Presentation Layer Unification System (PLUS)

This is the documentation for the genearized implementation of a College of Education presentation layer for print (pdf,rtf), web (html), etc. Additional documentation is at: \\files.ed.uiuc.edu\Ed Admin\OET\ITGroup\Projects\PresentationLayerUnificationSystem

Print Documents

Print documents are generated by converting pphtml to xsl-fo and then processing the xsl-fo with FOP.

> Application xml → application xslt → pphtml → pphtml2xslfosimplified → xsl-fo → pdf or rtf

Affecting All print documents.

The behavior of **all** print documents implementing PLUS can be changed by editing `pphtml2xslfosimplified.xslt` and `defaultxslfoattributes.xml` in the \library\plus folder.

`pphtml2xslfosimplified.xslt` is the main transform engine. It has a couple of general templates for most inline and block html elements (inline_processor and block_processor) and templates for tricky html elments such as a,dd, cite, and font. Tables are more complex and their templates are in an included xslt `pphtml2fosimplified_tables.xslt`

`defaultxslfoattributes.xml` is an xml file imported as a variable in pphtml2xslfo.xslt. It is analogous to an html css style sheet.

An example is:

```
<h1>
    <applied_as_group font-size="16pt" line-height="20pt"
    keep-with-next="always" space-after="12pt"
    font-family="serif" font-weight="bold" />
    <applied_individually />
</h1>
```

The `defaultxslfoattributes.xml` document does not control what xsl-fo element the html element is mapped to. The pphtml2xslfosimplified document does this. For example the h1 html element is mapped the xsl-fo block element and the html p element is mapped to the xsl-fo block element. `defaultxslfoattributes.xml` does control what attributes are applied to the xsl-fo element. The attributes within the applied_as_group elements are taken as a whole from either the print_instructions in the

pphtml document or from `defaultxslfoattributes.xml` and added to the xsl-fo element.

The applied individually attributes are special cases. Putting applied_individually attributes in defaultxslfoattributes.xml won't necessarily make them appear in the xsl:fo element they are mapped to; only when a template looks to defaultxslfoattributes.xml to find a default attribute value when one is otherwise underivable, such as the border in a table cell. These are generally on elements whose attributes are complex to derive because of inheritance or other contextual information.

Affecting A Specific Application's print documents

The behaviour of documents from a specific application are affected by that application's print instructions in the html head element of the pphtml document. That is, specifications that override the general print behaviour can be included in the pphtml document. .

PDF table of Contents

The pdf table of contents is cofigured in the pphtml print instructions. For example:

```
<pdf_table_of_contents include="true" hdepth="4" name="OAR Table of
Contents" />
```

Indicates a table of contents should be used (include="true") and that it should include h1-h4 elements (hdepth="4"). The title of the table of conents is indicated in the name attribute. If the name attribute is left out, this defaults to table of contents.

PDF bookmarks

The pdf bookmarks are cofigured in the pphtml print instructions. For example:

```
<pdf_bookmarks include="true" hdepth="3" />
```

Indicates bookmarks should be used (include="true") and that it should include h1-h3 elements (hdepth="3"). The title of the booka

XSL-FO Layout Tables

Xsl-fo layout tables must be generated to control orphaning and widowing. This is because FOP only support the keep-with-next, keep-together, and keep-with-previous attributes in table row elements. Xsl-fo layout tables are controlled by the print instructions in the head element of the pphtml document. They are only useful for particular html and xsl-fo elements.

To use xsl-fo layout tables to control orphans and widowing, 2 things must be done:

1. map a parent html element to be a layout table

Any block html element that is mapped to an xsl-fo layout table, will have all its immediate descendent elements wrapped in xsl-fo row elements. To map an html element to be a layout table, add a `<layout  table="true"></layout>` element to the element in the pphtml print instructions such as:
`<div class="firsttiersection"><layout  table="true"></layout></div>`

Whoever is coding the preprint html document will need to note: all child elements of any block html element that is mapped to an xsl-fo layout table will become table rows which are block elements in xsl-fo, even inline elements.

So if a series of inline elements are beneath said element, they will become block elements. Text not within child elements will be discarded.

For example, if div.firstTierSection is mapped to an xslt layout table then:
`<div class="firstTierSection">I am <b>having</b> <i>fun</i>.  <p>You are too</p></div>`

Will transform to

Having
Fun
You are too

if div.firstTierSection is NOT mapped to an xslt layout table then:

`<div class="firstTierSection">I am <b>having</b> <i>fun</i>.  <p>You are too</p></div>`

Will transform to

I am **Having** *Fun*
You are too

I can get the xslt to group consecutive inline child elements into a single xsl fo table row, but there would be a considerable performance hit and we are better off just generating reasonable html.

2. map row attributes such as keep-together-always to the child hml elements.

All child rows of the layout table will get their orphan and widow control attributes from the print config of the element they wrap around. These will only have an effect if their parent html element is mapped to a layout table.

For example,

```
<div class="secondtiersection">
      <layout><row_attributes  keep-together="always" /></layout>
</div>
```

Will keep the entire row of the layout table which div.secondtiersection has been mapped to togther on a page.

Resources

Xsl-fo reference: http://www.w3schools.com/xslfo/xslfo_reference.asp
FOP xsl-fo compliance: http://xml.apache.org/fop/compliance.html
RTF Specification (1.6) http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnrtfspec/html/rtfspec.asp