Running FOP

1. System Requirements

The following software must be installed:

- Java 1.2.x or later Runtime Environment.
- FOP. The FOP distribution includes all libraries that you will need to run a basic FOP installation. These can be found in the xml-fop/lib directory. These libraries include the following:
  - Apache Xerces-J for XML parsing. You can use other XML parsers which support SAX and DOM.
  - Apache Xalan-J, an XSLT processor.
  - Apache Batik, an SVG library.

The following sofware is optional, depending on your needs:

- Graphics libraries. Support for some graphics formats requires additional packages. See FOP: Graphics Formats for details.
- PDF encryption. See FOP: PDF Encryption for details.

In addition, the following system requirements apply:

- If you will be using FOP to process SVG, you must do so in a graphical environment. See FOP: Graphics (Batik) for details.

2. Installation

2.1. Instructions

Basic FOP installation consists of first unzipping the `.gz` file that is the distribution medium, then unarchiving the resulting `.tar` file in a directory/folder that is convenient on your system. Please consult your operating system documentation or Zip application software documentation for instructions specific to your site.

2.2. Problems

Some Mac OSX users have experienced filename truncation problems using Stuffit to unzip and unarchive their distribution media. This is a legacy of older Mac operating systems,

which had a 31-character pathname limit. Several Mac OSX users have recommended that Mac OSX users use the shell command `tar -xzf` instead.

3. Starting FOP as a Standalone Application

The usual and recommended practice for starting FOP from the command line is to run the batch file fop.bat (Windows) or the shell script fop.sh (Unix/Linux). If you write your own scripts, be sure to review these standard scripts to make sure that you get your environment properly configured.

The standard scripts for starting FOP require that the environment variable JAVA_HOME be set to a path pointing to the appropriate Java installation on your system. Macintosh OSX includes a Java environment as part of its distribution. We are told by Mac OSX users that the path to use in this case is `/Library/Java/Home`. **Caveat:** We suspect that, as Apple releases new Java environments and as FOP upgrades the minimum Java requirements, the two will inevitably not match on some systems. Please see Java on Mac OSX FAQ for information as it becomes available.

```
fop [options] [-fo|-xml] infile [-xsl file] [-awt|-pdf|-mif|-pcl|-ps|-txt|-svg|-at|-print] <outfile>
```

[OPTIONS]

```
     -d             debug mode
     -x             dump configuration settings
     -q             quiet mode
     -c cfg.xml     use additional configuration file cfg.xml
     -l lang        the language to use for user information
     -s             (-at output) omit tree below block areas
     -txt.encoding  (-txt output encoding use the encoding for the output file.
                    The encoding must be a valid java encoding.
     -o [password]  pdf file will be encrypted with option owner password
     -u [password]  pdf file will be encrypted with option user password
     -noprint       pdf file will be encrypted without printing permission
     -nocopy        pdf file will be encrypted without copy content permission
     -noedit        pdf file will be encrypted without edit content permission
     -noannotations pdf file will be encrypted without edit annotation permission
```

[INPUT]

```
     infile            XSLFO input file (the same as the next)
     -fo  infile       xsl:fo input file
     -xml infile       xml input file, must be used together with -xsl
     -xsl stylesheet   xslt stylesheet
```

[OUTPUT]

```
     outfile           input will be rendered as pdf file into outfile
     -pdf outfile      input will be rendered as pdf file (outfile req'd)
```

```
      -awt                input will be displayed on screen
      -mif outfile        input will be rendered as mif file (outfile req'd)
      -pcl outfile        input will be rendered as pcl file (outfile req'd)
      -ps outfile         input will be rendered as PostScript file (outfile req'd)
      -txt outfile        input will be rendered as text file (outfile req'd)
      -svg outfile        input will be rendered as an svg slides file (outfile req'd)
      -at outfile         representation of area tree as XML (outfile req'd)
      -print              input file will be rendered and sent to the printer
                          see print specific options with "-print help"
```

[Examples]

```
      fop foo.fo foo.pdf
      fop -fo foo.fo -pdf foo.pdf (does the same as the previous line)
      fop -xsl foo.xsl -xml foo.xml -pdf foo.pdf
      fop foo.fo -mif foo.mif
      fop foo.fo -print or fop -print foo.fo
      fop foo.fo -awt
```

PDF encryption is only available if FOP was compiled with encryption support **and** if compatible encryption support is availabe at run time. Currently, only the JCE is supported. Check the Details.

4. Using Xalan to Check XSL-FO Input

FOP sessions that use -xml and -xsl input instead of -fo input are actually controlling two distinct conversions: Tranforming XML to XSL-FO, then formatting the XSL-FO to PDF (or another FOP output format). Although FOP controls both of these processes, the first is included merely as a convenience and for performance reasons. Only the second is part of FOP's core processing. If a user has a problem running FOP, it is important to determine which of these two processes is causing the problem. If the problem is in the first process, the user's stylesheet is likely the cause. The FOP development team does not have resources to help with stylesheet issues, although we have included links to some useful Specifications and Books/Articles. If the problem is in the second process, FOP may have a bug or an unimplemented feature that does require attention from the FOP development team.

Note:
The user is always responsible to provide correct XSL-FO code to FOP.

In the case of using -xml and -xsl input, although the user is responsible for the XSL-FO code that is FOP's input, it is not visible to the user. To make the intermediate FO file visible, the FOP distribution includes xalan.bat (Windows batch file) and xalan.sh (Unix/Linux script), which run only the first (transformation) step, and write the results to a file.

Note:
When asking for help on the FOP mailing lists, *never* attach XML and XSL to illustrate the issue. Always run the xalan script and send the resulting XSL-FO file instead. Of course, be sure that the XSL-FO file is correct before sending it.

The scripts are invoked the same way that Xalan is:

```
xalan -in xmlfile -xsl file -out outfile
```

Note that there are some subtle differences between the "fop" and "xalan" command lines.

5. Memory Usage

FOP can consume quite a bit of memory, even though this has been continually improved. This is partly inherent to the formatting process and partly caused by implementation choices. All FO processors currently on the market have memory problems with certain layouts.

If you are running out of memory when using FOP, here are some ideas that may help:

- Increase memory available to the JVM. See the -Xmx option for more information. (Warning: It is usually unwise to increase the memory allocated to the JVM beyond the amount of physical RAM, as this will generally cause significantly slower performance.)
- Avoid forward references. Forward references are references to some later part of a document. Examples include page number citations which refer to pages which follow the citation, tables of contents at the beginning of a document, and page numbering schemes that include the total number of pages in the document ("page N of TOTAL"). Forward references cause all subsequent pages to be held in memory until the reference can be resolved, i.e. until the page with the referenced element is encountered. Forward references may be required by the task, but if you are getting a memory overflow, at least consider the possibility of eliminating them. A table of contents might be eliminated, relying on PDF bookmarks instead. Or it might be moved to the end of the document without dimishing its value very much. Or, in some circumstances, the paper could even be reshuffled after printing.
- Avoid large images, especially if they are scaled down. If they need to be scaled, scale them in another application upstream from FOP. For many image formats, memory consumption is driven mainly by the size of the image file itself, not its dimensions (width*height), so increasing the compression rate may help.
- Use multiple page sequences. FOP starts rendering after the end of a page sequence is encountered. While the actual rendering is done page-by-page, some additional memory allocated for other purposes could be freed after the page sequence has been rendered.
- Break down large tables. If you don't use table headers and footers, just start a new table every N rows. With headers and footers, consider integrating them as normal table rows, or, if they are used at page breaks, try to put the information into static content. You can then use markers to change them.
- Clear the image cache. At the moment, images in the cache are not released automatically when an OutOfMemoryError is imminent. Starting with version 0.20.5 however, you can

call `org.apache.fop.image.FopImageFactory.resetCache()` to empty the image cache.

There are currently some bugs which cause FOP to go into a nonterminating loop, which will also often result in a memory overflow. A characteristic symptom is continuous box overflows in the log. Most of these loops are triggered by elements that do not fit in the available space, such as big images or an improperly specified width in nested block elements. The only workaround is to locate such problems and correct them.

One of FOP's stated design goals is to be able to process input of arbitrary size. Addressing this goal is one of the prime motivations behind the FOP Redesign.

6. Problems

If you have problems running FOP, please see the "How to get Help" page.