Ogg

Ogg is a free, open container format maintained by the Xiph.Org Foundation. The authors of the Ogg format state that it is unrestricted by software patents[3] and is designed to provide for efficient streaming and manipulation of high-quality digital multimedia. Its name is derived from "ogging", jargon from the computer game *Netrek*.[4]

The Ogg container format can multiplex a number of independent streams for audio, video, text (such as subtitles), and metadata.

In the Ogg multimedia framework, Theora provides a lossy video layer. The audio layer is most commonly provided by the music-oriented Vorbis format or its successor Opus. Lossless audio compression formats include FLAC, and OggPCM.

Before 2007, the .ogg filename extension was used for all files whose content used the Ogg container format. Since 2007, the Xiph.Org Foundation recommends that .ogg only be used for Ogg Vorbis audio files. The Xiph.Org Foundation decided to create a new set of file extensions and media types to describe different types of content such as .oga for audio only files, .ogv for video with or without sound (including Theora), and .ogx for multiplexed Ogg.[5]

As of November 7, 2017, the current version of the Xiph.Org Foundation's reference implementation is libogg 1.3.3.[6] Another version, libogg2, has been in development, but is awaiting a rewrite as of 2018.[7] Both software libraries are free software, released under the New BSD License. Ogg reference implementation was separated from Vorbis on September 2, 2000.[8]

Because the format is free, and its reference implementation is not subject to restrictions associated with copyright, Ogg's various codecs have been incorporated into a number of different free and proprietary media players, both commercial and non-commercial, as well as portable media players and GPS receivers from different manufacturers.

Ogg

Filename extension	`.ogg`, `.ogv`, `.oga`, `.ogx`, `.ogm`, `.spx`, `.opus`
Internet media type	`video/ogg`, `audio/ogg`, `application/ogg`
Magic number	OggS
Developed by	Xiph.Org Foundation
Initial release	May 2003
Type of format	Container format
Container for	Vorbis, Theora, Speex, Opus, FLAC, Dirac, and others.
Open format?	Yes
Free format?	Yes[1]

libogg

Developer(s)	Xiph.Org Foundation
Initial release	22 September 2004
Stable release	1.3.5 / 4 June 2021
Type	Reference implementation (multiplexer/demultiplexer)
License	BSD-style license[2]
Website	downloads.xiph.org/releases/ogg/ (http://downloads.xiph.org/releases/ogg/)

Naming

Ogg is derived from "ogging", jargon from the computer game *Netrek*, which came to mean doing something forcefully, possibly without consideration of the drain on future resources.[4] At its inception, the Ogg project was thought to be somewhat ambitious given the limited power of the PC hardware of the time.[9] Although it is sometimes assumed that the name "Ogg" comes from the character of Nanny Ogg in Terry

Pratchett's *Discworld* novels, the format's developers say that is not true.[9] Still, to quote the same reference: "Vorbis, on the other hand is named after the Terry Pratchett character from the book *Small Gods*".

The Ogg Vorbis project started in 1993. It was originally named "Squish" but that name was already trademarked, so the project underwent a name change. The new name, "OggSquish", was used until 2001 when it was changed again to "Ogg". Ogg has since come to refer to the container format, which is now part of the larger Xiph.org multimedia project. Today, "Squish" (now known as "Vorbis") refers to a particular audio coding format typically used with the Ogg container format.[10]

File format

The "Ogg" bitstream format, designed principally by the Xiph.Org Foundation, has been developed as the framework of a larger initiative aimed at producing a set of components for the coding and decoding of multimedia files, which are available free of charge and freely re-implementable in software and hardware.

The format consists of chunks of data each called an "Ogg page". Each page begins with the characters "OggS" to identify the file as Ogg format.

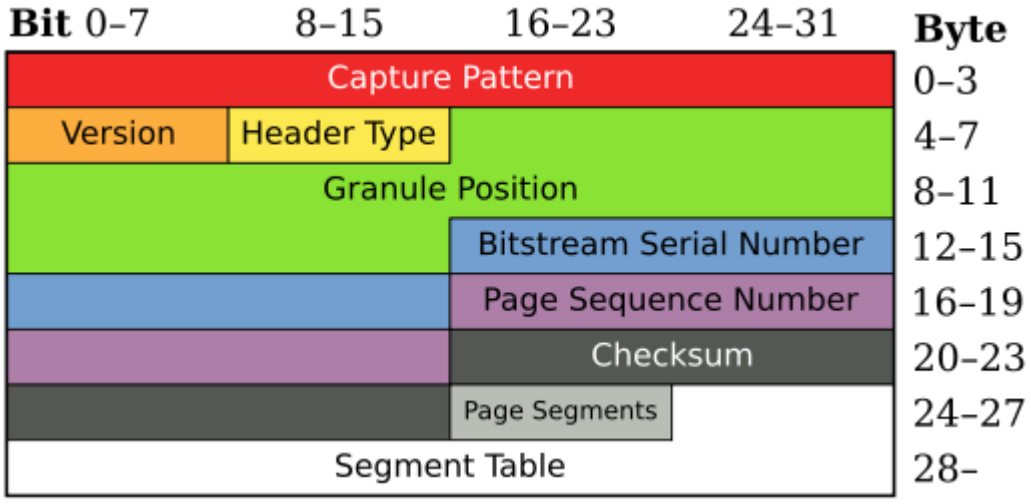
A "serial number" and "page number" in the page header identifies each page as part of a series of pages making up a bitstream. Multiple bitstreams may be multiplexed in the file where pages from each bitstream are ordered by the seek time of the contained data. Bitstreams may also be appended to existing files, a process known as "chaining", to cause the bitstreams to be decoded in sequence.

A BSD-licensed library, called "libvorbis", is available to encode and decode data from "Vorbis" streams. Independent Ogg implementations are used in several projects such as RealPlayer and a set of DirectShow filters.[11]

Mogg, the "Multi-Track-Single-Logical-Stream Ogg-Vorbis", is the multi-channel or multi-track Ogg file format.

Page structure

The following is the field layout of an Ogg page header:



Capture pattern – 32 bits

The capture pattern or sync code is a magic number used to ensure synchronization when parsing Ogg files. Every page starts with the four ASCII character sequence, "OggS". This

assists in resynchronizing a parser in cases where data has been lost or is corrupted, and is a sanity check before commencing parsing of the page structure.

Version – 8 bits

This field indicates the version of the Ogg bitstream format, to allow for future expansion. It is currently mandated to be 0.

Header type – 8 bits

This is an 8 bit field of flags, which indicates the type of page that follows.

Bit	Value	Flag	Page type
0	0x01	Continuation	The first packet on this page is a continuation of the previous packet in the logical bitstream.
1	0x02	BOS	Beginning Of Stream. This page is the first page in the logical bitstream. The BOS flag must be set on the first page of every logical bitstream, and must not be set on any other page.
2	0x04	EOS	End Of Stream. This page is the last page in the logical bitstream. The EOS flag must be set on the final page of every logical bitstream, and must not be set on any other page.

Granule position – 64 bits

A granule position is the time marker in Ogg files. It is an abstract value, whose meaning is determined by the codec. It may, for example, be a count of the number of samples, the number of frames or a more complex scheme.

Bitstream serial number – 32 bits

This field is a serial number that identifies a page as belonging to a particular logical bitstream. Each logical bitstream in a file has a unique value, and this field allows implementations to deliver the pages to the appropriate decoder. In a typical Vorbis and Theora file, one stream is the audio (Vorbis), and the other is the video (Theora)

Page sequence number – 32 bits

This field is a monotonically increasing field for each logical bitstream. The first page is 0, the second 1, etc. This allows implementations to detect when data has been lost.

Checksum – 32 bits

This field provides a CRC32 checksum of the data in the entire page (including the page header, calculated with the checksum field set to 0). This allows verification that the data has not been corrupted since it was authored. Pages that fail the checksum should be discarded. The checksum is generated using a polynomial value of 0x04C11DB7.

Page segments – 8 bits

This field indicates the number of segments that exist in this page. It also indicates how many bytes are in the segment table that follows this field. There can be a maximum of 255 segments in any one page.

Segment table

The segment table is an array of 8-bit values, each indicating the length of the corresponding segment within the page body. The number of segments is determined from the preceding page segments field. Each segment is between 0 and 255 bytes in length.

The segments provide a way to group segments into packets, which are meaningful units of data for the decoder. When the segment's length is indicated to be 255, this indicates that the following segment is to be concatenated to this one and is part of the same packet. When the segment's length is 0–254, this indicates

that this segment is the final segment in this packet. Where a packet's length is a multiple of 255, the final segment is length 0.

Where the final packet continues on the next page, the final segment value is 255, and the continuation flag is set on the following page to indicate that the start of the new page is a continuation of last page.

Metadata

VorbisComment is a base-level Metadata format initially authored for use with Ogg Vorbis. It has since been adopted in the specifications of Ogg encapsulations for other Xiph.Org codecs including Theora, Speex, FLAC and Opus. *VorbisComment* is the simplest and most widely-supported mechanism for storing metadata with Xiph.Org codecs.[12]

Notably, one or more `METADATA_BLOCK_PICTURE=...` in a *VorbisComment* for thumbnails and cover art have Base64-encoded values of the corresponding FLAC `METADATA_BLOCK_PICTURE`. In other words, FLAC stores thumbnails and cover art in binary blocks—outside of the FLAC tags in a little-endian `METADATA_BLOCK_VORBIS_COMMENT`.[13][14]

Other existing and proposed mechanisms are:[15]

- FLAC metadata blocks
- Ogg Skeleton[16]
- Continuous Media Markup Language (deprecated)

History



The *Play Ogg* web button

The Ogg project began with a simple audio compression package as part of a larger project in 1993.[9] The software was originally named *Squish* but due to an existing trade mark it was renamed to *OggSquish*. This name was later used for the whole Ogg project. In 1997, the Xiphophorus OggSquish was described as "an attempt both to create a flexible compressed audio format for modern audio applications as well as to provide the first audio format that is common on any and every modern computer platform".[17] The OggSquish was in 2000 referred to as "a group of several related multimedia and signal processing projects". In 2000, two projects were in active development for planned release: Ogg Vorbis format and libvorbis—the reference implementation of Vorbis. Research also included work on future video and lossless audio coding.[9][18][19][20][21] In 2001, OggSquish was renamed to *Ogg* and it was described as "the umbrella for a group of several related multimedia and signal processing projects".[22] Ogg has come to stand for the file format, as part of the larger Xiph.org multimedia project. Squish became just the name of one of the Ogg codecs.[10] In 2009, Ogg is described as "a multimedia container format, and the native file and stream format for the Xiph.org multimedia codecs".[23]

The Ogg reference implementation was separated from Vorbis on September 2, 2000.[8]

In May 2003, two Internet RFCs were published relating to the format. The Ogg bitstream was defined in RFC 3533 (https://datatracker.ietf.org/doc/html/rfc3533) (which is classified as 'informative') and its Internet content type (`application/ogg`) in RFC 3534 (https://datatracker.ietf.org/doc/html/rfc3534) (which

is, as of 2006, a proposed standard protocol). In September 2008, RFC 3534 was obsoleted by RFC 5334 (https://datatracker.ietf.org/doc/html/rfc5334), which added content types `video/ogg`, `audio/ogg` and filename extensions .ogx, .ogv, .oga, .spx.

OGM

In 2002, the lack of formal video support in Ogg resulted in the development of the *OGM* file format, a hack on Ogg that allowed embedding of video from the Microsoft DirectShow framework into an Ogg-based wrapper. OGM was initially supported only by closed source Windows-only tools, but the codebase was subsequently opened. Later, video (and subtitle) support were formally specified for Ogg but in a manner incompatible with OGM. Independently, the Matroska container format reached maturity and provided an alternative for people interested in combining Vorbis audio and arbitrary video codecs. As a result, OGM is no longer supported or developed and is formally discouraged by Xiph.org.[24] Today, video in Ogg is found with the .ogv file extension, which is formally specified and officially supported. Software and codecs that support .ogm files are available without charge.[25]

2006

Although Ogg had not reached anywhere near the ubiquity of the MPEG standards[26][27] (e.g., MP3/MP4), as of 2006, it was commonly used to encode free content (such as free music, multimedia on Wikimedia Foundation projects and Creative Commons files) and had started to be supported by a significant minority of digital audio players. Also supporting the Ogg format were many popular video game engines, including *Doom 3*, *Unreal Tournament 2004*, *Halo: Combat Evolved*, *Jets'n'Guns*, *Mafia: The City of Lost Heaven*, *Myst IV: Revelation*, *StepMania*, *Serious Sam: The Second Encounter*, *Lineage 2*, *Vendetta Online*, *Battlefield 2*, and the *Grand Theft Auto* engines, as well as the audio files of the Java-based game, *Minecraft*. The more popular Vorbis codec had built-in support on many software players, and extensions were available for nearly all the rest.

2007

On May 16, 2007, the Free Software Foundation started a campaign to increase the use of Vorbis "as an ethically, legally and technically superior audio alternative to the proprietary MP3 format".[28] People were also encouraged to support the campaign by adding a web button to their website or blog. For those who did not want to download and use the FSF's suggested Ogg player (VLC), the Xiph.Org Foundation had an official codec[29] for QuickTime-based applications in Windows and Mac OS X, such as iTunes players and iMovie applications; and Windows users could install a Windows Media Player Ogg codec.[30]

2009

By June 30, 2009, the Ogg container, through the use of the Theora and Vorbis, was the only container format included in Firefox 3.5 web browser's implementation of the HTML5 `<video>` and `<audio>` elements.[31][32] This was in accordance with the original recommendation outlined in, but later removed from, the HTML5 draft specification (see Ogg controversy).

2010

On March 3, 2010, a technical analysis by an FFmpeg developer was critical about the general purpose abilities of Ogg as a multimedia container format.[33] The author of Ogg later responded to these claims in an article of his own.[34]

Ogg codecs

Ogg is only a container format. The actual audio or video encoded by a codec is stored inside an Ogg container. Ogg containers may contain streams encoded with multiple codecs; for example, a video file with sound contains data encoded by both an audio codec and a video codec.

Being a container format, Ogg can embed audio and video in various formats[35][36] (such as Dirac, MNG, CELT, MPEG-4, MP3 and others) but Ogg was intended to be, and usually is, used with the following Xiph.org free codecs:

- Audio
  - Lossy
    - Speex: handles voice data at low bitrates (~2.1–32 kbit/s/channel)
    - Vorbis: handles general audio data at mid to high-level variable bitrates (≈16–500 kbit/s per channel)
    - Opus: handles voice, music and generic audio at low and high variable bitrates (≈6–510 kbit/s per channel)
  - Lossless
    - FLAC handles archival and high-fidelity audio data.
    - OggPCM allows storing standard uncompressed PCM audio in an Ogg container[37]
- Video
  - Lossy
    - Theora: based upon On2's VP3, it is targeted at competing with MPEG-4 video (for example, encoded with DivX or Xvid), RealVideo, or Windows Media Video.
    - Daala: a video coding format under development.
    - Tarkin: an experimental and now obsolete video codec developed in 2000, 2001 and 2002 utilizing discrete wavelet transforms in the three dimensions of width, height, and time.[20][38][39][40] It has been put on hold after Theora became the main focus for video encoding (in August 2002).[41]
    - Dirac: a free and open video format developed by the BBC. Uses wavelet encoding.[42]
  - Lossless
    - Dirac: a part of the specification of dirac covers lossless compression.
    - Daala: a video coding format under development.
- Text
  - Continuous Media Markup Language: a text/application codec for timed metadata, captioning, and formatting.
  - Annodex: A free and open source set of standards developed by CSIRO to annotate and index networked media.

- OggKate: An overlay codec, originally designed for karaoke and text, that can be multiplexed in Ogg.[43]

Media types

Ogg audio media is registered as IANA media type `audio/ogg` with file extensions `.oga`, `.ogg`, and `.spx`. It is a proper subset of the Ogg video media type `video/ogg` with file extension `.ogv`. Other Ogg applications use media type `application/ogg` with file extension `.ogx`; this is a superset of `video/ogg`.[36] The Opus media type `audio/opus` with file extension `.opus` was registered later in RFC 7587 (https://datatracker.ietf.org/doc/html/rfc7587) and 7845 (https://datatracker.ietf.org/doc/html/rfc7845).

See also

- Audio data compression
- Comparison of audio coding formats
- Comparison of container formats
- Commons:file types
- Use of Ogg formats in HTML5
- Wikimedia Commons' Theora video conversion help page
- Wikipedia:Media help (Ogg) Installing audio and video software

References

1. *Ogg File Format* (https://www.loc.gov/preservation/digital/formats/fdd/fdd000026.shtml) (Full draft). Sustainability of Digital Formats. Washington, D.C.: Library of Congress. 19 February 2008. Archived (https://web.archive.org/web/20211008114103/https://www.loc.gov/preservation/digital/formats/fdd/fdd000026.shtml) from the original on 8 October 2021. Retrieved 1 December 2021.
2. "Sample Xiph.Org Variant of the BSD License" (http://www.xiph.org/licenses/bsd/). Xiph.Org Foundation. Archived (https://web.archive.org/web/20200411163951/https://www.xiph.org/licenses/bsd/) from the original on 2020-04-11. Retrieved 2009-08-29.
3. "Vorbis.com: FAQ" (https://web.archive.org/web/20051001065649/http://vorbis.com/faq/#what). Archived from the original (http://www.vorbis.com/faq/#what) on 2005-10-01. Retrieved 2010-05-28.
4. "Ogging 101" (http://www.netrek.org/clueguides/ogging-101.html). Archived (https://web.archive.org/web/20171225161547/http://www.netrek.org/clueguides/ogging-101.html) from the original on 2017-12-25. Retrieved 2016-11-06. "3.3 Ogging: This is the art of killing a carrier, or potential carrier, by a suicide run."
5. "MIME Types and File Extensions" (http://wiki.xiph.org/index.php/MIME_Types_and_File_Extensions). *XiphWiki*. 2007-09-07. Archived (https://web.archive.org/web/20181117004309/https://wiki.xiph.org/index.php/MIME_Types_and_File_Extensions) from the original on 2018-11-17. Retrieved 2007-09-10.
6. Giles, Ralph (2017-11-07). "libogg 1.3.3 release" (http://lists.xiph.org/pipermail/ogg-dev/2017-November/001933.html). *ogg-dev* (Mailing list). Archived (https://web.archive.org/web/20180404073629/http://lists.xiph.org/pipermail/ogg-dev/2017-November/001933.html) from the original on 2018-04-04. Retrieved 2019-01-14.

7. Giles, Ralph (2008-01-05). "r14372 - in trunk/theora: . lib/dec lib/enc" (http://lists.xiph.org/pipermail/commits/2008-January/012792.html). *xiph-commits* (Mailing list). Archived (https://web.archive.org/web/20201003153343/http://lists.xiph.org/pipermail/commits/2008-January/012792.html) from the original on 2020-10-03. Retrieved 2019-01-14. "This library was never released and now looks like it needs to be redesigned."
8. Xiph.Org (2002-07-19) Ogg releases - libogg-1.0.tar.gz - CHANGES (http://downloads.xiph.org/releases/ogg/) Archived (https://web.archive.org/web/20170614174325/http://downloads.xiph.org/releases/ogg/) 2017-06-14 at the Wayback Machine. Retrieved 2009-09-01.
9. "Xiph.org: naming" (https://web.archive.org/web/20120227092222/http://www.xiph.org/xiphname). Xiph.org Foundation. 2006-01-07. Archived from the original (http://xiph.org/xiphname) on 2012-02-27. Retrieved 2008-02-16. "At the time Ogg was starting out, most personal computers were i386s and the i486 was new. I remember thinking about the algorithms I was considering, "Whoa, that's heavyweight. People are going to need a 486 to run that..." While the software ogged the music, there wasn't much processor left for anything else."
10. "Xiph.org: naming" (https://web.archive.org/web/20120227092222/http://www.xiph.org/xiphname). Xiph.org Foundation. 2006-01-07. Archived from the original (http://xiph.org/xiphname) on 2012-02-27. Retrieved 2009-09-02.
11. "Directshow Filters for Ogg Vorbis" (http://xiph.org/dshow/). Archived (https://web.archive.org/web/20150208215333/http://www.xiph.org/dshow/) from the original on 2015-02-08. Retrieved 2015-02-14.
12. "VorbisComment" (https://wiki.xiph.org/VorbisComment). 26 April 2016. Archived (https://web.archive.org/web/20181023234243/https://wiki.xiph.org/VorbisComment) from the original on 23 October 2018. Retrieved 23 October 2018.
13. "METADATA_BLOCK_PICTURE" (https://xiph.org/flac/format.html#metadata_block_picture). *flac free lossless audio codec*. Xiph.Org. 2014. Archived (https://web.archive.org/web/20170327065952/https://xiph.org/flac/format.html#metadata_block_picture) from the original on 2017-03-27. Retrieved 2019-12-03.
14. "Ogg Vorbis I format specification: comment field and header specification" (https://www.xiph.org/vorbis/doc/v-comment.html). Xiph.Org. 2005. Archived (https://web.archive.org/web/20191203005103/https://www.xiph.org/vorbis/doc/v-comment.html) from the original on 2019-12-03. Retrieved 2019-12-03.
15. "Metadata" (https://wiki.xiph.org/Metadata). *xiph.org Foundation*. 24 July 2013. Archived (https://web.archive.org/web/20181023234247/https://wiki.xiph.org/Metadata) from the original on 23 October 2018. Retrieved 23 October 2018.
16. "Ogg Skeleton 4" (https://wiki.xiph.org/Ogg_Skeleton_4). Xiph.Org. 2012. Archived (https://web.archive.org/web/20190813223619/https://wiki.xiph.org/Ogg_Skeleton_4) from the original on 2019-08-13. Retrieved 2019-12-03.
17. Montgomery, Christopher (1997). "Ogg 98.9" (http://people.xiph.org/~giles/historic/www.mit.edu/afs/sipb/user/mwhitson/Project/Ogg/98.9/). Xiphophorus company. Archived (https://web.archive.org/web/20160424045623/http://people.xiph.org/~giles/historic/www.mit.edu/afs/sipb/user/mwhitson/Project/Ogg/98.9/) from the original on 2016-04-24. Retrieved 2009-09-02.
18. Xiph.org (2000-01-18). "OggSquish Vorbis encoding format documentation" (https://web.archive.org/web/20000118132124/http://www.xiph.org/ogg/doc/vorbis.html). Archived from the original (http://www.xiph.org/ogg/doc/vorbis.html) on January 18, 2000. Retrieved 2008-09-02.
19. Xiph.org (2000-01-18). "OggSquish logical and physical bitstream overview" (https://web.archive.org/web/20000118122008/http://www.xiph.org/ogg/doc/oggstream.html). Archived from the original (http://www.xiph.org/ogg/doc/oggstream.html) on January 18, 2000. Retrieved 2008-09-02.

20. Xiphophorus company (2001-04-05). "The Ogg project homepage" (https://web.archive.org/web/20010405060456/http://www.xiph.org/ogg/index.html). Archived from the original (http://www.xiph.org/ogg/index.html) on April 5, 2001. Retrieved 2009-09-02.
21. Xiph.org (2001-12-04). "The Ogg project homepage" (https://web.archive.org/web/20011204194046/http://www.xiph.org/ogg/index.html). Archived from the original (http://www.xiph.org/ogg/index.html) on December 4, 2001. Retrieved 2008-09-02.
22. Xiphophorus company (2001-12-04). "The Ogg project homepage" (https://web.archive.org/web/20011204194046/http://www.xiph.org/ogg/index.html). Archived from the original (http://www.xiph.org/ogg/index.html) on December 4, 2001. Retrieved 2009-09-02.
23. "The Ogg container format" (http://xiph.org/ogg/). Xiph.Org Foundation. 2006-01-07. Archived (https://web.archive.org/web/20000520212458/http://xiph.org/ogg/) from the original on 2000-05-20. Retrieved 2009-09-02.
24. "OGM is not Ogg. Xiph.org does not support OGM" (https://web.archive.org/web/20120317061837/http://www.xiph.org/container/ogm.html). Backup at WayBack Machine. 2004-07-07. Archived from the original on March 17, 2012.
25. "K-Lite Codec Pack: Comparison of abilities and supported file formats" (https://www.codecguide.com/klcp_ability_comparison.htm). *Codec Guide*. Archived (https://web.archive.org/web/20180316214632/https://www.codecguide.com/klcp_ability_comparison.htm) from the original on 16 March 2018. Retrieved 16 March 2018.
26. "MP3's Loss, Open Source's Gain" (https://www.wired.com/entertainment/music/news/2007/02/72785?currentPage=all). *Wired.com*. 2007-02-23. Archived (https://web.archive.org/web/20100605091342/http://www.wired.com/entertainment/music/news/2007/02/72785?currentPage=all) from the original on 2010-06-05. Retrieved 2010-03-04.
27. "MP3's Loss, Open Source's Gain" (http://archive.wired.com/entertainment/music/news/2007/02/72785). *Wired.com*. 2007-02-23. Archived (https://web.archive.org/web/20150204090537/http://archive.wired.com/entertainment/music/news/2007/02/72785) from the original on 2015-02-04. Retrieved 2015-02-04.
28. " 'Play Ogg': FSF launches free audio format campaign" (http://www.fsf.org/news/playogg.html) (in Latin). Fsf.org. Archived (https://web.archive.org/web/20110608015711/http://www.fsf.org/news/playogg.html) from the original on 2011-06-08. Retrieved 2010-03-04.
29. "Foundation: Ogg QuickTime Components for iTunes and iMovie (Windows and Mac OS X)" (http://xiph.org/quicktime/). Xiph.org. Archived (https://web.archive.org/web/20081026081609/http://xiph.org/quicktime/) from the original on 2008-10-26. Retrieved 2010-03-04.
30. "Directshow Filters for Ogg Vorbis, Speex, Theora and FLAC" (http://www.xiph.org/dshow/). Xiph.org. 2010-02-22. Archived (https://web.archive.org/web/20130124155113/http://www.xiph.org/dshow/) from the original on 2013-01-24. Retrieved 2010-03-04.
31. "Mozilla Developer Center - HTML - Element - Video" (https://developer.mozilla.org/En/HTML/Element/Video). September 2009. Archived (https://web.archive.org/web/20100627072405/https://developer.mozilla.org/En/HTML/Element/Video) from the original on 2010-06-27. Retrieved 2009-12-28.
32. "Mozilla Developer Center - HTML - Element - Audio" (https://developer.mozilla.org/En/HTML/Element/Audio). September 2009. Archived (https://web.archive.org/web/20100627072053/https://developer.mozilla.org/En/HTML/Element/Audio) from the original on 2010-06-27. Retrieved 2009-12-28.
33. Rullgard, Mans (2010-03-03). "Ogg objections" (http://hardwarebug.org/2010/03/03/ogg-objections/). hardwarebug.org. Archived (https://web.archive.org/web/20100509211056/http://hardwarebug.org/2010/03/03/ogg-objections/) from the original on 2010-05-09. Retrieved 2010-05-02.

34. Montgomery, Christopher (2010-04-27). "Monty - In Defense of Ogg's Good Name" (https://web.archive.org/web/20100429111810/http://people.xiph.org/~xiphmont/lj-pseudocut/o-response-1.html). people.xiph.org/~xiphmont. Archived from the original (http://people.xiph.org/~xiphmont/lj-pseudocut/o-response-1.html) on 2010-04-29. Retrieved 2010-05-02.
35. "MIME Types and File Extensions" (http://wiki.xiph.org/index.php/MIME_Types_and_File_Extensions). XiphWiki. 2009-10-04. Archived (https://web.archive.org/web/20181117004309/https://wiki.xiph.org/index.php/MIME_Types_and_File_Extensions) from the original on 2018-11-17. Retrieved 2009-10-24.
36. I. Goncalves; S. Pfeiffer; C. Montgomery (2008). *Ogg Media Types* (https://datatracker.ietf.org/doc/html/rfc5334#section-10). sec. 10. doi:10.17487/RFC5334 (https://doi.org/10.17487%2FRFC5334). RFC 5334 (https://datatracker.ietf.org/doc/html/rfc5334).
37. "OggPCM" (https://wiki.xiph.org/index.php/OggPCM). XiphWiki. 2020-05-25. Archived (https://web.archive.org/web/20201003204000/https://wiki.xiph.org/index.php/OggPCM) from the original on 2020-10-03. Retrieved 2020-05-25.
38. Ralph Giles (2000-12-19) vorbis-dev - Tarkin video codec? (http://lists.xiph.org/pipermail/vorbis-dev/2000-December/002281.html) Archived (https://web.archive.org/web/20180124052905/http://lists.xiph.org/pipermail/vorbis-dev/2000-December/002281.html) 2018-01-24 at the Wayback Machine, Xiph.org vorbis-dev mailinglist. Retrieved 2009-09-06.
39. Jack Moffitt (2001-01-03) vorbis-dev - Tarkin at last (http://lists.xiph.org/pipermail/vorbis-dev/2001-January/002604.html) Archived (https://web.archive.org/web/20140930123907/http://lists.xiph.org/pipermail/vorbis-dev/2001-January/002604.html) 2014-09-30 at the Wayback Machine, Xiph.org vorbis-dev mailinglist. Retrieved 2009-09-06.
40. Chris Montgomery (2001-02-11) Tarkin developer mailing list (http://lists.xiph.org/pipermail/vorbis/2001-February/014147.html) Archived (https://web.archive.org/web/20140930202014/http://lists.xiph.org/pipermail/vorbis/2001-February/014147.html) 2014-09-30 at the Wayback Machine, Retrieved on 2009-09-06
41. Michael Smith (2005-08-29) Tarkin (http://lists.xiph.org/pipermail/vorbis-dev/2005-August/018213.html) Archived (https://web.archive.org/web/20160303194435/http://lists.xiph.org/pipermail/vorbis-dev/2005-August/018213.html) 2016-03-03 at the Wayback Machine. Retrieved 2009-09-06.
42. Dirac specification - integration of Dirac encoded video into commonly used container formats (http://diracvideo.org/specifications) Archived (https://web.archive.org/web/20100614050318/http://diracvideo.org/specifications/) 2010-06-14 at the Wayback Machine Retrieved on 2009-07-05
43. "OggKate" (https://wiki.xiph.org/OggKate). *wiki.xiph.org*. 2017. Archived (https://web.archive.org/web/20110718082812/http://wiki.xiph.org/OggKate) from the original on 2011-07-18. Retrieved 2019-12-03.

External links

- The Xiph.Org Foundation official webpage — Ogg (https://www.xiph.org/ogg/)
- Pfeiffer, Silva (May 2003). *The Ogg Encapsulation Format Version 0* (https://datatracker.ietf.org/doc/html/rfc3533). Internet Engineering Task Force. doi:10.17487/RFC3533 (https://doi.org/10.17487%2FRFC3533). RFC 3533 (https://datatracker.ietf.org/doc/html/rfc3533).
- Using Creative Commons Metadata in Ogg containers (https://wiki.creativecommons.org/OGG)
- Ogg etymology (http://www.catb.org/~esr/jargon/html/O/ogg.html) from the Jargon File
- Xiph.Org's official Ogg QuickTime Components for iTunes and iMovie (Windows and Mac OS X) (https://xiph.org/quicktime/)

- Windows Media Player codecs for Vorbis, Speex, Theora and FLAC (http://www.xiph.org/dshow/)
- ffmpeg2theora (http://v2v.cc/~j/ffmpeg2theora/) Ogg Theora encoder, commandline application for Linux and Win32

Retrieved from "https://en.wikipedia.org/w/index.php?title=Ogg&oldid=1173848626"

-