

# Apache FOP: Configuration

\$Revision: 357081 \$

## Table of contents

1 Configuration File Basics.....	2
1.1 Making Configuration Available to FOP.....	2
2 Summary of the General Configuration Options.....	2
3 Renderer configuration.....	4

## 1. Configuration File Basics

The FOP configuration file is an XML file containing a variety of settings that are useful for controlling FOP's behavior, and for helping it find resources that you wish it to use.

The easiest way to get started using a FOP configuration file is to copy the sample found at `{fop-dir}/conf/fop.xconf` to a location of your choice, and then to edit it according to your needs. It contains templates for the various configuration options, most of which are commented out. Remove the comments and change the settings for entries that you wish to use. Be sure to follow any instructions, including comments which specify the value range. Also, since the configuration file is XML, be sure to keep it well-formed.

### 1.1. Making Configuration Available to FOP

After creating your configuration file, you must tell FOP how to find it:

- If running FOP from the command-line, see the "-c" command-line option in [Running FOP](#).
- If running FOP as an embedded application, see [Embedding, Using a Configuration File](#).

See [Setting the Configuration Programmatically](#) for instructions on how to do so in an embedded environment.

## 2. Summary of the General Configuration Options

Element	Data Type (for the value)	Default Value
base	URL	Specifies the base URL based on which relative URL will be resolved.
source-resolution	Integer, dpi	Resolution in dpi (dots per inch) which is used internally to determine the pixel size for SVG images and bitmap images without resolution information.
target-resolution	Integer, dpi	Resolution in dpi (dots per inch) used to specify the output resolution for bitmap images generated by bitmap renderers (such as the TIFF renderer) and by bitmaps generated by

		Apache Batik for filter effects and such.
strict-validation	Boolean (true, false)	Setting this option to 'false' causes FOP to be more forgiving about XSL-FO validity, for example, you're allowed to specify a border on a region-body which is supported by some FO implementations but is non-standard. Note that such a border would currently have no effect in Apache FOP.
break-indent-inheritance	Boolean (true, false)	Setting this option to 'true' causes FOP to use an alternative rule set to determine text indents specified through margins, start-indent and end-indent. Many commercial FO implementations have chosen to break the XSL specification in this aspect. This option tries to mimic their behaviour. Please note that Apache FOP may still not behave exactly like those implementations either because FOP has not fully matched the desired behaviour and because the behaviour among the commercial implementations varies. The default for this option (i.e. false) is to behave exactly like the specification describes.
default-page-settings	n/a	Specifies the default width and height of a page if "auto" is specified for either or both values. Use "height" and "width" attributes on the default-page-settings element to specify the two values.
renderers	(see text below)	Contains the configuration for each renderer. See below.

This is an excerpt from the example configuration file coming with FOP:

```
<fop version="1.0">

  <!-- Base URL for resolving relative URLs -->
  <base>./</base>

  <!-- Source resolution in dpi (dots/pixels per inch) for determining the
size of pixels in SVG and bitmap images, default: 72dpi -->
  <source-resolution>72</source-resolution>
  <!-- Target resolution in dpi (dots/pixels per inch) for specifying the
target resolution for generated bitmaps, default: 72dpi -->
  <target-resolution>72</target-resolution>

  <!-- default page-height and page-width, in case
value is specified as auto -->
  <default-page-settings height="11in" width="8.26in"/>

  <!-- etc. etc..... -->
</fop>
```

### 3. Renderer configuration

Each Renderer has its own configuration section which is identified by the MIME type the Renderer is written for, ex. "application/pdf" for the PDF Renderer.

The configuration for the PDF Renderer could look like this:

```
<renderers>
  <renderer mime="application/pdf">
    <filterList>
      <!-- provides compression using zlib flate (default is on) -->
      <value>flate</value>
    </filterList>
    <font>
      <font metrics-url="arial.xml" kerning="yes" embed-url="arial.ttf">
        <font-triplet name="Arial" style="normal" weight="normal"/>
        <font-triplet name="ArialMT" style="normal" weight="normal"/>
      </font>
      <font metrics-url="arialb.xml" kerning="yes" embed-url="arialb.ttf">
        <font-triplet name="Arial" style="normal" weight="bold"/>
        <font-triplet name="ArialMT" style="normal" weight="bold"/>
      </font>
    </font>
  </renderer>

  <renderer mime="application/postscript">
    <!-- etc. etc..... -->
```

The details on the font configuration can be found on the separate [Fonts](#) page. Note especially

the section entitled [Register Fonts with FOP](#).