Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 21 August 2026

ssw. Whited, Ed.

17 February 2026

OGG Stem Files
draft-swhited-ogg-stems-00

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 21 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 least 5 stereo audio tracks

* The ability to syncronize multiple audio tracks

* The ability to store global metadata and per-stem metadata

3. Bitstream Layout

3.1. Audio Streams

> | TK: if we use Skeleton can we include syncronization data so
> | that the stems don't have to have the same length? Or will
> | this just make things harder to decode with no real benefit
> | (since FLAC or Opus would compress the silence)?

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 MUST be the final post-mix, mastered audio. This helps preserve backwards compatibility in media players which do not support a [Skeleton] bitstream. The remaining logical bitstreams will be the stems and MUST have the same audio length as 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. Skeleton

> | TK: Skeleton seems ideal for the stems use case, but I can't
> | figure out if it's still recommended by Xiph.Org or which
> | version we should use (the Xiph.Org website has a page for v3,
> | but the wiki has a v4 that it says it the latest). Maybe it
> | would be better to define our own stream/metadata type and keep
> | everything there? If we're just using Skeleton for per-strem
> | metadata it might be overkill anyways since we'll have to
> | define some sort of global metadata logical bitstream anyways
> | to store the DSP info.

Stem files MUST contain a [Skeleton] bitstream. For each fisbone secondary header packet describing a stem logical bitstream (ie. not the fisbone packet describing the first stream containing the post-mix audio) the following message headers are defined:

Message Header	Requirement Level	Description
Role	REQUIRED	MUST always be "audio/stem"
Title	REQUIRED	Free text, used for the stem name (eg. "Percussion")
Stem-color	OPTIONAL	Color representing this track in RGB hex format, eg. "#145374"

Table 1

The fisbone secondary header packet describing the first logical bitstream containing the main audio MUST set the "Role" message header to "audio/main".

4. Mixing

The stem track 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

> | TK: does it make sense to put these in their own OGG page
> | instead of just putting them in the vorbis comments with
> | everything else? It would make them less likely to be stripped
> | out by metadata editors. Maybe we define a different raw
> | VorbisComment logical bitstream, or use a JSON blob or similar
> | like the NI ones do?

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 stem file. 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.

Metadata used for configuring the compressor should be stored alongside the stem files global metadata (ie. in the primary VorbisComment).

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 2

5.2. Limiter Metadata

Metadata used for configuring the limiter should be stored alongside the stem files global metadata (ie. in the primary VorbisComment).

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 3

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document should not affect the security of the Internet.

8. References

8.1. Normative References

[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>.

[Skeleton] Xiph.Org Foundation, "OGG Skeleton 4", 18 February 2026, <https://wiki.xiph.org/Ogg_Skeleton_4>.

8.2. Informative References

[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>.

Author’s Address

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