FOP Development: Managing Documentation

$Revision: 239450 $

Table of contents

1. General Information

All raw documentation content is managed in the FOP SVN repository. Updates should be committed to the repository, then the repository files are used to generate usable output. The remaining discussions on this page assume that the SVN repository is the starting place for processing. The path to the documentation is xml-fop/src/documentation/content/xdocs.

Note:

All documentation is maintained on the trunk. Although we are currently maintaining two sets of code (trunk and maintenance), there is only one set of documentation. Most of the user and developer doc is common to the two environments, and differences are highlighted where necessary. The major exception to this rule is the design doc, which currently exclusively pertains to the trunk (redesign). Maintenance branch releases either copy the trunk content to the maintenance branch or use the trunk content directly for doc builds.

Basic documents are stored in XML files, and use DTDs provided by Apache Forrest.

2. Design Principles

These principles are not written in stone, but reflect the current philosophy, and are documented here primarily to help achieve consistency. These principles should be changed if better or more practical ones are found, but they should probably be discussed and changed by common consent.

2.1. Where

- To the extent possible, keep user content separate from developer content, primarily so the user doesn't have to filter out technical information.
- To the extent possible, try to document a topic exactly once, in the place the user is most likely to look for it, then link to that from other locations as appropriate. This is somewhat contrary to the principle above, which should be applied as a higher priority.

2.2. When

The documentation and the product are in a constant state of change, and there is some difficulty in deciding what product state the website content should reflect. The current thinking is that the website should reflect the current state of the repository code branch from which releases are made. Features or other documentation that applies to unreleased code should be marked in such a way within the content that the user can determine whether and how it applies to the version they are using. For example, "Feature xyz is first available in Release n.nn.n".

Other approaches were considered, but all seemed to have significantly higher costs both to the users and the developers. From the user's standpoint, the choice is either that they potentially have to look multiple places to get the information they need (which was rejected), or they have to filter out an occasional feature that is in code available subsequent to their release (which was accepted).

3. Website

3.1. Background

The FOP web site and documentation are generated using Apache Forrest.

The following table summarizes the flow of data to the FOP website in chronological order:

Process	Output	State	View(s)
Developer commits code to FOP repository.	FOP repository (SVN)	Raw XML and other content	ViewCVS
Developer builds and uploads documentation using ForrestBot.	/www/xmlgraphics.apach on cvs.apache.org	sync-ready	n/a
Cron job runs rsync to synchronize the website with the real web server (runs every few hours).	Infrastructure knows. :-)	web-ready	FOP Web Site

Note:
Server-side ForrestBot is currently not available for website publishing. We use it locally and with manual invocation.

3.2. ForrestBot "publish" Step-by-Step

We're using ForrestBot for build and deploy the FOP website. ForrestBot comes with Apache Forrest 0.7. The root directory of your FOP checkout contains the file "publish.xml" which is an Ant build file that manages the build and the deployment of the FOP website. Please look into this file for further instructions to set up ForrestBot on your machine. Basically, we're simply running ForrestBot manually by typing "ant -f publish.xml" once we're happy with our changes to the site. Be sure to set up the "deploy.settings" file as described in the "publish.xml" file.

Step-by-step instructions for the deployment process again:

- Modify the sources of the website and check locally with Forrest 0.7 (run "forrest run" or just "forrest").
- Once you're satisfied, run "ant -f publish.xml" to do a clean build of the website. If the build runs without problems, the website will be uploaded as a whole using SCP to cvs.apache.org.
- Wait for the next rsync cycle and check your changes in the live site. (Sorry, no manual rsync available ATM)

3.3. Using a Local Forrest

To use a local Forrest (during website development, not for deployment):

- download latest the Forrest release (currently Forrest 0.7)
- set environment variable FORREST_HOME=~/apache-forrest-0.7 where ~ is the directory where Forrest is installed (see http://forrest.apache.org/docs/your-project.html for details)
- set environment variable PATH=$PATH:$FORREST_HOME/bin
- cd to your local FOP checkout
- update your local FOP checkout (svn update)
- run forrest(.bat), which will build the web-site documents in xml-fop/build/site.

Note:
You can use "forrest run" to start a local web server. That improves development speed as you can simply refresh in the browser after a change.

3.4. Deleting Documentation Files

ForrestBot simply uploads the whole generated site. It doesn't delete obsolete files. You can do that manually in the /www/xmlgraphics.apache.org/fop folder on cvs.apache.org. Be careful when doing stuff like that.

Note:
Please make sure you always have **group rw permissions on all files** under the /www directory!