FLAC

FLAC (/flæk/; **Free Lossless Audio Codec**) is an audio coding format for lossless compression of digital audio, developed by the Xiph.Org Foundation, and is also the name of the free software project producing the FLAC tools, the reference software package that includes a codec implementation. Digital audio compressed by FLAC's algorithm can typically be reduced to between 50 and 70 percent of its original size[4] and decompresses to an identical copy of the original audio data.

FLAC is an open format with royalty-free licensing and a reference implementation which is free software. FLAC has support for metadata tagging, album cover art, and fast seeking.

Contents

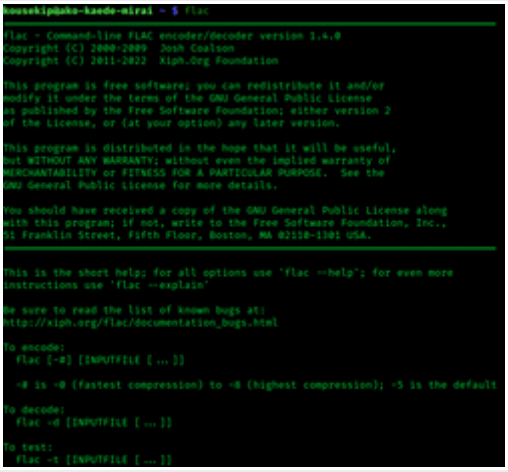
- History
- Composition
- Design
  - File structure
  - Encoding and decoding
  - Compression
- Comparison to other formats
- Adoption and implementations
- See also
- References
- External links

History

Development was started in 2000 by Josh Coalson.[5] The bit-stream format was frozen when FLAC entered beta stage with the release of version 0.5 of the reference implementation on 15 January 2001. Version 1.0 was released on 20 July 2001.[5]

On 29 January 2003, the Xiph.Org Foundation and the FLAC project announced the incorporation of FLAC under the Xiph.org banner. Xiph.org is home to other free compression formats such as Vorbis, Theora, Speex and Opus.[5][6][7]

Free Lossless Audio Codec



Developer(s)	Xiph.Org Foundation, Josh Coalson, Erik de Castro Lopo
Initial release	20 July 2001
Stable release	1.4.2[1] / 22 October 2022
Repository	gitlab.xiph.org/xiph/flac (https://gitlab.xiph.org/xiph/flac)
Written in	C, C++
Operating system	Cross-platform
Type	Codec
License	Command-line tools: GNU GPL Libraries: BSD
Website	xiph.org/flac (https://xiph.org/flac)

Free Lossless Audio Codec

Filename extension	`.flac`

Version 1.3.0 was released on 26 May 2013, at which point development was moved to the Xiph.org git repository.[8]

Internet media type	`audio/flac`
Uniform Type Identifier (UTI)	org.xiph.flac
Magic number	fLaC[2]
Type of format	Lossless audio
Standard	xiph.org/flac/format.html (https://xiph.org/flac/format.html)
Open format?	Yes[3]
Free format?	Yes

Composition

The FLAC project consists of:

- The stream formats
- A simple container format for the stream, also called FLAC (or *Native FLAC*)
- libFLAC, a library of reference encoders and decoders, and a metadata interface
- libFLAC++, an object-oriented wrapper around libFLAC
- `flac`, a command-line program based on libFLAC to encode and decode FLAC streams
- `metaflac`, a command-line metadata editor for .flac files and for applying ReplayGain
- Input plugins for various music players (Winamp, XMMS, foobar2000, musikCube, and many more)
- With Xiph.org incorporation, the Ogg container format, suitable for streaming (also called *Ogg FLAC*)

The specification of the stream format can be implemented by anyone without prior permission (Xiph.org reserves the right to set the FLAC specification and certify compliance), and neither the FLAC format nor any of the implemented encoding or decoding methods are covered by any patent. The reference implementation is free software. The source code for libFLAC and libFLAC++ is available under the BSD license, and the sources for `flac`, `metaflac`, and the plugins are available under the GNU General Public License.

In its stated goals, the FLAC project encourages its developers not to implement copy prevention features (DRM) or lossy compression of any kind.[9]

Design

File structure

A FLAC file consists of the magic number `fLaC`, metadata, and encoded audio.[10]

The encoded audio is divided into frames, which consists of a header, a data block, and a CRC16 checksum. Each frame is encoded independent of each other. A frame header begins with a sync word, used to identify the beginning of a valid frame. The rest of the header contains the number of samples, position of the frame, channel assignment, and optionally the sample rate and bit depth. The data block contains the audio information.[10]

Metadata in FLAC precedes the audio. Properties like the sample rate and the number of channels are always contained in the metadata. It may also contain other information, the album cover for example.[10] FLAC uses Vorbis comments for some types of metadata, like the title and artist name.

Encoding and decoding

The FLAC encoding algorithm consists of multiple stages. In the first stage, the input audio is split into blocks. If the audio contains multiple channels, each channel is encoded separately as a subblock. The encoder then tries to find a good mathematical approximation of the block, either by fitting a simple polynomial, or through general linear predictive coding. A description of the approximation, which is only a few bytes in length, is then written. Finally, the difference between the approximation and the input, called residual, is encoded using Rice coding. In many cases, a description of the approximation and the encoded residual takes up less space than using pulse-code modulation.[10]

The decoding process is the reverse of encoding. The compressed residual is first decoded.[11][12] The description of the mathematical approximation is then used to calculate a waveform. The result is formed by adding the residual and the calculated waveform.[13][14] As FLAC compresses losslessly, the decoded waveform is identical to the waveform before encoding.

For two-channel stereo, the encoder may choose to joint-encode the audio. The channels are transformed into a side channel, which is the difference between the two input channels, and a mid channel, the sum of the two input channels. In place of a mid channel, the left channel or the right channel may be encoded instead, which is sometimes more space-efficient.[15]

Even though the reference encoder uses a single block size for the whole stream,[10] FLAC allows the block size in samples to vary per block.

Compression

The amount of compression is determined by various parameters, including the order of the linear prediction model and the block size. Regardless of the amount of compression, the original data can always be reconstructed perfectly.

For user's convenience, the reference implementation defines 9 compression levels, which are presets of the more technical parameters to the encoding algorithm. The levels are labeled from 0 to 8, with higher numbers resulting in a higher compression ratio, at the cost of compression speed. The meaning of each compression level varies by implementation.[16][17]

FLAC is optimized for decoding speed at the expense of encoding speed. A benchmark has shown that, while there is little variation in decoding speed as compression level increases, beyond the default compression level 5, the encoding process takes up considerably more time with little space saved compared to level 5.[18]

Comparison to other formats

FLAC is specifically designed for efficient packing of audio data, unlike general-purpose lossless algorithms such as DEFLATE, which are used in ZIP and gzip. While ZIP may reduce the size of a CD-quality audio file by 10–20%, FLAC is able to reduce the size of audio data by 40–50% by taking advantage of the characteristics of audio.

The technical strengths of FLAC compared to other lossless formats lie in its ability to be streamed and decoded quickly, independent of compression level.

Since FLAC is a lossless scheme, it is suitable as an archive format for owners of CDs and other media who wish to preserve their audio collections. If the original media are lost, damaged, or worn out, a FLAC copy of the audio tracks ensures that an exact duplicate of the original data can be recovered at any time. An exact restoration from a lossy copy (e.g., MP3) of the same data is impossible. FLAC being lossless

means it is highly suitable for transcoding e.g. to MP3, without the normally associated transcoding quality loss between one lossy format and another. A CUE file can optionally be created when ripping a CD. If a CD is read and ripped perfectly to FLAC files, the CUE file allows later burning of an audio CD that is identical in audio data to the original CD, including track order and pregap, but excluding CD-Text and other additional data such as lyrics and CD+G graphics.[19]

Adoption and implementations

The reference implementation of FLAC is implemented as the *libFLAC* core encoder & decoder library, with the main distributable program `flac` being the reference implementation of the libFLAC API. This codec API is also available in C++ as libFLAC++. The reference implementation of FLAC compiles on many platforms, including most Unix (such as Solaris, BSD) and Unix-like (including Linux), Microsoft Windows, BeOS, and OS/2 operating systems. There are build-systems for autoconf/automake, MSVC, Watcom C, and Xcode. There is currently no multicore support in libFLAC, but utilities such as GNU parallel and various graphical frontends can be used to spin up multiple instances of the encoder.

FLAC playback support in portable audio devices and dedicated audio systems is limited compared to formats such as MP3[20] or uncompressed PCM. FLAC support is included by default in Windows 10, Android, BlackBerry 10 and Jolla devices.

In 2014, several aftermarket mobile electronics companies introduced multimedia solutions that include support for FLAC. These include the NEX series from Pioneer Electronics and the VX404 and NX404 from Clarion.

The European Broadcasting Union (EBU) has adopted the FLAC format for the distribution of high quality audio over its Euroradio network.[21] The Windows operating system has supported native FLAC integration since the introduction of Windows 10.[22] The Android operating system has supported native FLAC playback since version 3.1.[23][24] macOS High Sierra and iOS 11 add native FLAC playback support.[25]

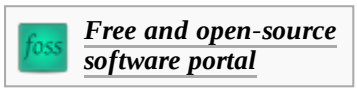
Among others the Pono music player and streaming service used the FLAC format.[26][27] Bandcamp insists on a lossless format for uploading, and has FLAC as a download option.[28] The Wikimedia Foundation sponsored a free and open-source online ECMAScript FLAC tool for browsers supporting the required HTML5 features.[29]

FLAC support by different operating systems

	Microsoft Windows	**macOS**	**Linux**	**Android**	**BlackBerry OS**	**iOS**
Codec support	Yes	Yes	Yes	Yes	Yes	Yes
Container support	FLAC (.flac) Matroska (.mka, .mkv)	FLAC (.flac) Core Audio Format (.caf)	FLAC (.flac) Matroska (.mka, .mkv) Ogg (.oga)	FLAC (.flac)	FLAC (.flac)	FLAC (.flac) Core Audio Format (.caf)
Notes	Support introduced in Windows 10.	Support introduced in High Sierra.	FLAC may need to be installed depending on the distribution.	Support introduced in Android 3.1	Support introduced in BlackBerry OS 5.0	Support introduced in iOS 11 (but depends on hardware used).

Various other containers are supported, independently from used operating system, depending on used playback software.

See also



- Comparison of audio coding formats

References

1. "FLAC 1.4.2 released" (https://xiph.org/flac/2022/10/22/flac-1-4-2-released.html).
2. Coalson, Josh. "FLAC – format" (https://xiph.org/flac/format.html#stream). Retrieved 4 April 2013. ""fLaC", the FLAC stream marker in ASCII, meaning byte 0 of the stream is 0x66, followed by 0x4C 0x61 0x43"
3. "PlayOgg!" (http://www.fsf.org/campaigns/playogg/en/). *Free Software Foundation*. 17 March 2010. Retrieved 1 October 2013.
4. "Looking at Flac Compression Ratios" (https://hbfs.wordpress.com/2012/02/07/looking-at-flac-compression-ratios/). Steven Pigeon. 7 February 2012. Retrieved 10 May 2018.
5. "News" (https://xiph.org/flac/news.html). FLAC. Retrieved 31 August 2009.
6. "FLAC Joins Xiph.org" (https://www.xiph.org/press/2003/flac/). *Xiph.org Foundation*. 29 January 2003. Retrieved 31 August 2009.
7. Plant, Emmett. "FLAC Joins Xiph!" (https://web.archive.org/web/20080529055542/http://xiph.org/ogg/flac.html). *Xiph.org Foundation*. Archived from the original (http://xiph.org/ogg/flac.html) on 29 May 2008. Retrieved 31 August 2009.
8. "FLAC – changelog" (https://xiph.org/flac/changelog.html). *Xiph.org Foundation*. Retrieved 15 October 2013.
9. "Developers" (https://xiph.org/flac/developers.html). *FLAC*. Retrieved 15 October 2013.
10. Coalson, Josh. "About the FLAC Format" (https://xiph.org/flac/documentation_format_overview.html). Retrieved 13 February 2022.
11. "stream_decoder.c" (https://gitlab.xiph.org/xiph/flac/-/blob/a2fe43f64e7ce8057fb274c64996568d69b301b8/src/libFLAC/stream_decoder.c). *libFLAC*. Retrieved 19 February 2022.
12. libFLAC/stream_decoder.c, line 2734.
13. "lpc.c" (https://gitlab.xiph.org/xiph/flac/-/blob/a2fe43f64e7ce8057fb274c64996568d69b301b8/src/libFLAC/lpc.c). *libFLAC*. Retrieved 19 February 2022.
14. libFLAC/lpc.c, lines 813–820.
15. "FLAC Format Specification" (https://xiph.org/flac/format.html). *FLAC*. Retrieved 19 February 2022.
16. "CUETools FLAC encoders comparison" (http://www.cuetools.net/wiki/CUETools_FLAC_encoders_comparison). *CUETools Wiki*. Retrieved 27 May 2013.
17. "Encoding Settings" (http://wiki.jriver.com/index.php/Encoding_Settings). *JRiver Media Centre*. Retrieved 27 May 2013.
18. "Lossless Codec Comparison" (https://web.archive.org/web/20090202063734/http://synthetic-soul.co.uk/comparison/lossless/index.asp). *Synthetic-soul.co.uk*. Archived from the original (http://synthetic-soul.co.uk/comparison/lossless/index.asp) on 2 February 2009. Retrieved 26 November 2016.
19. "FAQ" (https://xiph.org/flac/faq.html). *FLAC*. Retrieved 23 September 2014.
20. "Links" (https://xiph.org/flac/links.html#hardware). *FLAC*. Retrieved 24 March 2009.

21. "What is the EBU Musipop system?" (https://www.ebu.ch/en/radio/ops_rdo/faq/index.php). *EBU*. 24 August 2021.
22. "Audio snobs rejoice: Windows 10 will have system-wide FLAC support" (http://www.pcworld.com/article/2852595/audio-snobs-rejoice-windows-10-will-have-system-wide-flac-support.html). *PC World*. Retrieved 10 July 2015.
23. "Android Supported Media Formats" (https://developer.android.com/guide/appendix/media-formats.html). *Android.com*. 4 August 2011. Retrieved 27 February 2012.
24. "Issue 1461 – android – FLAC file support enhancement request" (https://code.google.com/p/android/issues/detail?id=1461). *Google Code*. Retrieved 5 August 2011.
25. "iOS 11 brings lossless FLAC audio playback to iPhone and iPad" (http://www.idownloadblog.com/2017/06/09/ios-11-brings-lossless-flac-audio-to-iphone-and-ipad/). *idownloadblog.com*. 9 June 2017. Retrieved 26 September 2017.
26. "FAQ" (http://www.ponomusic.com/#faq). *ponomusic.com*. 14 March 2014. Retrieved 14 March 2014.
27. "Home" (http://www.qobuz.com). *Qobuz.com*. Retrieved 10 April 2014.
28. "How and why should I upload lossless files?" (https://get.bandcamp.help/hc/en-us/articles/360007802254-How-and-why-should-I-upload-lossless-files-). *Bandcamp*. Retrieved 16 February 2022.
29. Rillke (2015). "JavaScript FLAC de- and encoder" (https://blog.rillke.com/flac.js/). Retrieved 9 February 2015.

External links

- Official website (https://xiph.org/flac/)
- Lossless audio formats comparison (http://www.bobulous.org.uk/misc/lossless_audio_2006.html): measuring FLAC against five other lossless audio formats
- Lossless comparison (http://wiki.hydrogenaud.io/index.php?title=Lossless_comparison): FLAC against seven other lossless audio formats on Hydrogenaudio
- GSMArena Phone Finder (http://www.gsmarena.com/results.php3?sName=&sFreeText=FLAC): all phones & tablets with FLAC support