Implementing Section 508 in XSL FO using RenderX XEP Software

This document serves as a Technical White Paper showcasing the enhanced capabilities of RenderX XEP software for creating Section 508 compliant PDF documents. The document itself was created using RenderX's software and it alone demonstrates much of the capabilities of Section 508 compliance of RenderX's software.

Within the document at various points there will be paragraphs that point out some characteristics of the tagging of some the material. These blocks are formatting red in color and marked as PDF Artifacts. Therefore, they do not appear in the structure of the document nor would they be read out loud in a proper screen reader.

This heading of this section demonstrates the capabilities of RenderX software for applying "actual-text" to allow for a clearer understanding of the content. The "508" and the word "RenderX" are formatted here and throughout this document using the "rx:actual-text" extension. This allows them to be more understandable to someone without vision. This extension will be describe later in the document. If you use a screen reader to read this document out loud you would hear the difference.

This headers and footers of this document are also marked as artifacts. They use the "rx:pdf-structure-tag" of "Artifact" and the "rx:pdf-artifact-type" of "Pagination". This is actually done automatically for you, there is no need to markup the content in the XSL FO as such. All content within "fo:static-content" is marked as such.

General Information

The heading above is marked as an H2 heading using the "rx:pdf-structure-tag" extension to override the default tagging for "fo:block".

RenderX's underlying formatting engine is XEP. This product is based on the industry standard XSL FO and allows for dynamic formatting of content from XML and XSL or XSL FO to PDF format. RenderX added support for tagged PDF several years ago and recently enhanced the tagged PDF abilities in XEP. These enhancements were driven by two primary customers:

1. United States Health and Human Services
2. The State of California Legislative Bureau

The list above is tagged properly as a list. The list item labels have been marked as PDF Artifacts using the "rx:pdf-structure-tag" and as such they are not read or seen in the document structure. You can choose this behavior if you wish, the labels could be marked with "Label" tags and the list item bodies are tagged "LBody". The power of RenderX software is that you have the choice to implement these in a consistent way through the rollmap. We will cover the rollmap later in this document.

Throughout that process, both of these customers constantly evaluated the output of RenderX software against their own requirements and provided feedback into our development cycle. The result is evident in this document. RenderX software is considered the leading software for formatting XML content to Section 508 compliant PDF documents. This is especially true when you combine the need for Section 508 compliance with the ability of RenderX software to dynamically compose and structure the information. Using XEP, you can create a solution for 100% dynamic composition of variable content that is perfectly tagged and Section 508 compliant.

Details of Section 508 Compliance Implementation

Our customers presented us with the challenge of creating one of the first dynamic formatting engines that could deliver completely and properly tagged PDF documents. For that, we implemented the solution in two parts.

A. Requirements that can be implemented within the engine.
B. Requirements outside the XSL FO specification to express or the engine to interpret.

The things we could do within the engine itself were mapping a known XSL FO construct to a known or intended PDF tagging construct. We could also implement an ability to roll-up certain structures to simplify the tagging. This second point is very important. If you examine the PDF tagging of many other vendors output, you will see a very complex, dense tagging scheme that mimics the structure of the XSL FO input exactly. This really violates Section 508 requirements in that the tag density and hierarchy is too deep. Many screen readers cannot handle this and our customers did not want it.

RenderX XEP Rollmap

Within our software, there is a separate, customizable file called a rollmap. This rollmap provides two different functions. First, it allows you to map common XSL FO structures to different PDF tags. Simple examples would be the "fo:root" becomes a "Document" or an "fo:block" becomes a "P". The rollmap allows you to define these for the output.

Second, it also allows you to define structures that are excluded from the tagging. This is a very important feature for compliance. Using the rollmap, you can specify XSL FO structures to exclude from the tagging of the output. These would be common FO structures that really do not contribute to the structure but are in place because of the XSL FO specification. Examples would be "fo:flow" and "fo:static-content" tags. This greatly simplifies the structure of the tagged PDF per the requirements of Section 508.

If you have an ability to examine the structure of the tags in this PDF, it is greatly encouraged. It has a minimal hierarchy and only includes some additional "Span" elements for handling proper pronunciation of terms like "RenderX" and "508".

XSL FO Extensions

For those capabilities required to meet Section 508 requirements that are outside XSL FO or cannot be interpreted through the rollmap, RenderX implemented extensions in XSL FO so that you can produce fully-compliant tagged PDF files.

PDF Structure Tag (rx:pdf-structure-tag)

RenderX implemented an extension attribute "rx:pdf-structure-tag" that is allowed at the lowest level where content is formatted to output. This attribute allows you to spot remap some piece of content to the proper (intended) PDF tag. The most common use is to map an "fo:block" to tagged PDF headings like "H1" to "H6". This is required because there is no direct equivalent in XSL FO, all of these are merely "fo:block" elements. There is no way to truely understand which "fo:block" is an

"H1" or an "H3" or a "P". Now, you can simply mark an "fo:block" with "rx:pdf-structure-tag" and set the value to your intended PDF tag like "H1". This document's headings demonstrate this functionality.

This same extension attribute also allows you to mark content as a PDF Artifact. By using the special value for "rx:pdf-structure-tag" of "Artifact", you can control information you do not wish to be included in the tag structure or reading of a document. This document's red paragraphs are the perfect demonstration using "rx:pdf-structure-tag" to mark content as an Artifact.

Just to emphasize this point, this content is marked as an Artifact. If you exmaine the tag structure for this document you will not see this paragraph nor will a proper screen reader cause the content to be read.

PDF Artifact Type and Subtype

RenderX has implemented two additional attributes that apply only to something classified as a PDF Artifact. These two extensions, "rx:pdf-artifact-type" and "rx:pdf-artifact-subtype" allow you to further classify Artifacts. It should be noted that these are optional according to the PDF Specification.

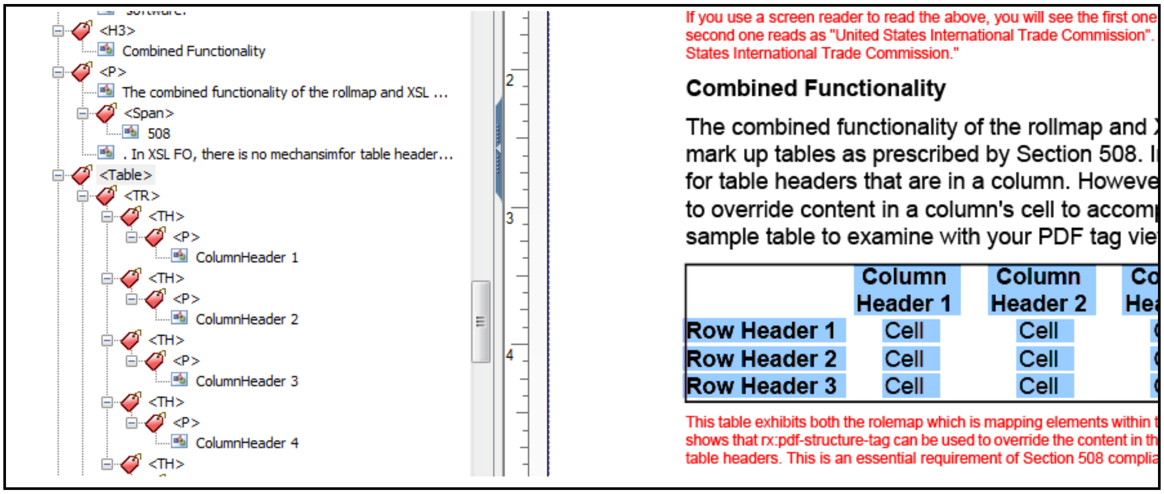
The valid values for the type of an Artifact are "Pagination", "Page" or "Layout". "Pagination" is used for Artifacts that are the direct result of the formatter's pagination of the document and they should (must) not be included in the PDF tag structure or reading. Running and headers and footers are a perfect example of this. "Page" Artifacts are normally used for something on a page like colored boxes and "Layout" Artifacts are used for things like table borders.

If the "rx:pdf-artifact-type" is "Pagination", then you can further classify a subtype. The valid values for "rx:pdf-artifact-subtype" are "Header", "Footer" or "Watermark".

As stated earlier in this document, the running headers and footers of this document showcase these. All of the content is force marked with "rx:pdf-structure-tag of "Artifact" and "rx:pdf-artifact-type" of "Pagination" during output of the tagged PDF.

Alternate Description

Normally used with images, the extension "rx:alt-description" allows you to assign alternate text to something that is not normally read out loud, like an image. The image below has text content applied to "rx:alt-description" so that this text is read when a screen reader encounters the image. You would use this to describe the image itself to the user.



```
<H3>
    Combined Functionality
<P>
    The combined functionality of the rollmap and XSL ...
    <Span>
        508
    . In XSL FO, there is no mechansimfor table header...
<Table>
    <TR>
        <TH>
            <P>
                ColumnHeader 1
        <TH>
            <P>
                ColumnHeader 2
        <TH>
            <P>
                ColumnHeader 3
        <TH>
            <P>
                ColumnHeader 4
        <TH>
```

If you use a screen reader to read the above, you will see the first one
second one reads as "United States International Trade Commission".
States International Trade Commission."

Combined Functionality

The combined functionality of the rollmap and X
mark up tables as prescribed by Section 508. I
for table headers that are in a column. Howeve
to override content in a column's cell to accom
sample table to examine with your PDF tag vie

	Column Header 1	Column Header 2	Co He
Row Header 1	Cell	Cell	C
Row Header 2	Cell	Cell	C
Row Header 3	Cell	Cell	C

This table exhibits both the rolemap which is mapping elements within t
shows that rx:pdf-structure-tag can be used to override the content in th
table headers. This is an essential requirement of Section 508 complia

The image above is marked using rx:alt-description= "An image showing the tagging structure of this document." This content would be read in lieu of the image by screen readers.

Actual Text

The extension "rx:actual-text" allows you to change what is read for some text content that is displayed in the PDF. This is many times used to read numbers in a special way, overriding the behavior of some screen readers from assuming some string of things is a number. Since this can be mis-interpreted, it is best to override what is read to the user. The following shows two ways of formatting some numbers that appear exactly the same in context of the PDF. If you allow a screen reader to read them out loud, you will see the difference.

No actual text: 12345678910

With actual text: 12345678910

The example with no actual text will attempt to read as a number. You would hear twelve billion, three hundred forty five million six hundred seventy eight thousand nine hunderd ten. The example with actual text is formatted with rx:actual-text = "1,2,3,4,5,6,7,8,9,10." and would be read exactly like it is intended ... one, two, three, ... Actual text is used throughout this document within content that is being read to clarify how it should be read, like for the terms Section 508 and RenderX.

Abbreviations

The extension rx:abbreviation allows you to mark content that is an abbreviation with an appropriate text to be read out loud. The following shows how you can use abbreviation.

US H&HS uses RenderX XEP software for Section 508 compliant PDFs.

US H&HS uses RenderX XEP software for Section 508 compliant PDFs.

If you use a screen reader to read the above, you will see the first one says exactly what is displayed "US H&HS" while the second one reads as "United States Health and Human Services". This is accomplished using rx:abbreviation = "United States Health and Human Services" as an attribute on the abbreviation in XSL FO.

Tables

The combined functionality of the rollmap and XSL FO extensions allows you to mark up tables as prescribed by Section 508.

Let's start with a simple table wiith only column based headings. RenderX internally already understands the content with the "table-header" XSL FO tag should be marked as "TH" in the tagged PDF result. The user does not need to do anything special except to use "table-header" structure in their XSL FO.

Column Header 1	Column Header 2	Column Header 3	Column Header 4	Column Header 5
Cell	Cell	Cell	Cell	Cell
Cell	Cell	Cell	Cell	Cell
Cell	Cell	Cell	Cell	Cell

In XSL FO, there is no mechanisim for marking table headers that are in a column. However, you can in HTML and the Section 508 specification allows for this. You can use "rx:pdf-structure-tag" to override content in a column's cell to accomplish just this. What follows is a sample table to examine with your PDF tag viewer. You will see that the first column's cells are also marked as "TH" tags.

Headings	Column Header 1	Column Header 2	Column Header 3	Column Header 4	Column Header 5
Row Header 1	Cell	Cell	Cell	Cell	Cell
Row Header 2	Cell	Cell	Cell	Cell	Cell
Row Header 3	Cell	Cell	Cell	Cell	Cell

This table exhibits both the rolemap which is mapping elements within this table header rows to be classified as "TH", but also shows that "rx:pdf-structure-tag" can be used to override the content in the first column of cells to map them also as row-based table headers. This is an essential requirement of Section 508 compliance.

Hints and Tips

There are a few other things to consider in the setup of RenderX XEP software for proper processing of Section 508 compliant documents. First, you should try to use the base 14 fonts and not use custom fonts. Because of the nature of custom font processing and requirements special spacing bewteen words, many custom fonts will cause document sizes to increase considerably over using built-in fonts (Helvetica, Times, Courier).

Font kerning should be turned off by setting KERN as "false" in the setup file. Font kerning causes many fragments of text in the output PDF as they must all be placed individually to account for the kerning.

You should avoid the use of nested "fo:block" elements as they are really unnecessary. This by nature would lead to nested "P" elements in the tagged PDF. While not a violation, it can certainly be avoided by structuring the input FO without nested "fo:block" elements.

Summary

RenderX has described in words and has demonstrated through this actual PDF document that our software meets all the requirements for Section 508 compliance. This document in both PDF and source format are available from http://www.renderx.com.