

FOP: Building from Source Code

\$Revision: 490012 \$

Table of contents

1 Do You Need To Build?.....	2
2 Set Up Your Environment.....	2
2.1 JDK.....	2
2.2 CLASSPATH.....	2
2.3 JAVA_HOME.....	2
3 Run the "build" Script.....	2
4 Troubleshooting.....	3

1. Do You Need To Build?

FOP distributions are either pre-compiled binary or source. If you are using a binary distribution, it is already built and there is no need to build it again. See the [Download Instructions](#) for information about whether a binary or source distribution is best for your needs.

2. Set Up Your Environment

2.1. JDK

Building FOP requires a minimum Java Development Kit (JDK/SDK) of 1.3 (A Java Runtime Environment is not sufficient).

2.2. CLASSPATH

There is generally no need to setup a classpath. All libraries needed to compile FOP are included in the source distribution and are referenced by the build script. You will only need to adjust the classpath if you build FOP in some other way. See the build scripts (build.bat for Windows, and build.sh for Unix) for details.

2.3. JAVA_HOME

The build script uses [Ant](#), a popular java-based build tool, which requires that the environment variable JAVA_HOME point to your local JDK root directory. This is true even if you use JDK 1.2 or above, which normally does not need this setting.

3. Run the "build" Script

Build FOP by executing the "build" script, which is located in the FOP root directory. The Windows batch file is build.bat, and the Unix shell script is build.sh. The examples below are for running the shell script, but except for the build file extension, the syntax is identical.

The file build.xml in the FOP root directory is the blueprint that Ant uses for the build. It contains information for numerous build targets, many of which are building blocks to more useful target, and others which are primarily used by the FOP developers. You may benefit from looking through this file to learn more about the various build targets. To obtain a complete list

of useful build targets:

```
build.sh -projecthelp
```

The most useful targets are:

- **package**: Generates the jar files (default). This is the normal build that produces a jar file usable for running FOP.
- **clean** : Cleans the build directory. This is useful for making sure that any build errors are cleaned up before starting a new build. It should not ordinarily be needed, but may be helpful if you are having problems with the build process itself.
- **javadocs**: Generates javadocs. This creates the FOP API documentation.

To run the build:

```
build.sh [target ...]
```

For example to do a normal build for the package target (which is the default):

```
build.sh
```

OR

```
build.sh package
```

To clean the build directory first:

```
build.sh clean package
```

4. Troubleshooting

If you have problems building FOP, please try the following:

- Run the build with the target of "clean", then rerun the build.
- Delete the build directory completely, then rerun the build.
- Make sure you do not have a non-FOP version of xerces.jar, xalan.jar, batik.jar, or another dependency product somewhere in your CLASSPATH.
- If the build still fails, see the [Getting Help](#) page for further help.