

HTTPbis
Bishop
Internet-Draft
Microsoft

M.

Intended status: Informational
2016
Expires: May 19, 2017

November 15,

HTTP/2 Extended SETTINGS Extension draft-bishop-httpbis-extended-settings-01

Abstract

HTTP/2 defines the SETTINGS frame to contain a single 32-bit value per setting. While this is sufficient to convey everything used in the core HTTP/2 specification, some protocols will require more complex values, such as arrays of code-points or strings.

For such protocols, this extension defines a parallel to the SETTINGS frame, EXTENDED_SETTINGS, where the value of a setting is not a 32-bit value, but a variable-length opaque data blob whose interpretation is subject entirely to the definition of the protocol using it.

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 <http://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 May 19, 2017.

Copyright Notice

Copyright (c) 2016 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

Bishop
1]

Expires May 19, 2017

[Page

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 1. Introduction
- 2 2. Detection of Support
- 3 3. Extension Frame Types
 - 3 3.1. EXTENDED_SETTINGS Frame
 - 3 3.1.1. EXTENDED_SETTINGS Format
 - 4 3.2. EXTENDED_SETTINGS_ACK Frame
- 5 4. Settings Synchronization
- 5 5. Security Considerations
- 6 6. IANA Considerations
 - 6 6.1. Signature Methods
 - 6 6.2. HTTP/2 SETTINGS_HTTP_CERT_AUTH Setting
 - 6 6.3. New HTTP/2 Frames
- 7 7. References
 - 7 7.1. Normative References
 - 7 7.2. Informative References
- 7 Author's Address
- 8

1. Introduction

In [[I-D.bishop-httpbis-http2-additional-certs](#)], values for which IANA registries already exist must be communicated between two HTTP/2 implementations. Since the SETTINGS frame constrains setting values to a 32-bit value, the existing version of that draft divides the 32-bit value into halves and dedicates bits to each currently-known value. This requires the creation of two duplicative IANA registries, and enormously constrains future extensibility since

each

future supported value will consume one of only sixteen bits. It also causes divergence from other places in the protocol where a bitmask is not required and a more sensible value can be used.

[\[MS-HTTP2E\]](#), likewise, defines a very limited bitmap in the 32-bit value - two bits are defined, all others are reserved (and not useful). The setting fits easily in a single byte, and need not consume a four-byte value every time it is transferred.

Alternately, a number of recent and in-progress HTTP/2 extensions describe properties of the connection that are informative to the peer ([\[RFC7838\]](#), [\[I-D.ietf-httpbis-origin-frame\]](#)). These are essentially settings that did not fit into a 32-bit value.

Each extension could define its own SETTINGS-equivalent frame to carry its own data, as these extensions already have, but to do so every time a new extension might require such a capability seems similarly wasteful, given the limited frame type space (also an IANA registry).

2. Detection of Support

An HTTP/2 peer that supports the EXTENDED_SETTINGS frame indicate this using the HTTP/2 "SETTINGS_EXTENDED_SETTINGS" (0xSETTING-TBD) setting.

The initial value for the "SETTINGS_EXTENDED_SETTINGS" setting is 0 (0x00), indicating that the peer does not support the EXTENDED_SETTINGS frame. A peer that is able to parse the EXTENDED_SETTINGS frame MUST set this value to 1 (0x01).

This setting MUST be sent before any of the frame types in [Section 3](#) are sent, but those frames MAY be sent before the setting is acknowledged and MAY be sent regardless of whether the peer has sent this setting.

3. Extension Frame Types

3.1. EXTENDED_SETTINGS Frame

The EXTENDED_SETTINGS frame (type=0xTBD1) conveys configuration parameters that affect how endpoints communicate, such as preferences

and constraints on peer behavior which occur in a form other than a 32-bit value. The EXTENDED_SETTINGS frame is also used to acknowledge the receipt of those parameters. Individually, an EXTENDED_SETTINGS parameter can also be referred to as a "setting".

EXTENDED_SETTINGS parameters are not negotiated; they describe characteristics of the sending peer, which are used by the receiving peer. However, a negotiation can be implied by the use of EXTENDED_SETTINGS - a peer uses EXTENDED_SETTINGS to advertise a set of supported values. The recipient can then choose which entries from this list are also acceptable and proceed with the value it has chosen. (This choice could be announced in a field of an extension frame, or in a value in SETTINGS.)

Different values for the same parameter can be advertised by each peer. For example, a server might support many different signing algorithms, while a resource constrained client has only one or two that it can validate.

Bishop
3]

Expires May 19, 2017

[Page

An EXTENDED_SETTINGS frame MAY be sent at any time by either endpoint over the lifetime of the connection.

Each parameter in an EXTENDED_SETTINGS frame replaces any existing value for that parameter. Parameters are processed in the order in which they appear, and a receiver of an EXTENDED_SETTINGS frame does not need to maintain any state other than the current value of its parameters. Therefore, the value of a EXTENDED_SETTINGS parameter is the last value that is seen by a receiver.

EXTENDED_SETTINGS parameters can request acknowledgement by the receiving peer. To enable this, the EXTENDED_SETTINGS frame defines the following flag:

REQUEST_ACK (0x1): When set, bit 0 indicates that this frame contains values which the sender wants to know were understood and applied. For more information, see [Section 4](#).

Like SETTINGS frames, EXTENDED_SETTINGS frames always apply to a connection, never a single stream. The stream identifier for an EXTENDED_SETTINGS frame MUST be zero (0x0). If an endpoint receives an EXTENDED_SETTINGS frame whose stream identifier field is anything other than 0x0, the endpoint MUST respond with a connection error ([Section 5.4.1](#)) of type PROTOCOL_ERROR.

The EXTENDED_SETTINGS frame affects connection state. A badly formed or incomplete EXTENDED_SETTINGS frame MUST be treated as a connection error ([Section 5.4.1](#)) of type PROTOCOL_ERROR.

3.1.1. EXTENDED_SETTINGS Format

The payload of a SETTINGS frame consists of zero or more parameters, each consisting of an unsigned 16-bit setting identifier and a length-prefixed binary value.

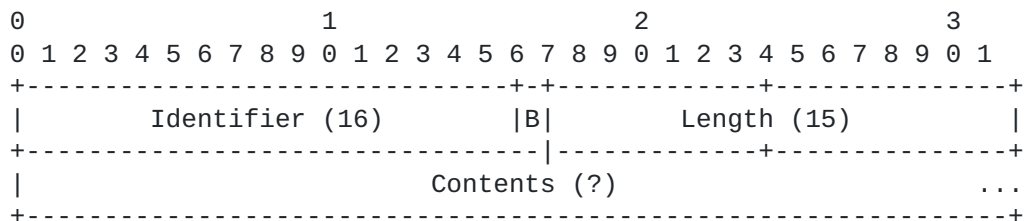


Figure 1: EXTENDED_SETTINGS frame payload

A zero-length content indicates that the setting value is a Boolean given by the B bit. If Length is not zero, the B bit MUST be zero,

Bishop
4]

Expires May 19, 2017

[Page

and MUST be ignored by receivers. The initial value of each setting is "false."

An implementation MUST ignore the contents for any EXTENDED_SETTINGS identifier it does not understand.

3.2. EXTENDED_SETTINGS_ACK Frame

The EXTENDED_SETTINGS_ACK frame acknowledges receipt and application of specific values in the peer's SETTINGS frame. It contains a list of EXTENDED_SETTINGS identifiers which the sender has understood and applied. This list MAY be empty.

Any EXTENDED_SETTINGS_ACK frame whose length is not a multiple of two

bytes MUST be treated as a connection error ([\[RFC7540\]](#) [section 5.4.1](#)) of type "FRAME_SIZE_ERROR".

4. Settings Synchronization

Some values in EXTENDED_SETTINGS benefit from or require an understanding of when the peer has received and applied the changed parameter values. In order to provide such synchronization timepoints, the recipient of a EXTENDED_SETTINGS frame MUST apply the

updated parameters as soon as possible upon receipt. The values in the EXTENDED_SETTINGS frame MUST be processed in the order they appear, with no other frame processing between values. Unsupported parameters MUST be ignored.

Once all values have been processed, if the REQUEST_ACK flag was set,

the recipient MUST immediately emit a EXTENDED_SETTINGS_ACK frame listing the identifiers whose values were understood and applied. (If none of the values were understood, the EXTENDED_SETTINGS_ACK frame will be empty, but MUST still be sent.) Upon receiving an EXTENDED_SETTINGS_ACK frame, the sender of the altered parameters can

rely on the setting having been applied.

If the sender of an EXTENDED_SETTINGS frame with the "REQUEST_ACK" flag set does not receive an acknowledgement from a peer that has sent the "SETTINGS_EXTENDED_SETTINGS" setting within a reasonable amount of time, it MAY issue a connection error ([\[RFC7540\]](#) [Section 5.4.1](#)) of type SETTINGS_TIMEOUT. This error MUST NOT be sent

if the peer has not previously advertised support for EXTENDED_SETTINGS.

Bishop
5]

Expires May 19, 2017

[Page

5. Security Considerations

Because these frames can be used to request that peers retain potentially-large state, implementations need to use caution in their retention policies. Values which are not understood MUST be discarded in order to protect against increased memory usage. Specifications which make use of EXTENDED_SETTINGS MUST include details about how the contents can be parsed and stored, and SHOULD include details about how the information can be compressed and when it can safely be discarded.

6. IANA Considerations

This draft establishes one new registry and add three entries across two existing registries.

The HTTP/2 "SETTINGS_EXTENDED_SETTINGS" setting is registered in [Section 6.2](#). Two frame types are registered in [Section 6.3](#).

6.1. Signature Methods

This document establishes a registry for HTTP/2 extended settings. The "HTTP/2 Extended Settings" registry manages a 16-bit space. The "HTTP/2 Extended Settings" registry operates under the "Expert Review" policy [[RFC5226](#)] for values in the range from 0x0000 to 0xffff, with values between and 0xf000 and 0xffff being reserved for Experimental Use.

New registrations are advised to provide the following information:

Name: A symbolic name for the setting. Specifying a setting name is optional.

Code: The 16-bit code assigned to the setting.

Specification: An optional reference to a specification that describes the use of the setting.

No entries are registered by this document.

6.2. HTTP/2 SETTINGS_HTTP_CERT_AUTH Setting

The "SETTINGS_EXTENDED_SETTINGS" setting is registered in the "HTTP/2 Settings" registry established in [[RFC7540](#)].

Name: SETTINGS_EXTENDED_SETTINGS

Code: 0xSETTING-TBD

Bishop
6]

Expires May 19, 2017

[Page

Initial Value: 0

Specification: This document.

6.3. New HTTP/2 Frames

Two new frame types are registered in the "HTTP/2 Frame Types" registry established in [RFC7540]. The entries in the following table are registered by this document.

Frame Type	Code	Specification
EXTENDED_SETTINGS	0xFRAME-TBD1	{{settings-frame}}
EXTENDED_SETTINGS_ACK	0xFRAME-TBD2	{{ack}}

Figure 2

7. References

7.1. Normative References

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", [RFC 7540](#), DOI 10.17487/RFC7540, May 2015, <<http://www.rfc-editor.org/info/rfc7540>>.

7.2. Informative References

- [I-D.bishop-httpbis-http2-additional-certs] Bishop, M. and M. Thomson, "Secondary Certificate Authentication in HTTP/2", [draft-bishop-httpbis-http2-additional-certs-02](#) (work in progress), October 2016.
- [I-D.ietf-httpbis-origin-frame] Nottingham, M. and E. Nygren, "The ORIGIN HTTP/2 Frame", [draft-ietf-httpbis-origin-frame-01](#) (work in progress), September 2016.

Bishop
7]

Expires May 19, 2017

[Page

[MS-HTTP2E]

"Hypertext Transfer Protocol Version 2 (HTTP/2)
Extension", October 2015,
<[http://download.microsoft.com/download/9/5/
E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/\[MS-HTTP2E\].pdf](http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/[MS-HTTP2E].pdf)>.

[RFC7838] Nottingham, M., McManus, P., and J. Reschke, "HTTP
Alternative Services", [RFC 7838](#), DOI 10.17487/RFC7838,
April 2016, <<http://www.rfc-editor.org/info/rfc7838>>.

Author's Address

Mike Bishop
Microsoft

Email: michael.bishop@microsoft.com

Bishop
8]

Expires May 19, 2017

[Page