

HTTP Working Group
Internet-Draft
Expires: May 15, 1998

Koen Holtman, TUE
Andrew Mutz, Hewlett-Packard
Ted Hardie, NASA NIC

The Alternates Header Field

[draft-holtman-http-alternates-00.txt](#)

STATUS OF THIS MEMO

This document is an Internet-Draft. 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".

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited. Though this work originated in the HTTP working group and comments should continue to be directed to that group, the authors expect that the proposed header will have considerable use outside of HTTP. Discussions about its potential uses outside of HTTP are particularly sought by the authors. Send comments to the HTTP working group at <http-wg@cuckoo.hpl.hp.com>. Discussions of the working group are archived at <URL:http://www.ics.uci.edu/pub/ietf/http/>. General discussions about HTTP and the applications which use HTTP should take place on the <www-talk@w3.org> mailing list.

ABSTRACT

This document proposes a header, Alternates, which is intended to provide a common method for Internet-based application

protocols to indicate that a particular resource exists in multiple forms. The Alternates header provides a machine-readable indication of the bases on which the different forms vary and how the the recipient of the header can select among them.

1 Introduction

Where a particular content provider has created more than one representation of a specific resource, the provider may wish to indicate the existence of the different representations, or "variants", to those who might want to select among them. Rather than consuming bandwidth by sending all available variants, the content provider can employ the Alternates header to indicate what variants are available, the bases on which the variants differ, and how to select a specific variant.

The functionality of the Alternates header is similar to that of the Multipart/Alternative MIME type[Ref] where the Multipart/Alternative object has all external message bodies. The Alternates header differs in that it does not rank the alternatives. Instead, it provides information which allows the recipient to decide among them based on the provider's estimation of their quality, local capabilities, and preferences.

2 Terminology

The words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in [RFC 2119](#) [Ref].

3 Notation

The version of BNF used in this document is taken from [Ref], and many of the nonterminals used are defined in [Ref].

4.1 Definition

The Alternates header field describes all available variants for a specific resource. The description for each variant includes an URI from which this variant can be retrieved.

Alternates = "Alternates" ":" variant-list

variant-list = 1#(variant-description ; see [section 5](#)
| fallback-variant

| negotiation-directive)

fallback-variant = "{" <"> URI <"> "}"

negotiation-directive = token ["=" (token | quoted-string)]

An example is

```
Alternates: {"http://x.org/paper.1" 0.9 {type text/html} {language en}},
            {"http://x.org/paper.2" 0.7 {type text/html} {language fr}},
            {"ftp://x.org/pub/paper.3" 1.0 {type application/postscript}
            {language en}},
            {"http://x.org/paper.html.en"}, x=y
```

A variant list may contain multiple differing descriptions of the same variant. This can be convenient if the variant uses conditional rendering constructs, or if the variant resource returns multiple representations using a multipart media type.

Only one fallback-variant field may be present. If the variant selection algorithm of the user agent finds that all variants described by variant-description fields are unacceptable, then it SHOULD choose the fallback variant, if present, as the best variant.

This specification does not define any specific negotiation directives for the Alternates header field. User agents SHOULD ignore all negotiation directives they do not understand. If a proxy receives an Alternates header field with an unknown negotiation directive, it SHOULD, whenever possible, forward the response towards the user agent instead of trying to take part in a negotiation process itself.

[4.2](#) Length of variant lists

As a general rule, variant lists in Alternates header fields should be as short as possible. The use of multiple differing descriptions of the same variant is supported, but should be limited to those resources which cannot easily be expressed without the use of conditional constructs or multipart media types. It is expected that a typical negotiable resource will have 2 to 10 variants. In situations where many more variants are available, more sophisticated methods than the Alternates header field should be used.

[5](#) Variant descriptions

[5.1](#) Syntax

A variant can be described in a machine-readable way with a variant

description.

```
variant-description =  
    "{" "<"> URI "<"> source-quality *variant-attribute"}"  
  
source-quality = qvalue  
  
variant-attribute = "{" "type" media-type "  
    | "{" "charset" charset "  
    | "{" "language" 1#language-tag "  
    | "{" "length" 1*DIGIT "  
    | extension-attribute  
  
extension-attribute = "{" extension-name extension-value "  
extension-name      = token  
extension-value     = *( token | quoted-string | LWS  
    | extension-specials )  
  
extension-specials =  
    <any element of tspecials except "<"> and "}">
```

Examples are

```
{"http://x.org/paper.2" 0.7 {type text/html} {language fr}}  
  
{"http://x.org/paper.5" 0.9 {type text/html} {length 1002}}  
  
{"http://x.org/paper.1" 0.001}
```

The various attributes which can be present in a variant description are covered in the subsections below. Each attribute may appear only once in a variant description.

[5.2](#) URI

The URI attribute gives the URI of the resource from which the variant can be retrieved with a GET request. It must be an absolute URI, in order to allow for this header to be passed to other MIME transport protocols without transformation. The variant resource may vary the content it sends (on the Cookie request header field, for example), but SHOULD NOT engage in content negotiation itself.

[5.3](#) Source-quality

The source-quality attribute gives the quality of the variant, as a representation of the negotiable resource, when this variant is rendered with a perfect rendering engine on the best possible output medium.

If the source-quality is less than 1, it often expresses a quality degradation caused by a lossy conversion to a particular data format. For example, a picture originally in JPEG form would have a lower source quality when translated to the XBM format, and a much lower source quality when translated to an ASCII-art variant. Note however, that degradation is a function of the source; an original piece of ASCII-art may degrade in quality if it is captured in JPEG form.

Servers should use the following table as a guide when assigning source quality values:

1.000	perfect representation
0.900	threshold of noticeable loss of quality
0.800	noticeable, but acceptable quality reduction
0.500	barely acceptable quality
0.300	severely degraded quality
0.000	completely degraded quality

In the source-quality values, servers should not account for the size of the variant and its impact on transmission and rendering delays; the size of the variant should be stated in the length attribute and any size-dependent calculations should be done by a variant selection algorithm in the user agent.

[5.4](#) Type, charset, language, and length

The type attribute of a variant description carries the same information as its Content-Type response header field counterpart defined in [\[1\]](#), except for any charset information, which MUST be carried in the charset attribute. For, example, the header field

```
Content-Type: text/html; charset=ISO-8859-4
```

has the counterpart attributes

```
{type text/html} {charset ISO-8859-4}
```

The language and length attributes carry the same information as their Content-* response header field counterparts in [\[1\]](#). The length attribute, if present, MUST thus reflect the length of the variant alone, and not the total size of the variant and any objects inlined or embedded by the variant.

Though all of these attributes are optional, it is often desirable to include as many attributes as possible, as this will increase the quality of the negotiation process.

Note: A server is not required to maintain a one-to-one

correspondence between the attributes in the variant description and the Content-* header fields in the variant response. For example, if the variant description contains a language attribute, the response does not necessarily have to contain a Content-Language header field. If a Content-Language header field is present, it does not have to contain an exact copy of the information in the language attribute.

[5.5](#) Extension-attribute

The extension-attribute allows future specifications to incrementally define new dimensions of negotiation, and eases content negotiation experiments. User agents conforming to this specification SHOULD treat all variants with extension attributes they do not recognize as unusable. Proxies SHOULD NOT do any negotiation processing for a response if an extension attribute unknown to them is present in the variant list. They SHOULD forward the response unchanged towards the user agent instead.

The extension names "features" and "description" are reserved by this specification for use in transparent content negotiation [\[4\]](#).

[6](#) Use of the Alternates header field

This section defines conventions and guidelines for the use of the Alternates header field.

[6.1](#) Use accompanying a variant

A resource provider may provide a recipient with a particular variant together with an Alternates header field which notifies the recipient that multiple variants are available. An HTTP example of such an exchange is:

```
HTTP/1.1 200 OK
Date: Tue, 11 Jun 1996 20:05:31 GMT
Content-Type: text/html
Content-Language: en
Last-Modified: Mon, 10 Jun 1996 10:01:14 GMT
Content-Length: 5327
Alternates: {"http://x.org/paper.1" 0.9 {type text/html} {language en}},
            {"http://x.org/paper.2" 0.7 {type text/html} {language fr}},
            {"http://x.org/paper.3" 1.0 {type application/postscript}
            {language en}}
Content-Location: http://x.org/paper.1
Vary: *
Expires: Thu, 01 Jan 1980 00:00:00 GMT
Cache-Control: max-age=604800
```

<title>A paper about

In this response, the Content-Location header field tells the user agent which variant was included. The Vary, Expires, and Cache-Control header fields ensure proper handling of the response by HTTP/1.0 and HTTP/1.1 caches.

When detecting that an Alternates header field is present, a user agent MAY choose to use a variant selection algorithm to select the best variant of the negotiable resource. If the best variant is not the same one as the one identified by the Content-Location header field, the user agent MAY use the URI indicated for the best version to obtain it.

6.2 Use when a variant is not present

A content provider may also inform a potential recipient that multiple variants exist without providing any particular variant. In such exchanges, the content provider should omit the Content-Location header. A message body MAY accompany the headers used to convey this information. An SMTP example of such an exchange is:

```
220 ESMTP spoken here
EHLO mail.x.org
250-mail.y.org Hello mail.x.org, pleased to meet you
MAIL FROM: lists@mail.x.org
250 lists@mail.x.org... Sender ok
RCPT TO: exploder-lists@mail.y.org
250 exploder-lists@mail.y.org... Recipient ok
DATA
354 Enter mail, end with "."on a line by itself
To: exploder-lists@mail.y.org
From: lists@mail.x.org
Date: Tue, 11 Jun 1996 20:02:21 GMT
Content-Type: text/html
Content-Length: 227
Alternates: {"http://x.org/paper.1" 0.9 {type text/html} {language en}},
            {"http://x.org/paper.2" 0.7 {type text/html} {language fr}},
            {"http://x.org/paper.3" 1.0 {type application/postscript}
            {language en}}
Vary: *

<h1>Multiple Choices:</h1>
<ul>
<li><a href=paper.1>HTML, English version</a>
<li><a href=paper.2>HTML, French version</a>
<li><a href=paper.3>Postscript, English version</a>
```


.
250 XAA13385 Message accepted for delivery

On receipt of such a message, the user agent MAY use a variant selection algorithm to select the best variant of the negotiable resource, and retrieve this variant. For compatibility with user agents which are not capable of handling the Alternates header field, any message body should normally allow the user to select the best variant manually.

6.3 Use of redirection to optimize availability

HTTP and related protocols allow an origin server to indicate that a resource is available but is not at the URL by which it was requested. As an optimization, the Alternates header can be used in combination with this redirection to allow an origin server to indicate the availability of variant resources without including a resource or excluding user agents which do not support the Alternates header. By providing the fallback variant's URL in the location header, the origin server can ensure that all user agents will have access to that variant, while the full list of variants is provided in the Alternates header for more capable user agents.

An example of this usage in an HTTP response is:

```
HTTP/1.1 302 Moved Temporarily
Date: Tue, 11 Jun 1996 20:05:31 GMT
Content-Type: text/html
Alternates: {"http://x.org/paper.1" 0.9 {type text/html} {language en}},
            {"http://x.org/paper.2" 0.7 {type text/html} {language fr}},
            {"http://x.org/paper.3" 1.0 {type application/postscript}
            {language en}}
Location: http://x.org/paper.1
Content-Length: 67
```

This document is available here.

Note the use of a Location header field instead of a Content-Location header field. On receipt of such a response, user agents which understand the Alternates header SHOULD use a variant selection algorithm to select the best variant of the negotiable resource, and retrieve this variant. When the Alternates header is used in protocols which do not provide redirection, the method outlined in [section 6.1](#) should be used when the content provider wishes to ensure at least one variant is made available to all recipients.

[6.4](#) User agent guidelines

Summarizing the three sections above, if an Alternates header field is present in the response, then

- * a recipient SHOULD use its variant selection algorithm to choose and retrieve the best variant if a Content-Location header field is absent,
- * and MAY use its variant selection algorithm to choose and retrieve the best variant if a Content-Location header field is present.

[6.4.1](#) Interactive user agent guidelines

For user agents which provide direct interaction with users, the following guidelines apply:

1. The user agent SHOULD make available through its user interface some indication that the resource being displayed is a negotiated resource instead of a plain resource. It SHOULD also allow the user to examine the variant list included in the Alternates header field. Such a notification and review mechanism is needed because of privacy considerations, see [section 7.1](#).
2. If the user agent shows the URI of the displayed information to the user, it SHOULD be the negotiable resource URI, not the variant URI that is shown. This encourages third parties, who want to refer to the displayed information in their own documents, to refer to the negotiable resource as a whole, rather than to the variant resource which happens to be shown. This abstract referencing is vital for the interoperability of content across sites. The user agent SHOULD also, however, provide a means for reviewing the URI of the particular variant which is currently being displayed.
3. Similarly, if the user agent stores a reference to the displayed information for future use, for example in a hotlist, it SHOULD store the negotiable resource URI, not the variant URI.

[6.4.2](#) Non-interactive user agent guidelines

Devices like printers, fax machines, or text processors may also be the recipients of the Alternates header. Since such devices typically function without direct interaction with a human operator, the user agent guidelines given above do not directly apply. Many of the same principals, however, can be employed. For example, a device which generates a log report or cover sheet

which contains information about the information received could provide the same information described above in that form.

[6.5](#) Negotiation on content encoding

Negotiation on the content encoding of a response is orthogonal to content negotiation based on the Alternates header field.

[6.6](#) Role of proxies

This specification does not define mechanisms by which proxies can use the Alternates header field, but does allow other specifications to define such mechanisms. To ensure extensibility of the Alternates header field, this specification does however define, in [section 4.1](#) and [section 5.5](#), that a proxy should not engage in a negotiation process when encountering an Alternates header field which has a component unknown to it.

[7](#) Security and privacy considerations

[7.1](#) Variants hiding the source of displayed material.

The Variants header may be used to allow a user to select among variants available from more than one location. In such cases, users may make incorrect assumptions about the origin of particular resources. The guidelines given in 6.4 alleviate this problem to a degree, but some implementors of the Alternates header may wish to provide further warnings or filters.

[7.2](#) User agent choices revealing information of a private nature

The automatic selection and retrieval of a variant by a user agent will reveal a preference for this variant to the server. A malicious service author could provide a page with 'fake' negotiability as a means of obtaining privacy-sensitive information. Such a plot would, however, be visible to an alert victim if the list of available variants and their properties is reviewed through a mechanism as described in [section 6.4.1](#).

[7.3](#) Security holes revealed by negotiation

Malicious servers could use content negotiation as a means of obtaining information about security holes which may be present in user agents.

[8](#) Acknowledgments

Work on HTTP content negotiation has been done since at least 1993. This specification builds on an earlier incomplete specification of the Alternates header field recorded in [2]. The authors wish to thank the individuals who have contributed to the work on content negotiation in the HTTP working group, including Brian Behlendorf, Daniel DuBois, Martin J. Duerst, Roy T. Fielding, Jim Gettys, Yaron Goland, Dirk van Gulik, Graham Klyne, Scott Lawrence, Larry Masinter, Jeffrey Mogul, Henrik Frystyk Nielsen, Frederick G.M. Roeber, Paul Sutton, and Klaus Weide.

[9](#) References

- [1] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol -- HTTP/1.1. [RFC 2068](#), HTTP Working Group, January 1997.
- [2] Roy T. Fielding, Henrik Frystyk Nielsen, and Tim Berners-Lee. Hypertext Transfer Protocol -- HTTP/1.1. Internet-Draft [draft-ietf-http-v11-spec-01.txt](#), HTTP Working Group, January 1996.
- [3] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol -- HTTP/1.0. [RFC 1945](#). MIT/LCS, UC Irvine, May 1996.
- [4] K. Holtman, A. Mutz. Transparent Content Negotiation in HTTP. Internet-Draft [draft-ietf-http-negotiation-04.txt](#), HTTP Working Group, September 1997.
- [5] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. [RFC 2119](#). Harvard University, March 1997.

[10](#) Authors' addresses

Koen Holtman
Technische Universiteit Eindhoven
Postbus 513
Kamer HG 6.57
5600 MB Eindhoven (The Netherlands)
Email: koen@win.tue.nl

Andrew H. Mutz
Hewlett-Packard Company
1501 Page Mill Road 3U-3
Palo Alto CA 94304, USA
Fax +1 415 857 4691
Email: mutz@hpl.hp.com

Ted Hardie
NASA NIC
Mail Stop 204-14
NASA Ames Research Center
Moffett Field, CA 94035
Email: hardie@nic.nasa.gov

11 Appendix: Example of a variant selection algorithm

A negotiating user agent will choose the best variant from a variant list with a variant selection algorithm. This appendix contains an example of such an algorithm.

The inputs of the algorithm are a variant list from an Alternates header field, and an agent-side configuration database, which contains

- a collection of quality values assigned to media types, languages, and charsets for the current request, following the model of the corresponding HTTP/1.1 [\[1\]](#) Accept- header fields,
- a table which lists 'forbidden' combinations of media types and charsets, i.e. combinations which cannot be displayed because of some internal user agent limitation.

The output of the algorithm is either the best variant, or the conclusion that none of the variants are acceptable.

11.1 Computing overall quality values

As a first step in the variant selection algorithm, the overall qualities associated with all variant descriptions in the list are computed.

The overall quality Q of a variant description is the value

$$Q = \text{round5}(q_s * q_t * q_c * q_l * q_a)$$

where `round5` is a function which rounds a floating point value to 5 decimal places after the point. It is assumed that the user agent can run on multiple platforms: the rounding function makes the algorithm independent of the exact characteristics of the underlying floating point hardware.

The factors q_s , q_t , q_c , q_l , and q_a are determined as follows.

q_s Is the source quality factor in the variant description.

q_t The media type quality factor is 1 if there is no type attribute in the variant description. Otherwise, it is the

quality value assigned to this type by the configuration database. If the database does not assign a value, then the factor is 0.

qc The charset quality factor is 1 if there is no charset attribute in the variant description. Otherwise, it is the quality value assigned to this charset by the configuration database. If the database does not assign a value, then the factor is 0.

ql The language quality factor is 1 if there is no language attribute in the variant description. Otherwise, it is the highest quality value the configuration database assigns to any of the languages listed in the language attribute. If the database does not assign a value to any of the languages listed, then the factor is 0.

qa The quality adjustment factor is 0 if the variant description lists a media type - charset combination which is 'forbidden' by the table, and 1 otherwise.

As an example, if a variant list contains the variant description

```
{"paper.2" 0.7 {type text/html} {language fr}}
```

and if the configuration database contains the quality value assignments

```
types:  text/html;q=1.0, type application/postscript;q=0.8
languages: en;q=1.0, fr;q=0.5
```

then the variant selection algorithm will compute the overall quality for the variant description as follows:

```
{"paper.2" 0.7 {type text/html} {language fr}}
      |           |           |
      |           |           |
      V           V           V
round5 ( 0.7 * 1.0 * 0.5 ) = 0.35000
```

With same configuration database, the variant list

```
{"paper.1" 0.9 {type text/html} {language en}},
{"paper.2" 0.7 {type text/html} {language fr}},
{"paper.3" 1.0 {type application/postscript} {language en}}
```

would yield the following computations:

```
round5 ( qs * qt * qc * ql * qa ) = Q
  ---  ---  ---  ---  ---
paper.1: 0.9 * 1.0 * 1.0 * 1.0 * 1.0 = 0.90000
```

```
paper.1:  0.7 * 1.0 * 1.0 * 0.5 * 1.0  = 0.35000
paper.3:  1.0 * 0.8 * 1.0 * 1.0 * 1.0  = 0.80000
```

[11.2](#) Determining the result

Using all computed overall quality values, the end result of the variant selection algorithm is determined as follows.

If all overall quality values are 0, then the best variant is the fallback variant, if there is one in the Alternates header field, else the result is the conclusion that none of the variants are acceptable.

If at least one overall quality value is greater than 0, then the best variant is the variant which has the description with the highest overall quality value, or, if there are multiple variant descriptions which share the highest overall quality value, the variant of the first variant description in the list which has this highest overall quality value.

[11.3](#) Ranking dimensions

Consider the following variant list:

```
{"paper.greek"    1.0 {language el} {charset ISO-8859-7}},
{"paper.english" 1.0 {language en} {charset ISO-8859-1}}
```

It could be the case that the user prefers the language "el" over "en", while the user agent can render "ISO-8859-1" better than "ISO-8859-7". The result is that in the language dimension, the first variant is best, while the second variant is best in the charset dimension. In this situation, it would be preferable to choose the first variant as the best variant: the user settings in the language dimension should take precedence over the hard-coded values in the charset dimension.

To express this ranking between dimensions, the user agent configuration database should have a higher spread in the quality values for the language dimension than for the charset dimension. For example, with

```
languages: el;q=1.0, en-gb;q=0.7, en;q=0.6, da;q=0, ...
```

```
charsets: ISO-8859-1;q=1.0, ISO-8859-7;q=0.95,
          ISO-8859-5;q=0.97, unicode-1-1;q=0, ...
```

the first variant will have an overall quality of 0.95000, while the second variant will have an overall quality 0.70000. This makes the first variant the best variant.

Expires: May 15, 1998