

## Apache Module `mod_negotiation`

<b>Description:</b>	Provides for content negotiation <sup>1</sup>
<b>Status:</b>	Base
<b>Module Identifier:</b>	<code>negotiation_module</code>
<b>Source File:</b>	<code>mod_negotiation.c</code>

### Summary

Content negotiation, or more accurately content selection, is the selection of the document that best matches the clients capabilities, from one of several available documents. There are two implementations of this.

- A type map (a file with the handler `type-map`) which explicitly lists the files containing the variants.
- A MultiViews search (enabled by the `MultiViews Options`), where the server does an implicit filename pattern match, and choose from amongst the results.

### Topics

Type maps.....	1
MultiViews.....	2
URI References .....	4

### Directives

<code>CacheNegotiatedDocs</code> .....	2
<code>ForceLanguagePriority</code> .....	3
<code>LanguagePriority</code> .....	4

### See also

- `Options`
- `mod_mime`
- Content Negotiation<sup>1</sup>

## Type maps

---

A type map has a format similar to RFC822 mail headers. It contains document descriptions separated by blank lines, with lines beginning with a hash character (`#`) treated as comments. A document description consists of several header records; records may be continued on multiple lines if the continuation lines start with spaces. The leading space will be deleted and the lines concatenated. A header record consists of a keyword name, which always ends in a colon, followed by a value. Whitespace is allowed between the header name and value, and between the tokens of value. The headers allowed are:

#### **Content-Encoding:**

The encoding of the file. Apache only recognizes encodings that are defined by an `AddEncoding` directive. This normally includes the encodings `x-compress` for compress'd files, and `x-gzip` for gzip'd files. The `x-` prefix is ignored for encoding comparisons.

#### **Content-Language:**

The language(s) of the variant, as an Internet standard language tag (RFC 1766<sup>2</sup>). An example is `en`, meaning English. If the variant contains more than one language, they are separated by a comma.

#### **Content-Length:**

---

## Apache Module mod\_negotiation

---

The length of the file, in bytes. If this header is not present, then the actual length of the file is used.

**Content-Type:**

The MIME media type of the document, with optional parameters. Parameters are separated from the media type and from one another by a semi-colon, with a syntax of `name=value`. Common parameters include:

**level**

an integer specifying the version of the media type. For `text/html` this defaults to 2, otherwise 0.

**qs**

a floating-point number with a value in the range 0.0 to 1.0, indicating the relative 'quality' of this variant compared to the other available variants, independent of the client's capabilities. For example, a jpeg file is usually of higher source quality than an ascii file if it is attempting to represent a photograph. However, if the resource being represented is ascii art, then an ascii file would have a higher source quality than a jpeg file. All `qs` values are therefore specific to a given resource.

**Example**

```
Content-Type: image/jpeg; qs=0.8
```

**URI:**

uri of the file containing the variant (of the given media type, encoded with the given content encoding). These are interpreted as URLs relative to the map file; they must be on the same server (!), and they must refer to files to which the client would be granted access if they were to be requested directly.

**Body:**

New in Apache 2.0, the actual content of the resource may be included in the type-map file using the Body header. This header must contain a string that designates a delimiter for the body content. Then all following lines in the type map file will be considered part of the resource body until the delimiter string is found.

**Example:**

```
Body:----xyz----
<html>
<body>
<p>Content of the page.</p>
</body>
</html>
----xyz----
```

## MultiViews

---

A MultiViews search is enabled by the `MultiViews Options`. If the server receives a request for `/some/dir/foo` and `/some/dir/foo` does *not* exist, then the server reads the directory looking for all files named `foo.*`, and effectively fakes up a type map which names all those files, assigning them the same media types and content-encodings it would have if the client had asked for one of them by name. It then chooses the best match to the client's requirements, and returns that document.

## CacheNegotiatedDocs Directive

---

<b>Description:</b> Allows content-negotiated documents to be cached by proxy servers
---

---

 Apache Module mod\_negotiation
 

---

<b>Syntax:</b>	CacheNegotiatedDocs On Off
<b>Default:</b>	CacheNegotiatedDocs Off
<b>Context:</b>	server config, virtual host
<b>Status:</b>	Base
<b>Module:</b>	mod_negotiation
<b>Compatibility:</b>	The syntax changed in version 2.0.

If set, this directive allows content-negotiated documents to be cached by proxy servers. This could mean that clients behind those proxys could retrieve versions of the documents that are not the best match for their abilities, but it will make caching more efficient.

This directive only applies to requests which come from HTTP/1.0 browsers. HTTP/1.1 provides much better control over the caching of negotiated documents, and this directive has no effect in responses to HTTP/1.1 requests.

Prior to version 2.0, `CacheNegotiatedDocs` did not take an argument; it was turned on by the presence of the directive by itself.

## ForceLanguagePriority Directive

---

<b>Description:</b>	Action to take if a single acceptable document is not found
<b>Syntax:</b>	ForceLanguagePriority None Prefer Fallback [Prefer Fallback]
<b>Default:</b>	ForceLanguagePriority Prefer
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	FileInfo
<b>Status:</b>	Base
<b>Module:</b>	mod_negotiation
<b>Compatibility:</b>	Available in version 2.0.30 and later

The `ForceLanguagePriority` directive uses the given `LanguagePriority` to satisfy negotiation where the server could otherwise not return a single matching document.

`ForceLanguagePriority Prefer` uses `LanguagePriority` to serve a one valid result, rather than returning an HTTP result 300 (MULTIPLE CHOICES) when there are several equally valid choices. If the directives below were given, and the user's `Accept-Language` header assigned `en` and `de` each as quality `.500` (equally acceptable) then the first matching variant, `en`, will be served.

```
LanguagePriority en fr de
ForceLanguagePriority Prefer
```

`ForceLanguagePriority Fallback` uses `LanguagePriority` to serve a valid result, rather than returning an HTTP result 406 (NOT ACCEPTABLE). If the directives below were given, and the user's `Accept-Language` only permitted an `es` language response, but such a variant isn't found, then the first variant from the `LanguagePriority` list below will be served.

```
LanguagePriority en fr de
ForceLanguagePriority Fallback
```

Both options, `Prefer` and `Fallback`, may be specified, so either the first matching variant from

`LanguagePriority` will be served if more than one variant is acceptable, or first available document will be served if none of the variants matched the client's acceptable list of languages.

### See also

- [AddLanguage](#)

## LanguagePriority Directive

---

<b>Description:</b>	The precedence of language variants for cases where the client does not express a preference
<b>Syntax:</b>	<code>LanguagePriority <i>MIME-lang</i> [<i>MIME-lang</i>] ...</code>
<b>Context:</b>	server config, virtual host, directory, .htaccess
<b>Override:</b>	FileInfo
<b>Status:</b>	Base
<b>Module:</b>	mod_negotiation

The `LanguagePriority` sets the precedence of language variants for the case where the client does not express a preference, when handling a MultiViews request. The list of *MIME-lang* are in order of decreasing preference.

### Example:

```
LanguagePriority en fr de
```

For a request for `foo.html`, where `foo.html.fr` and `foo.html.de` both existed, but the browser did not express a language preference, then `foo.html.fr` would be returned.

Note that this directive only has an effect if a 'best' language cannot be determined by any other means or the `ForceLanguagePriority` directive is not `None`. Correctly implemented HTTP/1.1 requests will mean this directive has no effect.

### See also

- [AddLanguage](#)

## URI References

---

[1] <http://httpd.apache.org/docs-2.1/content-negotiation.html>

[2] <http://www.ietf.org/rfc/rfc1766.txt>