

HTTP
Internet-Draft
Intended status: Standards Track
Expires: 6 April 2026

Y. Rosomakho
Zscaler
D. Schinazi
Google LLC
3 October 2025

Unbound DATA Frames in HTTP/3
draft-rosomakho-httpbis-h3-unbound-data-00

Abstract

This document defines a new HTTP/3 frame type, UNBOUND_DATA, and a corresponding SETTINGS parameter that enables endpoints to negotiate its use. When an endpoint sends an UNBOUND_DATA frame on a request or response stream, it indicates that all subsequent octets on that stream are interpreted as data. This applies both to message body data and to octets transmitted after CONNECT or extended CONNECT. The use of UNBOUND_DATA removes the need to encapsulate each portion of the data in DATA frames, reducing framing overhead and simplifying transmission of long-lived or indeterminate-length payloads.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://yaroslavros.github.io/draft-httpbis-h3-unbound-data/draft-rosomakho-httpbis-h3-unbound-data.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-rosomakho-httpbis-h3-unbound-data/>.

Discussion of this document takes place on the HTTP Working Group mailing list (<mailto:ietf-http-wg@w3.org>), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/>.

Source for this draft and an issue tracker can be found at <https://github.com/yaroslavros/draft-httpbis-h3-unbound-data>.

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 6 April 2026.

Copyright Notice

Copyright (c) 2025 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	2
2. Conventions and Definitions	3
3. Capability Negotiation	3
4. UNBOUND_DATA Frame	4
4.1. Frame Layout	4
4.2. Semantics	4
5. Stream State Transitions	5
6. Security Considerations	6
7. IANA Considerations	6
7.1. HTTP/3 Setting	6
7.2. HTTP/3 Frame Type	6
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Acknowledgments	8
Authors' Addresses	8

1. Introduction

[HTTP/3] transmits message content on client-initiated bidirectional QUIC streams. On these streams, the request and response messages are carried using a sequence of HTTP/3 frames. The DATA frame is used to encapsulate octets of a message body, or the opaque data associated with CONNECT and its extensions.

When the size of the body is unknown in advance, or when data is generated incrementally such as media streaming, [WebSockets], [WebTransport], or other tunneled protocols using CONNECT and its extensions, senders typically generate many DATA frames. Although DATA frames are lightweight, each adds framing overhead and requires the sender to manage frame boundaries. For long-lived or high-volume streams, this overhead is unnecessary because the end of the QUIC stream already provides a natural message boundary.

This document defines a new HTTP/3 frame type, UNBOUND_DATA, and a corresponding SETTINGS parameter that endpoints use to negotiate support. Once an UNBOUND_DATA frame is sent on a request or response stream, all subsequent octets on that stream are interpreted as data. This mechanism eliminates the need to encapsulate each portion of the body in separate DATA frames.

The goals of UNBOUND_DATA are:

- * Reduce framing overhead for large or indeterminate-length message bodies.
- * Simplify sender and receiver state machines by eliminating repeated DATA frame headers.
- * Enable efficient transport of long-lived data flows such as streaming APIs, media delivery, or tunneled protocols.

The use of UNBOUND_DATA does not alter HTTP semantics, flow control, or prioritization; it is strictly a framing optimization.

2. Conventions and Definitions

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.

3. Capability Negotiation

Endpoints indicate support for unbound data transmission by sending the SETTINGS_ENABLE_UNBOUND_DATA (0x282cf6bb) setting with a value of 1.

The valid values of the SETTINGS_ENABLE_UNBOUND_DATA setting are 0 and 1. If the SETTINGS_ENABLE_UNBOUND_DATA setting is received with a different value, the receiver MUST treat it as a connection error of type H3_SETTINGS_ERROR.

A value of 1 indicates that the sender of the SETTINGS frame is willing to receive UNBOUND_DATA frames.

Endpoints MUST NOT send an UNBOUND_DATA frame to a peer that has not advertised SETTINGS_ENABLE_UNBOUND_DATA with a value of 1. Endpoints that receive an UNBOUND_DATA frame without having advertised support MUST treat it as a connection error of type H3_FRAME_UNEXPECTED.

The SETTINGS_ENABLE_UNBOUND_DATA parameter is directional: each endpoint independently advertises whether it accepts receiving UNBOUND_DATA. An endpoint that has not indicated support cannot be assumed to understand or correctly process the frame.

4. UNBOUND_DATA Frame

The UNBOUND_DATA frame (type=0x2a937388) is used on request or response streams to indicate that all subsequent octets on the stream are interpreted as data. This data can represent an HTTP message body or the data stream as defined in Section 3.1 of [HTTP-DGRAM].

4.1. Frame Layout

```
UNBOUND_DATA Frame {  
  Type (i) = 0x2a937388,  
  Length (8) = 0,  
}
```

Figure 1: HTTP/3 UNBOUND_DATA Frame

The UNBOUND_DATA frame has no payload. The Length field of the frame MUST be zero. If a nonzero length is received, the endpoint MUST treat this as a connection error of type H3_FRAME_ERROR.

The UNBOUND_DATA frame is only valid on request or response streams. It is invalid on control streams, QPACK encoder/decoder streams, or push streams. If an endpoint receives an UNBOUND_DATA frame on a stream that isn't a client-initiated bidirectional stream, it MUST treat it as a connection error of type H3_FRAME_UNEXPECTED.

Similar to DATA frames, endpoints MUST send a HEADERS frame before sending an UNBOUND_DATA frame on a given stream. Receipt of an UNBOUND_DATA frame on a stream that hasn't received a HEADERS frame MUST be treated as a connection error of type H3_FRAME_UNEXPECTED.

4.2. Semantics

Upon receiving an UNBOUND_DATA frame on a request or response stream, the receiver enters unbound mode for that stream. In unbound mode:

- * All remaining octets on the stream, up to the QUIC FIN, are interpreted as data.
- * No further HTTP/3 frames (including DATA, HEADERS, or any extension frames) can be received on the stream.
- * The end of the data is indicated by the QUIC FIN on the stream.
- * If a Content-Length header was included, the recipient needs to ensure that the combined length of all received data (in both DATA and UNBOUND_DATA frames) matches the content length from the header.

5. Stream State Transitions

The use of the UNBOUND_DATA frame modifies the sequence of frames exchanged on request and response streams.

In normal operation, a request or response body is carried as a sequence of one or more DATA frames, followed optionally by a HEADERS frame containing trailers:

```

New bi-direcitonal QUIC stream ----> +-----+
                                       | HEADERS (headers) |
                                       +-----+
                                       | [ DATA ... ]      |
                                       +-----+
                                       | [ HEADERS (trailers) ] |
QUIC FIN ----> +-----+

```

Figure 2: Regular HTTP/3 Frame sequence on bi-directional stream

When UNBOUND_DATA is used, the sender signals that all subsequent octets on the stream are data. Regular DATA frames MAY be sent on a stream prior to the UNBOUND_DATA. After the UNBOUND_DATA frame, the sender cannot send any further HTTP/3 frames on the stream. The end of the body is signaled by the QUIC stream FIN:

```

New bi-directional QUIC stream ----> +-----+
                                       | HEADERS (headers) |
                                       +-----+
                                       | [ DATA ... ]      |
                                       +-----+
                                       | UNBOUND_DATA        |
                                       +-----+
                                       | Raw octets (data only) |
QUIC FIN ----> +-----+

```

Figure 3: HTTP/3 Frame sequence with UNBOUND_DATA

Since the recipient of an UNBOUND_DATA will no longer parse frame types on the stream after its receipt, it is not possible to send other frames after the UNBOUND_DATA. If that is required, for example if the sender wishes to send trailers, then the UNBOUND_DATA frame cannot be used.

6. Security Considerations

The introduction of UNBOUND_DATA does not alter the security properties of HTTP/3 or QUIC. It only changes how message bodies or tunneled octets are framed on request and response streams.

7. IANA Considerations

7.1. HTTP/3 Setting

This specification registers the following entry in the "HTTP/3 Settings" registry defined in [HTTP/3]:

- * Code: 0x282cf6bb
- * Setting Name: SETTINGS_ENABLE_UNBOUND_DATA
- * Default: 0
- * Status: provisional (permanent if this document is approved)
- * Reference: This document
- * Change Controller: Yaroslav Rosomakho (IETF if this document is approved)
- * Contact: yrosomakho@zscaler.com (HTTP_WG; HTTP working group; ietf-http-wg@w3.org if this document is approved)
- * Notes: None

7.2. HTTP/3 Frame Type

This specification registers the following entry in the "HTTP/3 Frame Types" registry defined in [HTTP/3]:

- * Value: 0x2a937388
- * Frame Type: UNBOUND_DATA

- * Status: provisional (permanent if this document is approved)
- * Reference: This document
- * Change Controller: Yaroslav Rosomakho (IETF if this document is approved)
- * Contact: yrosomakho@zscaler.com (HTTP_WG; HTTP working group; ietf-http-wg@w3.org if this document is approved)
- * Notes: None

8. References

8.1. Normative References

- [HTTP-DGRAM] Schinazi, D. and L. Pardue, "HTTP Datagrams and the Capsule Protocol", RFC 9297, DOI 10.17487/RFC9297, August 2022, <<https://www.rfc-editor.org/rfc/rfc9297>>.
- [HTTP/3] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <<https://www.rfc-editor.org/rfc/rfc9114>>.
- [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/rfc/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/rfc/rfc8174>>.

8.2. Informative References

- [WebSockets] Hamilton, R., "Bootstrapping WebSockets with HTTP/3", RFC 9220, DOI 10.17487/RFC9220, June 2022, <<https://www.rfc-editor.org/rfc/rfc9220>>.
- [WebTransport] Frindell, A., Kinnear, E., and V. Vasiliev, "WebTransport over HTTP/3", Work in Progress, Internet-Draft, draft-ietf-webtrans-http3-13, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-http3-13>>.

Acknowledgments

This specification originated from discussions with Christian Huitema and Alan Frindell, whose ideas and feedback helped shape the approach described in this document. The authors thanks them for their valuable contributions.

Authors' Addresses

Yaroslav Rosomakho
Zscaler
Email: yrosomakho@zscaler.com

David Schinazi
Google LLC
Email: dschinazi.ietf@gmail.com