



DELIVERABLE

Grant Agreement number: 250500

Project acronym: MULTILINGUALWEB

Project title: Advancing the Multilingual Web, Thematic Network

D03.2 Practical work items: Report on first implementation of internationalization checker

Revision: 1.0

Authors:

Richard Ishida (W3C/ERCIM)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	P
C	Confidential, only for members of the consortium and the Commission Services	

PROJECT DELIVERABLE REPORT

Project	
Grant Agreement number	250500
Project acronym:	<i>Multilingual Web</i>
Project title:	<i>Advancing the Multilingual Web, Thematic Network</i>
Funding Scheme:	<i>Thematic Network</i>
Date of latest version of Annex I against which the assessment will be made:	26/10/2009
Document	
Deliverable number:	D03.2
Deliverable title	Practical work items: Report on First implementation of internationalization checker
Contractual Date of Delivery:	April 2011
Actual Date of Delivery:	23 May 2011
Editor (s):	N/a
Author (s):	Richard Ishida
Reviewer (s):	Jessica Michel
Work package no.:	WP03
Work package title:	Authoring the multilingual Web Project
Work package leader:	ERCIM/W3C
Version/Revision:	1.0
Draft/Final:	Final
Total number of pages (including cover):	13
Deliverable objective:	To summarise progress on the internationalization checker and training materials to date, in order to provide input to the face-to-face meeting planned for WP4.

Table of Contents

Internationalization Checker.....	4
Bucharest review.....	4
Current status	5
Work in progress.....	11
Training curriculum	12
Revision History	13

We will refer to the Internationalization Checker as the ‘i18n checker’ in this document.

Internationalization Checker

People creating HTML pages can run their page through the i18n checker to check for issues in their markup related to internationalization. Anyone can also use the checker to find out information about any page on the Web, such as what character encoding or language has been declared (either in the page or on the server).

Bucharest review

An initial prototype of the i18n checker was developed prior to the face-to-face meeting in Bucharest in June 2010. During that meeting, ideas were sought from the partners about ways to improve the tool.

Ideas included the following:

- Create a package that enterprises could use in non-Web applications.
- Make it available in an offline mode.
- Compare the actual language being used with the HTTP language header.
- Say something about linguistic quality, eg. spell-checking ‘global English’
- Produce an XML-based report.
- Recursive checking may become problematic.
- Check for formatting and markup that cause problems in certain languages, such as italics in Japanese.
- Suggest alternatives for inline CSS, since it causes localization problems.
- Mention ITS in the tool.

No commitment was made to apply any of these suggestions, but they were all considered. Some of the suggestions are being addressed by current work. Others may be addressed in the future.

A number of people volunteered to provide translations for the text used in the checker.

- Felix, German

- Jirka, Czech
- Andrzej, Polish
- Tadej, Slovene
- Luis, Spanish
- Andrea, Italian
- Pål, Norwegian (Bokmål)
- Tomas, other languages

Current status

The checker is now available at <http://validator.w3.org/i18n-checker>, having been moved from it's previous development location. This is the same server used for other validators from the W3C.

The checker allows you to check a page on the web by entering the URI into the address field on the user interface. The checker then returns a report for that page that lists:

1. a summary of how many errors, warnings and recommendations were found.
2. a panel of information about the internationalisation aspects of the page.
3. a description, for each error, warning and recommendation found, that lists a short title for the issue, a longer explanation of what the issue is about and why it is important, what you should do next, and a set of links to further reading.
4. the source text of the page that was analysed.

Sections and descriptions can be expanded or collapsed to make it easier for the user to view the information they want.

The checker currently supports HTML analysis, but does not fully support XHTML 1.1 or HTML5 documents.

The following diagrams show the checker after it has been run on a test page. The full page is shown in three parts to fit onto the pages of this document. Only a few representative items in the detailed report section have been opened.

W3C I18n Checker

http://validator.w3.org/i18n-checker/index?docAddr=

W3C W3C Internationalization Checker (Prototype only!)
Is your Web site Internationalized?

▼ Address http://rishida.net/tools/i18nchecker/test.php

http://rishida.net/tools/i18nchecker/test.php

▼ Results

✖ 3
! 7
i 2

▼ Information XHTML 1.0 :: text/html

Character encoding		Code
HTTP Content-Type	No charset found.	Content-Type: text/html
Byte order mark (BOM)	UTF-8	
xml declaration	None found.	
content-type meta	iso-8859-1	<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
HTML5 meta charset	None found.	
Language		Code
<html lang=	kk	<html lang="kk" xml:lang="to" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
<html xml:lang=	to	<html lang="kk" xml:lang="to" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
HTTP Content-Language	ka, ta	Content-Language: ka, ta
meta content-language element	en, fr, sp	<meta http-equiv="Content-Language" content="en, fr, sp" />
Detected language		
Text direction		Code
Default direction	ltr	<html lang="kk" xml:lang="to" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
Class & id names		Code
Non-ascii class or id names	8	<input type="button" value="Show list"/>
Non-NFC class or id names	4	<input type="button" value="Show list"/>
Request headers		
Accept-Language	en-gb,en;q=0.5	
Accept-Charset	UTF-8,*	

1 TOP

▼ Detailed report

Severity	Description
✖	Conflicting character encoding declarations.

Explanation

The following character encoding declarations are inconsistent:

- a. Byte-order mark: UTF-8
- b. `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />`

Browsers will apply precedence rules to determine the character encoding to use for the page, but this may not be the encoding you intended.

What to do

Change the character encoding declarations so that they match. Ensure that your document is actually saved in the encoding you choose.

Further reading

[Character encodings explained](#)
[Choosing a character encoding](#)
[Changing the encoding of a document](#)

↑ TOP

**The lang attribute and the xml:lang attribute in the html tag have different values.****Explanation**

The lang value is `kk` and the xml:lang value is `to` in the html tag:

```
<html lang="kk" xml:lang="to" dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
```

What to do

Change one of the values by editing the markup.

Further reading

[Language declarations explained](#)
[Using attributes to declare language](#)

↑ TOP

**Incorrect values used for dir attribute.****Multiple encoding declarations using the meta tag.****UTF-8 BOM found at start of file.****Explanation**

The UTF-8 Byte Order Mark (BOM) was found at the beginning of the page. It can sometimes introduce blank spaces or short sequences of strange-looking characters (such as `ï»¿`).

What to do

Using an editor or an appropriate tool, remove the byte order mark from the beginning of the file. This can often be achieved by saving the document with the appropriate settings in the editor. On the other hand, some editors (such as Notepad on Windows) do not give you a choice, and always add the byte order mark. In this case you may need to use a different editor.

Further reading

[Handling the byte-order mark](#)

↑ TOP

**A lang attribute value did not match an xml:lang value when they appeared together on the same tag.****A language attribute value was incorrectly formed.**

Explanation

In the following tag or tags the language values of the lang and xml:lang attributes are not well-formed according to BCP47. Attributes values must contain a maximum of one language tag, and a language tag is composed of one or more subtags taken from the IANA Language Subtag Registry, separated by hyphens (eg. zh-Hans-SG).

- `<!--<link title="___VERSION DESCN IN FOREIGN LANG." type="text/html" rel="alternate" hreflang="___LANG" href="___AND THE HREF" lang="___LANG" xml:lang="___LANG" />`
- ``
- `<p title="Armenian : Armenian" lang="hy, my" xml:lang="hy" class="phrase">`
- `<p title="Canadian Syllabics : Inuktitut" lang="iu" xml:lang="iu_CA" class="phrase">`


What to do


Change the attribute values to conform to BCP47 syntax rules.


Further reading


[Language declarations explained](#)
[Choosing language values](#)


↑ TOP

▶  **A tag uses a lang attribute without an associated xml:lang attribute.**

▶  **A tag uses an xml:lang attribute without an associated lang attribute.**

▶  **Class or id names found that are not in Unicode Normalization Form C.**

▶  **Non-UTF8 character encoding declared.**

▼  ** tags found with no class attribute.**

Explanation

One or more tags that don't use a class attribute were found in the source code for this page. These tags may cause problems for localization if the content for which they are used has more than one semantic value.

Total number of tags: 1.
Number of tags without a class attribute: 1.

What to do

You should not use tags if there is a more descriptive and relevant tag available. If you do use them, it is usually better to add class attributes that describe the intended meaning of the markup, so that you can distinguish one use from another.

Further reading

[Using and <i> tags](#)

↑ TOP

▶ Source code

[Home](#) | [About...](#) | [Feedback](#)

Internationalization (I18n) Activity
Making the World Wide Web world wide!

COPYRIGHT © 1994-2010 W3C® (MIT, ERCIM, KEIO). ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

The code and user interface have been updated and various bugs fixed since the discussion in Bucharest. Additional tests have also been added.

The following is a list of tests currently supported by the checker. This first list concerns tests for information about the page:

- Character encoding
 - HTTP Content-Type: Any character encoding information passed by the HTTP header.
 - Byte-order mark (BOM): Whether or not a BOM is present at the start of a file, and if so, which (utf-8 or utf-16).

- XML declaration: Whether an XML declaration appears at the top of the page, and if so, what encoding, if any, is declared there.
- Content-type meta: Whether a content-type meta element is used in the document <head>, and if so, what encoding, if any, is declared there.
- HTML5 meta charset: Detects whether the new HTML5 syntax is used for declaring the character encoding in the document <head>, and if so, what encoding is declared.
- Language
 - <html lang=: If the lang attribute is used in the <html> tag, what its value is.
 - <html xml:lang=: If the lang attribute is used in the <html> tag, what its value is.
 - HTTP Content-language: If the content-language header is used in HTTP, what is its value.
 - Meta content-language element: If this element is used in the document <head>, what is its value.
 - Detected language: This field is currently not used.
- Text direction
 - Default direction: Detects whether a dir attribute is used on the <html> element, and if so, reports its value.
- Class & id names
 - Non-ascii class or id names: Reports any class or id names that use non-ascii characters. These need to be normalised in the same way as CSS selectors in order to match and produce the desired effect.
 - Non-NFC class or id names: Reports any id or class names that are not normalized using Unicode Normalization Form C – the recommended form for use on the Web.
- Request headers
 - Accept-Language: Lists the Accept-Language header sent by the browser when the document was requested.

- Accept-Charset: Lists the Accept-Charset information sent by the browser when the document was requested.

The next list details the tests that are performed on the page once the initial information has been gathered. If a test fails, an issue is added to the Detailed report section.

- Errors
 - Conflicting character encoding declarations
 - The lang attribute and the xml:lang attribute in the html tag have different values
 - Incorrect values used for dir attribute
- Warnings
 - No character encoding information
 - Multiple encoding declarations using the meta tag
 - UTF-8 BOM found at start of file
 - BOM found in content
 - No in-document encoding found
 - The html tag has no language attribute
 - This HTML file contains xml:lang attributes [when served as text/html]
 - A lang attribute value did not match an xml:lang value when they appeared together on the same tag
 - A language attribute value was incorrectly formed
 - A tag uses a lang attribute without an associated xml:lang attribute
 - A tag uses an xml:lang attribute without an associated lang attribute
 - This XHTML file contains lang attributes [when served as XML]
 - Class or id names found that are not in Unicode Normalization Form C

- Information
 - Non-UTF8 character encoding declared
 - < b> tags found with no class attribute
 - < i> tags found with no class attribute

Work in progress

Work is currently under way to produce a new version of the checker, which it is hoped will go live within a month of this report.

Most of the changes are architectural, and include the following:

- The checker will allow file upload. This helps if you are still developing your page and wish to check it before making it live on a server.
- The look and feel of the checker will conform more with other validators on the W3C site, although there will still be various innovations.
- The checker code will use a parser rather than regular expressions to check the target page. Not only will this make the checker more robust, but it will also allow us to check HTML5 pages and pages served as XML.
- Code rewrites now enable full localization of the checker. This localization is done using a crowd-sourcing model, and the W3C HTML validator has translations into around 40 languages.
- Code rewrite now enables information to be externalised from the checker in XML form. The prime target for this functionality at the moment is to incorporate the results into the W3C's Unicorn validator, which groups results from various checkers, such as HTML, CSS, MobileOK, etc.
- Code rewrites also make the navigation of HTTP when retrieving a page more robust.
- We are investigating how to use the Google API to detect the language of the page and compare that against the language declarations in the page – or if there are none, suggest appropriate markup.

Training curriculum

There was some discussion during the Bucharest meeting of the idea of a training curriculum and some agreement that it would be useful to put one together. Note that this relates to defining topics related to standards and best practices that would figure in an educational curriculum, and not to the development of curriculum materials.

An IRC log of the discussion can be found at <http://www.w3.org/2010/06/02-mlw-minutes#item02>.

The Bucharest discussion generated enthusiasm for the development of the curriculum outline to be handled by partners in a dedicated subgroup, rather than by the W3C. Tomas Carrasco Benitez volunteered to lead that subgroup.

Tomas subsequently set up a dedicated wiki page at <http://www.w3.org/International/multilingualweb/wiki/Curriculum>, but there has not been a great deal of activity since.

We should revisit this idea at the Bled face-to-face meeting and decide on how to move forward.

Revision History

Revision	Date	Author	Organisation	Description
0.0	11/05/11	R.Ishida	W3C	Initial draft
0.1	12/05/11	J.Michel	ERCIM	Edits
1.0	20/05/11	J.Michel	ERCIM	Reviewed by MLW TN, prepared for submission