

HanWEB Publishing Server

HTML Authoring Style Guide

(Last updated: July 1, 2013)

Overview

HanWEB Publishing Server is designed to process web pages written in syntactically correct HTML. However, many existing web pages contain minor syntax errors because of the lenient nature of the HTML specification. Like most web browsers, HanWEB either ignores or makes assumptions of the errors. This approach works with some of the errors, but the rest of them often lead to unexpected results. Therefore, web developers who design web pages to be translated by HanWEB are recommended to ensure the correctness of their web pages' syntax. More information on HTML, including a free validation service, is available at W3C:

- HTML 4.01 Specification: <http://www.w3.org/TR/html401/>
- HTML Validation Service: <http://validator.w3.org/>

HanWEB returns translated web pages in two different formats depending on the browser and its language support. These two modes of translation will be referred to as *text-to-text* translation and *text-to-image* translation respectively. Text-to-text translation occurs when the browser supports the translated language. In this case, plain text in the original page will be translated into the desired language. Text-to-image translation occurs when the browser does not support the translated language. As its name suggests, plain text in the original page will be translated into image (GIF) files for display.

The various sections of this guide will show web developers who design web pages for translation with HanWEB areas that they should pay attention to. The sections apply to both text-to-text and text-to-image translation, unless specified otherwise. Explanation of common errors are given and examples of workarounds are suggested.

Character Set

HanWEB Publishing Server is a language-sensitive application. Although not required in most cases, one should always include a META tag specifying the *correct* character set in the HEAD section of your document. For example, if the document is encoded in Big5, the following tag should be used:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Big5">
```

This tag should be placed before the TITLE tag so that the browsers can also use the correct font to display the text in the title bar of the browser window.

Note that the second attribute ("text/html; charset=Big5") in this META tag must *always* be quoted by a pair of double quote characters because it contains a blank space. Otherwise, translation might yield unexpected results. Also, one should always ensure that the character set defined in the META tag is the one being used in the body of the page.

For more information on META tags, please consult the following web site:

HTML META, REL and REV Tags: <http://vancouver-webpages.com/META/>

Text in Images

HanWEB Publishing Server is designed to process plain text only. Text appearing inside an image is, straightly speaking, a part of that image. Therefore it will remain unchanged after translation. However, HanWEB can be configured to replace specified URLs in translated pages. This way another set of images created for the Simplified Chinese version can be displayed to Simplified Chinese viewers. See the next section for more details.

URL Replacement

Web developers can utilize the URL replacement feature of HanWEB to display different graphic files, document files and application to different viewers. To activate this feature, set "URLReplace = on" in the configuration file (han.conf) and create a text file called replace.txt. In this file, each line represents a URL replacement to be done by HanWEB when translating the page, with the link to be used for Traditional Chinese viewing first, then a tab and finally the link to be displayed for Simplified Chinese viewing. Directories can be replaced in a similar manner.

The same principle can be applied to different file types, such as Flash animation files (*.swf), PSF files, wav files, ram files, etc because the URL replacement will be performed on all hyperlinks identified by HanWEB. The following is a simple example (the space between the file names shown below is a single tab):

```
/name_tc.jpg /name_sc.jpg  
/test/img_b5/ /test/img_gb/  
www.kanhan.com/trad/ www.kanhan.com/simp/  
www.kanhan.com/tc/\*.swf www.kanhan.com/sc/\*.swf  
http://www.kanhan.com/tc.gif http://www.kanhan.com/sc.gif  
http://www.kanhan.com/tc/\*.pdf http://www.kanhan.com/sc/\*.pdf
```

Note: The URL replacement feature does not work with all links expressed in JavaScript. It is because HanWEB is not always able to determine whether a JavaScript string is a link or part of a link. However, developers can tell HanWEB which arguments of which JavaScript functions are strings representing hyperlinks, and HanWEB will process them accordingly. See the next section, JavaScript, for more details.

JavaScript

HanWEB Publishing Server can translate JavaScript correctly in most case. However, some string literals that are supposed to be interpreted as URLs might not be detected by HanWEB. The use of `document.write` and `document.writeln` methods is a common source of problem. For example:

```
document.write("<a href=" + var1 + ">" + var2 + "</a>");
```

The current version of HanWEB assumes what follows the first quotation mark after the equal sign and before the next quotation mark as a string literal, but in fact it is a variable in this case. To avoid this problem, the value inside the brackets can be concatenated on a line prior to the `document.write` statement:

```
var temp = "<a href=" + var1 + ">" + var2 + "</a>";  
document.write(temp);
```

Also, developers should avoid using `document.write` or `document.writeln` in external JavaScript files to refer to other external style sheets. For example, the following code in an external JavaScript file will not have any effect as HanWEB processes a web page and its related external files separately in real-time:

```
document.write("<link href='style.css' rel='stylesheet' type='text/css'>");
```

More on URLs in JavaScript String Literals

All occurrences of "http://" and "https://" in JavaScript string literals are interpreted as the beginning of URLs. For example:

```
var str1 = "http://www.kanhan.com/";           // a link
var str2 = "Link http://www.kanhan.com/ here."; // also a link
```

If a link must not be processed by HanWEB, a simple workaround is to break up the string literal so that "http" or "https" are not immediately followed by "://". For example:

```
var str = "http" + "://" + "www.kanhan.com/"; // not detected
```

However, in some cases HanWEB automatically treats a string literal as a link if they appear in assignment statements where the l-value is `.src`, `.href`, `.action` or `.location`. For example, assuming that `obj` is any object:

```
obj.src = "foo";           // a link
obj.href = "foo";         // a link
obj.action = "foo";       // a link
obj.location = "foo";     // a link

obj.src = var1;           // not detected
obj.temp = "foo.html";   // not detected
```

A string literal that appears as the first argument of the method `window.open()` will also be treated as a link:

```
window.open("file.asp?abc=def"); // a link
window.open(var1);               // not detected
```

Forms

HanWEB is designed to handle information submitted by forms intelligently. The current version of HanWEB translates this information between Traditional Chinese (Big5) and Simplified Chinese (GB2312) seamlessly and automatically. When a user is viewing a Simplified Chinese page translated into Traditional Chinese, he/she can enter information in a form on this page with any Traditional Chinese input method. When the form is being submitted, HanWEB will convert the information into Simplified Chinese so that it can be processed by the original web server. Similarly, Simplified Chinese users need not to worry about using Traditional Chinese search engines, etc., as HanWEB will convert their input into Traditional Chinese before submitting their queries.

However, due to additional features of some web browsers and their different behaviour on different platforms, the translation of information entered into forms might fail in some cases. These additional features include the browsers' own translation mechanism, the operating system's ability to recognize "foreign" characters, etc. The following table summarizes the outcome of a few common situations:

Operating System	Translated Page	Original Page	Form submission
Big5	Big5	GB2312	no problem
Big5	GB2312	Big5	may have problem
GB2312	GB2312	Big5	no problem
GB2312	Big5	GB2312	may have problem

Table: Form submission via HanWEB

JavaScript and Form Submission

Developers should avoid using JavaScript to replace the original form submission routine of the browser (e.g. by concatenating a new URL from the values of form objects). It is because the values of JavaScript variables are directly placed in the URL, thus creating ambiguity and might lead to unpredictable results. However, calling the *submit* method is fine as it simply invokes the form submission routine of the browser.

On the server side, developers should make sure that scripts do not respond to a form submission with an HTML page containing a META tag that redirects the browser to an URL containing the original form data. It is because HanWEB will translate the encoded characters in this URL again and the final value of the data will be incorrect. To redirect the browser, the server-side script should respond with an appropriate HTTP redirection header.

Form Elements

[Applies to text-to-image translation only]

Except selection lists*, text in form elements such as buttons and text area cannot be displayed after text-to-image translation. They can be displayed only if the browser supports the translated language. For example, if an English browser without any language support is viewing a page in Traditional Chinese, all the text on that page will be translated except those in form elements. On the other hand, if the same browser has support for Simplified Chinese and the Traditional-to-Simplified Chinese translation option is chosen, then everything including the controls will be displayed properly.

* In text-to-image translation, items in a selection lists are displayed in a separate window.

Line Wrap

[Applies to text-to-image translation only]

All browsers automatically wrap text that goes beyond the window width onto the next line. However, HanWEB's text-to-image translation will convert the text into a series of GIF files. These GIF files are not wrapped if they are inside a table, and the increased width of the document will require the viewer to scroll some distance before reaching the other end of the document. Therefore, web developers are recommended to set table cells that contain text to an absolute width in order to force the browser to wrap the GIF files after translation.

For example, the following indicates a typical table without a defined width:

```
<TABLE>
  <TR>
    <TD>
      Some text here.
    </TD>
  </TR>
</TABLE>
```

To set the width of the cell, simply add the width attribute and assign an appropriate absolute value to it. Once the width is set, the browser will be forced to wrap the GIF files onto the next line:

```
<TABLE>
  <TR>
    <TD width="400">
      Some text here.
    </TD>
  </TR>
</TABLE>
```

Alternatively, HanWEB can be set to add one blank space between each character during translation. Turning this option on will change the layout of the page slightly although it will also allow the browser to wrap the GIF files automatically without the need to define fixed-width tables.

Java, Flash, documents, multimedia files and Other External Objects

Java applets, Flash, Shockwave, Word, Excel, PowerPoint, PDF documents, multimedia files etc. are objects external to the browser and are rendered by separate software. Therefore HanWEB does not have access to the text or hyperlinks embedded in these objects and cannot translate them.

However, developers can prepare the Simplified Chinese version of these files and make use of HanWEB's URL replacement (replace.txt) feature to link to the Simplified Chinese version of these files when the page is being translated. For more information, see the section on URL Replacement. Alternatively, developers using recent versions of Flash can place text outside the Flash animation, e.g. in some JavaScript variables or functions, so that it can be translated by HanWEB.

In addition, with the HanWEB Microsoft Office Fanjian Translation Server (installed with Microsoft Office 2000), HanWEB can real time convert traditional Chinese MS Word, Excel and PowerPoint documents to simplified Chinese version. Please kindly note that during traditional Chinese to simplified Chinese translation for MS Excel and PowerPoint documents, no HKSCS characters is supported.

Cookies

HanWEB passes cookies back and forth between browsers and servers transparently in most cases. However, developers are suggested not to set the domain and path parameters programmatically using JavaScript. It is because overriding the default domain and path will prevent HanWEB from passing the correct cookies back to the server.

HTTPS and Other Secure Connections

Starting from version 2.5, HanWEB supports both SSL 2.0 and 3.0. Users of earlier versions of HanWEB should note that all HTTPS and subsequent links bypass translation.

Cascading Style Sheets

[Applies to text-to-image translation only]

Most features in the Cascading Style Sheet Level 2 (CSS2) specification are supported by HanWEB. However, "at-rules" (for example: *@media*, *@page*) and rules containing multiple classes (for example: *.class1.class2 { ... }*) or the following selectors are not supported in text-to-image translation yet:

Type	Pattern
Descendant selectors	E F
Child selectors	E > F
:first-child pseudo-class	E:first-child
Link pseudo-classes	E:link E:visited
Dynamic pseudo-classes	E:active E:hover E:focus
:lang() pseudo-class	E:lang(c)
Adjacent selectors	E + F
Attribute selectors	E[foo] E[foo="warning"] E[foo~="warning"] E[lang]="en"]

More information on CSS, including a free validation service, is available at W3C:

- CSS2 Specification: <http://www.w3.org/TR/REC-CSS2/>
- CSS Validation Service: <http://jigsaw.w3.org/css-validator/>

XMP or BLINK Tags

[Applies to text-to-image translation only]

The rarely used XMP tag and the Netscape-supported BLINK tags do not cause any problem in text-to-text translation. However, they should not be used when text-to-GIF translation is required as the translated "text" (which is made up of GIF files) cannot be displayed correctly by these tags.

Limitations

The following is a summary of contents on web pages that cannot or may not be translated by HanWEB correctly:

- Text on pages with incorrectly defined character set information (see [Character Set](#))
- Text in images (see [Text in Images](#))
- Some links generated by the write and writeln methods of JavaScript (see [JavaScript](#))
- Text or links embedded in Java applets or content created by some Macromedia or similar products (see [Others](#))

In addition, the following elements on web pages cannot be processed during text-to-image translation:

- Text within form objects such as button, text area, etc. (see [Forms](#))
- Text formatted with some advanced CSS style rules (see [Others](#))
- Text enclosed by the XMP or BLINK tags (see [Others](#))