

Projects on the Move



Denis Tevekov, 123RF



Denis Tevekov, 123RF

DJs don't need expensive decks now that Mixxx offers a competitive computer-based alternative. The Liquidsoap programming language provides a fully automated approach to generating flexible streams.

By Carsten Schnober and Heike Jurzik

Users of Mixxx [1] software can make their dance floor audience happy with the virtual DJ deck that the free project offers (Figure 1). But keeping a crowd happy on the dance floor means doing more than just playing loud music; you have to set up the right track on the deck at the right time, cross-fading it nicely with the previous track without interrupting the groove. To do this perfectly, you have to match the beats of the two tracks.

Traditionally, the original equipment for disk jockeying was two turntables, a mixing desk, and a mike. With these, DJs could manually adjust the speed of the records they played. Later, techniques like scratching were added to integrate the sound of the needle on the record with the beat of the music. Today's more sophisticated decks have three or four audio outputs, so DJs can cue the next track via headphones while the music continues to play on the dance floor.

Two Turntables and a Microphone

New technologies have since conquered the world of the DJ, and today, many DJs use software to supplement or completely replace traditional decks or CD players, which have also made inroads into the market. This software offers a couple of advantages: Gone are the days of lugging around heavy turntables or cases full of records and CDs; you can even do without turntables. Instead DJs now use laptops with a well-sorted collection of MP3s and a piece of software.

Some DJs don't like to sacrifice any options offered by the real thing, even with a virtual deck. This desire prompted early developers of Mixxx, Ken and Tue Haste Andersen (both of whom are amateur DJs), to code the most important elements into their Java program. Work on the software mixer began in 2001 when very few tools of this kind existed, either as free or proprietary software. Since then, the two developers have stopped working on the project and handed it over to the community under

the GPLv2 license. The project managers and maintainers are now Adam Davison and Albert Santoni, who head a large team.

Mixxx uses four virtual channels – two for loudspeakers and two for headphones – to emulate legacy record decks. DJs no longer need to cue records, though; instead, they select digitized tracks and mix them for output to their audience's taste. Mixxx channels, which are individually configurable, can also serve as input channels for MIDI controllers or physical turntables, so the tool is also an excellent choice if you need a mixing deck (Figure 2).

Mixxx also supports DJ control hardware. These devices emulate turntables both visually and with respect to their controls, although they actually control the output of digitized tracks. If you wire up a device of this kind to your own computer, you can use it to control Mixxx, rather than relying on the mouse and the keyboard.

Featuring ...

Newbies to the world of virtual disk-jockeying will find it easy to get started with Mixxx. The beginner's task of playing one track after another is easily mastered, then you can explore other features one by one, such as cross-fading between tracks. The hot cue function lets you tag the specific spot in a track to fade in or out.

Advanced users will enjoy the sound wave display and the BPM meter, which helps achieve that perfect cross-fade. As the name suggests, the meter shows the beats per minute count, a vital criterion in the world of electronic music. This feature lets you choose the next tracks on the basis of identical BPM counts. At the same time, Mixxx has a function that changes the speed of a track without changing its key, which would happen otherwise according to the laws of acoustics. However, these subtle changes to the original music are exactly what the DJ's artistry is all about, so the program also offers a vinyl emulations mode that changes the speed and key at the same time.

Another important element that DJs have used for years is the loop. Loops repeat certain sections of a song, known as samples, that can be mixed into other songs. Mixxx can help you do this. Click-

ing *IN* marks the start, whereas *OUT* marks the end of the loop.

Mixxx then repeats the loop until the DJ stops it. Loops can also be manipulated like any other output.

This Is the End

If Mixxx reaches the end of a track without intervention by the DJ, the program identifies this as the default behavior for the track next time it is played. Your options here are *Stop* (stops playing at the end of the track), *Loop* (plays the track again), and *Next* (which goes to the next entry in the playlist).

The integrated mixer also has basic functions like a fader for smooth cross-fading and an equalizer that adjusts the volume to reflect the sound frequency. Thanks to its plugin infrastructure, Mixxx is infinitely extensible, so tracks that use exotic coding or new hardware don't need to be insurmountable obstacles.

If you enjoy using Mixxx and would like to contribute to the development of the mixer or add-ons, the contact info (IRC channel and mailing lists) is listed on the project's website.

Radio DJ

All over the world, the powers that be have tried to clamp down on web radio with royalties, checks, and all kinds of bureaucratic obstacles. Despite all these efforts, popular private stations continue to flourish. Music fans stream their favorite tracks into the ether, whereas Internet radio stations are more like spoken blogs, from which the makers vocalize their thoughts on various topics.

Whatever the content, a web stream always comprises three components from a technical point of view: the audio

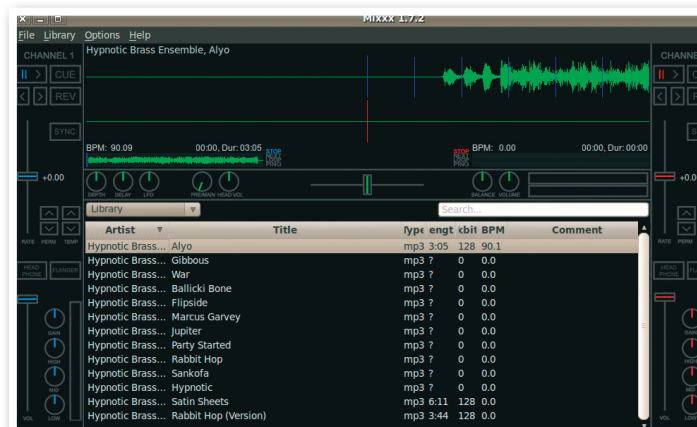


Figure 1: If you prefer to avoid lugging tons of vinyl and special record decks around with you, Mixxx gives you a set of virtual DJ equipment with all the modern conveniences.

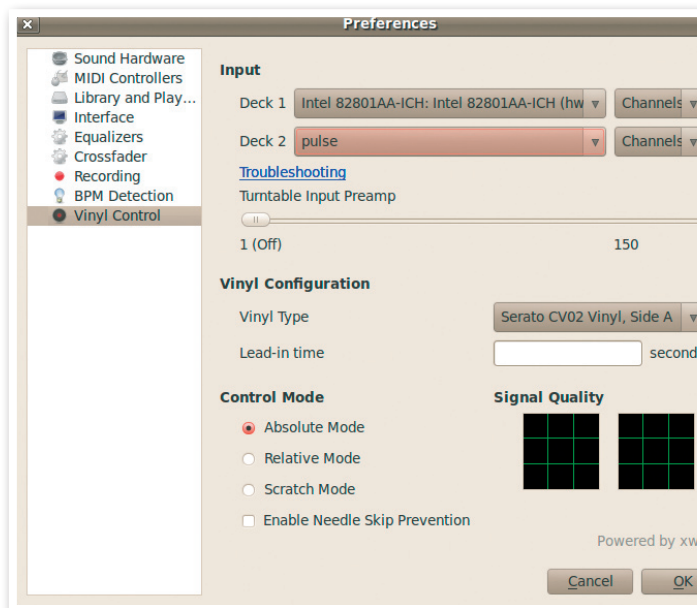


Figure 2: Mixxx not only plays tracks off the local machine, it also accesses external record decks and other devices.



Figure 3: The Liquidsoap graphical developer environment generates programs for the Liquidsoap scripting language.

data provided by some source, a streaming server that accepts this data and serves it up on the Internet, and the client on the listening end that accesses the data and typically outputs it via the computer's sound card. Icecast [2] and Shoutcast [3] are two of the most popular open source streaming servers. And the client-side market is huge, with more or

less every desktop now offering integrated player software capable of playing streams.

Liquidsoap [4] facilitates the task of putting together web streams from audio files. The software can combine a number of input sources, such as local files and network resources, and build the ready-to-broadcast audio stream from them.

The program then sends the stream to a server, such as Icecast. The whole thing is based on a special, custom scripting language created by the developers of Liquidsoap for this purpose.

This approach gives the stream provider the freedom to implement the required features individually with minimal overhead, instead of resorting to existing solutions (with all of their restrictions).

At the Source

A typical line in a Liquidsoap script references the Source; the resulting stream is typically compiled from a single source object (e.g., a single track on the local hard disk), a playlist, or a sound file on the network. Additionally, output commands push the stream to a streaming server.

Alternative output options include devices such as sound cards, although this capability is not commonly required in the web radio field.

The Icecast server supports multiple streams, and Liquidsoap takes advantage of this feature. Home DJs can send sound objects to different streams ac-

ording to metadata – such as the genre or artist – and modify technical details – such as the bitrate or the format – as needed. A basic command line creates a sound object from a single local Ogg Vorbis file:

```
single("file.ogg")
```

The user can store this item in a variable and then output it via the `output.icecast()` function. The call controls a local or network-based Icecast server. Other functions let you configure a fallback option in case a problem occurs when accessing resources. Stream programmers can use the Boolean `fallible` tag to indicate that access is uncertain. Liquidsoap also handles issues that occur above the network layer. For example, if the program is only receiving silence from a source, it automatically enables a fallback option.

Interface

The Liquidsoap developers point to the benefits their flexibly programmable solution offers compared with a static interface. Although a static interface might be easier to use, it would restrict the user in other areas.

If you prefer not to do without a GUI, check out LiGUIdsoap [4]. Liquidsoap is a graphical development environment that lets users click Liquidsoap scripts together (Figure 3).

The Liquidsoap GUI provides buttons for the most frequently used Liquidsoap commands. Users can open them to define the parameters in a separate window, which means you can configure a streaming server for output or create a playlist with your favorite tracks. If necessary, the GUI tool will also launch a local Icecast server. Although the GUI doesn't support the full feature scope of Liquidsoap, it does give users rapid access to the material and helps newbies set up their own point-and-click web radio.

Liquidsoap is implemented in the Objective Caml (OCaml) [5] language developed by the French research institute, Inria. Numerous plugins offer interfaces for embedding in websites with Django, Perl, PHP, Plone, and more.

If you are interested in the latest development, check out the project's bug tracker [6]. ■■■

INFO

[1] Mixxx: <http://www.mixxx.org>

[2] Icecast: <http://www.icecast.org>

[3] Shoutcast:
<http://www.shoutcast.com>

[4] Liquidsoap and LiguIdsoap:
<http://savonet.sourceforge.net>

[5] The Caml Language:
<http://caml.inria.fr>

[6] Liquidsoap bug tracker: <http://dev.sourceforge.org/browse/LS>