HSSF Serializer (2.1 legacy document)

Table of contents

Table of contents

Warning:
This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. HSSF Serializer

The HSSF serializer catches SAX events and creates a spreadsheet in the XLS format used by Microsoft Excel (but the output looks just dandy in Gnumeric or OpenOffice.org as well).

The HSSF Serializer supports most of the functionality supplied by the HSSF API which is part of the Jakarta POI project.

- Name: xls
- Class: org.apache.cocoon.serialization.HSSFSerializer
- Cacheable: no

2. Usage

Using the HSSF Serializer is fairly simple. You'll need a sitemap of course. Once you have that, well, you're half there. Add

```
<map:serializer name="xls" src="org.apache.cocoon.serialization.HSSFSerializer"
locale="us"/>
```

into the <map:serializers/> section of your sitemap. The locale is optional and is used only to validate numbers. Please note that numbers not in US-default format may not be compatible with Gnumeric (it's less cosmopolitan then the HSSF Serializer ;-)). Setting the locale lets you use default number formats from other locales. Set this to a two letter lowercase country code. See java.util.Locale for details.

Next, set up an entry for each URL or set of URLs (via matching rules) resembling this:

```
<map:match pattern="hello.xls"> <map:generate src="docs/samples/hello-page.xml"/>
<map:transform src="stylesheets/page/simple-page2xls.xsl"/> <map:serialize type="xls"/>
</map:match>
```

The most important question is now, which what XML the serializer is fed. You will get it answered in the next paragraph.

3. Required XML Format

The HSSF Serializer expects data in the same XML language as the popular spreadsheet progam Gnumeric. You have different possibilities to get such a document:

- You can create and save a spreadsheet using Gnumeric.
- You can write it yourself using the Schemas or DTDs available at the Gnumeric's website or in Gnumeric's CVS repository.
- You can take one of the samples delivered with Cocoon and start from there.
- Or you use the empty workbook from appendix A in The Gnumeric File Format PDF (all further references to 'PDF' mean this file), which can further simplified to the following:

```
<gmr:Workbook xmlns:gmr="http://www.gnome.org/gnumeric/v7"> <gmr:SheetNameIndex>
<gmr:SheetName>Sheet1</gmr:SheetName> </gmr:SheetNameIndex> <gmr:Sheets>
<gmr:Sheet> <gmr:Name>Sheet1</gmr:Name> <gmr:MaxCol>-1</gmr:MaxCol>
<gmr:MaxRow>-1</gmr:MaxRow> <gmr:Cells> <!-- add your cells here --> </gmr:Cells>
</gmr:Sheet> </gmr:Sheets> </gmr:Workbook>
```

While HSSFSerializer ignores the bulk of the elements, it is suggested you provide at a minimum the basic elements as in the list below. As a general rule, if Gnumeric in the versions 0.7 - 1.04 will load the XML (provided it is tar'd and gzipped as expected), then the HSSFSerializer should be able to handle it.

As a general guideline the following elements are supported in this release. For the nesting have a look into the sample files or the PDF.

- gmr:Workbook - Required. Basically the root element.
- gmr:Sheets - Required. Container for the spreadsheets.
- gmr:Sheet - Required for each sheet. For the attributes have a look at the example above or into the PDF.
- gmr:Name - Required? Defines the sheet's name as it appears on the little tabs under the workbook in your favorite GUI spreadsheet application.
- gmr:MaxRow, gmr:MaxCol - Used to set the dimensions for the sheet. This can be wrong and your spreadsheet application may not care, but some other ports depend upon this, so we set it to be compatible.
- gmr:Rows, gmr:Cols - Used to determine the default row or column width in points via the attribute DefaultSizePts.
- gmr:RowInfo, gmr:ColInfo - Used to determine the row height/column width for a specific row/column in points.
  Attributes:
  - No - row/column number
  - Unit - row/column height

  The count of the rows/columns starts with 0.
- gmr:Cells - Required. Container for all cells.
- gmr:Cell - Defines the actual column and row number as well as the data type.
  Attributes:
  - Row - row number
  - Col - col number
  - ValueType - the data type
    If you don't specify the data type, the cell content will not be shown! The type is determined by a numerical key, where the following are known: 10 - empty, 20 - boolean, 30 - integer, 40 - float, 50 - error, 60 - string, 70 - cell range, 80 - array
- gmr:Content - Defines the start of the value contained in the cell. This is obsolete as of Gnumeric 1.03. It's not recommended to use it, because it may not be supported in future versions. With POI release 1.5.1 I didn't use gmr:Content, but I had to specify '10' as ValueType on empty cells. Otherwise I got strange output.
- gmr:Styles - Required if you want to use styles. Container for gmr:StyleRegion's.
- gmr:StyleRegion - Defines the region that the style applies to.
  Attributes:
  - startRow - self-explanatory
  - startCol - self-explanatory
  - endRow - self-explanatory
  - endCol - self-explanatory

  Again: The count of the rows/columns starts with 0.
- gmr:Style - Specifies the style for a StyleRegion.
  Attributes:
  - HAlign - specifies the horizontal alignment.
    Possible values: 1 - general, 2 - left, 4 - right, 8 - center, 16 - fill, 32 - justify, 64 - center across

selection
    - VAlign - specifies the vertical alignment.
      Possible values: 1 - top, 2 - bottom, 4 - center, 8 - justify
    - WrapText - specifies whether to wrap text around or not
      Possible values: 0 - don't wrap, 1 - do wrap
    - Shade - kind a stupid flag
      If you're setting a background color and want it filled ... use Shade="1".
    - Format - number format to use.
      Generally, Excel and Gnumeric have the same formats.
- gmr:Font - Defines the font used for the style region.
  Attributes:
    - Bold - self-explanatory
    - Italic - self-explanatory
    - Underline - self-explanatory
    - StrikeThrough - self-explanatory

  Set the values of the attributes to 0 or 1 to disable or enable a specific font style.
- gmr:StyleBorder - Defines the borders that are used for a style region. It contains one element for each possible border specifying the style and the color of the border.

For more specific information on the Gnumeric file format, especially on some more interesting attributes or attribute values or the nesting of the elements, I only can recommend you to read the PDF or to have a look at the sample files. If you want it more complicated, you can also get the information from the Schema file (look above for the link).

4. Automatic Excel Spreadsheet Generation

Hmm, I don't want to say to much on this. I showed you the XML the serializer wants to have. Now it's up to you to generate this XML dynamically. The best way to do this is using a XSLT stylesheet. You need some information on this? Hmm, have a look at the W3C's XSL page. From there you get many links to tutorials, articals, the surprisingly readable XSLT spec and so on.

5. Future Features

So HSSF Serializer is well on its way to being darn near everything you need to create fancy smancy reports in Excel or OpenOffice. (And you can just serialize the output from your stylesheets as XML for Gnumeric version).

- Add support for formulas. (not yet supported by HSSF)
- Add support for custom data formats. (not yet supported by HSSF)

1. Comments

add your comments