telling us your ideas on the Launchpad bug tracker is a big help too! We cannot fix problems we do not know about, so please let us know if you find any issues with Mixxx. If you would like to help translate Mixxx into another language, refer to the Internationalization wiki page. Of course, more programmers could always help. Read the Developer Documentation on the wiki for tips on getting started contributing code to Mixxx.

We hope you have as much fun with Mixxx as we do!

For a full list of new features and bugfixes, check out the [2.1.0 milestone on Launchpad](#).

### 16.8.18 2.0.0 (2015-12-31)

- 4 Decks with Master Sync
- New Effects Framework with 4 Effect Units and 5 Built-in Effects:
  - Flanger, Bit Crusher, Reverb, Echo, Filter
  - More to come!
- Configurable, Resizeable User Interface with 3 Brand New Skins
- Cover Art Display
- Music Key Detection and Shifting
- Vinyl Audio Pass-Through
- 4 Microphone inputs and 4 Auxiliary inputs
- MIDI Mapping GUI and Improved Learning Wizard
- MusicBrainz metadata fetching
- RGB Musical Waveforms
- Hundreds of Bug Fixes and Improvements
- New Pitch-Independent Algorithm for Better-Sounding Key-lock

For a full list of new features and bugfixes, check out the [2.0.0 milestone on Launchpad](#).

**Viz také:**

For an overview of previous versions, [take a look](#) at the timeline.

### 17.1 Controllers

Mixxx supports the following *controllers* out of the box. Additional third-party controller mappings that were contributed by the community (but are not yet included in Mixxx) can be found at Mixxx's [Controller Mapping Forum](#).

**Viz také:**

*Using MIDI/HID Controllers* describes how to configure Mixxx to use controllers.

#### 17.1.1 Akai LPD8

- [Manufacturer's product page](#)
- [Forum thread](#)

Nové ve verzi 1.10.1.

---

**Poznámka:** Unfortunately a detailed description of this controller mapping is still missing. If you own this controller, please consider [contributing one](#).

---

#### 17.1.2 Akai MPD24

- [Manufacturer's product page](#)
- [Forum thread](#)

Nové ve verzi 1.8.

**Poznámka:** Unfortunately a detailed description of this controller mapping is still missing. If you own this controller, please consider [contributing one](#).

### 17.1.3 Allen & Heath Xone:K2

The Allen & Heath Xone:K2 is a flexible controller that is mapped to control Mixxx's decks, effects, or both. It has an integrated 4 channel audio interface with 2 RCA jacks and an 1/8" stereo headphone jack. The Xone K1 is the same as the Xone K2 but without the built in audio interface and does not come with the EVA travel case that the K2 comes with.

- [Manufacturer's product page](#)
- [Forum thread](#)
- [Manufacturer's User Guide](#)
- [Blank template diagrams](#)

Nové ve verzi 1.11.

#### Audio Setup

The Xone:K2 is setup in Mixxx > Preferences > Sound Hardware > Output

Output	Device	Channel
Master	XONE:K2	Channel 3 - 4
Headphones	XONE:K2	Channel 1 - 2

Note: This Channel setup is the opposite of what most DJ controllers with an integrated audio interface use. The design choice to not use Channel 1 - 2 for Master Output is known to cause problems, when users want to use the Xone:K2 as default system audio output as the operating system will always use Channel 1 - 2. Thus system audio will go to the headphones.

There are no hardware controls for the volume so they are always at max volume. Adjusting the volume of master and headphone outputs is done by adjusting the gain for those outputs in Mixxx using the controller mapping.

#### Setup

**Requirement:** The Xone K2 must have Latching Layers turned off, which is the default. Refer to the [Xone K2 manual](#) page 12 for details. The K1 does not have Latching Layers.

This mapping can be used with one or multiple Xone K2s/K1s. Multiple Xone K2s/K1s can be connected to each other via X-Link with one of them connected to the computer via USB. Alternatively, when using 2 K2s/K1s, they can both be connected with their own USB cable and this same mapping can be loaded for each.

The layout of the mapping depends on the configured MIDI channel of the controller. Change the MIDI channel of the controller by pressing the bottom right encoder (labeled "Power On Setup/Scroll/Set") while plugging in the controller. Scroll with the encoder to select a MIDI channel. The letter in parentheses corresponds to the last lit button when selecting the channel:

- Channel 15 (O, default out of the box): two decks + two effect units with decks in the middle
- Channel 14 (N): two decks + two effect units with effect units in the middle
- Channel 13 (M): two decks + two effect units with decks on left

- Channel 12 (L): two decks + two effect units with decks on right
- Channel 11 (K): four decks ordered 3 1 2 4
- Channel 10 (J): four decks ordered 1 2 3 4
- Channel 9 (I): four effect units ordered 3 1 2 4
- Channel 8 (H): four effect units ordered 1 2 3 4

## Global Controls

These are available on any configuration with decks, but not the 4 effect unit layout.

- Bottom left encoder:
  - adjust tempo of all decks with sync enabled
  - press and turn: PFL/master mix in headphones
  - shift: headphone gain
  - press with shift: toggle split cue mode
- Bottom right encoder
  - scroll through tracks in library
  - press and release: load selected track into first stopped deck
  - press and hold: load selected track into a deck by pressing the play button of the deck
  - shift: master gain

## Decks

The bottom right button is the shift button. The bottom left button toggles the bottom button grid between a loop layer (amber) and a hotcue layer (red). Holding shift then holding the bottom left layer button at the same time activates supershift mode.

- Top encoder: jog
  - shift: gain
  - supershift: adjust key
- Top encoder press: reset gain
  - shift: master sync
  - supershift: reset key
- Knobs: high/mid/low equalizer knobs
- Top button 1: headphones/PFL
  - shift: reset tempo
  - supershift: set beatgrid to current position
- Top button 2: cue
  - shift: go to beginning of track and stop
  - supershift: keylock

- Top button 3: play
  - shift: reverse
  - supershift: quantize
- Fader: volume
- Bottom buttons (intro/outro cue layer, amber): Intro/outro cues from top to bottom are ordered: - Intro start - Intro end - Outro start - Outro end
  - press: jump to that cue or sets it if it is not set
  - shift: seek forward/back. The top two (intro) buttons seek quickly; the bottom two (outro) seek slowly
  - supershift: delete cue
- Bottom buttons (hotcue layer, red): Hotcues are ordered 1-4 from top to bottom
  - press: jump to that hotcue or sets it if it is not set
  - shift: seek forward/back. The top two buttons seek quickly; the bottom two seek slowly
  - supershift: delete hotcue
- Bottom buttons (loop layer, green):
- Bottom button 1 (red): reloop/disable loop
  - shift: jump to to beginning of loop, stop playback, and activate loop
  - supershift: set loop in point. Hold to move loop in point with play position.
- Bottom button 2 (green): activate loop of selected size
  - shift: activate rolling loop of selected size
  - supershift: set loop out point. Hold to move loop out point with play position.
- Bottom button 3 (amber): double loop size
  - shift: beatjump forward by selected size if no loop is enabled. If loop is enabled, move the loop forward by the beatjump size.
  - supershift: double beatjump size
- Bottom button 4 (amber): halve loop size
  - shift: beatjump backward by selected size if no loop is enabled. If loop is enabled, move the loop backward by the beatjump size.
  - supershift: halve beatjump size

## Effects

The top part of the column uses the [Standard Effects Mapping](#). Pressing the top encoder acts as the effect focus button. When no effect is focused, the buttons are red. When holding the top encoder to choose an effect to focus, the buttons are green. When an effect is focused, the buttons are amber.

The fader acts as the mix knob.

The bottom buttons assign the effect unit to different input channels and light up red. On the two deck layouts, from top to bottom, they assign the effect unit to deck 1, deck 2, master mix, and headphones. On the four effect unit layouts, they assign the effect unit to decks 1-4 going down the column. You can look down a column to see which decks an effect unit is assigned. You can look across a row to see which effect units are assigned to a deck. When shift is pressed, the bottom two buttons switch to controlling the routing buttons for the master and headphones channels and light up amber.

The bottom encoders are not mapped in the 4 effect unit layout.

## Effect unit focusing

In addition to focusing one effect in a unit at a time with the [Standard Effects Mapping](#), the Xone K2/K1 has another mode for focusing a whole effect unit. This allows for controlling the parameters of all 3 effects in the unit at a time. This mode is only available on the 4 effect unit layouts. To access it, press the Layer button in the bottom left. Press one of the top encoders to choose which effect unit to focus.

In this mode, each horizontal row of knobs and buttons controls one effect. The knobs control the parameters of the effects. Turning any of the knobs with shift loads different effects. The button in the leftmost column controls the enable button of the effect and turns amber when it is on. The rest of the buttons control the button parameters of the effect and turn green when active (not all effects have button parameters). The faders still control the dry/wet knobs and the bottom button grid still controls the routing buttons.

To get back to controlling all 4 effect units, press the Layer button again. The next time Layer is pressed, the effect unit that was focused before will be remembered (but it will not be remembered after restarting Mixxx).

## Troubleshooting

If general functionality or supershift are not working, double check that Latch Layers is turned off and MIDI CH is selected. Refer to the [Xone K2 manual](#) page 12 for details.

## Known Issues

There are a few known issues with the Xone K2/K1 firmware:

- Since there is no way for Mixxx to ask the controller about the positions of all the knobs and faders when Mixxx starts, they are out of sync with the state of Mixxx until they are first moved.
- When the USB connection is interrupted and the controller stays powered on, such as leaving the controller plugged into a powered USB hub and disconnecting the hub from the computer, the controller will not send MIDI input until it is powered down (unplugged) and plugged back in.

If you would like these issues to be fixed, please let [Allen & Heath Support](#) know.

### 17.1.4 American Audio Radius 2000

A CD and USB media player that can control Mixxx over USB with MIDI. It can also play timecode from CDs or USB drives which can control Mixxx when they are connected by a sound card.

- [Manufacturer's product page](#)

Nové ve verzi 1.10.

---

**Poznámka:** Unfortunately a detailed description of this controller mapping is still missing. If you own this controller, please consider [contributing one](#).

---



### 17.1.5 American Audio VMS2

The American Audio VMS2 is a 2-deck all-in-one controller. It is a USB class compliant MIDI and audio device that works with GNU/Linux, Mac OS X, and Windows. It features a 4 channel input and 4 channel output sound card with 2 phono preamps. The main output has XLR and RCA outputs (do not use both at once). There is a separate RCA booth output. The VMS2 can also be used as a stand-alone mixer with analog sources without a computer by setting the USB/Analog switches on the front of the device to “Analog”.

- [Manufacturer’s product page](#), provides manual and drivers for download.
- [Forum thread](#), for discussion of mapping options.
- [Pull Request](#), which this wiki page describes.

Nové ve verzi 1.11.

#### Audio Setup

No driver needs to be installed on GNU/Linux or Mac OS X. However, on Windows, it is recommended to install the driver from the manufacturer to be able to use the *ASIO sound API*.

Unlike most controllers with built in sound cards, which rely on Mixxx to do all mixing in software, the VMS2 mixes signals from the sound card in hardware. To use it with Mixxx:

- Bypass the built-in hardware equalizer of the VMS2 (i.e. use Post-EQ Mode). Hold the forward search button on the right deck as you turn the VMS2 on to switch between Post-EQ and Pre-EQ modes. See section 15 of the [manufacturer’s manual](#) for details.
- Set the USB/Analog switches on the front side of the VMS2 to “USB”
- In Mixxx’s Sound Hardware Preferences:
  - Set the sample rate to 48000 Hz
  - Select channels 1-2 for Deck 1 output
  - Select channels 3-4 for Deck 2 output

*Note:* You cannot use the preview deck to pre-listen in this setup, as Mixxx routes the preview deck to Mixxx’s headphone output, which is not mixed with the Deck 1/2 outputs. You could use a separate soundcard and route the headphone output there to also pre-listen using the preview deck. Of course, if you do so, attach the headphones to that other soundcard instead of the VMS2.

#### Input and Recording

The analog inputs are captured by the built-in soundcard as input signals. They can be used for timecode signals (e.g. timecode vinyl), as the VMS2 also features built-in phono preamps. *FIXME* I did not try that, please verify!

The microphone input is mixed directly into the master output signal of the VMS2 in hardware and cannot be captured through software. If you want to record voice over using the Mixxx software, you will need a different solution. You can use a separate microphone attached to the computer, but that signal will not be routed to the VMS2 and therefore not be on the VMS2 master output (but in the Mixxx recording from the software master/record output).

Mixxx’s Deck 1/2 outputs are affected by the Mixxx software EQs. However, the VMS2’s volume faders and crossfader (as well as the headphone buttons) control the VMS2’s hardware mixer and do not affect Mixxx’s Deck 1 and Deck 2 output signals. Therefore, a recording using the Mixxx software will sound different from what is played through the VMS2 master output, as the crossfader curve and signal mixing are not the same in hardware and software. If you require a recording that captures exactly what the audience will hear, use a separate soundcard and recording software to record from the VMS2 booth output.

## Mapping description

### Mixer section

All main functions are mapped straightforwardly:

VMS2 Control	Mixxx Control
Crossfader	Crossfader
Volume Fader	Volume Fader
Cue (PFL)	PFL
Cue Mix	Cue Mix (PFL / Master)
Headphone Gain	Headphone Gain
Channel Gain	Channel Gain
Master Gain	Master Gain

As noted in the *#Audio Setup* section, these control the VMS2's hardware mixer. Although the state of the controls is reflected on screen in Mixxx, the actual mixing is done by the VMS2's hardware mixer.

### Deck Control

Deck control is straightforward, too:

VMS2 Control	Mixxx Control
Play	Toggle deck play/pause
Pause	Pause the deck
Cue	Cue Point (configure behavior in software)
Pitch +/-	Temporary pitch bend +/-
Pitch Fader	Pitch Fader
Range (Shift+Sync)	Cycle pitch fader range (+-8/10/30/100%)
Sync	Sync to other deck
Search <</>>	Search through currently loaded track
Keylock (Shift+Vinyl)	Toggle pitch independent time stretch (KeyLock)
Vinyl	Toggle between Scratch mode and Pitch mode
Platter	Touch sensitive platters! Scratch or Pitchbend
Shift + Platter	Scroll through library quickly

In Pitch mode, moving the platters from either the top or side will only bend the pitch of the deck. In Scratch mode, the platters act like vinyl turntables and can be used for scratching. Touching and holding them from the top stops the deck. Moving them from the edge does not stop the deck.

### Library and Track loading

Use the encoder in the center of the controller to browse through the library. Press the encoder to switch between library main window and sidebar. Unfortunately there seems to be no way to expand entries in the sidebar through the controller script.

Use the [LOAD] buttons to load the currently selected track into either the left or right deck.

The four directional buttons around the knob also control the library:

Button	Library function
Up	Previous library entry
Down	Next library entry
Left	Previous sidebar entry
Right	Next sidebar entry

If you hold Shift and then rotate a platter, you can scroll through the library much faster.

## Equalizer

The per deck EQ rotaries are mapped to their software counterparts. The VMS2 has no dedicated kill switches for the EQ. However, in Mixxx, pressing the following buttons with Shift acts as EQ kill switches.

VMS2 Control	Mixxx Control
Shift+IN	Kill Switch Low
Shift+OUT	Kill Switch Mid
Shift+RELOOP	Kill Switch High

## Hotcues

The VMS2 can control 6 hotcues per deck.

VMS2 Control	Mixxx Control
1 / 2 / 3	Set/Jump HotCue 1 / 2 / 3
Vinyl + 1 / 2 / 3	Delete HotCue 1 / 2 / 3
4 / 5 / 6 = (Shift + 1 / 2 / 3)	Set/Jump HotCue 4 / 5 / 6
Vinyl + 4 / 5 / 6 = (Vinyl + Shift + 1 / 2 / 3)	Delete HotCue 4 / 5 / 6

Pressing the vinyl/keylock button will not toggle scratch mode or keylock as long as a hotcue is deleted before releasing the vinyl/keylock button.

## Loops

VMS2 Control	Mixxx Control
IN	Mark beginning of loop
OUT	Mark end of loop
RELOOP	Leave / Reenter current loop
LOOP	Start a 4 Beat loop from current position
Smart (Shift + Loop)	Toggle quantize
(:2) / (*2)	Halve or double the current loop length

As Mixxx currently only supports one active loop per deck, the secondary loop controls have been remapped to EQ kill switches.

## 17.1.6 American Audio VMS4/4.1

The original VMS4 has problems with sound quality. The microphone input impedance is off and the output volume is low. The VMS4.1 is the same controller, except that those hardware issues have been fixed. The Traktor edition has different labels on some buttons & knobs, but it's otherwise the same hardware as the VMS4.1.

- VMS4
- VMS4.1
- VMS4.1 Traktor edition

Nové ve verzi 1.9.

### Setup

---

**Poznámka:** Mixxx expects the VMS4 to be set to “Post EQ” mode for best sound quality. Do this by holding down the Headphone Cue button on Midilog 4 while powering on the unit. You only need to do this once. (Each time you do, it changes the mode back and forth.) Consult the [user manual](#) for more information.

To check the status of this in Linux, at a console, issue the command `lsusb -v|grep 'iSerial\|iProduct'` and look at the serial number under the VMS4 device per the instructions in the user manual. As of this writing, the leading digit should be **1**.

---

1. Make sure the VMS4 is off
2. Slide the switch on the front of the VMS4 to “8 OUT”
  - If you're using vinyl control or aux devices (or Mixxx 1.11 & below) set the switch to “4 OUT” for 2-deck output and 2-deck input
3. Turn on the unit (and plug in the USB cable if you haven't yet)
4. Start Mixxx
5. Open Preferences
6. Click Sound Hardware. In the right pane:
  1. Set the sample rate to **44100Hz**
  2. Set the Master output to **None**
  3. Set the Headphone output to **None**
  4. Set the Deck 1 output to the **VMS4** device and **Channel 3-4** (may show as “USB Audio Device” on Windows)
  5. Set the Deck 2 output to the **VMS4** device and **Channel 5-6**
  6. Set the Deck 3 output to the **VMS4** device and **Channel 1-2**
  7. Set the Deck 4 output to the **VMS4** device and **Channel 7-8**
    - If you're using 4 OUT mode, (for vinyl control/aux input or Mixxx 1.11 & below):
      1. Set the Deck 1 output to the **VMS4** device and **Channel 1-2**
      2. Set the Deck 2 output to the **VMS4** device and **Channel 3-4**
      3. For vinyl control, set the Vinyl Control 1 input to the **VMS4** device and **Channel 1-2**, connect a turntable to Midilog 1, and set it to Analog.

4. For vinyl control, set the Vinyl Control 2 input to the **VMS4** device and **Channel 3-4**, connect a turntable to Midilog 4, and set it to Analog.
7. Plug your headphones into the VMS4's jack on the front. You will use the VMS4's CUE buttons and knobs for headphone control.
8. Still in the Preferences, expand "Controllers" on the left
9. Select the "VMS4 MIDI" device (may show as "USB Audio Device" on Windows)
  - Do not choose the the HID one. That's for the little mouse pad and button area.
10. Click the Enable checkbox in the right pane
11. Click the drop-down and choose the *American Audio VMS4* preset
12. Click OK and the controller should light up. (In 1.9.x, the controller will light up when you load a track to a deck.)
13. Continue reading below to know how everything is mapped

## Direct deck output implications

This setup uses direct deck outputs because mixing takes place in the VMS4 hardware. The down side to this is that you won't be able to use Mixxx features that play to Master or Cue buses like the Preview Deck or Samplers. Using these features with a hardware + MIDI mixer like the VMS4 is not really feasible unless you used a separate sound interface or controlled Mixxx's internal mixer another way.

## Control

This guide explains how the VMS4 is mapped by default in Mixxx.

**Easy customization:** We have provided the following customization variables at the top of the script you can set to your liking:

1. **RateRanges:** Set the pitch slider range each time you toggle it with Shift+Sync.

Just open the `midi/American-Audio-VMS4-scripts.js` file in your favorite text editor (Wordpad works too) and you'll see these variables right near the top. Edit & save, then restart Mixxx and enjoy.

## Deck controls

---

**Poznámka:** There are a number of different face plates in the wild so these images and control descriptions may not exactly match yours. The locations of the controls are the key things to pay attention to.

---

*The controls are the same on both sides of the controller. The left side controls Deck 1 and the right side controls Deck 2.*

- **Sync/Range** button - Changes the BPM of this deck to match that of the other. When shifted, toggles the pitch slider range. (See top of page to customize.)
- **Hot cue buttons** - Press to set or recall a hot cue. The buttons light up red when one is set. Hold Shift and press to access hot cues 5-8. (The buttons light up blue for these.) Hold Vinyl and press to delete the cue. (Note: there is a bug in some firmware versions that causes hot cues 5 and 6 to be deleted simultaneously.) the cue.
- **Pitch Bend +/-**
- Momentarily speeds up or slows down the deck while the button is held down.
- Hold Shift and press to adjust the key (pitch) of the song (independent of the tempo.)

- **Loop In** - Set the in point of a loop
- **Loop Out** - Set the out point of a loop
- **Reloop** - Toggle a previously-set loop. Lights red when a loop is active.
- **Loop/Smart** Start a 4-beat loop. Hold Shift and press to toggle quantization (locking to the nearest beat.)
- **Vinyl/Keylock** - (*Does nothing on its own when un-shifted at the moment.*) When shifted, toggles key lock.
- **<< Search/ /2** - Fast-rewind Hold Shift and press to halve the current loop length
- **Search >>/ \*2** - Fast-forward Hold Shift and press to double the current loop length
- **Touch strip** Scroll through the respective Library panes. (Left for the folders/crates list, right for the track list.) Hold Shift and touch: - to affect the Dry/Wet knob on the respective effect unit while the deck is playing - for Needle drop - search through the track while the deck is stopped
- **Wheel** Move the wheel while touching the top to scratch the current track like a vinyl record Move the wheel without touching the top (so on the sides) to perform a temporary pitch bend
- **CUE** - Operates according to the Cue mode set in Mixxx's preferences Hold Shift and press to play immediately from the cue point (known as Cue+Play or CUP.)
- **Play** - Press to toggle deck playback Hold Shift and press to set this deck as the master for syncing with another
- **Pause** - Stops the deck playback

## Effects section

The left side controls effect unit 1 and the right side controls effect unit 2.

- **Select knob**
  - Rotate to choose the effect chain preset
  - Press to toggle whether the effect unit is on or not
- **Control knob**
  - Adjusts the wet/dry ratio
  - When Parameter is on, this becomes the Wonder Knob, adjusting all effect parameters at once
  - **On/Off** - Toggles the effect for the deck matching the effect unit number
  - **Parameter** - Toggles what the above Control knob adjusts

## Sample section

The left side of the controller controls Sampler 1, and the right, Sampler 2.

- **Select knob** Rotate to move the highlight in the library. Press to load the currently highlighted track into the sampler. Hold Shift and press to eject the current track from the sampler (when the sampler is not playing.)
- **Volume knob** - Adjusts the volume of the sampler
- **Play** Press to play the sample from the beginning. Press while playing for a stutter-play effect (play again from the beginning.) Hold Shift and press to stop playing.
- **Rec** - *Currently does nothing*

## Mixer controls

The volume sliders don't control the ones in Mixxx because direct Deck outputs are affected by Mixxx's internal ones.

- **Midilog 1:** This strip controls Deck 3/C
- **Midilog 2:** This strip controls Deck 1/A
- **Midilog 3:** This strip controls Deck 2/B
- **Midilog 4:** This strip controls Deck 4/D

### 17.1.7 Behringer BCD2000

The Behringer B-Control DeeJay BCD2000 features a USB 4-channel audio interface plus 24-bit A/D and D/A converters.

This controller has been discontinued. The *Behringer BCD3000* is its successor.

- [Manufacturer's product page](#)
- [Forum thread](#)

Nové ve verzi 1.11.

---

**Poznámka:** Unfortunately a detailed description of this controller mapping is still missing. If you own this controller, please consider [contributing one](#).

---

### 17.1.8 Behringer BCD3000

- [Manufacturer's page](#)

The B-Control DeeJay BCD3000 controller is a 4-channel audio interface. It offers a fully fitted controller console with 24-bit A/D and D/A converters, a USB audio interface with a microphone preamplifier, a 3-band kill EQ for each channel, faders, a crossfader and talkover function.

By default, the input channels 1-2 use the RCA phono inputs for the source signal. On Windows, this can be switched to the microphone in the ASIO driver control panel. On GNU/Linux and Mac OS X, a [particular MIDI message must be sent](#) to switch channels 1-2 to the microphone input.

Nové ve verzi 1.6.

---

**Poznámka:** Unfortunately a detailed description of this controller mapping is still missing. If you own this controller, please consider [contributing one](#).

---

### 17.1.9 Behringer BCR2000

- [Manufacturer's product page](#)
- [Forum thread](#)

The B-CONTROL BCR2000 is a general-purpose *USB MIDI* controller containing 20 buttons, 24 rotary encoders and 8 push encoders. Every control is backed by LEDs that show the current value. As a pure MIDI controller it contains no interfaces for audio or microphones.

The controller can be configured extensively. It offers 32 presets that may be programmed by the user.

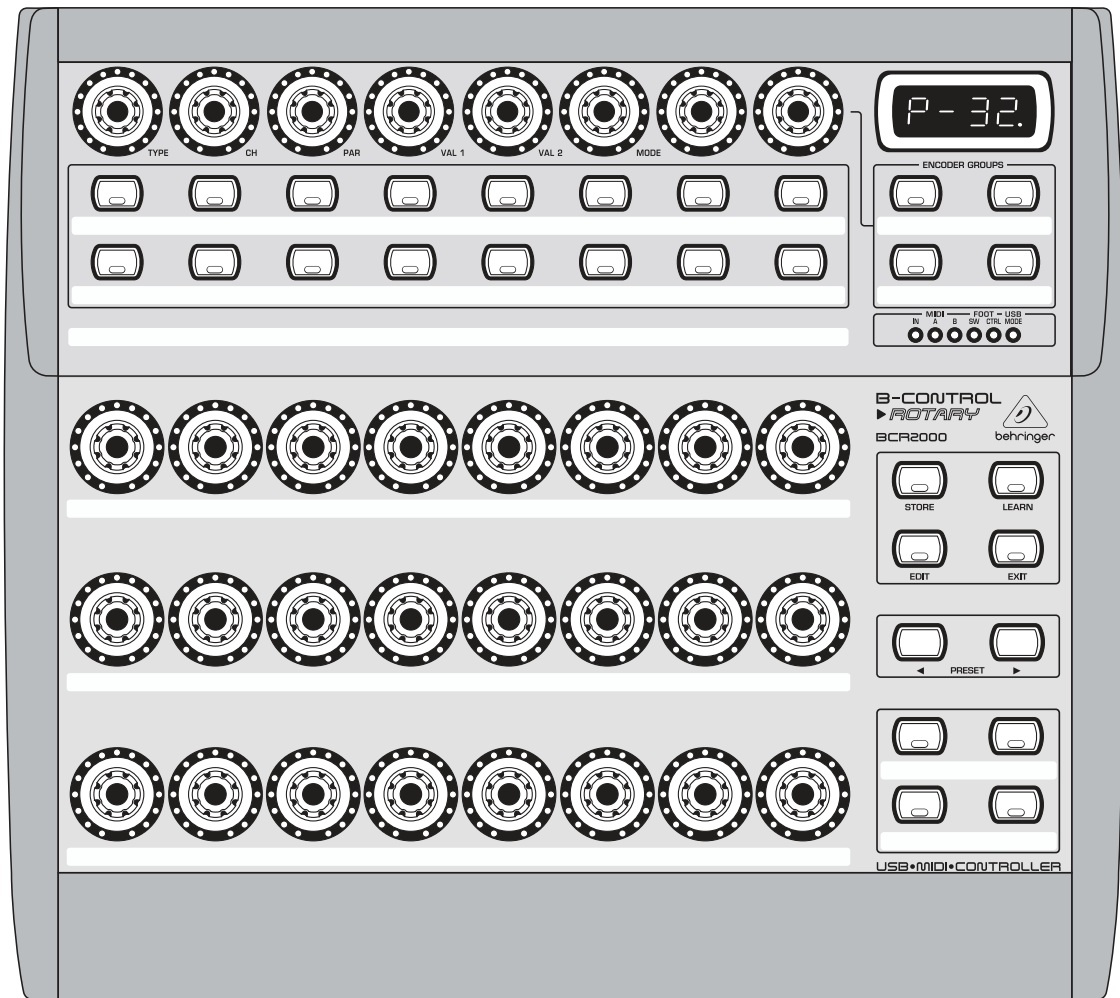


Figure1: Behringer BCR2000 (schematic view)