Apache FOP: Fonts

Version 627324

by Jeremias Märki, Tore Engvig, Adrian Cumiskey

Table of contents

1 Summary

> **Note:**
> The FOP Font subsystem is currently undergoing a significant change. The details provided here especially related to the generation of FOP Font Metrics files and the FOP Font configuration are likely to change substantially in the future.

The following table summarizes the font capabilities of the various FOP renderers:

Renderer	Base-14	AWT/OS	Custom	Custom Embedding
PDF	yes	no	yes	yes
PostScript	yes	no	yes	yes
TXT	yes (used for layout but not for output)	no	yes (used for layout but not for output)	no
AWT	if available from OS	yes	yes	n/a (display only)
Print	if available from OS	yes	yes	controlled by OS printer driver
RTF	n/a (font metrics not needed)	n/a	n/a	n/a
MIF	n/a (font metrics not needed)	n/a	n/a	n/a
SVG	if available from OS	yes	no	no
XML	yes	no	yes	n/a

2 Base-14 Fonts

The Adobe PDF Specification specifies a set of 14 fonts that must be available to every PDF reader: Helvetica (normal, bold, italic, bold italic), Times (normal, bold, italic, bold italic), Courier (normal, bold, italic, bold italic), Symbol and ZapfDingbats.

3 AWT/Operating System Fonts

The AWT family of renderers (AWT, Print, SVG), use the Java AWT libraries for font metric information. Through operating system registration, the AWT libraries know what fonts are available on the system, and the font metrics for each one.

4 Custom Fonts

Support for custom fonts is added by creating font metric files (written in XML) from the actual font files, and registering them with FOP. Currently only Type 1 and TrueType fonts can be added. More information about fonts can be found at:

- Adobe font types

- Adobe Font Technote

4.1 Type 1 Font Metrics

FOP includes PFMReader, which reads the PFM file that normally comes with a Type 1 font, and generates an appropriate font metrics file for it. To use it, run the class org.apache.fop.fonts.apps.PFMReader:

Windows (on JDK 1.4 and later):

```
java -cp build\fop.jar;lib\avalon-framework.jar;lib\commons-logging.jar;lib\commons-io.jar
          org.apache.fop.fonts.apps.PFMReader [options] pfm-file xml-file
```

Windows (on JDK 1.3.x):

```
java -cp build\fop.jar;lib\avalon-framework.jar;lib\commons-logging.jar;lib\commons-io.jar;lib\xml-
apis.jar;
         lib\xercesImpl.jar;lib\xalan.jar;lib\serializer.jar
          org.apache.fop.fonts.apps.PFMReader [options] pfm-file xml-file
```

Unix (on JDK 1.4 and later):

```
java -cp build/fop.jar:lib/avalon-framework.jar:lib/commons-logging.jar:lib/commons-io.jar
          org.apache.fop.fonts.apps.PFMReader [options] pfm-file xml-file
```

Unix (on JDK 1.3.1):

```
java -cp build/fop.jar:lib/avalon-framework.jar:lib/commons-logging.jar:lib/commons-io.jar:lib/xml-
apis.jar:
         lib/xercesImpl.jar:lib/xalan.jar:lib/serializer.jar
          org.apache.fop.fonts.apps.PFMReader [options] pfm-file xml-file
```

PFMReader [options]:

- **-fn <fontname>** By default, FOP uses the fontname from the .pfm file when embedding the font. Use the "-fn" option to override this name with one you have chosen. This may be useful in some cases to ensure that applications using the output document (Acrobat Reader for example) use the embedded font instead of a local font with the same name.

> **Note:**
> The classpath in the above example has been simplified for readability. You will have to adjust the classpath to the names of the actual JAR files in the lib directory. xml-apis.jar, xercesImpl.jar, xalan.jar and serializer.jar are not necessary for JDK version 1.4 or later.

> **Note:**
> The tool will construct some values (FontBBox, StemV and ItalicAngle) based on assumptions and calculations which are only an approximation to the real values. FontBBox and Italic Angle can be found in the human-readable part of the PFB file or in the AFM file. The PFMReader tool does not yet interpret PFB or AFM files, so if you want to be correct, you may have to adjust the values in the XML file manually. The constructed values however appear to have no visible influence.

4.2 TrueType Font Metrics

FOP includes TTFReader, which reads the TTF file and generates an appropriate font metrics file for it. Use it in a similar manner to PFMReader. For example, to create such a metrics file in Windows from the TrueType font at c:\myfonts\cmr10.ttf:

```
java -cp build\fop.jar;lib\avalon-framework.jar;lib\commons-logging.jar;lib\commons-io.jar
          org.apache.fop.fonts.apps.TTFReader [options]
            C:\myfonts\cmr10.ttf ttfcm.xml
```

TTFReader [options]:

- **-d <DEBUG | INFO >** Sets the debug level (default is INFO).
- **-fn <fontname>** Same as for PFMReader.
- **-ttcname <fontname>** If you're reading data from a TrueType Collection (.ttc file) you must specify which font from the collection you will read metrics from. If you read from a .ttc file without this option, the fontnames will be listed for you.
- **-enc ansi** Creates a WinAnsi-encoded font metrics file. Without this option, a CID-keyed font metrics file is created. The table below summarizes the differences between these two encoding options as currently used within FOP. Please note that this information only applies to TrueType fonts and TrueType collections:

Issue	WinAnsi	CID-keyed
Usable Character Set	Limited to WinAnsi character set, which is roughly equivalent to iso-8889-1.	Limited only by the characters in the font itself.
Embedding the Font	Optional.	Mandatory. Not embedding the font produces invalid PDF documents.

Warning:

You may experience failures with certain TrueType fonts, especially if they don't contain the so-called Unicode "cmap" table. TTFReader can currently not deal with font like this.

4.3 TrueType Collections Font Metrics

TrueType collections (.ttc files) contain more than one font. To create metrics files for these fonts, you must specify which font in the collection should be generated, by using the "-ttcname" option with the TTFReader.

To get a list of the fonts in a collection, just start the TTFReader as if it were a normal TrueType file (without the -ttcname option). It will display all of the font names and exit with an Exception.

Here is an example of generating a metrics file for a .ttc file:

```
java -cp build\fop.jar;lib\avalon-framework.jar;lib\commons-logging.jar;lib\commons-io.jar
          org.apache.fop.fonts.apps.TTFReader -ttcname "MS Mincho"
            msmincho.ttc msminch.xml
```

4.4 Register Fonts with FOP

You must tell FOP how to find and use the font metrics files by registering them in the FOP Configuration. Add entries for your custom fonts, regardless of font type, to the configuration file in a manner similar to the following:

```
<fonts>
  <!-- register a particular font -->
  <font metrics-url="file:///C:/myfonts/FTL_____.xml" kerning="yes"
      embed-url="file:///C:/myfonts/FTL_____.pfb">
     <font-triplet name="FrutigerLight" style="normal" weight="normal"/>
  </font>

  <!-- register all the fonts found in a directory -->
  <directory>C:\MyFonts1</directory>

  <!-- register all the fonts found in a directory
       and all of its sub directories (use with care) -->
  <directory recursive="true">C:\MyFonts2</directory>

  <!-- automatically detect operating system installed fonts -->
  <auto-detect/>
</fonts>
```

Note:

Review the documentation for FOP Configuration for instructions on making the FOP configuration available to FOP when it runs. Otherwise, FOP has no way of finding your custom font information.

- URLs are used to access the font metric and font files. Relative URLs are resolved relative to the font-base property (or base) if available. See FOP: Configuration for more information.
- If relative URLs are specified, they are evaluated relative to the value of the "font-base" setting. If there is no "font-base" setting, the fonts are evaluated relative to the base directory.
- Either an "embed-url" or a "metrics-url" must be specified for font tag configurations.
- The font "kerning" attribute is optional.
- If embedding is off, the output will position the text correctly (from the metrics file), but it will not be displayed or printed correctly unless the viewer has the applicable font available to their local system.
- When setting the "embed-url" attribute for Type 1 fonts, be sure to specify the PFB (actual font data), not PFM (font metrics) file that you used to generate the XML font metrics file.
- The fonts "directory" tag can be used to register fonts contained within a single or list of directory paths. The "recursive" attribute can be specified to recursively add fonts from all sub directories.
- Fonts registered with "font" tag configurations override fonts found by means of "directory" tag definitions.
- Fonts found as a result of a "directory" tag configuration override fonts found as a result of the "auto-detect" tag being specified.
- The fonts "auto-detect" tag can be used to automatically register fonts that are found to be installed on the native operating system.
  - On Unix platforms the autodetect feature looks in java user.home + "/.fonts", "/usr/local/fonts", "/usr/share/fonts" and "/usr/X11R6/lib/X11/fonts" for fonts it is able to use.

- On Mac platforms the autodetect feature looks in java user.home + "/Library/Fonts/, "/Library/Fonts/", "/System/Library/Fonts/" and "/Network/Library/Fonts/" for fonts it is able to use.
- On Windows platforms the autodetect feature attempts to determine the Windows fonts directory (usually C:\WINDOWS\FONTS) and also the existence of a PSFONTS directory for fonts it is able to use.

4.5 Embedding

> **Note:**
> The PostScript renderer does not yet support TrueType fonts, but can embed Type 1 fonts.

> **Note:**
> The font is simply embedded into the PDF file, it is not converted.

Font embedding is enabled in the userconfig.xml file and controlled by the embed-url attribute. If you don't specify the embed-url attribute the font will not be embedded, but will only be referenced.

> **Warning:**
> Omitting the embed-url attribute for CID-encoded TrueType fonts will currently produce invalid PDF files! If you create the XML font metric file using the "-enc ansi" option, you can omit the embed-url attribute for TrueType fonts but you're restricted to the WinAnsi character set.

When FOP embeds a font, it adds a prefix to the fontname to ensure that the name will not match the fontname of an installed font. This is helpful with older versions of Acrobat Reader that preferred installed fonts over embedded fonts.

When embedding PostScript fonts, the entire font is always embedded.

When embedding TrueType fonts (ttf) or TrueType Collections (ttc), a subset of the original font, containing only the glyphs used, is embedded in the output document.

4.6 Explicitly embedding the base 14 fonts

There are cases where you might want to force the embedding of one or more of the base 14 fonts that can normally be considered available on the target platform (viewer, printer). One of these cases is PDF/A which mandates the embedding of even the base 14 fonts. Embedding a font such as Helvetica or Courier is straight-forward. The "Symbol" and "ZapfDingbats" fonts, however, currently present a problem because FOP cannot correctly determine the encoding of these two single-byte fonts through the PFM file. FOP now correctly interprets the "encoding" value in the XML font metrics file, but the PFMReader application writes "UnknownEncoding" to the generated XML file. In order to embed "Symbol" and "ZapfDingbats" you have to manually change the XML font metrics file and specify "SymbolEncoding" or "ZapfdingbatsEncoding" encoding respectively as the value for the "encoding" element.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<font-metrics type="TYPE1">
  <font-name>Symbol</font-name>
  <embed/>
```

```
   <encoding>SymbolEncoding</encoding>
   <cap-height>673</cap-height>
   <x-height>766</x-height>
   [..]
```