

Dynamic PDF Generation using ColdFusion MX, XSL-FO, and FOP

Posted At : March 22, 2004 11:51 AM | Posted By : Steven Erat

Related Categories: Java, ColdFusion

The Apache **XML FOP Project** utilizes XML based stylesheets written to the XSL-FO schema as the input to render as output any of a variety of document types, but most commonly PDF. ColdFusion MX can easily be used to generate dynamic content in a pre-defined XSL-FO stylesheet, where the stylesheet is then transformed by a FOP servlet to produce they customized PDF as output to the client. From the Apache website, FOP is described as:

FOP (Formatting Objects Processor) is the world's first print formatter driven by XSL formatting objects (XSL-FO) and the world's first output independent formatter. It is a Java application that reads a formatting object (FO) tree and renders the resulting pages to a specified output. Output Formats currently supported include PDF, PCL, PS, SVG, XML (area tree representation), Print, AWT, MIF and TXT. The primary output target is PDF.

Here's what I did to generate my own dynamic PDFs with ColdFusion MX:

- **Download** FOP
- Build the FOP web application in as described in `{FOP_HOME}examplerservletREADME [More]`
- Note: `ant.jar` was supposed to be with the other jars required for the build according to `build.bat`, but it was missing from `%LIBDIR%` where `LIBDIR` evaluates to `../../lib` or `{FOP_HOME}/lib`. I grabbed an `ant.jar` from another project and plugged it in there, then the build worked.
- Extract the `{FOP_HOME}examplerservletuildfop.war` to the same directory
- Copy the `FopServlet.class` from `{FOP_HOME}examplerservletuildfopWEB-INFclasses` to `{CFMX_HOME}wwwrootWEB-INFclasses`
- Copy all the jar files from `{FOP_HOME}examplerservletuildfopWEB-INFlib` to `{CFMX_HOME}wwwrootWEB-INFlib`
- Add a Fop Servlet definition and mapping to ColdFusion's deployment descriptor, `{CFMX_HOME}wwwrootWEB-INFweb.xml`.

In `web.xml`, after the very last servlet definition, the one for RDS, add the FopServlet definition:

```
Fop
FopServlet
```

In `web.xml`, after the very last servlet mapping, the one for `CFCServlet/*.jws`, add the Fop servlet mapping:

```
Fop
/Fop
```

- Restart ColdFusion MX.
- Create a location for your FOP test files
- Build a test that dynamically generates XSL-FO with dynamic contents [[EXAMPLE FILES](#)]

```
[cfile action="WRITE"
nameconflict="OVERWRITE"
output="#xslfoResult#"
file="#expandpath('.')#xsl-fo_result.fo">
```

Forward location of the .fo file to the FOP Servlet

```
[cfscript>
currentDir = expandPath('.');
getPageContext().forward('/Fop?fo=#currentDir#/xsl-fo_result.fo');
[/cfscript>
```

PDF output is automatically rendered in the browser

Example PDF output from one of my tests can be viewed here [[PDF](#)]. I hope to have a demonstration here on TalkingTree.com very soon so that you can fill out a form to generate a PDF to display your results on the fly.

The **XSL-FO syntax** is not terribly difficult to learn. It's basically just a hierarchical set of containers to hold content and provide visual layout and style. I used a working **example** that was provided with FOP as a starting point. Then I just deconstructed it to a simpler form, devoid of content, and finally mixed in the appropriate CFML to dynamically generate the XSL-FO file.

Good Luck!

[[Example CFM Source](#)] [[Example PDF Result](#)]