

HyBi Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 29, 2011

T. Yoshino
Google, Inc.
April 27, 2011

WebSocket Per-frame DEFLATE Extension
draft-tyoshino-hybi-websocket-perframe-deflate-00

Abstract

This specification defines a per-frame DEFLATE compression extension for the WebSocket protocol. This extension enables compressing only the Application Data part of WebSocket frames using DEFLATE compression while leaving the other parts such as opcode, payload length, the Extension Data part untouched.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 29, 2011.

Copyright Notice

Copyright (c) 2011 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Background	3
1.2.	Extension Overview	3
2.	Conformance Requirements	4
3.	Extension Negotiation	5
4.	Detailed Specification	6
5.	Security Considerations	7
6.	IANA Considerations	8
6.1.	Registration of the "deflate-application-data" WebSocket Extension Keyword	8
7.	Acknowledgements	9
8.	References	10
8.1.	Normative References	10
8.2.	Informative References	10
	Author's Address	11

1. Introduction

1.1. Background

This section is non-normative.

The WebSocket protocol is designed to allow for extensions [[I-D.ietf-hybi-thewebsocketprotocol](#)]. This enables use of compression extension to reduce the amount of data to be sent over network.

The WebSocket protocol specification contains built-in compression extension "deflate-stream" that applies DEFLATE to everything including header octets such as opcode and payload length. It's simple, however it requires intermediaries that are not interested in the contents of the Application Data part to decompress received octets using DEFLATE to see opcode, etc.

1.2. Extension Overview

This section is non-normative.

Per-frame DEFLATE extension adds another DEFLATE [[RFC1951](#)] based compression option to the WebSocket protocol. Per-frame DEFLATE extension compresses only the octets in the Application Data part of a frame. Header octets are unchanged by this extension.

To align the end of compressed data to octet boundary, this extension uses the algorithm described in the [Section 2.1](#) of the PPP Deflate Protocol [[RFC1979](#)]. The frame sender keeps using the same LZ77 sliding window across multiple frames [[LZ77](#)].

This extension doesn't use any of reserved bits, opcodes or Extension Data part.

The simplest "Sec-WebSocket-Extensions" header in the opening handshake from the client that negotiates only per-frame DEFLATE extension is the following:

Sec-WebSocket-Extensions: deflate-application-data

The server also sends the same header to accept the use of this compression extension.

2. Conformance Requirements

Everything in this specification except for sections explicitly marked non-normative is normative.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

3. Extension Negotiation

The registered extension token for this extension is "deflate-application-data".

To use per-frame DEFLATE extension, a user-agent MUST send an opening handshake request with the "deflate-application-data" extension token included in its "Sec-WebSocket-Extensions" header.

To accept use of per-frame DEFLATE extension requested by the user-agent, a server MUST send an opening handshake response with the "deflate-application-data" extension token included in its "Sec-WebSocket-Extensions" header.

Any extension parameter MUST NOT be attached to "deflate-application-data" extension token.

Once per-frame DEFLATE extension is accepted, both endpoints MUST use the algorithm described in [Section 4](#) in sending frames.

4. Detailed Specification

This extension does not use any of reserved bits, opcodes or the Extension Data part in a WebSocket frame.

This extension operates only on Data Frames, and only on the Application Data therein (it does not affect the Extension Data portion of the Payload Data).

A sender using this extension MUST compress all octets of the Application Data part using DEFLATE [[RFC1951](#)]. The sender MAY use multiple blocks to compress data for one frame. The sender MAY use blocks with any effective "BTYPE" defined in [[RFC1951](#)] for one frame. "BFINAL" bit MUST NOT be set for any block. The sender MUST use the algorithm described in the Data paragraph of the [Section 2.1](#) of the PPP Deflate Protocol [[RFC1979](#)] to align the end of the compressed data to octet boundary. That is that the sender MUST append an empty uncompressed block to the tail of the compressed data, and then remove 4 octets from the tail of the concatenated data. The sender MUST put this data (the octets left after removing 4 octets) in the Application Data part instead of the original data. The payload length field of the frame MUST be the number of octets in the Extension Data part and the Application Data part after this process. The senders MAY keep using the same LZ77 sliding window across multiple frames [[LZ77](#)].

A receiver MUST append 4 octets of 0x00 0x00 0xff 0xff to the tail of the received octets in the Application Data part and decompress the concatenated octets using DEFLATE to get the original octets for the Application Data part. The receiver MUST keep using the same LZ77 sliding window for all frames on the same WebSocket connection. The receiver MUST use 32768 byte LZ77 sliding window.

5. Security Considerations

There's no security concern for now.

6. IANA Considerations

6.1. Registration of the "deflate-application-data" WebSocket Extension Keyword

Name of token.

deflate-application-data

Author/Change controller

IETF <iesg@ietf.org>

Contact

HYBi <hybi@ietf.org>

References

RFC XXX

The "deflate-application-data" token is used in the "Sec-WebSocket-Extensions" header in the WebSocket opening handshake to negotiate use of per-frame DEFLATE compression extension.

[7.](#) Acknowledgements

Special thanks to Patrick McManus who wrote up the initial version of DEFLATE-based compression extension specification for the WebSocket protocol which I referred when I write this specification.

8. References

8.1. Normative References

- [I-D.ietf-hybi-thewebsocketprotocol]
Fette, I., "The WebSocket protocol",
[draft-ietf-hybi-thewebsocketprotocol](#) (work in progress),
April 2011.
- [RFC1951] Deutsch, P., "DEFLATE Compressed Data Format Specification
version 1.3", [RFC 1951](#), May 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [LZ77] Ziv, J. and A. Lempel, "A Universal Algorithm for
Sequential Data Compression", IEEE Transactions on
Information Theory, Vol. 23, No. 3, pp. 337-343.

8.2. Informative References

- [RFC1979] Woods, J., "PPP Deflate Protocol", [RFC 1979](#), August 1996.

Author's Address

Takeshi Yoshino
Google, Inc.

Email: tyoshino@google.com