

Workgroup: Internet Engineering Task Force
Internet-Draft: draft-valin-opus-extension-00
Updates: [6716](#) (if approved)
Published: 8 March 2023
Intended Status: Standards Track
Expires: 9 September 2023
Authors: JM. Valin T. Terriberry
 Amazon Amazon

Extension Formatting for the Opus Codec

Abstract

This document proposes a mechanism to extend the Opus codec (RFC6716) in a way that maintains inter-operability, while adding optional functionality.

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 9 September 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
 - [1.1. Requirements Language](#)
- [2. Extension Format](#)
 - [2.1. ID 0: Original Padding](#)
 - [2.2. ID 1: Separator](#)
 - [2.3. IDs 2-119: Unassigned](#)
 - [2.4. IDs 120-127: I-D Experimental](#)
- [3. IANA Considerations](#)
- [4. Security Considerations](#)
- [5. References](#)
 - [5.1. Normative References](#)
- [Authors' Addresses](#)

1. Introduction

This document proposes a mechanism to extend the Opus codec [RFC6716] in a way that maintains inter-operability, while adding optional functionality.

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. Extension Format

The Opus padding mechanism provides a safe way to extend the Opus codec while preserving interoperability and without having to transmit any extra packets. [RFC6716] specifies that all padding bytes "MUST be set to zero" by the encoder, while the decoder "MUST accept any value for the padding bytes". In that way, any non-zero padding will indicate to an extended decoder that an extension is present and can be processed. On the other hand, for any all-zero padding, the decoder will just discard the padding like any non-extended decoder. A non-extended decoder receiving a packet with an extension will simply discard the extension and proceed as if none was present.

An extension starts with a byte that signals a 7-bit ID, as well as a binary flag L for length signalling. For extension IDs 1 through 31, L=0 means that no data follows the extension, whereas L=1 means that exactly one byte of extension data follows. For IDs 32 to 127, L=0 signals that the extension data takes up the rest of the padding, and L=1 signals that a length indicator follows. For ID 0, L=0 has the same meaning as for IDs 32 to 127, but L=1 signals a

length of zero (no length indicator follows). In any given packet containing padding, the "rest of the padding" cannot appear more than once. When a length indicator is signalled, the following byte contains a length value from 0 to 254. If the length byte is 255, then the length is 255 plus the length signaled from the next byte, with 255 case being allowed to repeat as long as the size of the padding is not exceeded. Any extension signalled with a length that would cause the decoder to read beyond the bounds of the packet MUST be ignored by the decoder.

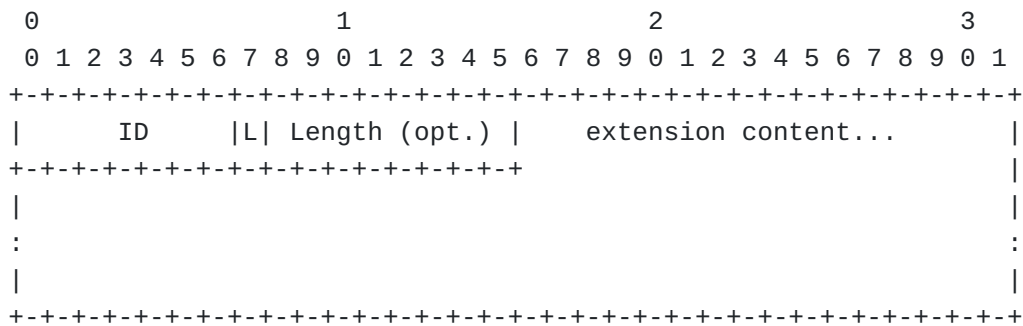


Figure 1: Extension framing

A decoder MUST ignore any extension it does not know, decoding the rest of the packet as if the extension was not present. Additionally, a decoder MAY ignore any other extension even if it technically supports it. An encoder MUST NOT alter the way it encodes the non-extension part of an Opus packet in such a way as to noticeably reduce its quality when decoded with a non-extended decoder.

Open questions:

*should we allow more than one of the same extension ID for the same frame?

*Any additional signalling on 120-127

2.1. ID 0: Original Padding

For compatibility reasons, an ID of 0 means that the content of the extension is actual padding, as originally defined in [\[RFC6716\]](#). As in its original definition, the padding bytes MUST be set to zero by the encoder, while the decoder MUST ignore any non-zero padding. In the case where the L flag is set, the 0x01 header byte is simply skipped and extension decoding continues from the next byte. This can be useful as a way to insert padding one byte at a time, since

appending zeros at the end may cause an increase in size from having to signal a multi-byte length indicator for the last extension.

2.2. ID 1: Separator

In the case where multiple frames are packed inside the same packet, there may be a need to specify which extension(s) apply to which frame. By default, all extensions apply to the first frame in the packet. Any time a separator with $L=0$ is encountered when parsing extensions sequentially, the associated frame is increased by one. If $L=1$ is used, the following data byte indicates the absolute value of the new associated frame. That value **MUST NOT** exceed the bound equal to the number of frames in the packet, minus one (indexing starts at zero). Similarly, $L=0$ separators **MUST NOT** cause the associated frame to exceed the above bound. The decoder **MUST** ignore all extensions associated with an out-of-bound frame index.

2.3. IDs 2-119: Unassigned

These extensions are to be define in their own respective documents and the IDs are to be assigned by IANA. Note that the definition of the L flag is already defined for all these unassigned IDs because a decoder must know how to skip extensions it doesn't know about. Due to potential for interaction between extensions, new extensions are to be assigned with the "Standards Action" policy defined by [\[RFC8126\]](#).

2.4. IDs 120-127: I-D Experimental

We reserve these 8 IDs for experimental extensions, such that extensions defined in Internet-Drafts can be tested before they become RFC without causing possible interoperability issues should their bitstream definitions change.

3. IANA Considerations

This document defines a new registry "Opus Extension IDs" that allocates individual IDs to individual extensions to be defined in the future. Moreover, this document already defines the following IDs:

Extension ID	Description	Reference
0	Original padding definition	Defined in Section 2.1
1	Frame separator	Defined in Section 2.2 .
2-119	Unassigned	To be assigned with the "Standards Action" policy [RFC8126]

Extension ID	Description	Reference
120-127	Experimental Internet-Draft implementations	Defined in Section 2.4 , following the "Experimental Use" policy [RFC8126]

Table 1

Note that for forward compatibility, any extension defined in the future MUST use the definition of the L flag that is dictated ([Section 2](#)) by its ID value.

4. Security Considerations

This document does not add security considerations beyond those already documented in [[RFC6716](#)]. Future Opus extensions may have their own security implications.

5. References

5.1. Normative 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>>.
- [[RFC6716](#)] Valin, JM., Vos, K., and T. Terriberry, "Definition of the Opus Audio Codec", RFC 6716, DOI 10.17487/RFC6716, September 2012, <<https://www.rfc-editor.org/info/rfc6716>>.
- [[RFC8126](#)] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

Authors' Addresses

Jean-Marc Valin
Amazon
Canada

Email: jmvalin@amazon.com

Timothy B. Terriberry
Amazon

United States of America

Email: territim@amazon.com