# WordWalkingStick Flat-OPC-to-HTML Conversion

This \*.docx file was generated by OpenXmlUtility.GenerateOfficeDocument(package) which implies that it has no local styles or effects. It depends *entirely* on the styles and settings of the opening Word application. Eric White’s original HtmlConverter (used in [Open XML Power Tools](http://powertools.codeplex.com/) for PowerShell) has been enhanced and is included in the WordWalkingStick utility (a VSTO add-in). This utility will convert a Word 2010 document to HTML adding support for the following:

* Selected Word 2010 Document Formatting
* Selected Word 2010 Document Styles
* Selected Characters
* Content Control “Micro-Formats”

## Selected Word 2010 Document Formatting

|  |  |
| --- | --- |
| Bold | The word **here** should be bold. |
| Italic | The word *here* should be italic. |
| Underline | The word here should be underline. |
| Strikethrough | ~~This entire sentence should be marked strikethrough.~~ |
| Subscript/Superscript | This should be the 1st superscript. And the variable xi should have *i* as a subscript. |
| Small caps | The next word, Microsoft, should be in small caps. |
| Combinations | The word ***here*** should be bold *and* italic.  The word **here** should be underline *and* bold.  The last word in this sentence should have “everything”: ***~~here~~***. |

## Selected Word 2010 Document Styles

|  |  |
| --- | --- |
| Block Text | This Paragraph Style translates into the blockquote element:  The W3C recommendation states that web page authors should not type quotation marks in the text when they’re using blockquote—  To support the cite attribute of blockquote see the Content Control “Micro-Formats” below. |
| HTML Cite | This Character Style is translated into the cite element:  The book One is about the second Arabic numeral.  The cite element is also used in the Content Control “Micro-Formats” (see below). |
| HTML Code | This Character Style is translated into the code element. |
| HTML Definition | This Character Style is translated into the dfn (“[defining instance](http://reference.sitepoint.com/html/dfn)”) element:  The word one represents the famous numeral of unity. |
| HTML Preformatted | This Paragraph Style is translated into the pre (“[preformatted](http://reference.sitepoint.com/html/pre)”) element:  We can indent words with spaces—  one  two  three |
| HTML Sample | This Character Style is translated into the samp (“[sample](http://reference.sitepoint.com/html/samp)”) element.  When counting, we can use words like: one, two or three. |
| ~~HTML Typewriter~~ | The assumption here is that this Word style maps to the tt (“[teletype](http://reference.sitepoint.com/html/tt)”) element. It is recommended not to support this element so the Word style will be ignored as well. |
| HTML Variable | This Character Style is translated into the var (“[variable](http://reference.sitepoint.com/html/var)”) element. According to reference.sitepoint.com:  The var element is used to indicate that the text is a variable and shouldn’t be taken literally. Instead, it’s a placeholder where the contents should be replaced with your own value.  It follows that the var style can be used inside the code style:  We have one, <your number>, three in the sequence. |
| List Bullet | This style is translated into an unordered list (ul):   * One * Two * Three |
| List Paragraph | This style is translated into an ordered list (ol):   1. One 2. Two 3. Three |
| Quote | This Character Style is translated into the q (“[quote](http://reference.sitepoint.com/html/q)”) element:  When counting, he said the words, “One, two, three…” |

### Warning: Character Styles Clash with Hyperlinks

In Microsoft Word, a Character Style like “HTML Code” will clash with the “Hyperlink” Character Style when text marked as HTML Code has a hyperlink assigned to it. The resultant Open XML Word Processing Markup Language might look like this:

<w:hyperlink>  
 <w:r w:rsidRPr="002B526C">  
 <w:rPr>  
 <w:rStyle w:val="Hyperlink" />  
 <w:rFonts w:ascii="Consolas"  
 w:hAnsi="Consolas" w:cs="Consolas" />  
 <w:noProof />  
 </w:rPr>  
 <w:t>samp</w:t>  
 </w:r>  
</w:hyperlink>

References to the font Consolas are the only clues that this Hyperlink style was once the HTML Code style.

## Selected Characters

|  |  |
| --- | --- |
| Line Break (w:br or w:cr) | This sentence should break here and then end. |
| No-break hyphen. | This sentence contains a word in italics, *censor‑ious* that should not break at the *r* and *i*. |
| No-break space. |  |

## Content Control “Micro-Formats”

Modern word processing file formats need a standard way to store metadata. The research of [Peter Sefton](http://academic.research.microsoft.com/Author/3445875.aspx), namely his work, “[Embedding metadata and other semantics in word processing documents](http://academic.research.microsoft.com/Paper/4937165.aspx),” details this issue. After [litigation in 2010](http://kintespace.com/rasxlog/?p=2236), Office Word 2010 has only *one* (legal) way to store metadata through the use of [Content Controls](http://msdn.microsoft.com/en-us/library/bb157891.aspx). Moreover, Office Word 2010 effectively stands alone, providing metadata entry *and* a robust API for access and manipulation.

The WordWalkingStick utility supports “[micro-formats](http://microformats.org/about)” that transform Content Controls into HTML. The following table summarizes:

|  |  |
| --- | --- |
| Acronym | **VSTO** |
| Amazon.com Product Image | **Amazon Product**  **ASIN:** 0735627045  **CSS:** margin:16px;  **Image Path:** http://kintespace.com/bitmaps/clr\_via\_csharp\_book.jpg |
| Image | **Web Image**  **Image Path:** http://farm4.static.flickr.com/3388/4627255732\_26e5fed444\_m.jpg  **Image Alternative Text:** New Books in the Labor Camp  **CSS:** margin:16px;  **Image Link:** http://www.flickr.com/photos/wilhite/4627255732/ |
| CSS Block | The next block of text is a CSS Block:  This is a block of text (*“rich”* text) in a Content Control that should export as ‘clean’ XHTML inside of a div element with a CSS class attribute of class="Note".  This comes after the special block of text. |