Getting started with

FOP

Patrick Paul <paulpatrick@gmail.com>
2006-10-12, TH8

US 2006

Topics

- Capabilities
- Project Status
- Integrating FOP
- Developing documents
- Q & A

XSL

- eXtensible Stylesheet Language
- Consists of two parts
  - XSLT – Transformations
  - XSL-FO – Formatting Objects
- Apache FOP implements XSL-FO
  - A good subset of XSL-FO 1.0
  - Some elements from XSL-FO 1.1 (CR!)

US 2006

Compliance

- FOP tries to be a reference implementation
- See http://xmlgraphics.apache.org/fop/compliance.html
- Extensions
  - General extensions (fox: prefix)
  - Output format specific extensions

US 2006

Document Types

- Business documents
  - Invoices, insurance policies, letters etc.
- Reports
  - Tabular data
- Book-like documents
  - Books
  - Papers
  - DocBook

Trying to do too much?

- Conflict of interest:
  - Business docs, reports: Speed
  - Books, Papers: Quality
- XSL-FO is feature-rich but still lacking for certain tasks
- XSL-FO is no catch-all solution!

Alternatives

- CSS in simpler situations
- TeX especially for scientific docs
- Proprietary formatters
  - High-speed for business docs
  - Specialized tools: FrameMaker & Co.
- ODF (Open Document Format)
- etc. etc.

Output Formats

- Page-oriented
  - Stable: PDF, PostScript, Plain Text
  - Almost: Java2D/AWT, Print, PNG, TIFF
  - Sandbox/New: AFP/MO:DCA, PCL 5
- Flow-oriented
  - RTF (optimized for MS Word)
- FOP is extensible: your format!

Non-FO content

- fo:external-graphic
  - SVG, bitmap images (PNG, JPEG, GIF etc.)
- fo:instream-foreign-object
  - SVG (through Apache Batik)
  - Barcodes (through Barcode4J)
  - MathML (through JEuclid)
  - FOP is extensible: your format!
- Others: XMP metadata

Special Features

- PDF encryption (PDF 1.3 level only)
- PDF/A-1b (not 100% complete)
- PDF/X (coming up)
- Intermediate Format (Area Tree XML)

Project History

- FOP contributed to the ASF by James Tauber in 1999
- Famous FOP 0.20.5 in July 2003
- Batik and FOP form the XML Graphics project in October 2004
- Loooong redesign phase from Oct 2001 until November 2005 with FOP 0.90alpha
- FOP 0.91beta in December 2005
- FOP 0.92beta in April 2006 (last beta)

What's new?

- Completely new layout engine
  - Layout approach borrowed from Donald Knuth (TeX)
- Improved architecture including support for flow-oriented formats
- New API!
- Much improved compliance
- Greater coverage of the FO spec

What's missing?

- Optimizations for large documents
- Floats
- Auto-table layout
- Collapsing border model
- A lot of smaller things...

Prospects

- FOP 1.0 imminent
- Important missing features are now being attacked.
- Live codebase is interesting for investments. New contributors are always welcome!!!

Integrating FOP

- Formatting Process
- Integration Approaches

Hello World in XSL-FO

```xml
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
 <fo:layout-master-set>
   <fo:simple-page-master master-name="A4"
            page-height="29.7cm" page-width="21cm"
            margin="2cm">
     <fo:region-body/>
   </fo:simple-page-master>
 </fo:layout-master-set>
<fo:page-sequence master-reference="A4">
   <fo:flow flow-name="xsl-region-body">
     <fo:block>Hello World!</fo:block>
   </fo:flow>
</fo:page-sequence>
</fo:root>
```
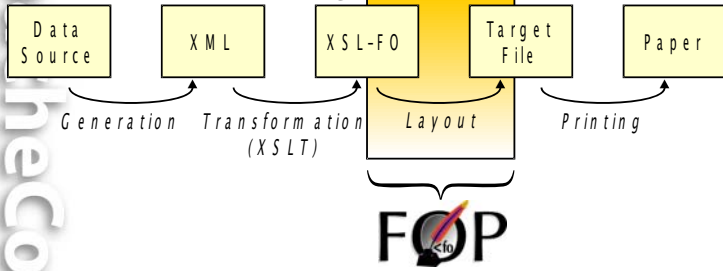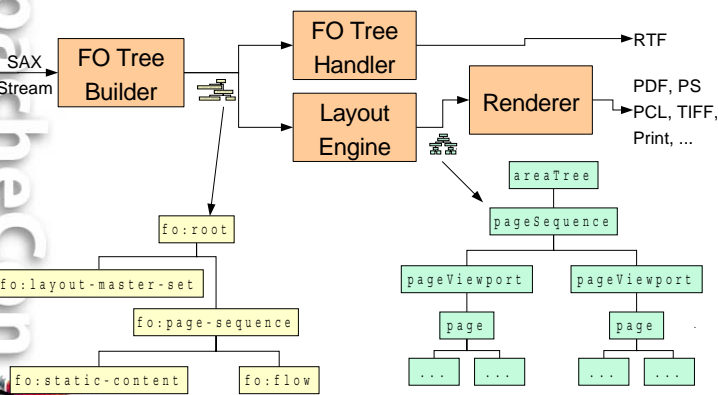
Formatting Process

Data Source		XML		XSL-FO		Target File		Paper

Generation *Transformation (XSLT)* *Layout* *Printing*

FOP

FOP is only a part of the transformation chain!

How FOP works

- **Input:** XSL-FO (as a SAX stream)
- Direct conversion for flow-oriented formats
- Layout Engine (Pagination) for page-oriented formats
- **Output:** Any of the supported output formats

Data Flow inside FOP

SAX Stream → FO Tree Builder → FO Tree Handler → RTF

Layout Engine → Renderer → PDF, PS, PCL, TIFF, Print, ...

fo:root
- fo:layout-master-set
- fo:page-sequence
  - fo:static-content
  - fo:flow

areaTree
- pageSequence
  - pageViewport
    - page
      - ... ...
  - pageViewport
    - page
      - ... ...

Integrating FOP

- Requirements:
  - Java Runtime Environment (1.3.1 or later)
- Usage:
  - Command-line
  - From Java (embedded)
  - Ant Task
  - Servlet
  - etc. etc.

Your Skills!

- Know your XML!
  - Namespaces are important to keep XSLT and XSL-FO apart.
- Know your XSLT and XSL-FO!
- At least some basic knowledge about Java
  - Controlling a class path (-cp)
  - Setting the VM heap size (-Xmx 256M)

Command-line

- Use in scripts
- For stylesheet development/debugging
- Slow! (Class loading, JIT, each time)
- Restricted functionality
- Easy to use:

```
fop -xml mydata.xml -xsl my2fo.xsl -pdf
   out.pdf
```

Ant Task

- Useful for generating documentation in a project
- Useful for batch processing

```xml
<target name="generate-multiple-pdf"
    description="Generates multiple PDF files">
<fop format="application/pdf" outdir="${pdf.dir}">
<fileset dir="${fo.dir}">
    <include name="*.fo"/>
</fileset>
</fop>
</target>
```

Servlet

- Sample servlet included in the distribution
  - Don't use the sample servlet in production!
  - It's only a simple example and a starting point.
- Fast
- Guard against DoS attacks!
  - Restrict concurrency!
  - Be in control what gets rendered!

Embedding in Java

- For any custom integration work
- Requires Java knowledge (obviously)
- Requires JAXP knowledge
- FOP's API tries to reuse most of the basic JAXP Transformer usage pattern.
- Coupling XSLT and FOP using SAX
- Step-by-step example on the website!

Other Possibilities

- Apache Cocoon
  - May be a bit complicated at first but handles the whole transformation chain for you!
- Some have written WebServices
  - Return PDFs as attachments
- Working on a .NET integration for FOP (using IKVM)

Developing Documents

- Skills
- Approaches
- Tips
- Troubleshooting

Your Skills!

- Again XML, XSLT and XSL-FO!
- XSLT is a programming language, but it's not like Pascal or C or Java.
- The XSL specification is a complex beast but don't be afraid to look at it.

Approaches

- WYSIWYG or WYSINWIG Editors
  - Ideal for simple documents
- Structural Editors
  - Allows for more complex documents
- XSLT programming by hand
  - Full flexibility
- Mixed development
  - The best of both worlds
- Editing in non-FO formats (DocBook)

Experience

(This mostly applies to business docs only!)

- Many start with WYSIWYG Editors

- Many end up writing XSLT

- You may need to use both approaches.

- It all depends on your requirements and on the people doing the development.

A few tips

- Install GhostScript/GhostView
  - Displays and auto-reloads PDF/PS files
- Or open the PDF in the browser instead of directly in Acrobat Reader
  - File is not locked this way. Just press F5.
- Don't use the JDK's parser and XSLT implementation (too buggy)
  - "Endorsed standards override mechanism"

Endorsed Standards Override

- http://java.sun.com/j2se/1.4.2/docs/guide/standards/
- Download the latest Xerces-J and Xalan-J (or SAXON)
- Put the JAR files in the "endorsed" directory
  - JRE: <jre-home>/lib/endorsed
  - JDK: <jdk-home>/jre/lib/endorsed
- Or use "-Xbootclasspath/p:"

When writing XSLT...

- Make use of the "import" facility.
- Extract common templates into "library" stylesheets (address formatting, for example)
- Avoid "spaghetti code" and nested for-each.
- Use "attribute-sets" to define styles.
- Refactoring helps, even in XSLT
- Make use of FO's property inheritance

Identifying problems

- Split the transformation chain.
- Write the generated XSL-FO to a file.
  - "-foout" on the command-line
- Comment out portions of the XML/XSLT to narrow down the cause.
- You get line numbers if you feed FOP FO instead of XML+XSLT.

Problem in XSLT or FOP?

- Many people mix XSL transformation and FO processing in their brains.
  - Example: You don't have access to page numbers during XSLT!
    - That's what page-number(-citation) are here for. FOP fills in the page numbers later.
- Step 1: XSLT
- Step 2: FOP

Getting help

- Is your problem about XSLT or FOP?
- FOP website contains links to forums and mailing lists on XSLT
- "fop-users" mailing list helps you with Apache FOP.
- Be sure to check the FAQ and the mailing list archives first.

When asking for help...

- Post an example but don't send XSLT files! Send scaled-down FO files!
- Smart questions: quicker answers
- **ALWAYS** state:
  - FOP and Java version
  - Operating System
  - How you use FOP (command-line, servlet etc.)
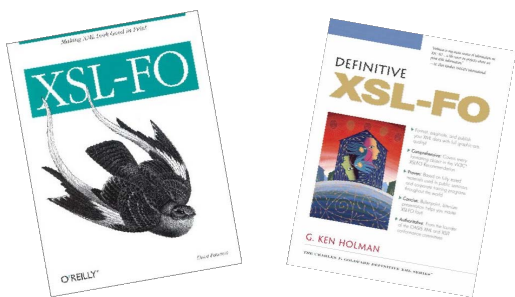 - Application server if applicable

Stuck? Need help?

Contact us by subscribing to
fop-users@xmlgraphics.apache.org

(To subscribe, send an empty mail to
fop-users-subscribe@xmlgraphics.apache.org)

(Forum-style access through GMane and Nabble.)

Questions?

Thank you!!!

Feedback? Comments? Suggestions?

Help wanted in the
XML Graphics project!