

# Ant task

\$Revision: 828999 \$

## Table of contents

1 Description.....	2
2 Parameters for FOP Ant task.....	2
3 Examples.....	4

Apache FOP provides an Ant task for automating the document build process.

## 1. Description

The FOP Ant task will convert XSL-FO documents to PDF, PS, PCL etc. output (see [Output formats](#) for available formats).

To call FOP tasks within Ant, first add a FOP task definition to your Ant build file. One method of defining the task is as follows:

```
<property name="fop.home" value="....path to your FOP HOME directory..." />

<taskdef name="fop"
  classname="org.apache.fop.tools.anttasks.Fop">
  <classpath>
    <fileset dir="${fop.home}/lib">
      <include name="*.jar" />
    </fileset>
    <fileset dir="${fop.home}/build">
      <include name="fop.jar" />
      <include name="fop-hyph.jar" />
    </fileset>
  </classpath>
</taskdef>
```

Then create FOP tasks within your Ant build file, using the FOP task parameters listed below.

## 2. Parameters for FOP Ant task

Attribute	Description	Required
fofile	XSL-FO file to be rendered	Yes, if no fileset nested element is used
xmlfile	XML input file	Yes, if no fofile is specified
xsltfile	XSLT input file	Yes, if no fofile is specified
outfile	Output filename	Yes, when fofile is used. (This attribute is not valid for filesets.)
format	Possible output formats: application/X-fop-awt-pr application/X-fop-print application/X-fop-areatr application/pdf application/postscript application/mif application/rtf, text/richtext, text/rtf application/x-pcl, application/vnd.hp-PCL	No, defaults to application/pdf

	application/x-afp, application/vnd.ibm.modc text/plain image/svg+xml image/gif image/png image/tiff	
outdir	Output directory	Required if a fileset is used to specify the files to render; optional for fofile. (Can alternatively specify the full path in the fofile value.)
force	Recreate target files, even if they are newer than their corresponding source files. Note: This attribute is available in post-0.20.5 versions (0.20.x nightly build and 1.0dev) only; target files are always generated (i.e., force=true) in 0.20.5 release.	No, default is false
basedir	Base directory to resolve relative references (e.g., graphics files) within the FO document.	No, for single FO File entry, default is to use the location of that FO file.
relativebase	For fileset usage only. A value of true specifies using the location of each .fo file as the base directory for resolving relative file references located within that .fo file. A value of false specifies using the value of basedir for all files within the fileset, or just the current working directory if basedir is not specified.	No, default is false.
userconfig	User configuration file (same as the FOP "-c" command line option).	No
messagelevel	Logging level Possible values: error, warn, info, verbose, debug. <b>Currently doesn't work in FOP Trunk!!!</b>	No, defaults to verbose
logFiles	Controls whether the names of the files that are processed are logged (true) or not (false). <b>Currently doesn't work in</b>	No, default is true

	<b>FOP Trunk!!!</b>	
throwexceptions	Controls whether or not an exception is thrown if an error occurs during rendering.	Default is true

**Table 1: Parameters specified as attributes**

Attribute	Description	Required
fileset	<a href="#">FileSets</a> are used to specify multiple XSL-FO files to be rendered.	Yes, if no fofile attribute is supplied

**Table 2: Parameters specified as nested elements**

### 3. Examples

The following example converts a single XSL-FO file to a PDF document:

```
<target name="generate-pdf" description="Generates a single PDF file">
  <fop format="application/pdf"
        fofile="c:\working\foDirectory\foDocument.fo"
        outfile="c:\working\pdfDirectory\pdfDocument.pdf" />
</target>
```

This example converts all XSL-FO files within an entire directory to PostScript:

```
<target name="generate-multiple-ps"
        description="Generates multiple PostScript files">
  <fop format="application/postscript"
        outdir="${build.dir}" messagelevel="debug">
    <fileset dir="${fo.examples.dir}">
      <include name="*.fo"/>
    </fileset>
  </fop>
</target>
```

The following example transforms and converts a single XML and XSLT file to an AFP document:

```
<target name="generate-afp-from-transform" description="Generates a single AFP file
from an XSLT stylesheet">
  <fop format="application/x-afp"
        xmlfile="c:\working\foDirectory\Document.xml"
        xsltfile="c:\working\foDirectory\Document.xslt"
        outfile="c:\working\afpDirectory\Document.afp" />
</target>
```

This example transforms and converts all XML files within an entire directory to PostScript:

```
<target name="generate-multiple-ps-from-transform" description="Generates multiple
PostScript files using an XSLT stylesheet">
  <fop format="application/postscript"
        xsltfile="c:\working\foDirectory\Document.xslt"
```

```
        outdir="${build.dir}" messagelevel="debug">
        <fileset dir="${test.dir}">
            <include name="*.xml"/>
        </fileset>
    </fop>
</target>
```