Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 24 August 2026

ssw. Whited, Ed.

20 February 2026

Ogg Stem Files
draft-swhited-ogg-stems-01

Abstract

This document defines a multi-track profile of the Ogg container format for storing stems that is also backwards compatible with existing media players.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 24 August 2026.

Table of Contents

1. Introduction

Stem are recordings of individual instruments, or clusters of instruments, used by DJs and music producers for live mixing of music. Historically stem files have been stored as individual audio files, or using patent-encumbered or vendor specific proprietary container formats. The Ogg file format developed by the Xiph.Org Foundation was formally specified in [RFC3533] and [RFC5334] and is ideally situated as a container for stems. This specification documents a profile for the Ogg container format that allows it to store lossless or lossy stems as well as metadata about the stems for use in DJ applications or Digital Audio Workstations.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Requirements

STEM files have a few basic requirements:

* Backwards compatibility with existing media players

* The ability to store at 5 audio tracks

* The ability to syncronize playback of multiple audio tracks

* The ability to store file-level metadata and per-stem metadata

3. Bitstream Layout

3.1. Audio Streams

Each stem file may contain an arbitrary number of logical bitstreams containing audio and MUST include at least 3 streams (the original audio and at least two stems). Each stream MUST be encoded using the same codec with the same parameters including bitrate, channel number, channel layout, and sample rate.

The first logical bitstream containing audio data MUST be the final post-mix, mastered audio. This helps preserve backwards compatibility in media players which do not support this format (which typically play the first audio stream found). The remaining logical bitstreams will be individual stems and MUST have the same audio length as the first logical bitstream such that playing each stem stream from the beginning would result in the same audio (excluding mastering) as the final mix present in the first logical bitstream.

For example, if the original logical bitstream is 3 minutes long and the stem file includes a percussion track but the percussion does not start until minute 2 the percussion stem would still be 3 minutes long but would contain a minute of silence at the start of the track.

3.2. Stem Metadata

The following tags MUST be stored in the Vorbis comment block encapsulated in the individual FLAC or Opus audio stream representing each stem. Keys for these tags rae case insensitive.

Tag	Description	Example
STEM:TITLE	Free text, used for the stem name	Percussion
STEM:COLOR	Color representing this track in RGB hex format	#145374

Table 1

3.3. DSP Metadata

For metadata that applies to all the stems it is not desirable to include it in the individual stream metadata blocks for several reasons:

1. In the absence of a standard many applications only store information on the first stream, but in the case of stems this is the one stream to which none of this metadata applies

2. Applications meant for writing general metadata may remove unknown values in the first streams metadata

3. Some stem metadata should be associated with all stem streams, but not the main mix stream and storing it on every stream is not ideal

To work around these limitations stem files store metadata that applies to all stems (notably information about configuring a basic Digital Signal Processor or DSP) in a separate logical bitstream, the first packet of which is structured according to the following table:

Data	Description
8 bytes	0x53 0x74 0x65 0x6d 0x4d 0x65 0x74 0x61 ("StemMeta")
2 bytes	Version number of the metadata logical bitstream (notably this is not the version of the metadata stored in the mapping). These bytes are 0x01 0x00, meaning version 1.0 of the mapping.

Table 2

The remainder of the logical bitstream comprises a Vorbis comment metadata block containing human-readable information coded in UTF-8. The name "Vorbis comment" points to the fact that the Vorbis codec stores such metadata in almost the same way (see [Vorbis]). A stem file MUST NOT contain more than one Vorbis comment metadata block The Vorbis comment metadata block is defined to be identical to the Vorbis comment metadata block defined in [RFC9639] section 8.6, "Vorbis Comment".

The Vorbis comment metadata block SHOULD NOT be used for arbitrary metadata that is unrelated to stems (ie. a track title or author). Vendor specific tags MAY be included in the metadata block. Vendor

specific tags in the block MUST use a vendor specific namespace and MUST NOT prefix their tags with "STEM:". Specific keys for the Vorbis comment metadata block are defined in the "Mastering" section.

4. Mixing

The stem tracks SHOULD NOT have any gain normalization applied. Instead they should retain the same levels as they would have in the final mix present in the first track so that if all stems were played at unity gain the levels would be equivalent to the final mix.

5. Mastering

Because mastering happens post-mix and the stems are pre-mix audio the stem tracks SHOULD NOT have any mastering steps applied. Instead, metadata for configuring a compressor and limiter SHOULD be included in the previously defined Vorbis comment metadata block. After mixing the stems applications MAY choose to feed the mix through a Digital Signal Processor configured with the limiter and compressor settings read from the metadata.

5.1. Compressor Metadata

> TK: I’m not really sure how this works for the NI stems, presumably they have a value range, but that probably depends on the specific compressor used and that’s not likely something we can do in a standard format. Instead we’d have to define exactly how the DSP works and say that you might need to normalize values for specific DSP’s? Unclear how best to handle this.

Tag	Requirement Level	Values
STEM:COMPRESSOR:ENABLED	REQUIRED	"TRUE" or "FALSE"
STEM:COMPRESSOR:RATIO	OPTIONAL	TODO
STEM:COMPRESSOR:OUTPUT_GAIN	OPTIONAL	TODO
STEM:COMPRESSOR:THRESHOLD	OPTIONAL	TODO
STEM:COMPRESSOR:ATTACK	OPTIONAL	TODO
STEM:COMPRESSOR:INPUT_GAIN	OPTIONAL	TODO
STEM:COMPRESSOR:RELEASE	OPTIONAL	TODO
STEM:COMPRESSOR:HP_CUTOFF	OPTIONAL	TODO
STEM:COMPRESSOR:HP_DRY_WET	OPTIONAL	TODO

Table 3

5.2. Limiter Metadata

Tag	Requirement Level	Values
STEM:LIMITER:ENABLED	REQUIRED	"TRUE" or "FALSE"
STEM:LIMITER:RELEASE	OPTIONAL	TODO
STEM:LIMITER:THRESHOLD	OPTIONAL	TODO
STEM:LIMITER:CEILING	OPTIONAL	TODO

Table 4

6. Use with Ogg Skeleton

Ogg [Skeleton] is a format designed to provide structuring information for multi-track Ogg files. Its use is not defined for stem files, however, if a Skeleton logical bitstream is present each fisbone secondary header packet describing a logical bitstream containing a stem track SHOULD set the "role" header to the value "audio/stem". Similarly, the fisbone secondary header packet describing the first logical bitstream containing the main audio SHOULD set the "Role" message header to "audio/main".

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

This document should not affect the security of the Internet.

9. References

9.1. Normative

[RFC3533] Pfeiffer, S., "The Ogg Encapsulation Format Version 0", RFC 3533, DOI 10.17487/RFC3533, May 2003, <https://www.rfc-editor.org/info/rfc3533>.

[RFC5334] Goncalves, I., Pfeiffer, S., and C. Montgomery, "Ogg Media Types", RFC 5334, DOI 10.17487/RFC5334, September 2008, <https://www.rfc-editor.org/info/rfc5334>.

[RFC9639] van Beurden, M.Q.C. and A. Weaver, "Free Lossless Audio Codec (FLAC)", RFC 9639, DOI 10.17487/RFC9639, December 2024, <https://www.rfc-editor.org/info/rfc9639>.

9.2. Informative

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[Vorbis] Xiph.Org Foundation, "Vorbis I specification", 4 July 2020, <https://xiph.org/vorbis/doc/Vorbis_I_spec.html>.

[Skeleton] Xiph.Org Foundation, "Ogg Skeleton 4", 23 May 2016,
<https://wiki.xiph.org/Ogg_Skeleton_4>.

Author’s Address

Sam Whited (editor)
Email: sam@samwhited.com
URI: https://blog.samwhited.com