

Network Working Group

J. Reschke

[TOC](#)

Internet-Draft

greenbytes

Intended status: Standards Track April 04, 2008

Expires: October 6, 2008

The Hypertext Transfer Protocol (HTTP) GET-Location header draft-reschke-http-get-location-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 6, 2008.

Abstract

Several hypertext transfer protocol (HTTP) extensions use methods other than GET to expose information. This has the drawback that this kind of information is harder to identify (missing a URL to which a GET request could be applied) and to cache.

This document specifies a simple extension header through which a server can advertise a substitute URL that an HTTP client subsequently can use with the GET method.

Editorial Note (To be removed by RFC Editor before publication)

Distribution of this document is unlimited. Please send comments to the Hypertext Transfer Protocol (HTTP) mailing list at ietf-http-wg@w3.org, which may be joined by sending a message with subject "subscribe" to ietf-http-wg-request@w3.org.

Discussions of the HTTP working group are archived at <http://lists.w3.org/Archives/Public/ietf-http-wg/>.

XML versions, latest edits and the issues list for this document are available from <http://greenbytes.de/tech/webdav/#draft-reschke-http-get-location>.

Table of Contents

- [1.](#) Introduction
- [2.](#) Notational Conventions
- [3.](#) The 'GET-Location' Header
- [4.](#) Security Considerations
- [5.](#) IANA Considerations
- [6.](#) Acknowledgments
- [7.](#) References
 - [7.1.](#) Normative References
 - [7.2.](#) Informative References
- [Appendix A.](#) Examples
 - [A.1.](#) WebDAV Collection Membership
 - [A.2.](#) WebDAV Custom Properties
 - [A.3.](#) DeltaV Version History Report
- [Appendix B.](#) Related HTTP features
 - [B.1.](#) Status 303
 - [B.2.](#) Content-Location Header
 - [B.3.](#) Location header
- [Appendix C.](#) Alternate Approaches
 - [C.1.](#) Link Relation
 - [C.2.](#) Multistatus Body Extension
- [Appendix D.](#) Open Issues
 - [D.1.](#) Content Negotiation on GET-Location
 - [D.2.](#) Using URI Templates rather than URIs
 - [D.3.](#) Extensions
- [Appendix E.](#) Change Log (to be removed by RFC Editor before publication)
 - [E.1.](#) Since draft-reschke-http-get-location-00
- [Appendix F.](#) Resolved issues (to be removed by RFC Editor before publication)
 - [F.1.](#) status-codes
 - [F.2.](#) non-get
- [Appendix G.](#) Open issues (to be removed by RFC Editor prior to publication)
 - [G.1.](#) edit
 - [G.2.](#) content-location
- [§](#) Author's Address
- [§](#) Intellectual Property and Copyright Statements

1. Introduction

Several HTTP ([\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#)) extensions use methods other than GET to expose information. This has the drawback that this kind of information is harder to identify (missing a URL to which a GET request could be applied) and to cache.

This document specifies a simple extension header through which a server can advertise a substitute URL that an HTTP client subsequently can use with the GET method.

2. Notational Conventions

[TOC](#)

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 [\[RFC2119\] \(Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.\)](#).

The terminology used here follows and extends that in the HTTP specification [\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#).

3. The 'GET-Location' Header

[TOC](#)

The GET-Location entity header identifies a substitute resource that can be used in subsequent requests for the same information, but using the GET method.

Note that, by definition, the GET-Location header can only used on responses to safe methods.

Syntax (using the the augmented Backus-Naur Form (BNF) defined in Section 2.1 of [\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#)):

```
GET-Location = "GET-Location" ":" "<" Simple-ref ">"
               *( ";" location-directive ) )
```

```
location-directive = "etag=" entity-tag
                    | "max-age" "=" delta-seconds
                    | location-extension
```

```
location-extension = token [ "=" ( token | quoted-string ) ]
```

```
Simple-ref      = absolute-URI | ( path-absolute [ "?" query ] )
```

```
absolute-URI   = <defined in [RFC3986], Section 4.3>
delta-seconds  = <defined in [RFC2616], Section 3.3.2>
entity-tag     = <defined in [RFC2616], Section 3.11>
path-absolute  = <defined in [RFC3986], Section 3.3>
quoted-string  = <defined in [RFC2616], Section 2.2>
query         = <defined in [RFC3986], Section 3.4>
token         = <defined in [RFC2616], Section 2.2>
```

Where:

Simple-ref Contains either the URI or the absolute path of the location.

etag The server can include the entity tag for the returned entity, if it would have been retrieved by a GET request to the substitute resource. Note that this is different from the value of the "ETag" header which could also be included in the response, but which would apply to the resource identified by the Request-URI.

max-age Specifies a lifetime for the information returned by this header. A client MUST discard any information related to this header after the specified amount of time.

The freshness lifetime for the information obtained from the GET-Location header does not depend on the cacheability of the response it was obtained from (which, in general, may not be cacheable at all). The "max-age" directive allows the server to specify after how many seconds a client should discard knowledge about the alternate resource. In absence of that header, clients SHOULD discard the information after 3600 seconds.

There is no direct relation between the status code of the HTTP response that included GET-Location and the status codes for subsequent GET requests on the substitute resource. For instance, GET-Location could be included in a 207 response to PROPFIND ([\[RFC4918\] \(Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning \(WebDAV\)," June 2007.\)](#), Section 9.1), but the response code for a successful GET on the substitute resource would usually be 200.

Note that servers may, but are not required to support methods other than GET or head on the substitute resource.

4. Security Considerations

[TOC](#)

This specification introduces no new security considerations beyond those discussed in Section 15 of [\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#).

5. IANA Considerations

[TOC](#)

This document specifies the new HTTP header listed below, to be added to the permanent registry (see [\[RFC3864\] \(Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields," September 2004.\)](#)).

Header field name: GET-Location

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document: [Section 3 \(The 'GET-Location' Header\)](#) of this specification

6. Acknowledgments

[TOC](#)

This document has benefited from thoughtful discussion by Stefan Eissing and Henrik Nordstrom.

7. References

[TOC](#)

7.1. Normative References

[TOC](#)

- [RFC2119] [Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels,"](#) BCP 14, RFC 2119, March 1997.
- [RFC2616]

- [Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2616, June 1999.](#)
- [RFC3986] [Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier \(URI\): Generic Syntax," RFC 3986, January 2005.](#)

7.2. Informative References

[TOC](#)

- [RFC2068] [Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2068, January 1997.](#)
- [RFC3253] [Clemm, G., Amsden, J., Ellison, T., Kaler, C., and J. Whitehead, "Versioning Extensions to WebDAV," RFC 3253, March 2002.](#)
- [RFC3864] [Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields," BCP 90, RFC 3864, September 2004.](#)
- [RFC4918] [Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning \(WebDAV\)," RFC 4918, June 2007.](#)
- [draft-gregorio-uritemplate] [Gregorio, J., Ed., Hadley, M., Ed., Nottingham, M., Ed., and D. Orchard, "URI Template," draft-gregorio-uritemplate-03 \(work in progress\), March 2008.](#)

Appendix A. Examples

[TOC](#)

A.1. WebDAV Collection Membership

[TOC](#)

In this example the client uses the WebDAV PROPFIND method ("HTTP Extensions for Web Distributed Authoring and Versioning", [\[RFC4918\] \(Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning \(WebDAV\)," June 2007.\)](#), Section 9.1) to get a list of all collection members, along with their DAV:resourcetype property ([\[RFC4918\] \(Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning \(WebDAV\)," June 2007.\)](#), Section 15.9):

```
>>Request
```

```
PROPFIND /collection/ HTTP/1.1
Host: example.com
Depth: 1
Content-Type: application/xml
```

```
<propfind xmlns="DAV:">
  <prop>
    <resourcetype/>
  </prop>
</propfind>
```

The response contains the requested information, plus the GET-Location header, identifying a separate resource which can provide the same information using the HTTP GET method:

>>Response

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml
GET-Location: <http://example.com/collection;/members>; etag="123";
              max-age=3600
```

```
<multistatus xmlns="DAV":>
  <response>
    <href>/collection/</href>
    <propstat>
      <prop>
        <resourcetype><collection/></resourcetype>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
  <response>
    <href>/collection/member</href>
    <propstat>
      <prop>
        <resourcetype/>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

The response provided the URL of the substitute resource, so when the client wishes to refresh the collection information, it uses that URI. The response contained the entity tag for the data being returned, so it can make the request conditional:

>>Request

```
GET /collection;/members HTTP/1.1
Host: example.com
Accept: application/xml
If-None-Match: "123"
```

The information did not change, so the server does not need to return new data:

>>Response

```
HTTP/1.1 304 Not Modified
```

Later on, the client tries again. This time, however, a second member has been added:

>>Request

```
GET /collection;/members HTTP/1.1
Host: example.com
Accept: application/xml
If-None-Match: "123"
```

>>Response


```
HTTP/1.1 200 OK
Content-Type: application/xml
ETag: "124"
```

```
<multistatus xmlns="DAV":>
  <response>
    <href>/collection/</href>
    <propstat>
      <prop>
        <resourcetype><collection/></resourcetype>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
  <response>
    <href>/collection/member</href>
    <propstat>
      <prop>
        <resourcetype/>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
  <response>
    <href>/collection/member2</href>
    <propstat>
      <prop>
        <resourcetype/>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

Finally, the collection has been removed by somebody else. The client tries a refresh:

```
>>Request
```

```
GET /collection;/members HTTP/1.1
Host: example.com
Accept: application/xml
If-None-Match: "124"
```

```
>>Response
```

```
HTTP/1.1 404 Not Found
```

Note that it may be hard to compute entity tags for more complex PROPFIND responses. For instance, most properties depend on the state of the collection member, not the state of the collection itself, and thus the response will change even though the state of the collection itself did not change.

This is why this extension leaves it to the server whether to return a GET-Location at all, and if so, whether to return cache validators along with it.

A.2. WebDAV Custom Properties

[TOC](#)

Here, the client uses the WebDAV PROPFIND method ([\[RFC4918\] \(Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning \(WebDAV\)," June 2007.\)](#), Section 9.1) to obtain a custom property:

>>Request

```
PROPFIND /collection/member HTTP/1.1
Host: example.com
Depth: 0
Content-Type: application/xml

<propfind xmlns="DAV:">
  <prop>
    <title xmlns="http://ns.example.com/" />
  </prop>
</propfind>
```

>>Response

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml
GET-Location: </collection/member;prop=title>; etag="1"

<multistatus xmlns="DAV":>
  <response>
    <href>/collection/member</href>
    <propstat>
      <prop>
        <title xmlns="http://ns.example.com/"
          >Document Title</title>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

>>Request

```
GET /collection/member;prop=title HTTP/1.1
Host: example.com
If-None-Match: "1"
```

>>Response

```
HTTP/1.1 304 Not Modified
```

Later, the request is repeated after the title property indeed changed...:

>>Request

```
GET /collection/member;prop=title HTTP/1.1
Host: example.com
If-None-Match: "1"
```

>>Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
ETag: "2"
```

```
<multistatus xmlns="DAV":>
  <response>
    <href>/collection/member</href>
    <propstat>
      <prop>
        <title xmlns="http://ns.example.com/"
          >New Document Title</title>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

Although this example may look like every WebDAV property would need a separate entity tag, this is of course not the case. For instance, a server that stores all custom properties in a single place (like a properties file) could use the same computation for the entity tag for all properties. Also, it could implement resources representing multiple custom property values the same way.

A.3. DeltaV Version History Report

[TOC](#)

Here, the client uses the DeltaV DAV:version-tree report ("Versioning Extensions to WebDAV", [\[RFC3253\] \(Clemm, G., Amsden, J., Ellison, T., Kaler, C., and J. Whitehead, "Versioning Extensions to WebDAV,"](#)

[March 2002.](#)), Section 3.7) to obtain the members of the version history of a version-controlled resource.

>>Request

```
REPORT /collection/member HTTP/1.1
Host: example.com
Depth: 0
Content-Type: application/xml
```

```
<version-tree xmlns="DAV:">
  <prop>
    <resourcetype/>
  </prop>
</version-tree>
```

>>Response

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml
GET-Location: </version-storage/12345/;justmembers>
```

```
<multistatus xmlns="DAV":>
  <response>
    <href>/version-storage/12345/V1</href>
    <propstat>
      <prop>
        <resourcetype><collection/></resourcetype>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
  <response>
    <href>/version-storage/12345/V2</href>
    <propstat>
      <prop>
        <resourcetype><collection/></resourcetype>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

Note that in this case, the substitute resource can be almost identical to the one from the PROPFIND/Depth:1 example: the only difference being that the report result does not contain a DAV:response element for the collection itself.

Appendix B. Related HTTP features

This section discusses some related HTTP features and explains why the author feels that they can't be used for the given use case.

B.1. Status 303

[TOC](#)

Section 10.3.4 of [\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#) defines the status code 303 (See Other):

The response to the request can be found under a different URI and SHOULD be retrieved using a GET method on that resource. This method exists primarily to allow the output of a POST-activated script to redirect the user agent to a selected resource. The new URI is not a substitute reference for the originally requested resource. The 303 response MUST NOT be cached, but the response to the second (redirected) request might be cacheable.

On first glance, it may look as if this addresses exactly the given use case. However:

1. It says: "The new URI is not a substitute reference for the originally requested resource. The 303 response MUST NOT be cached, but the response to the second (redirected) request might be cacheable." That is, the information about the alternate resource is not cacheable.
2. Servers returning a 303 status instead of the one expected by the client, such as 207 Multistatus, would likely break existing clients.

B.2. Content-Location Header

[TOC](#)

Section 14.14 of [\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#) states:

The Content-Location value is not a replacement for the original requested URI; it is only a statement of the location of the resource corresponding to this particular entity at the time of the request. (...)

However, the purpose of "GET-Location" is to enable the server to provide a permanent replacement URI.

B.3. Location header

[TOC](#)

Section 14.30 of [\[RFC2616\] \(Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.\)](#) states:

The Location response-header field is used to redirect the recipient to a location other than the Request-URI for completion of the request or identification of a new resource. (...)

Neither of these cases ("redirect to a location for completion of the request" and "identification of a new resource") matches the use case "GET-Location" covers.

Appendix C. Alternate Approaches

[TOC](#)

C.1. Link Relation

[TOC](#)

An alternative to introducing a new header would be to re-use an existing header, such as the Link header defined in Section 19.6.2 of [\[RFC2068\] \(Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," January 1997.\)](#). Note that this still would require registering a link relation. The example from [Appendix A.1 \(WebDAV Collection Membership\)](#) would then read like this:

```
Link: </collection;/members>;  
      rel=getlocation; etag="123"; max-age=3600
```

C.2. Multistatus Body Extension

[TOC](#)

Observing that the whole proposal tries to deal with WebDAV related shortcomings, it may make sense to constrain the solution to WebDAV response bodies, thereby not having to introduce anything that would be visible outside WebDAV.

A very simple approach would be to embed the information in the DAV:multistatus ([\[RFC4918\] \(Dusseault, L., Ed., "HTTP Extensions for](#)

[Web Distributed Authoring and Versioning \(WebDAV\)," June 2007.](#)),
Section 14.16) response body.
Re-using the example in [Appendix A.1 \(WebDAV Collection Membership\)](#),
this could look like this:

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml

<multistatus xmlns="DAV":>
  <gl:get-location
    xmlns:gl="http://purl.oclc.org/NET/webdav/mount/getlocation">
    <href>/collection;/members</href>
    <getetag>"123"</getetag>
    <gl:max-age>3600</gl:max-age>
  <gl:get-location
  <response>
    <href>/collection/</href>
    <propstat>
      <prop>
        <resourcetype><collection/></resourcetype>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
  <response>
    <href>/collection/member</href>
    <propstat>
      <prop>
        <resourcetype/>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

Appendix D. Open Issues

[TOC](#)

D.1. Content Negotiation on GET-Location

[TOC](#)

Should it be possible to use Content Negotiation on the resource identified by GET-Location? A use case could be a metadata provider that would support different formats, such as WebDAV's multistatus format (MIME type missing!), RDF, JSON, whatever.

This could be done using a location-extension specifying the Accept header for the GET operation.

D.2. Using URI Templates rather than URIs

[TOC](#)

Should we allow servers to return URI templates ([draft-gregorio-uritemplate] (Gregorio, J., Ed., Hadley, M., Ed., Nottingham, M., Ed., and D. Orchard, "URI Template," March 2008.)), so that clients can compute substitute URLs for other requests as well? For instance, this could be done by allowing a URI template instead of the Simple-ref, and to return another template specifying how to derive the template variable from the Request-URI:

>>Request

```
PROPFIND /documents/a/b HTTP/1.1
Host: example.com
Depth: 0
Content-Type: application/xml
```

>>Response

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml
GET-Location: </metadata/{path};members>; path-template=</a/b/{path}>
...

```

So in this case, the actual URI to be used would be <http://example.com/metadata/a/b;members>.

D.3. Extensions

[TOC](#)

Do we need a registry for new location-directive values?

Appendix E. Change Log (to be removed by RFC Editor before publication)

[TOC](#)

E.1. Since draft-reschke-http-get-location-00

[TOC](#)

Add and resolve issues "non-get" and "status-codes". Add issue "content-location". Add "Acknowledgments" Section. Update uri-template

reference. Discuss more alternative approaches: Link header, Multistatus body extension.

Appendix F. Resolved issues (to be removed by RFC Editor before publication)

[TOC](#)

Issues that were either rejected or resolved in this version of this document.

F.1. status-codes

[TOC](#)

In Section 3:

Type: edit

julian.reschke@greenbytes.de (2007-07-31): Explain the relation between the status code GET-Location comes with, and the status codes for GET requests on the substitute resource.

Resolution (2008-03-14): Add clarification.

F.2. non-get

[TOC](#)

In Section 3:

Type: edit

julian.reschke@greenbytes.de (2007-08-05): Say something about non GET/HEAD requests to the substitute resource.

Resolution (2008-03-14): Add clarification.

Appendix G. Open issues (to be removed by RFC Editor prior to publication)

[TOC](#)

G.1. edit

[TOC](#)

Type: edit

julian.reschke@greenbytes.de (2007-07-27): Umbrella issue for editorial fixes/enhancements.

G.2. content-location

[TOC](#)

In Section B.2:

Type: edit

julian.reschke@greenbytes.de (2008-03-14):

In <http://lists.w3.org/Archives/Public/ietf-http-wg/2007JulSep/0184.html>, Roy Fielding points out that the statement about Content-Location is out-of context; and that Content-Location indeed should be used for this use case.

In this case, other information defined in GET-Location would still be missing, such as validity and etag information (see response in <http://lists.w3.org/Archives/Public/ietf-http-wg/2007JulSep/0195.html>). So would it make sense to use Content-Location instead, and move the optional fields defined in GET-Location somewhere else? Or would Content-Location alone be sufficient (which proper instructions how to use it with methods like PROPFIND and REPORT)?

Author's Address

[TOC](#)

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany

Phone: +49 251 2807760

Email: julian.reschke@greenbytes.de

URI: <http://greenbytes.de/tech/webdav/>

Full Copyright Statement

[TOC](#)

Copyright © The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the

procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.