How browsers support Multilingual Content

Arun Viswanathan (aviswana@ieee.org) 01/23/09

Multilingual content is supported using internationalization (i18n) support in browsers. In simple terms its the support for displaying information in various languages. Every language is supported by specifying its characters as a **Coded Character Set**. The coded character set has a set of characters with a unique number (called a **code point**) assigned to each character. For example, English will have a character set, Hindi will have a character set and so on. The character ‘A’ in English has the unique identifier (or code point) of 65. **Character encoding** is the way these abstract characters are mapped to bytes for manipulation in a computer. For example, the character ‘A’ is represented in a byte in the ISO-8859-1 encoding while it can be represented in 1, 2, or 4 bytes in the Unicode Encoding Formats. It is important to note the distinction between the character set and its encoding. *There is only one character set but there can be multiple encodings.*

For XML and HTML (from version 4.0 onwards) the document character set is defined to be the Universal Character Set (UCS) or Unicode Standard. What this means is that HTML and XML documents will *use the Unicode Character Set but can have any Character Encodings.*[1]

There are many Character Encodings in use today. One easy way to find them out is to open the Firefox browser and look under View → Character Encodings. It should be noted that the Unicode Character Set and its associated encodings (UTF-8/UTF-16/UTF-32) are supposed to encompass all the existing character encodings in future. Also, what Unicode allows is to use multiple languages on the same page (say Hebrew, English and Chinese). This is something which is difficult to support with say the ISO-8559 series of encodings because they only encode the Western European Languages.

---

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit [http://creativecommons.org/licenses/by-nc-sa/3.0/](http://creativecommons.org/licenses/by-nc-sa/3.0/) or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
These are ways in which one can define the encodings for HTML/XML/CSS pages.

1) Define encoding in the HTTP Header sent by the server
   
   Content-Type: text/html; charset=iso-8859-1

2) For HTML or XHTML document (with content-type: text/html)
   
   `<meta http-equiv="Content-type" content="text/html; charset=UTF-8"/>
   <link href="okinawa.html" rel="parent" charset="euc-jp"/>

3) For XML
   
   `<?xml version="1.0" encoding="UTF-8"?>

4) For CSS Style Sheets
   
   @charset "utf-8";

Values for the encoding attribute can be found in the IANA registry [3]. Though these are called charset, they refer to the encodings and not the character sets.

These are the Precedence Rules that browsers will use if there are encodings specified in multiple places.

1. HTTP header Content-Type
2. XML declaration
3. `<meta ... charset=...> declaration
4. `<link ... charset=...> declaration

For external, linked CSS style sheets the precedence rules are:

1. HTTP header Content-Type
2. @charset rule
3. `<link charset=".." rel="stylesheet" … />

Experiments using Firefox

Loading the website [http://www.xinhuanet.com](http://www.xinhuanet.com) automatically displays the page in Chinese characters. This is because the encoding has been defined correctly by the authors of the page and my firefox browser understands the encoding.

HTTP capture of one of the GET requests shows that the response (in red) contains the charset GB2312. This helps the browser in identifying the encoding used in the page.

[http://www.xinhuanet.com/gundong_bobao.htm](http://www.xinhuanet.com/gundong_bobao.htm)

GET /gundong_bobao.htm HTTP/1.1
Host: www.xinhuanet.com

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit [http://creativecommons.org/licenses/by-nc-sa/3.0/](http://creativecommons.org/licenses/by-nc-sa/3.0/) or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.5) Gecko/2008121622
Ubuntu/8.10 (intrepid) Firefox/3.0.5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
If-Modified-Since: Fri, 23 Jan 2009 05:51:33 GMT
If-None-Match: "8608a-197d-fb30c340"
Cache-Control: max-age=0

HTTP/1.1 304 Not Modified
Content-Type: text/html; charset=GB2312
Last-Modified: Fri, 23 Jan 2009 05:51:33 GMT
Etag: "8608a-197d-fb30c340"
Date: Fri, 23 Jan 2009 05:49:34 GMT
Connection: keep-alive
Vary: Accept-Encoding

Additionally, the source code of the page reveals that the page contains the following meta tag.

```html
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
```

Also, firefox automatically displays the current character encoding that it is using under View → Character Encoding. As you can see below, the encoding is displayed as Chinese Simplified (GB2312)

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
Another experiment I did was to change the encoding to UTF-8 by using View → Character Encodings and selecting UTF-8. Unexpectedly, the page turned to gibberish. According to definitions, the document is supposed to correctly display in UTF-8 too but it does not. It turns out the UTF-8 (inspite of its Chinese support) may not be the most efficient encodings for Chinese fonts. The GB2312 encoding is more efficient (according to [4]) and hence is the preferred choice of encoding in mainland China.

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
An Indian website (shown above) in the Marathi language uses the UTF-8 encoding by default. This ensures maximal compatibility as every browser would at the least support the UTF-8 encoding.

Supporting Multilingual options on the same page

There can be two possible solutions:
1) Use different URLs for multiple languages and put them on top of the page. Like for example, xinhuanet gives you the same news in English/French/Russian/Arabic/Latin etc. Clicking on the links takes you to a different webpage where the encodings are specified differently.

Something interesting I found out about this after clicking on each link:

<table>
<thead>
<tr>
<th>Language</th>
<th>Encoding in HTTP header</th>
<th>Encoding specified in Index.html</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese (GB) Version</td>
<td>GB2312</td>
<td>GB2312</td>
</tr>
<tr>
<td>Chinese (Big5) Version</td>
<td>Big5</td>
<td>Big5</td>
</tr>
<tr>
<td><strong>English</strong></td>
<td><strong>GB2312</strong></td>
<td>ISO-8859-1</td>
</tr>
<tr>
<td>Russian</td>
<td>utf-8</td>
<td>utf-8</td>
</tr>
<tr>
<td>Arabic</td>
<td>utf-8</td>
<td>utf-8</td>
</tr>
<tr>
<td><strong>French</strong></td>
<td><strong>Utf-8</strong></td>
<td>iso-8859-1</td>
</tr>
<tr>
<td>Latin</td>
<td>Utf-8</td>
<td>utf-8</td>
</tr>
</tbody>
</table>

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
There is a discrepancy in the encodings specified for the English and French sites. And according to priority the encoding specified in the HTTP header would be used. But it seems GB2312 also supports the basic ASCII set and hence the English version still displays correctly. But i think this is a bug and cause problems.

2) The other solution is to use UTF-8 as the default encoding as it allows one to use a combination of character sets (thats what it is for). One could use javascript to hide display information according to language selected. This is the recommended way i guess.

References