Converting and modifying audio files

FORMAT ARTISTS

Your new media player might match your new outfit, but does it use the audio files formats you have? SoundKonverter helps you easily reformat your music collection. **BY JAN RÄHM**

The Qt-based tool has a relative in the Gtk camp: SoundConverter [2], with a 'C'. A quick glance at the two applications reveals that the Qt-based software is far superior to its Gtk-based counterpart with respect to its options and usability. SoundConverter only supports four formats and it does not have granular options for customizing the conversion process.

The number of formats that soundKonverter supports depends directly on the number of encoders and decoders you install. If your system has a full set of available encoders, the list of decodable audio and video formats has more than 30 entries, including some proprietary types; soundKonverter will encode some 20 formats.

After installing the libraries for tag support, the tool also parses the track information on conversion and converts the tags into matching entries for the new format, assuming the new format supports them. The documentation that accompanies the source code package [3] includes a list of supported back ends and libraries. The "Installation" box has more details on just that.

After you complete the installation, there is nothing to keep you from launching the program. SoundKonverter comes up with a clearly structured interface. Your first steps should take you to the Settings menu, where you will find four entries and a bunch of options. In *General*, you will probably want to configure the *Process priority of the backends* and the *Number of files to convert at once*. This has a big influence on the tool's performance.

If you have a powerful system, you can leave the priority setting as is: *Normal*. If you have an older and less powerful computer, you might prefer to change the priority to *Low*; otherwise,

Sound-Konverter [1] is the Swiss Army Knife of digital audio data conversion in the free software world. The program provides a graphical front end for numerous command-line tools, giving users an easy option for controlling various encoders and decoders and a number of optimization tools – such as MP3Gain or AACGain – for volume modification. Useful conversion options round off the feature set. Plus, soundKonverter is a useful ripping tool for audio CDs.

Installation

If you use a package manager, installing soundKonverter should be fairly trivial. Pre-built packages are available for openSUSE, Fedora, Mandriva, Ubuntu, and derivates such as Slackware. The package manager will help you install the required and optional back ends and plugins.

A manual install from the source code [3] is far more complex. Manually building the software means resolving a number of dependencies up front, and you are advised to install a complete KDE desktop. On top of this, you need all the back ends and tools you want to include; *taglib-devel* and Cdparanoia are mandatory. The developers also point out that for some systems, such as Ark Linux, you need to remove the *cdparanoia check* entry from the *configure.in.in* file. For more information on the installation process, read the *README* file in the source code package.
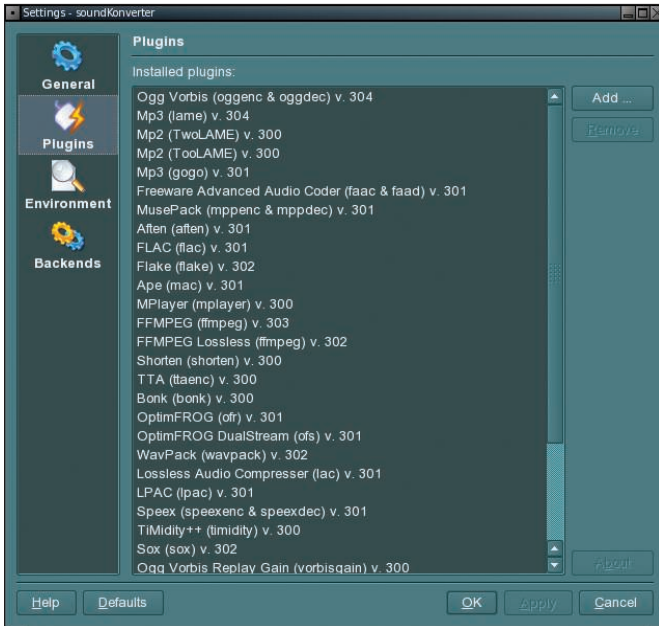
Figure 1: The soundKonverter Plugins dialog shows you the available plugins.
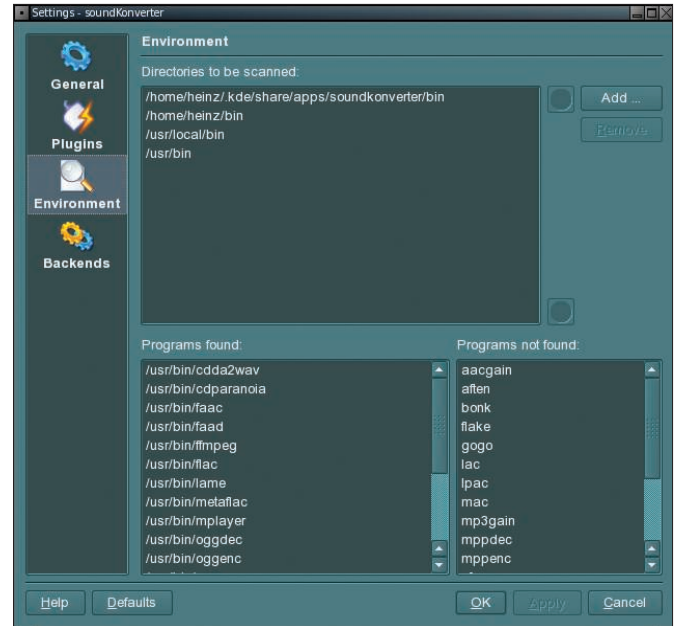

Figure 2: The Environment dialog lists the resources used and those that weren't used.

the system could respond very slowly during the conversion process.

For best results, it is a good idea to set the number of files to be converted simultaneously to one file for each processor core.

Encoders process the data in a linear manner from start to finish. Tools such as LAME will only process one file per core and even run on a multi-processor system. On the other hand, with other approaches, encoders use multiple cores for each process [4]. Unfortunately, this caused problems in our lab.

The Plugins menu tells you which plugins the tool can access (Figure 1). The *Add* button lets you add more plugins. The *Environment* sub-item lets you specify the search path for back ends and plugins. In the lower part of the window, soundKonverter lists the resources it uses and those it doesn't (Figure 2). The fourth menu item lets you decide which back ends to use for encoding and decoding of each format (Figure 3).

SoundKonverter has three file conversion modes, starting with a *Simple* mode, in which you select the source and target folders, the desired target format, and the quality. In *Detailed* mode, you also have the option of specifying the bit rate to use within the encoder's range.

With *Advanced Options*, you can modify the options available to each encoder at the command line. SoundKonverter thus lets users specify a full set of commands for the conversion process.

No matter which mode you opt for, after you click *Start*, soundKonverter will launch into the file conversion process. At the bottom edge of the screen, you can see the progress, the current data throughput, and the remaining time.

Conclusions

If you need to convert audio files from one format to another quickly and with little interaction, consider soundKonverter. New users should be able to find their way around without a steep learning curve.

As long as the source files are not copy protected, the tool works very reliably. If soundKonverter fails to identify a format, you can quickly resolve the issue by adding the appropriate back ends before going on to process the file in the desired way. ∎
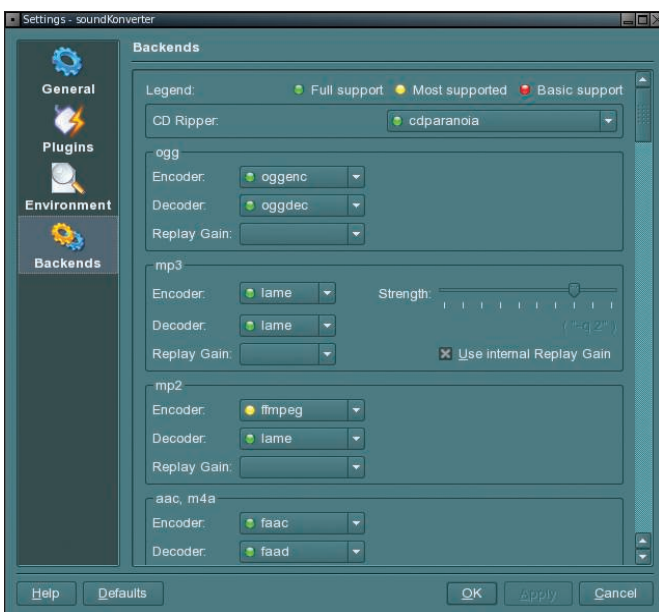

Figure 3: For each format, you can specify which back end to use for encoding and decoding.

INFO
[1] soundKonverter: *http://www.kde-apps.org/content/ show.php?content=29024*
[2] SoundConverter: *http://soundconverter.berlios.de/*
[3] Source code package: *http://www.kde-apps.org/content/ download.php?content=29024& id=1&tan=25499450*
[4] Multi-threaded LAME engine: *http://softlab.technion.ac.il/project/ LAME/html/lame.html*