Gary Griswold
513-508-6127
gary@shortsands.com
http://shortsands.com
March 11, 2020

# USX & HTML

## An Implementation of USX in HTML

The USFM (Unified Standard Format Markers) is a standard for formatting Bible text using format tags that are embedded in the Bible text. These embedded format tags are preceded by a "\" (back-slash) and separated from the actual text by a single space before and after each format code. I believe this standard was developed by SIL (Wycliffe) translators, but has been adopted as a standard by the United Bible Society and many others. The technique of using "\" delineated format codes originates from early computer science work on formatted text, such as the LaTex system developed by Donald Knuth at Stanford. Documentation on the USFM format can be found at: https://ubsicap.github.io/usfm/

USX (Unified Standard XML) was developed as a modern variant, which uses the identical format codes, except the format codes are embedded in XML in a style attribute. The Digital Bible Library has adopted this second format as their storage format for Bible content, but I have interacted with some Bible copyright owners, who still use USFM, and not USX. Fortunately, the SIL program ParaText is capable of translating content between these two formats. Documentation on the USX format can be found at:

https://app.thedigitalbiblelibrary.org/static/docs/usx/index.html

The following is an example USX content for Psalms 1. Notice, that the USFM style codes become a "style" attribute in this XML format.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<usx version="2.0">
  <chapter number="1" style="c" />
  <para style="ms">Book 1</para>
  <para style="mr">(Psalms 1 – 41)</para>
  <para style="q1">
    <verse number="1" style="v" />Great blessings belong to those</para>
  <para style="q2">who don’t listen to evil advice,</para>
  <para style="q1">who don’t walk on Sinners Road</para>
  <para style="q2">or join in making fun of what is good.</para>
  <para style="q1">
    <verse number="2" style="v" />Instead, they love the LORD’s teachings</para>
  <para style="q2">and think about them day and night.</para>
  <para style="q1">
    <verse number="3" style="v" />So they grow strong,</para>
  <para style="q2">like a tree planted by a stream—</para>
  <para style="q1">a tree that produces fruit when it should</para>
  <para style="q2">and has leaves that never fall.</para>
  <para style="q1">Everything they do is successful.</para>
  <para style="b" />
  <para style="q1">
    <verse number="4" style="v" />But the wicked are not like that.</para>
  <para style="q2">They are like chaff that the wind blows away.</para>
  <para style="q1">
    <verse number="5" style="v" />God’s judgement will find the wicked guilty.</para>
  <para style="q2">He will not give sinners a place among those he considers right.<note caller="+"
style="f"><char style="fr" closed="false">1:5 </char><char style="ft" closed="false">Or “The wicked will not
be allowed to sit as judges nor sinners in the meeting of good people.”</char></note></para>
  <para style="q1">
    <verse number="6" style="v" />The LORD makes sure his people go the right way,</para>
  <para style="q2">but the wicked follow the way that leads to ruin.</para>
  <para style="b" />
</usx>
```

The DBL provides a CSS stylesheet that shows how each of these styles is to be represented in a Web Browser. I have not been able to locate the original USX CSS files that I used as a source, but I offer the following tid-bit from my own code, which shows the CSS for the first line of a poetry verse, i.e. q1.

```
/* line 506, Poetry First Line */
p.q, p.q1 {
   text-align: start;
   text-indent: -2.5rem; /* std -1.0in */
   margin-left: 3.0rem; /* std 1.25in */
   margin-right: 0;
   margin-top: 0; /* non-std */
   margin-bottom: 0; /* non-std */
}
```

One way to develop a Bible App would be to develop a rendering engine that can parse USX and display the formatted text, but the USX is so close to HTML it makes sense to translate the USX to HTML so that an existing Web rendering engine can be used to present the text.  For example, the following is an example of the HTML that can be produced using the above USX.

```
<section id="PSA:1">
 <p class="c">1</p>
 <p class="ms">Book 1</p>
 <p class="mr">(Psalms 1 – 41)</p>
 <p class="q1">
  <span id="PSA:1:1" class="v">1&nbsp;</span>Great blessings belong to those</p>
 <p class="q2">who don’t listen to evil advice,</p>
 <p class="q1">who don’t walk on Sinners Road</p>
 <p class="q2">or join in making fun of what is good.</p>
 <p class="q1">
  <span id="PSA:1:2" class="v">2&nbsp;</span>Instead, they love the LORD’s teachings</p>
 <p class="q2">and think about them day and night.</p>
 <p class="q1">
  <span id="PSA:1:3" class="v">3&nbsp;</span>So they grow strong,</p>
 <p class="q2">like a tree planted by a stream—</p>
 <p class="q1">a tree that produces fruit when it should</p>
 <p class="q2">and has leaves that never fall.</p>
 <p class="q1">Everything they do is successful.</p>
 <p class="b" />
 <p class="q1">
  <span id="PSA:1:4" class="v">4&nbsp;</span>But the wicked are not like that.</p>
 <p class="q2">They are like chaff that the wind blows away.</p>
 <p class="q1">
  <span id="PSA:1:5" class="v">5&nbsp;</span>God’s judgement will find the wicked guilty.</p>
 <p class="q2">He will not give sinners a place among those he considers right.<span id="PSA1-1" class="topf" caller="+" onclick="bibleShowNoteClick('PSA1-1');"> ☞ <span class="fr" closed="false" hidden="1:5 "></span><span class="ft" closed="false" style="display:none">Or “The wicked will not be allowed to sit as judges nor sinners in the meeting of good people.”</span></span></p>
 <p class="q1">
  <span id="PSA:1:6" class="v">6&nbsp;</span>The LORD makes sure his people go the right way,</p>
 <p class="q2">but the wicked follow the way that leads to ruin.</p>
 <p class="b" />
</section>
```

Notice that the style attributes in the USX file have all become class attributes in the HTML. This makes it possible to format the text using CSS. For example, in CSS syntax “p.q” means an HTML element whose tag name is “p” and whose class attribute is “q”. Notice that the USX “para” tags in this example, have all been changed to the HTML standard “p” tag, and the “verse” tags have been changed to the standard “span” HTML tags. Notice that the span elements for each verse contain an “id” attribute that identifies the verse, such as id=“PSA:1:1”. These verse markers make it possible to scroll the text to the position of the verse, and make it possible to jump to any verse in the Bible. Notice that the chapter is wrapped in a section element (it could have been a div element), which has an id attribute to identify the chapter. The chapter id makes it possible to scroll to the beginning of a chapter when a web view contains multiple chapter files.

One aspect of USX that seems confusing on first glance is that the verse number is simply an element within the text. It seems like a good idea to have the verse become a div which wraps the content of the entire verse. Unfortunately, this solution will only work well most of the time. At other times, a verse might only be the part of a sentence.

All of the above transformations from USX to HTML are quite straightforward, but the handling of footnotes requires a bit more creativity, because in the web app some kind of interactivity is desirable. Possibly the App will display a footnote marker in the text, which displays the full footnote. Or, possibly the App will display a footnote marker at the bottom of a page or chapter. And, possibly the App will be given the ability to jump to a verse.

What has been done in this example, is to hide the footnote, but display a character icon of a hand pointing, which when clicked will display the footnote. To accomplish this the footnote has been given a unique id of "PSA1-1". This id is similar to a verse id, but intentionally different, because it identifies a footnote and not a verse. The footnote has been given a class="topf", which is not a standard class, but one which identifies to the App the top element of a footnote. That element also has a function to call when the element is clicked. And notice, that the elements below have the attribute "hidden", and style="display:none" which prevents them from being displayed. The function whose name is mentioned will modify those attributes in order to display the footnote.

The transformation of USX to HTML could be done inside an App at the moment of the request. Possibly, it could even be done using XSLT. But there is a testing advantage in performing the transform outside the App, so that automated tests can be written to verify that the transform has correctly converted all chapters from USX to HTML. If a transform is performed only inside an App, then validation would be limited to human review of a limited number of screens.

The code that I have written to generate the HTML, is available in GitHub and available to be freely used given the MIT style license that covers the repository. It is written in javascript node. The code that generates HTML chapter files is embedded within a 2000 line program that prepares chapters, verses, table of contents, and concordance from the USX text, and puts all of these into relational tables in a sqlite3 database, where a single database contains one Bible. The code would require slight modification because of hard-coded directory names, but otherwise could be readily used. It can be found at: https://github.com/garygriswold/SafeBible/tree/master/Publisher where it is part of the "SafeBible" repository. I should move it to a repository of its own, where it could be more easily cloned or downloaded. Also, available in the same repository are a collection of test programs that were used to test the publisher program results. This code can be found at: https://github.com/garygriswold/SafeBible/tree/master/Validation