

The Reality of Web Encoding Identification, or Lack Thereof

--- *What to trust to determine the character encoding of a web page?* ---

KUROSAKA Teruhiko¹, Internationalization Architect, IONA Technologies²

1. Introduction

In order for the web browser to render the contents of the web pages, it has to interpret the byte stream that comes as an HTTP response. Interpretation of the bytes depends on the character encoding of the web page, which must be announced (declared) explicitly to the browser, somehow inferred, or guessed by the browser. There are two methods in the Internet standards that can be used to announce the character encoding. These will be introduced in the next subsections.

Another aspect of the web page, closely related to the character encoding, but not as important, is the natural language in which the page is written. The Internet standards also give three different ways to declare the natural language of the web page, and these, too, will be introduced shortly.

The problem is that there is more than one standard method of achieving the same purpose, and use of them is not mandatory. The author wanted to know, *in practice*:

- Do we have to deal with all of these methods?
- What methods should we focus on?
- Are all of them reliable?

To answer these questions, the author examined representative sites in Japanese and other languages.

2. Internet Standards

The following two Internet standards define character encoding and language announcement mechanisms:

- RFC 2616: Hypertext Transfer Protocol—HTTP/1.1
<<http://www.ietf.org/rfc/rfc2616.txt>>.
- HTML 4.01 Specification <<http://www.w3.org/TR/html401/>>: Section 5.2. <<http://www.w3.org/TR/html401/charset.html#encodings>> discusses the character encoding.

¹ In the Japanese *locale*, the surname precedes the given name.

² This paper was proposed, and part of the research was performed, when the author worked for Basis Technologies Corporation as Senior Software Engineer.

2.1. Charset Attribute of Content-Type HTTP Header

RFC2616 in Sections 3.4., 3.7.1, and 14.17 discusses the *charset* attribute of the *Content-Type* header. The charset attribute is used as in the following examples to announce the character encoding of the data:

```
Content-Type: text/html; charset=EUC-JP
```

```
Content-Type: text/html; charset=Shift_JIS
```

The charset attribute is optional. When the Content-Type is missing the charset attribute, ISO-8859-1 is assumed, according to Section 3.7.1 of RFC 2616 that reads:

The *charset* parameter is used with some media types to define the character set (section 3.4) of the data. When no explicit charset parameter is provided by the sender, media subtypes of the "text" type are defined to have a default charset value of "ISO-8859-1" when received via HTTP. Data in character sets other than "ISO-8859-1" or its subsets MUST be labeled with an appropriate charset value. See section 3.4.1 for compatibility problems.

2.2. Charset Attribute of Content-Type in META Tag

The HTML 4.01 standard, in Subsection 7.4.4.2, provides a way for an HTML author to specify the HTTP header information through use of the *http-equiv* attribute within a META tag (or *element*, formally speaking). In this way the HTML author can specify the charset attribute of the Content-Type header using the http-equiv attribute. Below are some examples:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=EUC-JP">
```

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Shift_JIS">
```

Note that although the standard seems to suggest that HTTP servers will use this information to generate the HTTP headers using the META http-equiv information, most HTTP servers, in reality, do not do this. It is up to the web browsers to interpret the META http-equiv information.

2.3. Lang attribute of HTML tag

Section 8.1 of the HTML 4.0.1 standard (<http://www.w3.org/TR/html401/>) defines a way of announcing the natural language for the whole or a part of an HTML document using the *lang* attribute within tags. Although the standard allows the lang attribute to appear in various types of tags, in practice it is usually used only in the HTML start tag, <HTML>, declaring the language for the whole document. Below are some examples:

```
<HTML lang="en"> <!-- English -->
```

```
<HTML lang="en_US"> <!-- American English -->
```

```
<HTML lang="ja"> <!-- Japanese -->
```

2.4. Content-Language HTTP Header

RFC2616, in Section 14.12, defines an HTTP header called Content-Language as:

The Content-Language entity-header field describes the natural language(s) of the intended audience for the enclosed entity.

Below are some usage samples of the Content-Language header:

```
Content-Language: en
Content-Language: en_US
Content-Language: ja
```

2.5. Content-Language in META Tag

As is the case for the Content-Language header, the META tag and the http-equiv attribute can be used to substitute the real headers in HTTP communication. Below are some usage samples:

```
<META HTTP-EQUIV="Content-Language" CONTENT="en">
<META HTTP-EQUIV="Content-Language" CONTENT="en_US">
<META HTTP-EQUIV="Content-Language" CONTENT="ja">
```

3. How Are These Standards Used

As described in the previous section, there are two ways of announcing the character encoding of the web document, and three ways of announcing the language.

The rest of the paper describes the results of the author's survey of real world use of these methods.

3.1. Method of Survey

We visited the home pages of the top 50 popular web sites (per domain) as of May 2000, as listed in Appendix 2 of the research paper titled *Cultural Differences in E-Commerce: A Comparison Between the U.S. and Japan*, by Ms. Kumiko Aoki³, which is available at:

http://www.firstmonday.dk/issues/issue5_11/aoki/#appendix2

Internet Explorer 6.0 from Microsoft Corporation was used to visit the site. The Language Preference was set to have Japanese (ja) as the first preference and American English (en-US) as the second preference. The page was saved using the *Save As...* command, and the saved page was viewed using a multi-lingual port of GNU Emacs 20.7.1 to Windows platform called Meadow. The HTTP headers were captured using a HTTP filtering tool called *muffin* (available from <http://muffin.doit.org>) using its *snoop* filter.

³ Ms. Aoki writes her name in the Westernized given name-surname order.

Similar survey was conducted against Chinese (both Traditional and Simplified) and French language sites in a smaller scale.

10 Chinese sites were randomly picked from the list of popular Chinese language sites found at:

<http://www.chinasite.com/Media/Magazine.html>

5 French language sites were taken from the links found in:

http://www.suite101.com/article.cfm/secrets_of_paris/32877

and other sources.

3.2. Results for Japanese Sites

Although 50 web sites were tried from the list, the results from seven sites were not considered for the purpose of this research because:

- At four sites, the home pages were in English.
- Two sites merely redirected to, or contain the exact same contents of another site that was checked. (This was due to a merger of these sites.)
- One site was closed. (The Nikkei BP list contained new and old domain names in transition; the old one had been phased out.)

Of the remaining 43 valid sites, some results are listed in Table 1. The original spellings of the encodings and languages are shown in the second to sixth columns. The last column shows the encodings in the normalized spellings.

Table 1. Japanese Sites(Excerpt)

URL (http://www. omitted)	Content-Type header	Content-Type using Meta tag	<HTML lang="...">	Content-Lang header	Content-Lang using Meta tag	Actual encoding
yahoo.co.jp	euc-jp	-	-	-	-	EUC-JP
microsoft.com/ japan	Shift_JIS	UTF-8 (wrong)	-	-	-	Shift_JIS
biglobe.ne.jp	-	SHIFT_JIS	ja	-	-	Shift_JIS
geocities.co.jp	-	euc-jp	-	-	-	EUC-JP
lycos.co.jp	Shift_JIS	-	-	-	-	Shift_JIS
asahi-net.or.jp	-	Shift_JIS	ja-JP	-	-	Shift_JIS
rim.or.jp	-	Shift_JIS and x-sjis ⁴	ja-JP	-	-	Shift_JIS
ijj4u.or.jp	-	-	-	-	-	EUC-JP
infoseek.co.jp	EUC-JP	EUC-JP	-	-	-	EUC-JP
rakuten.co.jp	-	x-euc-jp	-	-	ja	EUC-JP
sakura.ad.jp (sakura.ne.jp)	Shift_JIS	Shift_JIS	ja	-	-	Shift_JIS

⁴ Two META tags with http-equiv attribute for Content-Type exist in the same HTML page.

After analyzing the data gathered, the following can be said regarding character encoding identification:

- Use of META tags with http-equiv attribute for Content-Type is the most popular approach. 38 sites, or 88%, use this method.
- Content-Type header is used by 7 sites (16%).
- 5 sites (11%) use both the header and the META http-equiv methods. One of the sites, however, put inconsistent information, Shift-JIS in the header, which was correct, and UTF-8 in the META http-equiv tag, which was wrong.
- 2 sites (5%) use only the Content-Type header.
- 3 sites (7%) do not use any method of character encoding announcement. One of them actually uses Shift_JIS while the other two use EUC-JP. This is against the rule quoted in Section 2.1, according to which, ISO 8859-1 must be assumed.

Though it is not the purpose of this research, it should be noted that 32 sites (74%) use Shift_JIS and the other 11 sites (26%) use EUC-JP. No sites that use ISO-2022-JP or UTF-8 were found among the sites in this research. It does not mean, however, such sites do not exist. Sun Microsystems' Japanese site <<http://www.sun.co.jp>>, for example, uses the ISO-2022-JP encoding.

Regarding natural language announcement, the following can be said for the Japanese sites surveyed:

- 30 sites (70%) use no language announcement mechanism.
- 12 sites (28%) use the lang attribute of HTML tag.
- Only 1 site uses the Content-Lang within META http-equiv tag.
- The Content-Lang header is not used at all.

3.3. Results for Chinese Sites

Of all of the 6 sites that use GB2312 and 4 sites that use Big5, they all use the META tag with http-equiv attribute for Content-Type for the encoding announcement. None use any form of natural language announcement mechanisms.

Table 2. Chinese Sites

URL (http://www omitted)	Content- Type header	Content-Type in Meta tag	<HTML lang="...">	Content -Lang header	Content- Lang in Meta tag	Actual encoding	Country
qikan.com	-	charset=gb2312	-	-	-	GB2312	CN
xinhua.org	-	charset=gb2312	-	-	-	GB2312	CN
nanfangdaily.com.cn	-	charset=gb2312	-	-	-	GB2312	CN
qingyun.com	-	charset=gb2312	-	-	-	GB2312	CN
otm.com.cn/	-	charset=gb2312	-	-	-	GB2312	CN
peopledaily.com.cn	-	charset=gb2312	-	-	-	GB2312	CN
next.atnext.com	-	charset=big5	-	-	-	BIG5	HK

netvigator.com	-	charset=big5	-	-	-	BIG5	HK
hkcyber.com	-	charset=big5	-	-	-	BIG5	HK
worldscreen.com.tw	-	charset=big5	-	-	-	BIG5	TW

3.4. Results for French Sites

Of all of the 5 French sites, they all use the META tag with http-equiv attribute for Content-Type for the encoding announcement. None use any form of natural language announcement mechanisms.

Table 3. French Sites

URL (http://www omitted)	Content-Type header	Content-Type in Meta tag	<HTML lang="...">	Content-Lang header	Content-Lang in Meta tag	Actual encoding
canalplus.fr	-	charset=iso-8859-1	-	-	-	ISO-8859-1
agoride.com	-	charset=iso-8859-1	-	-	-	ISO-8859-1
parisbalades.com	-	charset=iso-8859-1	-	-	-	ISO-8859-1
service-public.fr	-	charset=iso-8859-1	-	-	-	ISO-8859-1
zdnnet.fr	-	charset=iso-8859-1	-	-	-	ISO-8859-1

4. Conclusion and Recommendation

4.1. Character Encoding Identification

For the purpose of identifying the character encoding, the META http-equiv tag is the most widely used approach. This virtually makes it *mandatory* for any applications that interpret the HTML pages to be able to obtain the charset information from the META http-equiv tag.

For Chinese and French sites, this seems to be the only character encoding announcement mechanism that developers need to worry about, and although it is reliable; a larger scale survey should be conducted to support this claim.

For Japanese sites, the situation is more complicated:

Sites that use *only* the HTTP Content-Type header with charset attribute *without* the same information presented using META http-equiv are rare but not negligible. The applications should take this into consideration.

There are major sites that do not use any character encoding announcement mechanism. There is also a case where the one encoding announcement was simply wrong. In order to build a robust application that functions properly in such cases, use of statistical and/or heuristic algorithms to identify the character encoding is necessary. There are some commercial and open-source solutions available.

Rosette Language Identifier from Basis Technology <<http://www.basistech.com>> is an example of commercial solutions. It is a C++ library that performs the encoding and language detection among the popular encoding and language combinations. With 64 byte input text, it can detect Japanese and Chinese at almost 100% accuracy, and French at 99% accuracy. The accuracy increases to 100% for the three languages with 128 byte input text.

On the other end of the spectrum is Java 2's built-in pseudo encoding "JISAutoDetect". This can be used if the possible encodings are limited to be one of the three popular Japanese encodings (ISO-2022-JP, Shift_JIS, EUC-JP), no tuning is needed, and the application does not need to know the source encoding. For C and Perl applications, a popular open-source software library called nkf <<http://www.vector.co.jp/soft/win95/util/se031296.html>> can be used.

4.2. Natural Language Identification

A surprising result of this survey is that natural language announcement is rarely used; only small number of Japanese language sites use the lang attribute within the HTML tag. Content-Language, whether in the HTTP header or in the META tag within HTML, is never used.

Because use of any of the three natural language announcement mechanisms is rare, the application cannot rely on these mechanisms in practice. For languages that can be unambiguously determined from the character encoding, such as Japanese, Korean, Thai, etc. this is not a real problem. For the languages which share a character encoding with many other languages, such as French, German, Spanish etc. which all use ISO-8859-1 (or its newer version, ISO-8859-15), we cannot infer a language from the character encoding. In such case, a statistical/heuristic algorithm seems to be the only solution.

4.3. Recommendation to Site Developers

This paper was mainly written with the Web application (applications that consume web pages, rather than those on the server-side) developers as audience in mind. This section, however, is written for the Web site developers.

When developing and managing Web sites, the author recommends to spend enough effort to achieve the followings:

- The Content-Type tag in HTTP protocol for the HTML page (text/html) includes the correct charset attribute, if possible. If that is not possible, charset attribute should be dropped entirely, so than wrong information will not be supplied.

- Every HTML page should have the META http-equiv tag for the Content-Type header with charset attribute.
- Every HTML page should use the lang attribute in its HTML start tag.
- Accept-Language header in request should be interpreted so that the requested language version of the page can be returned. (Note: very few web server products, however, implement this feature.)

4.4. Comments on the Standards and Trends

As noted, the combination of the two standards, HTTP/1.1 (RFC2616) and HTML 4.01, introduces several methods of achieving the same goal, identification of the character encoding and language. This duplication might have contributed confusion to the site developers and web server developers. The author hopes that the standard bodies work together to eliminate the duplication, and/or issue a guideline which method should be used.

Content-Language should probably be eliminated. This is not used by any implementation as far as the survey shows. The lang attribute in the HTML start tag alone would serve the purpose. This is also supported by the XML standard.

Whether the encoding announcement should be within the domain of the HTML language, or the HTTP protocol, is less clear. The author, however, feel that this should rather be done in the HTTP Content-Type header.

One reason for this is that the current use of META http-equiv seems not what the HTML specification originally intended. The specification reads:

HTTP servers may use the property name specified by the http-equiv attribute to create an [RFC822]-style header in the HTTP response.

in `<http://www.w3.org/TR/html401/struct/global.html#adef-http-equiv>`; the web server should interpret this META tag and generates the corresponding Content-Type (or other specified) header, and the client should only need to interpret the Content-Type header. (The client still needs to interpret META tags for other attributes than http-equiv.)

The latest version of Servlet specification, version 2.3, `<ftp://ftp.java.sun.com/pub/servlet/135790dwh/servlet-2_3-fcs-spec.pdf>` requires use of the charset attribute in the Content-Type header. If use of the Servlet technology gains popularity, this might lead to wider support of this encoding identification method by the servers.

The author suggests that the web server vendors to support the http-equiv attribute of the META tag at the server side so that the HTTP responses always have the correct charset identification. This could be achieved without degrading performance by parsing an HTML page once only after the HTML page is updated, and keeping the Content-Type information in some kind of cache.

Acknowledgement

This research evolved from initial investigation I conducted for a project when I worked at Basis Technology. Basis Technology encouraged me to submit a proposal for this paper to the Conference Committee. Basis Technology has been gracious to agree to allow me to continue this research after leaving the company.

I am also very grateful to IONA Technologies, my current employer, which let me continue this research and complete the paper. Without its support, this research could not be completed.