

The ORIGIN HTTP/2 Frame

Abstract

This document specifies the ORIGIN frame for HTTP/2, to indicate what origins are available on a given connection.

Status of this Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 7841](#)¹.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8336>².

Copyright Notice

Copyright (c) 2018 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>³) 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.

¹ <https://www.rfc-editor.org/rfc/rfc7841.html#section-2>

² <https://www.rfc-editor.org/info/rfc8336>

³ <https://trustee.ietf.org/license-info>

Table of Contents

1 Introduction.....	3
1.1 Notational Conventions.....	3
2 The ORIGIN HTTP/2 Frame.....	4
2.1 Syntax.....	4
2.2 Processing ORIGIN Frames.....	4
2.3 The Origin Set.....	5
2.4 Authority, Push, and Coalescing with ORIGIN.....	5
3 IANA Considerations.....	7
4 Security Considerations.....	8
5 References.....	9
5.1 Normative References.....	9
5.2 Informative References.....	9
Appendix A Non-Normative Processing Algorithm.....	10
Appendix B Operational Considerations for Servers.....	11
Authors' Addresses.....	12

1. Introduction

HTTP/2 [RFC7540] allows clients to coalesce different origins [RFC6454] onto the same connection when certain conditions are met. However, in some cases, a connection is not usable for a coalesced origin, so the 421 (Misdirected Request) status code ([RFC7540], [Section 9.1.2](#)) was defined.

Using a status code in this manner allows clients to recover from misdirected requests, but at the penalty of adding latency. To address that, this specification defines a new HTTP/2 frame type, "ORIGIN", to allow servers to indicate for which origins a connection is usable.

Additionally, experience has shown that HTTP/2's requirement to establish server authority using both DNS and the server's certificate is onerous. This specification relaxes the requirement to check DNS when the ORIGIN frame is in use. Doing so has additional benefits, such as removing the latency associated with some DNS lookups.

1.1. Notational Conventions

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. The ORIGIN HTTP/2 Frame

This document defines a new HTTP/2 frame type ([RFC7540], [Section 4](#)) called ORIGIN, that allows a server to indicate what origin(s) [RFC6454] the server would like the client to consider as members of the Origin Set ([Section 2.3](#)) for the connection within which it occurs.

2.1. Syntax

The ORIGIN frame type is 0xc (decimal 12) and contains zero or more instances of the Origin-Entry field.

```
+-----+-----+
|          Origin-Entry (*)          ...
+-----+-----+
```

An Origin-Entry is a length-delimited string:

```
+-----+-----+-----+
|          Origin-Len (16)          | ASCII-Origin?          ...
+-----+-----+-----+
```

Specifically:

Origin-Len: An unsigned, 16-bit integer indicating the length, in octets, of the ASCII-Origin field.

Origin: An OPTIONAL sequence of characters containing the ASCII serialization of an origin ([RFC6454], [Section 6.2](#)) that the sender asserts this connection is or could be authoritative for.

The ORIGIN frame does not define any flags. However, future updates to this specification MAY define flags. See [Section 2.2](#).

2.2. Processing ORIGIN Frames

The ORIGIN frame is a non-critical extension to HTTP/2. Endpoints that do not support this frame can safely ignore it upon receipt.

When received by an implementing client, it is used to initialize and manipulate the Origin Set (see [Section 2.3](#)), thereby changing how the client establishes authority for origin servers (see [Section 2.4](#)).

The ORIGIN frame MUST be sent on stream 0; an ORIGIN frame on any other stream is invalid and MUST be ignored.

Likewise, the ORIGIN frame is only valid on connections with the "h2" protocol identifier or when specifically nominated by the protocol's definition; it MUST be ignored when received on a connection with the "h2c" protocol identifier.

This specification does not define any flags for the ORIGIN frame, but future updates to this specification (through IETF consensus) might use them to change its semantics. The first four flags (0x1, 0x2, 0x4, and 0x8) are reserved for backwards-incompatible changes; therefore, when any of them are set, the ORIGIN frame containing them MUST be ignored by clients conforming to this specification, unless the flag's semantics are understood. The remaining flags are reserved for backwards-compatible changes and do not affect processing by clients conformant to this specification.

The ORIGIN frame describes a property of the connection and therefore is processed hop by hop. An intermediary MUST NOT forward ORIGIN frames. Clients configured to use a proxy MUST ignore any ORIGIN frames received from it.

Each ASCII-Origin field in the frame's payload MUST be parsed as an ASCII serialization of an origin ([RFC6454], [Section 6.2](#)). If parsing fails, the field MUST be ignored.

Note that the ORIGIN frame does not support wildcard names (e.g., `*.example.com`) in Origin-Entry. As a result, sending ORIGIN when a wildcard certificate is in use effectively disables any origins that are not explicitly listed in the ORIGIN frame(s) (when the client understands ORIGIN).

See [Appendix A](#) for an illustrative algorithm for processing ORIGIN frames.

2.3. The Origin Set

The set of origins (as per [\[RFC6454\]](#)) that a given connection might be used for is known in this specification as the Origin Set.

By default, the Origin Set for a connection is uninitialized. An uninitialized Origin Set means that clients apply the coalescing rules from [Section 9.1.1](#) of [\[RFC7540\]](#).

When an ORIGIN frame is first received and successfully processed by a client, the connection's Origin Set is defined to contain an initial origin. The initial origin is composed from:

- Scheme: "https"
- Host: the value sent in Server Name Indication (SNI) ([\[RFC6066\]](#), [Section 3](#)) converted to lower case; if SNI is not present, the remote address of the connection (i.e., the server's IP address)
- Port: the remote port of the connection (i.e., the server's port)

The contents of that ORIGIN frame (and subsequent ones) allow the server to incrementally add new origins to the Origin Set, as described in [Section 2.2](#).

The Origin Set is also affected by the 421 (Misdirected Request) response status code, as defined in [\[RFC7540\]](#), [Section 9.1.2](#). Upon receipt of a response with this status code, implementing clients MUST create the ASCII serialization of the corresponding request's origin (as per [\[RFC6454\]](#), [Section 6.2](#)) and remove it from the connection's Origin Set, if present.

Note: When sending an ORIGIN frame to a connection that is initialized as an alternative service [\[RFC7838\]](#), the initial Origin Set ([Section 2.3](#)) will contain an origin with the appropriate scheme and hostname (since RFC 7838 specifies that the origin's hostname be sent in SNI). However, it is possible that the port will be different than that of the intended origin, since the initial Origin Set is calculated using the actual port in use, which can be different for the alternative service. In this case, the intended origin needs to be sent in the ORIGIN frame explicitly.

For example, a client making requests for "https://example.com" is directed to an alternative service at ("h2", "x.example.net", "8443"). If this alternative service sends an ORIGIN frame, the initial origin will be "https://example.com:8443". The client will not be able to use the alternative service to make requests for "https://example.com" unless that origin is explicitly included in the ORIGIN frame.

2.4. Authority, Push, and Coalescing with ORIGIN

[Section 10.1](#) of [\[RFC7540\]](#) uses both DNS and the presented Transport Layer Security (TLS) certificate to establish the origin server(s) that a connection is authoritative for, just as HTTP/1.1 does in [\[RFC7230\]](#).

Furthermore, [Section 9.1.1](#) of [\[RFC7540\]](#) explicitly allows a connection to be used for more than one origin server, if it is authoritative. This affects what responses can be considered authoritative, both for direct responses to requests and for server push (see [\[RFC7540\]](#), [Section 8.2.2](#)). Indirectly, it also affects what requests will be sent on a connection, since clients will generally only send requests on connections that they believe to be authoritative for the origin in question.

Once an Origin Set has been initialized for a connection, clients that implement this specification use it to help determine what the connection is authoritative for. Specifically, such clients MUST NOT consider a connection to be authoritative for an origin not present in the Origin Set, and they SHOULD use the connection for all requests to origins in the Origin Set for which the connection is authoritative, unless there are operational reasons for opening a new connection.

Note that for a connection to be considered authoritative for a given origin, the server is still required to authenticate with a certificate that passes suitable checks; see [Section 9.1.1](#) of [RFC7540] for more information. This includes verifying that the host matches a `dNSName` value from the certificate `subjectAltName` field (using the rules defined in [RFC2818]; see also [RFC5280], [Section 4.2.1.6](#)).

Additionally, clients MAY avoid consulting DNS to establish the connection's authority for new requests to origins in the Origin Set; however, those that do so face new risks, as explained in [Section 4](#).

Because ORIGIN can change the set of origins a connection is used for over time, it is possible that a client might have more than one viable connection to an origin open at any time. When this occurs, clients SHOULD NOT emit new requests on any connection whose Origin Set is a proper subset of another connection's Origin Set, and they SHOULD close it once all outstanding requests are satisfied.

The Origin Set is unaffected by any alternative services [\[RFC7838\]](#) advertisements made by the server. Advertising an alternative service does not affect whether a server is authoritative.

3. IANA Considerations

This specification adds an entry to the "HTTP/2 Frame Type" registry.

- Frame Type: ORIGIN
- Code: 0xc
- Specification: RFC 8336

4. Security Considerations

Clients that blindly trust the ORIGIN frame's contents will be vulnerable to a large number of attacks. See [Section 2.4](#) for mitigations.

Relaxing the requirement to consult DNS when determining authority for an origin means that an attacker who possesses a valid certificate no longer needs to be on path to redirect traffic to them; instead of modifying DNS, they need only convince the user to visit another website in order to coalesce connections to the target onto their existing connection.

As a result, clients opting not to consult DNS ought to employ some alternative means to establish a high degree of confidence that the certificate is legitimate. For example, clients might skip consulting DNS only if they receive proof of inclusion in a Certificate Transparency log [\[RFC6962\]](#) or if they have a recent Online Certificate Status Protocol (OCSP) response [\[RFC6960\]](#) (possibly using the "status_request" TLS extension [\[RFC6066\]](#)) showing that the certificate was not revoked.

The Origin Set's size is unbounded by this specification and thus could be used by attackers to exhaust client resources. To mitigate this risk, clients can monitor their state commitment and close the connection if it is too high.

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>>.
- [RFC2818] Rescorla, E., "[HTTP Over TLS](#)", RFC 2818, [DOI 10.17487/RFC2818](#), May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "[Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List \(CRL\) Profile](#)", RFC 5280, [DOI 10.17487/RFC5280](#), May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6066] Eastlake 3rd, D., "[Transport Layer Security \(TLS\) Extensions: Extension Definitions](#)", RFC 6066, [DOI 10.17487/RFC6066](#), January 2011, <<https://www.rfc-editor.org/info/rfc6066>>.
- [RFC6454] Barth, A., "[The Web Origin Concept](#)", RFC 6454, [DOI 10.17487/RFC6454](#), December 2011, <<https://www.rfc-editor.org/info/rfc6454>>.
- [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "[Hypertext Transfer Protocol Version 2 \(HTTP/2\)](#)", RFC 7540, [DOI 10.17487/RFC7540](#), May 2015, <<https://www.rfc-editor.org/info/rfc7540>>.
- [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>>.

5.2. Informative References

- [RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "[X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP](#)", RFC 6960, [DOI 10.17487/RFC6960](#), June 2013, <<https://www.rfc-editor.org/info/rfc6960>>.
- [RFC6962] Laurie, B., Langley, A., and E. Kasper, "[Certificate Transparency](#)", RFC 6962, [DOI 10.17487/RFC6962](#), June 2013, <<https://www.rfc-editor.org/info/rfc6962>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "[Hypertext Transfer Protocol \(HTTP/1.1\): Message Syntax and Routing](#)", RFC 7230, [DOI 10.17487/RFC7230](#), June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [RFC7838] Nottingham, M., McManus, P., and J. Reschke, "[HTTP Alternative Services](#)", RFC 7838, [DOI 10.17487/RFC7838](#), April 2016, <<https://www.rfc-editor.org/info/rfc7838>>.
- [RFC8288] Nottingham, M., "[Web Linking](#)", RFC 8288, [DOI 10.17487/RFC8288](#), October 2017, <<https://www.rfc-editor.org/info/rfc8288>>.

Appendix A. Non-Normative Processing Algorithm

The following algorithm illustrates how a client could handle received ORIGIN frames:

1. If the client is configured to use a proxy for the connection, ignore the frame and stop processing.
2. If the connection is not identified with the "h2" protocol identifier or another protocol that has explicitly opted into this specification, ignore the frame and stop processing.
3. If the frame occurs upon any stream except stream 0, ignore the frame and stop processing.
4. If any of the flags 0x1, 0x2, 0x4, or 0x8 are set, ignore the frame and stop processing.
5. If no previous ORIGIN frame on the connection has reached this step, initialize the Origin Set as per [Section 2.3](#).
6. For each `Origin-Entry` in the frame payload:
 1. Parse `ASCII-Origin` as an ASCII serialization of an origin ([RFC6454], [Section 6.2](#)), and let the result be `parsed_origin`. If parsing fails, skip to the next .
 2. Add `parsed_origin` to the Origin Set.

Appendix B. Operational Considerations for Servers

The ORIGIN frame allows a server to indicate for which origins a given connection ought be used. The set of origins advertised using this mechanism is under control of the server; servers are not obligated to use it or to advertise all origins that they might be able to answer a request for.

For example, it can be used to inform the client that the connection is to only be used for the SNI-based origin, by sending an empty ORIGIN frame. Or, a larger number of origins can be indicated by including a payload.

Generally, this information is most useful to send before sending any part of a response that might initiate a new connection; for example, "Link" response header fields [\[RFC8288\]](#), or links in the response body.

Therefore, the ORIGIN frame ought be sent as soon as possible on a connection, ideally before any HEADERS or PUSH_PROMISE frames.

However, if it's desirable to associate a large number of origins with a connection, doing so might introduce end-user-perceived latency, due to their size. As a result, it might be necessary to select a "core" set of origins to send initially, and expand the set of origins the connection is used for with subsequent ORIGIN frames later (e.g., when the connection is idle).

That said, senders are encouraged to include as many origins as practical within a single ORIGIN frame; clients need to make decisions about creating connections on the fly, and if the Origin Set is split across many frames, their behavior might be suboptimal.

Senders take note that, as per [Section 4](#), Step 5, of [\[RFC6454\]](#), the values in an ORIGIN header need to be case-normalized before serialization.

Finally, servers that host alternative services [\[RFC7838\]](#) will need to explicitly advertise their origins when sending ORIGIN, because the default contents of the Origin Set (as per [Section 2.3](#)) do not contain any alternative services' origins, even if they have been used previously on the connection.

Authors' Addresses

Mark Nottingham

E-Mail: mnot@mnot.net

URI: <https://www.mnot.net/>

Erik Nygren

Akamai Technologies

E-Mail: erik+ietf@nygren.org