Roundup

» Every month we compare tons of software so you don't have to!

Audio codecs

Daniel James looks at the options for long-term storage of your music collection. Can you fit it all in the space available, and still get good quality?

How we tested...

Key factors in our test included audio quality and time taken for encoding. Most modern PCs will have no trouble decoding these formats in real time, which allows for skip-free playback directly from the compressed file. A faster processor or multiple CPU cores should make an obvious improvement in encoding time, particularly if you have a lot of material to work on.

Three different instrumental tracks were used for the encoding tests, to balance the effect that the sonic content of material has on compressed file size and quality.

» GNU/Linux box specification

- Tyan S2875 motherboard.
- Dual Opteron 240 CPUs.
- 1GB OCZ registered RAM
- Two Seagate 7200RPM disks (80GB root, 200GB /home).
- On-board AC97 audio chipset.
- 64 Studio 2.0 AMD64 distro.

Our selection

During the 1980s, research into digital audio broadcasting was funded by the European Union's Eureka initiative. Lossy compression techniques were developed that enabled large portions of audio data to be thrown away, and yet leave a listenable result. These techniques found their way into the MPEG-1 Layer III standard, which became known as MP3.

The MP3 lossy format was an obvious candidate for the internet, since it could achieve file compression of around 10:1 (from the original WAV size), without sounding too nasty. MP3 quickly became the *de facto* standard for internet audio, and had things gone differently, it might have remained so. The problem was that some of the research organisations that had worked on MPEG encoders and decoders (abbreviated to 'codecs') decided to cash in on the internet boom of the late 1990s by claiming patent royalties on the formats, including MP3. That move can be held responsible for creating the fractured media landscape that we inherit today.

Rather than simply give in to the patent holders' demands and cough up the cash in perpetuity, many software companies opted to create alternatives to MP3. Real Audio took an early lead in the streaming market, while MP3 remained popular for static downloads. Microsoft and Apple created their own proprietary audio formats, not only to avoid paying royalties on MP3, but to generate codec patent income of their own, and lever in DRM restrictions at the same time.

The upshot of all this is that most proprietary media formats aren't supported on Linux out of the box. But that doesn't mean you can't use them, because with a bit of work you can listen to most audio formats out there on the web. In this roundup, we're going to look at the codec question from the point of view of a user who is choosing a format for their own music collection.

MP3

The best-known of all audio codecs, but is it still any use?

In September 1998, Germany's publicly owned and funded Fraunhofer-Gesellschaft research organisation sent letters to many MP3 software developers, demanding licence fees be paid. Despite the fact that research in this area was originally funded by the taxpayers of the European Union, several organisations came together under the banner of **www.mp3licensing.com** to collect royalties from every significant user of the technology.

A number of other organisations also claim patents on the MP3 format, as Microsoft found to its cost when it was sued by Alcatel-Lucent in 2006. Microsoft was ordered to pay US $1.52 billion in damages to Alcatel-Lucent in February 2007. The Redmond company is currently appealing against the judgement, on the grounds that it had already paid millions of dollars to Fraunhofer for the right to use MP3 in its products.

Our MP3 test began with *Lame 3.97* (because of the developer's fear of a punitive patent lawsuit, the recursive acronym stands for *Lame* Ain't an MP3 Encoder). Despite the name, once you've built the binary, you can encode valid MP3s with it, at bit rates from 32 to 320kbps. Constant bit rate MP3s are sometimes required for backwards compatibility, but variable bit rate encoding is more efficient when it comes to reducing file size. This is because more data can be thrown away for some parts of the audio file, without impacting quality too much.

Bit rate race

The general consensus is that below about 100kbps, MP3 compression starts to have an effect which is too obvious, and it's perhaps for this reason that the default encoding bit rate in *Lame* is 128kbps. However, using *Lame* on the command line presents the user with a great variety of options to adjust bit rate and quality. You can also specify different algorithms to speed up encoding at the expense of audio quality.

Most of these options are hidden when you use GUI front-ends for CD ripping and encoding, so it's well worth checking out the man page if you need to tweak your MP3s to the max. Fortunately, there are some quality presets available for encoding, including 'standard', 'extreme', and 'insane'. The insane mode encodes the audio file without regard to size, and since the whole point of MP3 is to bring file size down, it is aptly named.

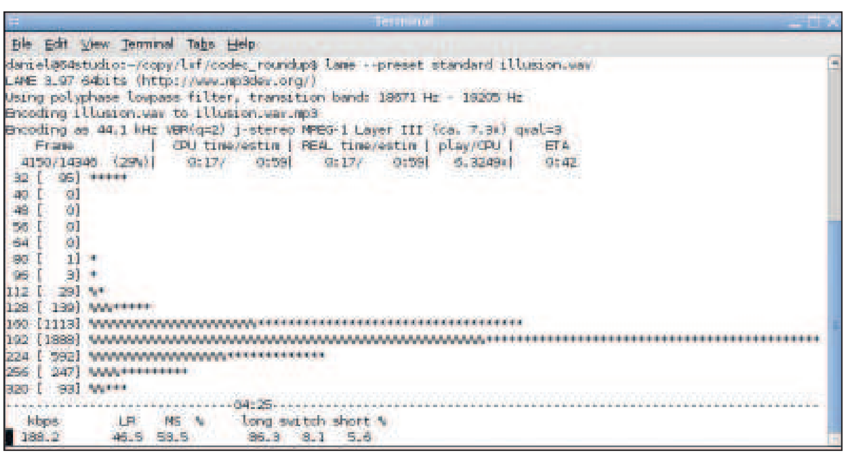
In this test, *Lame* was used on the command line, with the **--preset standard** option. This equated to a modal average bit rate of between 160kbps and 192kbps, depending on the material encoded. Using this preset, *Lame* draws a graph in the terminal or console to show how the bit rate varies during the encoding stage.

> **"If you're determined to use MP3, Lame is the tool to encode it with."**

Not so lame after all

The resulting MP3 files were played back using the LGPL'd *Totem* and the Fluendo plugin for MP3 support in *GStreamer*. This plugin is not free software, but it is free as in beer and can be downloaded from the **https://shop.fluendo.com** site. Unlike the more commonly used *libmad* decoder, Fluendo's plugin is patent-licensed, which means that it can be used in commercial products with some defence against legal threats.

Subjectively, the MP3 files sounded good enough for everyday, casual listening, as you would expect at this bit rate. Compatibility would have to be tested against a range of hardware MP3 devices before settling on this preset for encoding your music collection, because of the use of the variable bit rate method. But if you're determined to use MP3, *Lame* is the tool to encode it with.



› **The FLAC encoder is faster than *Oggenc* or *Lame*, taking care of an almost four-minute file in just six seconds.**

The stats: MP3

Name	Illusion.wav	Square.wav	Unobtanium.wav
WAV file size (MB)	63.0	45.2	39.5
Encoding time (minutes:seconds)	1:04	0:43	0:37
MP3 file size (MB)	8.1	5.1	5.4
Compression ratio (approximate)	8:1	9:1	7:1

MP3 generates reasonably small files, but at the cost of audio quality.

LINUX FORMAT Verdict

MP3

Version: Lame 3.97
Web: http://lame.sourceforge.net/
Price: Free software under the GNU LGPL

» *MP3 may be the grandad of online audio formats, but the backwards compatibility can't be beaten.*

Rating 7/10

LXF104.round 31 19/2/08 11:41:19

Vorbis

The original Free Software contender for MP3's crown?

The Vorbis codec was created as a direct result of the nasty-gram that arrived on the doormat of MP3 software projects in 1998. A plucky band of Free Software developers decided that codecs were a basic part of the Internet, and reasoned that access to this technology should not be controlled by corporate interests. Under the umbrella of **www.xiph.org** they created a set of Free Software codecs that could be used by anyone without having to pay patent royalties, including the lossy audio codec Vorbis. The Xiph project became an official not-for-profit foundation, similar to the Mozilla Foundation, which produces *Firefox*.

The Vorbis codec is almost always used in an Ogg file container, which is why many users assume that Ogg Vorbis is the name of the codec itself. In fact, an Ogg container can contain any one of a number of Free Software codecs. One of the strengths of Vorbis is that subjectively it sounds better than MP3 at lower bit rates. That amounts to saying that at the same file size, Vorbis will sound better than MP3. But if we're encoding our own music collection, we're unlikely to be using bit rates below 100kbps, because hard disks and flash storage are pretty cheap now. Currently, 500GB SATA II drives work out at under 15 pence per gigabyte, and 4GB of Flash memory on a USB stick can be had for well under £25, so there's no need to sacrifice quality unless resources are very constrained.

Oggenc is the command line encoder that appears in the **vorbis-tools** package of most GNU/Linux distros. As with *Lame*, there are any number of CD-ripping GUI front-ends that can make the encoding process more user friendly. The Gnome desktop includes the *Sound Juicer* application, which has support for encoding your CDs to Vorbis as standard.

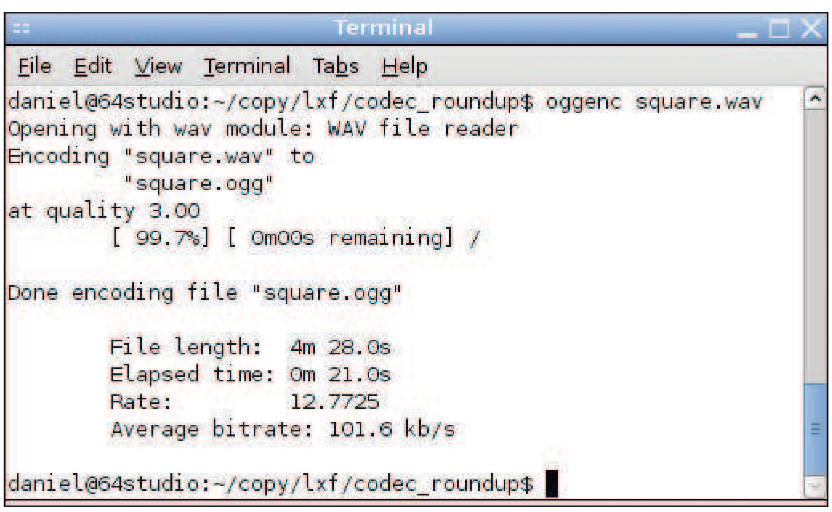
Burn faster

Using *Oggenc*, quality presets are available from zero to 10, where 10 is the highest quality. Material encoded at quality zero sounds surprisingly good, despite its low bit rate, but as we're testing codecs for long-term use, we're going to use the default setting of three. One drawback to using lossy codecs is that when the data is gone, it's gone for good, unless you keep the source material around forever, which would defeat the purpose of encoding the collection in the first place.

The Vorbis files were played back-to-back with the MP3 versions in *Totem*, using the *GStreamer 0.10* plugins to decode Ogg files. Subjectively, there was little to tell the MP3 and Vorbis files apart. This was despite the fact that the average bit rate of the Vorbis files was between 102kbps and 107kbps – considerably lower than that of their MP3 counterparts. The difference in bit rate explains the difference in file size, with the Vorbis files being between 35% and 43% smaller. More surprisingly, the Vorbis encoding time per file was only about half that for the MP3 equivalents.

> **"Weirdly, many cheap unbranded 'MP3' players have Vorbis support."**

These significant performance advantages over MP3 reflect the fact that Vorbis is a more modern codec, but it's only fair to point out that Vorbis does not enjoy the breadth of support on portable, home and in-car players that MP3 does. Having said that, there are now dozens of Flash and hard disk-based players on the market which do support Vorbis, listed at the **http://wiki.xiph.org/VorbisHardware** site. Weirdly, many cheap unbranded 'MP3' and 'MP4' players have Vorbis support, even though they do not mention this on the packaging or in the manual. There are also a few products where Vorbis support was originally available, but has been removed from versions of the firmware in later models, so you need to do some research before ordering products.



› *Oggenc* is faster than *Lame*, encoding this four-and-a-half-minute file in just 21 seconds.

The stats: Vorbis

Name	Illusion.wav	Square.wav	Unobtanium.wav
WAV file size (MB)	63.0	45.2	39.5
Encoding time (minutes:seconds)	01:29	0:21	0:19
Vorbis file size (MB)	4.7	3.3	3.1
Compression ratio (approximate)	13:1	14:1	13:1

Super-quick and extraordinarily high compression makes Vorbis a top codec.

Linux Format Verdict

Vorbis

Version: libvorbis 1.1.2.
Web: www.vorbis.com
Price: Free under BSD-style licence

» *Vorbis clearly beats MP3 when it comes to compression and speed, but hardware support isn't as widespread.*

Rating 9/10

FLAC

Lossless compression in a Free Software format?

FLAC is different from the other codecs on test, because it is designed to be lossless – the uncompressed file should be identical to the original WAV. It's more like *gzip* than a traditional audio compressor, except that the Free Lossless Audio Codec is specialised for audio, so it achieves much smaller output files than *gzip* in the kind of test we're doing.

For instance, running *gzip* on the 63MB file **illusion.wav** results in a zipped file of 55.8MB, where as encoding the same WAV file with default FLAC options produces an output file of 31.4MB. When using FLAC, the trade-off is between output file size and encoding time; using the **--fast** switch is equivalent to a compression-level setting of **-0**, and the **--best** switch is equivalent to a compression-level setting of **-8**. Using the **--best** option, we can just about shave another 0.2MB from the file, but encoding takes fifty-one seconds, instead of just nine seconds with the default compression setting of **-5**.

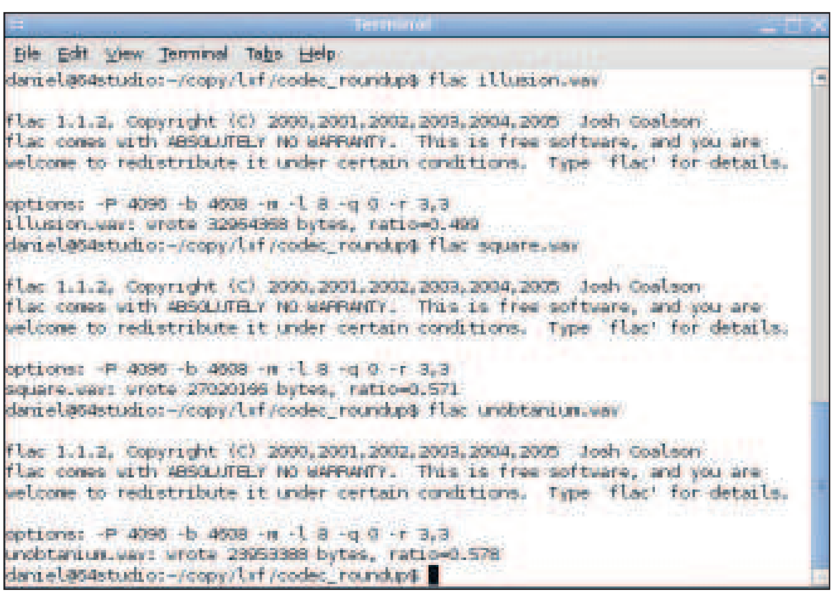
Another advantage of FLAC over general file compression methods is that some newer audio hardware is able to play FLAC files directly – see **http://flac.sourceforge.net/links.html#hardware** for a list of products available. Now that broadband internet connections are more common, and the extra megabytes aren't such a big deal, a few online music stores offer FLAC format downloads. Soft-rock superstars the Eagles released their latest album, *Long Road Out Of Eden*, in FLAC format; they even charge a dollar more for it, compared to the MP3 256kbps version.

A flask of Ogg

In January 2003, the FLAC project joined the Xiph Foundation; you can now specify on the command line that you would like to have your FLAC file wrapped in an Ogg container. This produces an output file with the extension **.ogg**, but because the codec used is just the same, you can decode Ogg FLAC files with the native FLAC decoder.

In the subjective test, FLAC files were played back using *Totem* with the *GStreamer* plugin. On this equipment at least, with this material, it's hard to hear much difference between the FLAC and the Vorbis or MP3 file. The FLAC file is over six times larger than the Vorbis file of the same material, using the default encoding options in both cases. But it's good to know that none of the audio data is being thrown away, making FLAC more suitable for long-term archival of material, or high-quality playback systems. At least if you keep the FLAC file, you have the option to make a smaller lossy Vorbis file later, whereas doing the conversion the other way round produces no sonic benefit.

> **"It's good to know that none of the audio data is being thrown away."**



› **The FLAC encoder is faster than *Oggenc* or *Lame*, taking care of an almost four-minute file in just six seconds.**

Radio Free Europe

It should be pointed out that other lossless audio formats have appeared over the last few years, under a variety of software licences. These include Apple's proprietary lossless codec, Windows Media Audio Lossless, WavPack, Monkey's Audio and Shorten. However it's FLAC that has the killer combination of patent-unencumbered Free Software and the strongest application and hardware support. The European Broadcasting Union uses FLAC in its Euroradio network (see **www.ebu.ch/en/radio/ops_rdo/erc**) indicating that FLAC is the true lossless audio standard, despite any contradictory claims from proprietary and less well-known formats.

The stats: FLAC

Name	Illusion.wav	Square.wav	Unobtanium.wav
WAV file size (MB)	63.0	45.2	39.5
Encoding time (minutes:seconds)	0:09	0:07	0:06
FLAC file size (MB)	34.4	25.8	22.8
Compression ratio (approximate)	2:1	2:1	2:1

FLAC is as fast as they come, but the trade-off is compression, thanks to its lossless nature.

LINUX FORMAT Verdict

FLAC

Version: FLAC 1.1.2
Web: http://flac.sourceforge.net/
Price: Free under BSD-style licence

» *FLAC offers the highest audio quality possible, with very fast encoding, but it does require significantly more storage.*

Rating 9/10

AAC

MPEG fights back with its own successor.

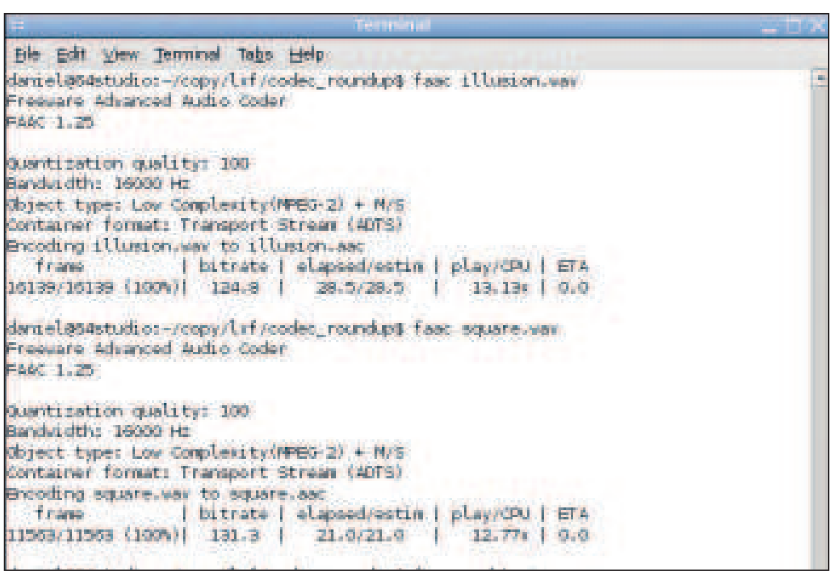
AAC is a newer codec from the MPEG stable, with the acronym standing for Advanced Audio Coding. It's defined in both MPEG-2 Part 7 and MPEG-4 Part 3, but is usually described as part of MPEG-4. Best known for its proprietary DRM-locked implementation in Apple products like the iPod, AAC is also found on the Sony PlayStation 3, Nintendo Wii, and in MPEG-4 video clips. It's part of the new DAB+ and Digital Radio Mondiale systems, neatly returning the story to the origins of lossy audio compression research.

MPEG-4 is a software patent gravy train, perhaps even more so than MPEG-1 Layer 3. Every company and research institute involved in the MPEG standards process is aiming to earn a slice of the internet and broadcast media pie (to go with the gravy, one assumes). An advantage over MP3 is that royalties are not required for AAC streaming and distribution, which must be very convenient if you're the boss of the iTunes Music Store. However, royalties are most definitely due on encoders and decoders, making AAC a difficult format to support in Free Software distributions.

AAC tools

There are tools for working with the AAC codec available for GNU/Linux users, thanks to **www.audiocoding.com**. *Faac* is a Free Software AAC encoder, released under the GNU LGPL license. The author concedes that the quality of *Faac* is not up to par with other AAC encoders available. It is complemented by *Faad2*, an AAC decoder licensed under GNU GPLv2. Both packages are, like *Lame*, distributed as source code only, and the home pages of *Faac* and *Faad* explicitly state that a patent licence is required to distribute binaries.

The default *Faac* quality setting of 100 approximates to 120kbps variable bit rate, for a normal WAV file ripped from an audio CD. The maximum quality value is 500, and the minimum is 10.



› ***Faac*** **is just as fast as Vorbis, but the output files are slightly larger.**

Encoding times were identical to the results achieved with Vorbis, but the AAC output files were larger. In the subjective test, the AAC files sounded perfectly good for casual listening, but no better or worse than the smaller Vorbis files. The last second or so of the file sounded like it had been truncated in *Totem* with the Gstreamer back-end, although the decoded WAV file produced by *FAAD* finished correctly. This would appear to indicate slightly buggy support for AAC in *Totem* or Gstreamer, rather than a bug in the *FAAC* encoder.

> **"The AAC files sounded no better or worse than the smaller Vorbis files."**

In theory, due the support of AAC by Apple and other proprietary software companies and music stores, AAC should be a long-lived format with plenty of hardware players available. The problem with that line of reasoning is that the Free Software implementation of the encoder, *FAAC*, is developed independently of the companies effectively controlling the format. It doesn't support any current or future DRM scheme that hardware players may require in order to permit the playback of audio content. This makes future compatibility something of a lottery, but there is the option to flash some iPods with a Free Software OS, such as the firmware from **www.rockbox.org**. These firmware replacements also have the side-effect of adding support to your player for Free Software formats like Vorbis and FLAC.

FAAC and *FAAD* are useful tools nonetheless, particularly if you have to exchange audio files with users of *iTunes* and other AAC software – as long as you can keep DRM out of the picture.

The stats: AAC

Name	Illusion.wav	Square.wav	Unobtanium.wav
WAV file size (MB)	63.0	45.2	39.5
Encoding time (minutes:seconds)	0:29	0:21	0:19
AAC file size (MB)	5.6	4.2	3.9
Compression ratio (approximate)	11:1	11:1	10:1

AAC is a safe all-rounder in terms of speed and compression; shame about the patents!

LINUX FORMAT Verdict

AAC

Version: FAAC 1.25
Web: www.audiocoding.com/faac.html
Price: Free under GNU LGPL

» *AAC offers no advantage over Vorbis, and because of DRM, compatibility with hardware players is not guaranteed.*

Rating 5/10

Codecs The verdict

FLAC 9/10

As GNU/Linux distributions began to add multimedia support, they naturally included the free Vorbis and FLAC encoders and decoders, which have become well integrated with Free Software desktop applications. MP3 and AAC support remains problematic; the distros don't want to get sued like Microsoft did, and upstream projects working on the codecs are still concerned about legal action. The *Lame* development team argue that their software is an educational tool for learning about MP3 encoding, not an actual encoder in the binary sense. While some distros include the *libmad* decoding library for MP3, most do not ship *Lame* binaries, for fear of legal reprisals – and a format that you can only decode, not encode, is only half as useful. Fluendo's plugins for *GStreamer* are a partial solution, but they cannot be linked against programs released under the GNU GPL because of their proprietary nature.

Real Audio

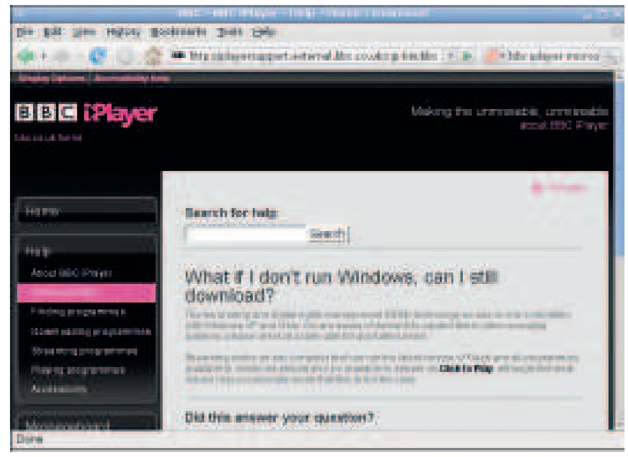
Real arrived somewhat half-heartedly on the scene, making an open source release of its player through the *Helix* project but leaving out the all-important Real Audio codecs. In the meantime, Real Audio lost its hegemony in the streaming market, with even former staunch Real supporter the BBC now looking like Microsoft's poodle in the *iPlayer* fiasco. Adobe has become significant in the streaming arena, through its acquisition of Macromedia. With *Flash*-based media players that sit on a website, the end user doesn't know or care which codec is being used, because the media is handled inside the *Flash* plugin. At least Adobe supports GNU/Linux on x86, for now, with its binary-only player.

The proprietary codec companies have little interest in supporting Linux, unless they can figure out some way of monetising its use. There are still no vendor implementations of Windows Media or *Apple iTunes* codecs for GNU/Linux, though third-party companies and independent Free Software developers have found various ways to support these formats. We did not test Windows Media codecs in this Roundup because they are completely proprietary, not even having the benefit of being published MPEG standards.

When it comes to encoding your own material from CD or the studio, there are a number of considerations. Not only is there a quality/size trade-off, but you've got to think about support for the format too. A music collection can last a lifetime, so do you really want to be hunting around for a binary decoder that will run on the computers of 20 years from now? Several proprietary codecs have gone from the internet during the last decade, including Liquid Audio and Sony's ATRAC. Free Software formats should have a natural advantage here, as they can outlast the company or developer that created them – as long as they remain popular enough to be maintained. But when it comes to portable audio devices, many of the manufacturers have lock-in deals with proprietary software vendors. In the case of Apple's iPod and Microsoft's Zune, the manufacturer and the software house are one and the same, leading to total vendor lock-in. Fortunately, there are enough open-minded hardware companies out there that you can have full Free Software compatibility, if you shop carefully.

> **"The BBC is now looking like Microsoft's poodle in the iPlayer fiasco."**

The overall winner of this Roundup is FLAC, for its to audio quality, impressive encoding speed and growing list of supported devices. True, it does take up many times the storage space of the alternatives tested here, but when storage space or bandwidth are limited, Vorbis complements FLAC very well. You can keep an archive of FLAC files on a hard disk, then encode from these to Vorbis files for a small capacity *Flash* player, or an *Icecast* streaming server. For a better guarantee of longevity, you should make sure that hard disc is backed up, just as you would with any other valuable data store. LXF



› Did we mention that *iPlayer* sucks? Send your emails to the BBC and see if you can get them to hurry up the Linux support!

Over to you

Which file formats have you chosen for your music collection? Which hardware players have you tried with Free Software format support? And does one codec really sound better than another? Email your feedback on this Roundup to **lxf.letters@futurenet.co.uk**

Table of features

Name	Licence	Patent trouble?	Audio quality	File compression	Encoder speed	Distro integration
Faac	LGPL	Yes	Good	Good	Good	Poor
FLAC	1.0.0	No	Excellent	Poor	Excellent	Good
Lame	LGPL	Yes	Good	Average	Poor	Average
Oggenc	BSD-style	No	Good	Good	Good	Good