

Transparent Content Negotiation

Start Presentation

Table of Contents

1. [Transparent Content Negotiation](#)
2. [Content Negotiation?](#)
3. [Introduction](#)
4. [Variant Selection](#)
5. [Dimensions of Negotiation](#)
6. [Why using Content Negotiation?](#)
7. [Content Negotiation Support](#)
8. [HTTP/1.0 Content Negotiation](#)
9. [Type Negotiation](#)
10. [Charset Negotiation](#)
11. [Encoding Negotiation](#)
12. [Language Negotiation](#)
13. [Apache Negotiation Algorithm](#)
14. [Fiddling with Quality Factors](#)
15. [Transparent Content Negotiation](#)
16. [TCN related HTTP Headers](#)
17. [Negotiate Header](#)
18. [Alternates Header](#)
19. [Variant Information](#)
20. [TCN Header](#)
21. [Vary Header](#)
22. [Variant-Vary Header](#)
23. [Accept-Feature Header](#)
24. [Feature Negotiation](#)
25. [Remote Variant Selection Algorithm](#)

Author: Lars Eilebrecht

E-Mail: lars@apache.org

26. [RVSA Example](#)
 27. [List Response](#)
 28. [Type-Map and MultiViews Configuration](#)
 29. [Type-Maps](#)
 30. [Type-Map Variant Fields](#)
 31. [Type-Map Example \(1\)](#)
 32. [Type-Map Example \(2\)](#)
 33. [MultiViews](#)
 34. [Filename Structure](#)
 35. [MultiViews](#)
 36. [Example: Language Configuration](#)
 37. [MultiViews and CGI](#)
 38. [MultiViews and Type-Maps](#)
 39. [Limitations of Content Negotiation](#)
 40. [The End](#)
-



Transparent Content Negotiation



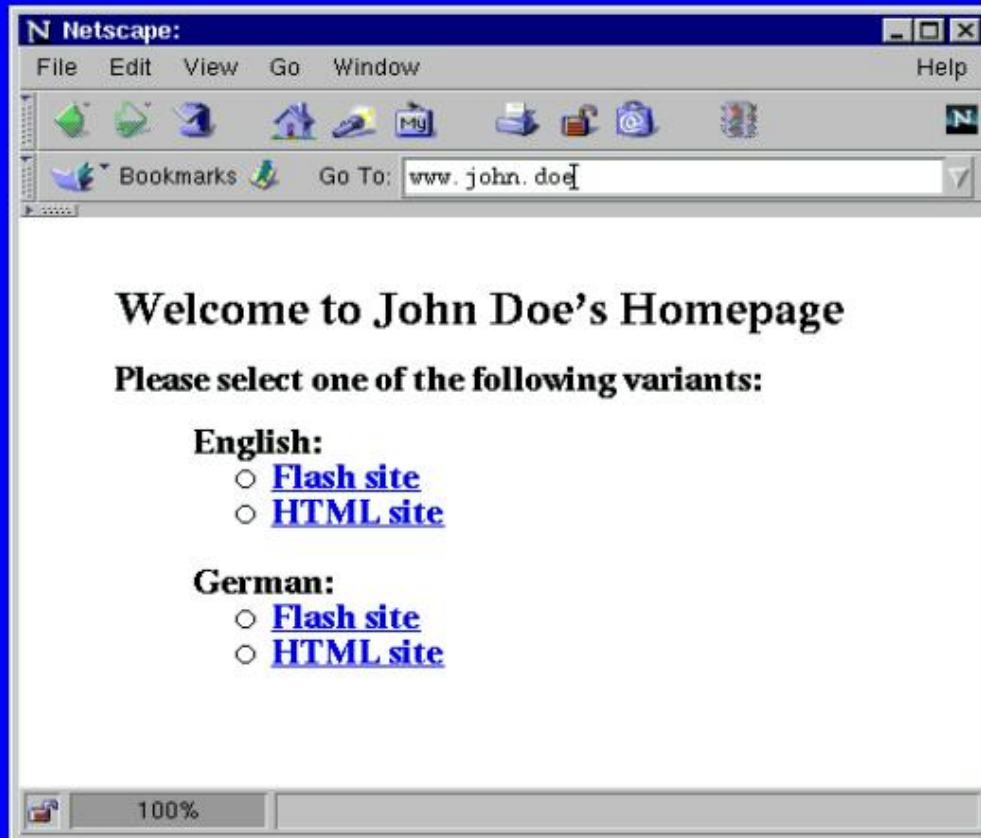
Lars Eilebrecht
<lars@apache.org>

ApacheCon 2000 - Lars Eilebrecht



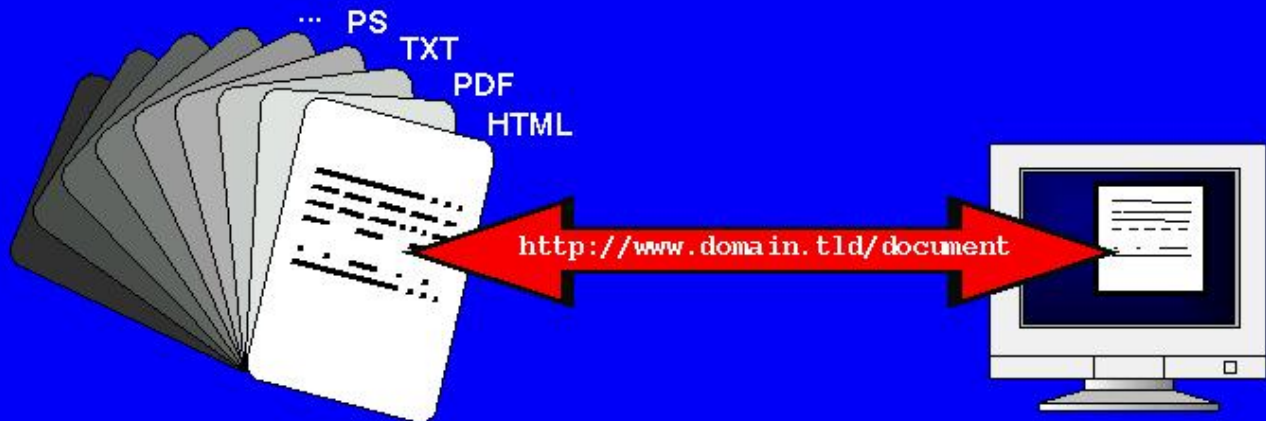


Content Negotiation?





Introduction



- Multiple representations of the same resource are *hidden* behind a single URL
- Each representation is called a »variant«



Variant Selection

- The *best* variant is automatically selected if a client requests a resource.
- The selection is done based on the client's preferences and capabilities.
- This mechanism is called »Content Negotiation«.
- Content Negotiation is an extension of HTTP.





Dimensions of Negotiation

- Media type
- Charset
- Language
- Encoding





Why using Content Negotiation?

- Eliminating the need to manually choose between different representations of a resource.
- Graceful deployment of new data formats.
- Eliminating error-prone and cache-unfriendly user-agent based negotiation.
- Providing multi-lingual content without a bias towards one language.





Content Negotiation Support

- Apache module: *mod_negotiation*
- HTTP/1.0 Content Negotiation
[available since the first Apache version]
- Transparent Content Negotiation
(TCN) [available since version 1.3.4]





HTTP/1.0 Content Negotiation

- Server-driven negotiation
- Negotiation related HTTP headers:
 - Accept
 - Accept-Charset
 - Accept-Encoding
 - Accept-Language
- These HTTP headers express the user agent capabilities and user preferences





Type Negotiation

- The user agent submits a list of acceptable media types in the »Accept« HTTP header
- Example:

```
Accept: image/png,  
        image/gif;q=0.5,  
        image/jpeg;q=0.7,  
        image/*;q=0.01
```
- »q« specifies the »quality factor«





Charset Negotiation

- The »Accept-Charset« header contains a list of acceptable character sets.

- Example:

```
Accept-Charset:  
    iso-8859-1,  
    unicode-1-1;q=0.9
```





Encoding Negotiation

- The »Accept-Encoding« header contains a list of acceptable encoding types.
- Example:
`Accept-Encoding:
gzip, compress;q=0.1`





Language Negotiation

- The »Accept-Language« header contains a list of acceptable language definitions as defined in ISO 639.
- Example:
`Accept-Language:
en, de;q=0.9, es;q=0.2`





Apache Negotiation Algorithm

- Check Accept headers and assign quality factor to each variant by multiplying the quality factor from the header by the source quality factor of the variant.
- Order of selection: media type, language, character set, encoding
- Select the variant with the smallest content length
- Send best variant or »Not acceptable« message to the client.





Fiddling with Quality Factors

- If no quality factors are given in an Accept header the sequence is taken into account (the second tag is set to a slightly smaller quality factor, etc.)
- Wildcard media types:
 - »*media-type*/*« is given a factor of 0.02.
 - »*/*« is given a factor of 0.01.
- Variants without language identification are given a source quality of 0.001.





Transparent Content Negotiation

- TCN is defined in RFC 2295 (experimental), Koen Holtman and Arthur Mutz, March 1998.
- TCN combines server-driven and agent-driven content negotiation.
- Servers with TCN support must implement the »Remote Variant Selection Algorithm« (RVSA).
- RVSA/1.0 is defined in RFC 2296.





TCN related HTTP Headers

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Accept-Features
- Negotiate
- Alternates
- TCN
- Vary
- Variant-Vary





Negotiate Header

- The »Negotiate« header is used by the user agent to signal TCN support and to request agent- or server-driven negotiation
- Values:
 - trans
 - vlist
 - guess-small
 - (rsva-version, e.g. 1.0)
 - *





Alternates Header

- The »Alternates« header is used by the server and contains a list of all available variants.
- Example:

Alternates:

```
["page.html.en" 1 {type text/html}
 {language en} {length 1877}},
["page.html.ja.jis" 0.9 {type text/html}
 {charset iso-2022-jp} {language ja}
 {length 1623}}
```





Variant Information

- URI
- Source quality
- Type
- Charset
- Encoding
- Language
- Length
- Features
- Description
- Extension attributes
- List directive
- Example:

```
{"page.html.ja" 1 {type text/html}
 {language ja} {length 1877}
 {description "Japanese documentation"}},
 {"page.html.en" }
```





TCN Header

- The »TCN« header is used by the server to signal that the resource is transparently negotiated.
- Values:
 - list
 - choice
 - adhoc
 - re-choose
 - keep





Vary Header

- The »Vary« header is used by the server to signal that the response is cacheable (by a HTTP/1.1 cache)
- Example:

```
Vary: negotiate, accept-language, accept-charset
```





Variant-Vary Header

- The »Variant-Vary« header can be used by the server to record any vary information which applies to the variant data contained in the response.





Accept-Feature Header

- The »Accept-Feature« header is used by the user agent and contains a list of »feature tags«.
- Feature negotiation is intended to provide for all areas of negotiation not covered by the normal dimensions of negotiation.





Feature Negotiation

- Examples:
 - HTML extensions (e.g., frames, javascript) and extensions of other media types
 - Special capabilities of the user agent (e.g., color or sound support)
 - Screen size
 - Special user preferences
- Feature tag registry (RFC 2506)
- Apache doesn't yet support Feature Negotiation.





Remote Variant Selection Algorithm

- Input:
 - The list of all variants and their information
 - Information from the user agent's Accept headers
- Output:
 - Choice or list response





RVSA Example

- Available variants:
 - »page.html.en« (source quality: 1.0)
 - »page.html.de« (source quality: 0.9)
- Request-URI: »page.html«
- Examples:
 - Accept-Language: de, en;q=0.5
 - Accept-Language: de, en;q=0.9
 - Accept-Language: fr





List Response

300 Multiple Choices

Multiple Choices

Available variants:

```
* index.html.ca , type text/html, language ca
* index.html.cz , type text/html, language cz
* index.html.de , type text/html, language de
* index.html.dk , type text/html, language da
* index.html.ee , type text/html, language et
* index.html.en , type text/html, language en
* index.html.es , type text/html, language es
* index.html.fr , type text/html, language fr
* index.html.it , type text/html, language it
* index.html.lu , type text/html, language ltz
* index.html.nl , type text/html, language nl
* index.html.pt , type text/html, language pt
* index.html.se , type text/html, language sv
```

Apache/1.3.12 Server at www.apache.org Port 80





Type-Map and MultiViews Configuration

- Apache-Module: *mod_negotiation*
- Handler: type-map
- Directives:
 - CacheNegotiatedDocs
 - LanguagePriority
 - AddLanguage
 - DefaultLanguage
 - AddCharset
 - AddDefaultCharset
 - AddEncoding
 - AddType
 - ForceType
 - AddHandler
 - SetHandler
 - RemoveHandler





Type-Maps

- A type-map is a plain text file defining all available variants of a specific resource.
- The name of the type-map is used to request the resource (e.g., »page.var«).
- Bind handler to a filename extension:
`AddHandler type-map .var`





Type-Map Variant Fields

- URI
- Content-Type
- Content-Length
- Content-Language
- Content-Encoding
- Description





Type-Map Example (1)

```
#
# type-map: photo.var
#

URI: photo.png
Content-Type: image/png; qs=1
Description: "Truecolor PNG image"

URI: photo.gif
Content-Type: image/gif; qs=0.9
Description: "256color GIF image"

URI: photo.jpg
Content-Type: image/jpeg; qs=0.7
Description: "Truecolor JPEG image
              with 70% quality level"
```





Type-Map Example (2)

```

#
# type-map: extreme.var

URI: extreme.iso.html.de
Content-Type: text/html; qs=...
Content-Language: de
Description: "German HTML document, unicode-1-1"

URI: extreme.iso.html.en
Content-Type: text/html; qs=...
Content-Language: en
Description: "English HTML document, unicode-1-1"

URI: extreme.iso.txt.de
Content-Type: text/plain; qs=...
Content-Language: de
Description: "German plain text document, unicode-1-1"

URI: extreme.iso.txt.en
Content-Type: text/plain; qs=...
Content-Language: en
Description: "English plain text document, unicode-1-1"
...

URI: extreme.uni.html.de
Content-Type: text/html; qs=0.9; charset=unicode-1-1
Content-Language: de
Description: "German HTML document, unicode-1-1"

URI: extreme.uni.html.en
Content-Type: text/html; qs=0.9; charset=unicode-1-1
Content-Language: en
Description: "English HTML document, unicode-1-1"

URI: extreme.uni.txt.de
Content-Type: text/plain; qs=0.2; charset=unicode-1-1
Content-Language: de
Description: "German plain text document, unicode-1-1"

URI: extreme.uni.txt.en
Content-Type: text/plain; qs=0.2; charset=unicode-1-1
Content-Language: en
Description: "English plain text document, unicode-1-1"

```





MultiViews

- Internal on-the-fly type-map creation
- Find variants via »MultiViews search«
- The file name of a valid variant starts with the request-URI and has one or more known extensions.
- The information about media type, language, charset and encoding is taken from the file extensions.





Filename Structure

Name.Type.Language.Charset.Encoding

- Ordering of extensions is not relevant
- Examples:
 - page.html.en
 - page.en.html
 - page.html.ja.jis
 - page.gz.html.ja.jis
- Unknown extensions have to be defined in the server configuration





MultiViews

- Advantage:
 - No need to create and maintain type-maps.
- Drawbacks:
 - Reduced flexibility (no quality of source factor, no description)
 - CPU- and I/O overhead (low)





Example: Language Configuration

```
<Location />  
Options +MultiViews  
</Location>
```

```
AddLanguage en .en  
AddLanguage de .de  
AddLanguage fr .fr
```

```
LanguagePriority en de fr
```

```
DefaultLanguage en
```





MultiViews and CGI

- If one of the variants is a CGI script (and it is either a POST request or GET request with a query-string) it is given an extremely high quality factor.
- For all other requests an extremely low quality factor is given to the CGI script.





MultiViews and Type-Maps

- Both features can be combined.
- If Apache finds a type-map during a MultiViews search the variant information is taken from the type-map.





Limitations of Content Negotiation

- TCN is still experimental
- Insufficient user agent support
- Misconfigured user agents
- Browser implementations are broken
- Users are confused by list responses
- File name handling of HTML editors and publishing tools





<http://www.apache.org/httpd>