

## **Review of OOXML Accessibility Guidelines**

The following is a critique of the January 8, 2008 draft of the document “Accessibility Guidelines for Ecma-376 Office Open XML (DIS29500)”, published online by Microsoft at <http://www.openxmldeveloper.org/articles/OpenXMLAccessibilityGuidelines.aspx> (henceforth referred to as “the Guidelines document”). In particular, the Guidelines document is considered for potential inclusion as an appendix to the proposed Office Open XML (OOXML) document standard and in light of Canadian criticisms of OOXML with respect to accessibility.

### **Purpose and Intended Audience of the Guidelines Document**

There seems to be a great deal of misunderstanding regarding the purpose and intended audience of the Accessibility Guidelines Document. The document as it is written is not for Assistive Technology (AT) developers. AT developers have no choice but to process the entire proposed standard as, unlike other tool developers, assistive technologies must be prepared to process and interoperate with any data type, markup, format, interaction or media contained in the proposed standard. For an AT developer there are no optional parts of the standard as the AT must be able to process whatever anyone may create using the document format.

The Guidelines document as it is written is ostensibly for authors who wish to create accessible documents and for authoring tool developers who wish to create authoring tools that support the creation of accessible documents.

### **Completeness of the Guidelines Document**

As a guideline for authors and authoring tool developers the document is woefully incomplete. The guidelines do not address the most common accessibility challenges nor the more complex accessibility challenges; the majority of these are completely absent from the document. Among the glaring omissions are:

- 1) How to provide access to any type of multi-media, including access to video or audio, and the need for captions of audio (both speech and non-speech) and the description of visual information in videos, and how this can be synchronized with the video through markup such as SMIL;
- 2) How to make embedded animations of any kind accessible;
- 3) How to author accessible hyperlinks;
- 4) Providing access to graphs and charts;
- 5) How to provide access to embedded scripts, applets and macros; and
- 6) How to create image maps (or image map equivalents) accessibly.

As a general, very serious omission, the document does not address the importance of semantics or associating consistent meaningful labels to document components and functions (beyond alt-text, which does not enforce consistent semantic labels because its values are unrestricted). This is fundamental to accessibility.

## **Structure vs. Styling**

In viewing any document, sighted users glean a great deal of information about the document before reading it. We can determine the type of document it is, the function of the document (e.g., a form, a table, a presentation), the length of the document, how many sections it has, and where the important parts of the document are. Much of this is communicated through visual metaphors and visually based styling that communicates the structure of the document visually. Consider a document with all formatting removed, including layout. Now consider viewing this document through a window that shows only one word at a time. Imagine viewing a complex and critically important government document or high-stakes test in this way. This is how individuals who cannot rely on vision would perceive a document. Document structure is essential for understanding, processing and responding to a document. For someone who cannot rely on vision this structure is even more important, as the structure is also used as a method of navigating through the document, to achieve the equivalent of glancing at or scanning a document and to find a specific location in a document.

Structure is not the equivalent of styling, although visual styling conventions are frequently used to communicate the structure of a document. Styling conventions are not consistent across documents and the names of styles applied to specific structural elements are completely inconsistent and/or arbitrary. It is therefore completely inadequate to rely on styling labels or style configuration to communicate structure. The Guidelines document conflates style with structure, offering styling as a replacement for structure. (For example, this is the case for Titles, Headings, Lists, Headers and Footers.)

The author should be guided to provide consistent labels for structural elements such that it is clear, for example, that an element is a level 3 header or that another element is a chapter title. This is essential for producing useful Daisy XML documents as well. A convert-to-Daisy utility is useless if the output has no comprehensible structure. A consistently labeled comprehensible structure is made possible by authoring tools that guide the author in providing the structural labels. Authors and authoring tools cannot create a comprehensible structure if there are no consistent structural labels provided by the proposed standard.

There are many assistive technologies that are dependent on consistent, semantically explicit structural information. These include audio style sheets and screen readers that convert structure to either spoken labels or changes in pitch, rate or voice, and Braille markup tools that codify structural information into tactile output. These will not work if there are no consistent structural labels.

## **UI Automation not yet usable**

The guidelines make an unwarranted assumption about UI Automation in the “Programmatic Access” section. UI Automation has yet to be supported by the major vendors of assistive technologies to date. Merely claiming something is an accessibility

API and declaring one that has actually been proven to support interoperability are different. Referring to a non-normative document is fine, but the document referenced, under Programmatic Access, should cite APIs that have been proven to work with assistive technologies.

### **Caption support inadequate**

First, the word “caption” will be used here in the same sense in which it was used in the Guidelines document, although this is narrow and confuses things with respect to multimedia captions discussed above; this narrower sense really should be something like “long description” for accessibility purposes. Such descriptions are essential to provide alternative ways for a person with a visual disability to understand the contents of figures and tables via screen readers. However, a caption is not the same as summary, which is a distinct concept with respect to accessibility; both must be supported. Yet, in OOXML no distinction is offered between summaries and captions. Further, table captions are inadequately supported, as there appears to be no way to specify in markup the table to which a caption refers.

The section of the Guidelines document that covers captions misses these points entirely. Instead, it discusses the “autoCaption” and “autoCaptions” tags, which have nothing to do with linking captions to the object or the label. Further, use of these mean that captions will not be explicitly stored in markup with the tables to which they belong (instead being automatically calculated on-the-fly). This is almost exactly opposite the accessibility goal, as it makes the captioning more opaque to assistive technology.

### **Alternative Text conflated with Descriptions**

As described in the previous section on captions, the Guidelines and proposed standard sometimes combine the concepts of alternative text with text descriptions, and further do not distinguish between long and short descriptions. These distinctions are required to be supported by accepted accessibility guidelines, such as WCAG, and are important for accessibility as each may be needed in various contexts, or they all three may be needed jointly.

As a further example of the confusion and inconsistency here, in the case of Frames discussed in the Guidelines document, it is seen that each frame can have a “title” and a “longDesc”. Note that in this case, the “title” field is doing double duty as alternative text. These sorts of inconsistencies not only make it difficult to design modular or portable AT, but they also gloss over important semantic distinctions for labeling different sorts of content.

### **Page Numbers**

While the discussion and examples in the Guidelines document related to page numbering are appreciated, it appears that there still is not a way for assistive technology to figure out the real page number of any given page. The point here is that AT needs to

know the bounds of a printed page so users (e.g., of “talking book” software) can synchronize in real time with someone, such as a teacher, who is referring to portions of the printed book. To do this, the actual number is needed.

### **Better Examples Needed**

While the few XML examples that are given in the Guidelines document are appreciated and much needed, there is still need for more in-depth and complex examples of the sort that might be found in real-world situations. This is particularly true for Lists, Tables, Frame relationships, Forms, and specifying logical navigation order through form fields.

#### Lists

For numbered lists, the example in the Guidelines document does not show what is in the external “numbering.xml” file. This is needed to fully understand the example. Further, the following sentence (from the bottom of pg. 17) is very unclear: “The start element in the abstract numbering definition handles the identifier.” An example “numbering.xml” is needed for clarification of this as well. Finally, since the list level (and list point numbering) are part of the paragraph properties (and since paragraph style could be doing double duty here, representing both style and structure), the example should include list items that span multiple, different-style paragraphs, with sub-lists within at least one of the interior paragraphs.

#### Frames

As frames are often used to group structurally related items within a document, it is important to give information as to how each frame fits into the document context and relates to other frames (including the main text frame) in terms of navigation order and other semantic relationships. There does not seem to be a way to specify relationships (e.g. semantic, navigation order, layout, nestings, etc.) among frames in OOXML. An example for specifying these sorts of complicated relationships, if they are possible, would be very helpful.

For example, the following form is extracted from the U.S. Form 1040 Tax Booklet (2006). If the differently shaded regions (grey and white) were to each constitute a different frame in the document, such that in this example there are 6 frames total, how would the relationships among these frames be specified? Specifically, how would the navigation order through these frames be specified along with the semantic relationship that each “grey-white pair” has with each other (each grey region corresponds with one white region). It is unclear if or how OOXML could represent such a structure and make it accessible. (More complicated examples can easily be imagined as well.)

<p><b>Part 1</b></p> <p><b>All Filers Using Worksheet A</b></p>	<p>1. Enter your earned income from Step 5 on page 48. <span style="float: right; border: 1px solid black; padding: 2px 10px;">1</span></p> <hr/> <p>2. Look up the amount on line 1 above in the EIC Table on pages 53–59 to find the credit. Be sure you use the correct column for your filing status and the number of children you have. Enter the credit here. <span style="float: right; border: 1px solid black; padding: 2px 10px;">2</span></p> <p style="text-align: center;">If line 2 is zero,  You cannot take the credit. Enter "No" on the dotted line next to line 66a.</p> <hr/> <p>3. Enter the amount from Form 1040, line 38. <span style="float: right; border: 1px solid black; padding: 2px 10px;">3</span></p> <hr/> <p>4. Are the amounts on lines 3 and 1 the same?</p> <p><input type="checkbox"/> <b>Yes.</b> Skip line 5; enter the amount from line 2 on line 6.</p> <p><input type="checkbox"/> <b>No.</b> Go to line 5.</p>
<p><b>Part 2</b></p> <p><b>Filers Who Answered "No" on Line 4</b></p>	<p>5. If you have:</p> <ul style="list-style-type: none"> <li>• No qualifying children, is the amount on line 3 less than \$6,750 (\$8,750 if married filing jointly)?</li> <li>• 1 or more qualifying children, is the amount on line 3 less than \$14,850 (\$16,850 if married filing jointly)?</li> </ul> <p><input type="checkbox"/> <b>Yes.</b> Leave line 5 blank; enter the amount from line 2 on line 6.</p> <p><input type="checkbox"/> <b>No.</b> Look up the amount on line 3 in the EIC Table on pages 53–59 to find the credit. Be sure you use the correct column for your filing status and the number of children you have. Enter the credit here. Look at the amounts on lines 5 and 2. Then, enter the <b>smaller</b> amount on line 6. <span style="float: right; border: 1px solid black; padding: 2px 10px;">5</span></p>
<p><b>Part 3</b></p> <p><b>Your Earned Income Credit</b></p>	<p>6. This is your earned income credit. <span style="float: right; border: 1px solid black; padding: 2px 10px;">6</span></p> <p style="text-align: right; font-size: small;">Enter this amount on Form 1040, line 66a.</p> <p><b>Reminder—</b></p> <p>✓ If you have a qualifying child, complete and attach Schedule EIC. </p> <hr/> <p style="text-align: center;"> <i>If your EIC for a year after 1996 was reduced or disallowed, see page 49 to find out if you must file Form 8862 to take the credit for 2006.</i></p>

**Figure 1: Example Frames from U.S. Form 1040 Tax Booklet (2006)**

Tables

In the example XML for a table on page 14 of the Guidelines document, for the cell with “x1” in it, it is unspecified in the markup that “HeaderA” is a *column* header and “HeaderC” is a *row* header. Note that in this simple example, the AT could “figure it out”, but in a more complicated table structure with multiple heading levels for both the rows and the columns, it is unclear whether or not the cell should point to all levels of its headers or if they should be “chained”. For example, consider the following table:

**Table 1: Example of multiple table heading levels**

		Column Heading 1		Column Heading 2	
		Col Subheading 1a	Col Subheading 1b	Col Subheading 2a	Col Subheading 2b
Row Heading 1	Row Subheading 1a	x1	x2	x3	x4
	Row Subheading 1b	x5	x6	x7	x8
Row Heading 2	Row Subheading 2a	x9	x10	x11	x12
	Row Subheading 2b	x13	x14	x15	x16

There are at least two ways this might be represented in OOXML.

Option 1:

```

<tbl>
  <tr>
    <trPr><tblHeader/></trPr>
    <tc/>
    <tc/>
    <tc id="CH1"><p><r><t>"Column Heading 1"</t></r></p></tc>
    <tc id="CH2"><p><r><t>"Column Heading 2"</t></r></p></tc>
  </tr>
  <tr>
    <trPr><tblHeader/></trPr>
    <tc/>
    <tc/>
    <tc id="CSH1a"><tcPr><headers><header val="CH1"/></headers></tcPr>
    <p><r><t>"Col Subheading 1a"</t></r></p></tc>
    <tc id="CSH1b"><tcPr><headers><header val="CH1"/></headers></tcPr>
    <p><r><t>"Col Subheading 1b"</t></r></p></tc>
    <tc id="CSH2a"><tcPr><headers><header val="CH2"/></headers></tcPr>
    <p><r><t>"Col Subheading 2a"</t></r></p></tc>
    <tc id="CSH2b"><tcPr><headers><header val="CH2"/></headers></tcPr>
    <p><r><t>"Col Subheading 2b"</t></r></p></tc>
  </tr>
  <tr>
    <tc id="RH1"><p><r><t>"Row Heading 1"</t></r></p></tc>
    <tc id="RSH1a"><tcPr><headers><header val="RH1"/></headers></tcPr>
    <p><r><t>"Row Subheading 1a"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH1a"/><header val="RSH1a"/></headers>
    </tcPr><p><r><t>"x1"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH1b"/><header val="RSH1a"/></headers>
    </tcPr><p><r><t>"x2"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH2a"/><header val="RSH1a"/></headers>
    </tcPr><p><r><t>"x3"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH2b"/><header val="RSH1a"/></headers>
    </tcPr><p><r><t>"x4"</t></r></p></tc>
  </tr>
  ...
</tbl>

```

Option 2:

```

<tbl>
  <tr>
    <trPr><tblHeader/></trPr>
    <tc/>
    <tc/>
    <tc id="CH1"><p><r><t>"Column Heading 1"</t></r></p></tc>
    <tc id="CH2"><p><r><t>"Column Heading 2"</t></r></p></tc>
  </tr>
  <tr>
    <trPr><tblHeader/></trPr>
    <tc/>
    <tc/>
    <tc id="CSH1a"><p><r><t>"Col Subheading 1a"</t></r></p></tc>
    <tc id="CSH1b"><p><r><t>"Col Subheading 1b"</t></r></p></tc>
    <tc id="CSH2a"><p><r><t>"Col Subheading 2a"</t></r></p></tc>
    <tc id="CSH2b"><p><r><t>"Col Subheading 2b"</t></r></p></tc>
  </tr>
  <tr>
    <tc id="RH1"><p><r><t>"Row Heading 1"</t></r></p></tc>
    <tc id="RSH1a"><p><r><t>"Row Subheading 1a"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH1a"/><header val="CH1"><header
      val="RSH1a"/><header val="RH1"></headers></tcPr>
      <p><r><t>"x1"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH1b"/><header val="CH1"><header
      val="RSH1a"/><header val="RH1"></headers></tcPr>
      <p><r><t>"x2"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH2a"/><header val="CH2"><header
      val="RSH1a"/><header val="RH1"></headers></tcPr>
      <p><r><t>"x3"</t></r></p></tc>
    <tc><tcPr><headers><header val="CSH2b"/><header val="CH2"><header
      val="RSH1a"/><header val="RH1"></headers></tcPr>
      <p><r><t>"x4"</t></r></p></tc>
  </tr>

```

Because nested headers were not explicitly considered in the examples, many assumptions had to be made to generate these two options.

This sort of table structure is common. For example, consider the following tax table (Figure 2) extracted from the U.S. Form 1040 Tax Booklet (2006). This particular table spans 11 pages of the booklet, 3 columns per page. For such a large table, it is imperative that assistive technology be able to provide “orientation” to users with visual disabilities.

If line 43 (taxable income) is —		And you are —			
At least	But less than	Single	Married filing jointly *	Married filing sepa- rately	Head of a house- hold
		Your tax is —			
2,700	2,725	271	271	271	271
2,725	2,750	274	274	274	274
2,750	2,775	276	276	276	276
2,775	2,800	279	279	279	279
2,800	2,825	281	281	281	281
2,825	2,850	284	284	284	284
2,850	2,875	286	286	286	286
2,875	2,900	289	289	289	289
2,900	2,925	291	291	291	291
2,925	2,950	294	294	294	294
2,950	2,975	296	296	296	296
2,975	3,000	299	299	299	299
<b>3,000</b>					
3,000	3,050	303	303	303	303
3,050	3,100	308	308	308	308
3,100	3,150	313	313	313	313
3,150	3,200	318	318	318	318
3,200	3,250	323	323	323	323
3,250	3,300	328	328	328	328
3,300	3,350	333	333	333	333
3,350	3,400	338	338	338	338

Figure 2: Example Table from page 67 of U.S. Form 1040 Tax Booklet (2006)

Figure 3 shows an even more complicated table from the same booklet. Note that in this case, the sub-heading one comes to first when scanning up a column from a particular cell spans three columns, each of which has a subheading above the spanning subheading first encountered. It is unclear whether and/or how such a table might be represented in OOXML (but such a table seems to preclude the use of Option 1 above). Examples such as these are sorely needed given their use in important government documents.

**2006 Earned Income Credit (EIC) Table – Continued**

If the amount you are looking up from the worksheet is –		And your filing status is –					
		Single, head of household, or qualifying widow(er) and you have –			Married filing jointly and you have –		
		No children	One child	Two children	No children	One child	Two children
At least	But less than	Your credit is –			Your credit is –		
5,000	5,050	384	1,709	2,010	384	1,709	2,010
5,050	5,100	388	1,726	2,030	388	1,726	2,030
5,100	5,150	392	1,743	2,050	392	1,743	2,050
5,150	5,200	396	1,760	2,070	396	1,760	2,070
5,200	5,250	400	1,777	2,090	400	1,777	2,090
5,250	5,300	404	1,794	2,110	404	1,794	2,110
5,300	5,350	407	1,811	2,130	407	1,811	2,130
5,350	5,400	412	1,828	2,150	412	1,828	2,150
5,400	5,450	412	1,845	2,170	412	1,845	2,170
5,450	5,500	412	1,862	2,190	412	1,862	2,190
5,500	5,550	412	1,879	2,210	412	1,879	2,210
5,550	5,600	412	1,896	2,230	412	1,896	2,230
5,600	5,650	412	1,913	2,250	412	1,913	2,250

**Figure 3: Example Table from page 54 of U.S. Form 1040 Tax Booklet (2006)**

Forms

More complicated form examples are also direly needed. For example, XML could be provided for forms such as those in Figures 1 and 4, which were extracted from the U.S. Form 1040 Tax Booklet (2006). Note such forms include complicated relationships between fields and their labels (and related icons), complicated navigations sequences through the form fields (sometimes depending on the value in entered into the form field), and structural relationships encoded in the vertical and horizontal alignment of the form fields with one another. For example, in Figure 4, on line 1, the text “X \$1000” needs to be associated with the blank line preceding it, as does the text “Number of qualifying children:”, and all of these need to be associated with the box labeled **1** to the right of them.

1. Number of qualifying children: \_\_\_\_\_ × \$1,000.  
Enter the result. 1

---

2. Enter the amount from Form 1040, line 46. 2

---

3. Add the amounts from Form 1040:

Line 47 \_\_\_\_\_

Line 48 + \_\_\_\_\_

Line 49 + \_\_\_\_\_

Line 50 + \_\_\_\_\_


Line 51 + \_\_\_\_\_

Line 52\* + \_\_\_\_\_ Enter the total. 3

\*Include only the amount, if any, from Form 5695, line 12.

---

4. Are the amounts on lines 2 and 3 the same?

**Yes.** 

You cannot take this credit because there is no tax to reduce. However, you may be able to take the **additional child tax credit**. See the **TIP** below.

**No.** Subtract line 3 from line 2. 4

---

5. Is the amount on line 1 more than the amount on line 4?

**Yes.** Enter the amount from line 4. Also, you may be able to take the **additional child tax credit**. See the **TIP** below.


**No.** Enter the amount from line 1. } **This is your child tax credit.** 5

Enter this amount on Form 1040, line 53.

---

**TIP** You may be able to take the **additional child tax credit** on Form 1040, line 68, if you answered "Yes" on line 4 or line 5 above.

- First, complete your Form 1040 through line 67.
- Then, use Form 8812 to figure any additional child tax credit.



1040

**Figure 4: Example form fields from page 43 of U.S. Form 1040 Tax Booklet (2006)**

While the Guidelines do show how to specify tabindex values, one complication that is not addressed is how the document order is produced. For example, is document order merely determined by the order in which elements are added to the document? This is particularly important in a presentation slide. We would like to see informative text as to how an office application would help the author in assigning the tab sequence. The author should not be required to remember tab numbers

### **Translation to other formats might be precluded**

Regarding the translation of OOXML into other formats, such as the DAISY book format, described at the end of the Guidelines document, there is a danger that the conversion will have unusable output because (1) there is no support to label structure consistently and thereby provide the structural markup critical for Daisy (as described above), and (2) OOXML allows characters forbidden by the XML 1.0 standard, such as from the C0 control character range, to be encoded in a non-standard way in OOXML strings (in the ST\_xString type). Guidance needs to be given as to how such special characters will and should be addressed in converters such as the aforementioned DAISY converter.

Further, in light of the response to CA-0019, while the modifications by ECMA have removed the specificity of a particular browser, the proposed wordings do not state that accessibility features of the markup must be preserved in the conversion process if similar accessibility features are supported in the target solution. This leaves open a big hole. Examples of things that should be preserved include structural markup, alternative text, table captions, etc. The details of meeting this could be spelled out in the Guidelines document.

### **Conclusion**

The Guidelines document is offered as a response to Canadian comments CA- 0075, which states:

CA Comment: It is recommended that this standard interoperate with accessibility-vetted approaches, such as XLinks, XForms and SMIL

CA Proposed resolution: Provide an “Informative” annex which would provide information on accessibility features and hooks in the development of assistive technologies. Canada recommends a full review of the standard for Accessibility for future versions of this standard.

as well as to CA-0078 which states:

CA Comment: An accessible standard must make it easy to create a compatible assistive technology (this is referred to as access-system friendliness). This requires a standard that is easy to follow, that is consistent (without a large number of exceptions), that uses other open standards wherever possible, that harmonizes with other standards in the domain, and that makes necessary semantics programmatically available to the assistive technology. It is not clear how this standard meets the above criteria.

CA proposed resolution: Provide an “Informative” annex which would provide information on accessibility features and hooks in the development of assistive technologies. Canada recommends a full review of the standard for Accessibility for future versions of this standard.

The Guidelines document addresses neither the Canadian comments nor the proposed resolutions. The document does not provide adequate information on accessibility features or hooks for assistive technology, nor does it provide a full accessibility review of the standard. Indeed, as detailed above, there are several accessibility issues within and detracting from the proposed guidelines themselves! And at a more general level, and thus even more importantly for accessibility, the substance of comments CA-0075 and CA-0078 are completely ignored in the Guidelines document. The document does not make the standard easy to follow or consistent. The Guideline document does not address the fact that the standard does not use other open standards wherever possible, does not harmonize with other standards in the domain and does not make necessary semantics programmatically available to assistive technologies. Without addressing these requirements the proposed standard will make the task of an assistive technology developer much more complex if not impossible. This strains an already inadequate support system for people with disabilities and reduces the overall accessibility of documents. This has huge repercussions, as documents form the basis of essential services and functions in education, government, commerce, health and employment; services and functions people with disabilities require and have the right to access equally.