

# Managing FOP Documentation

## 1. General Information

All raw documentation content is managed in the FOP CVS 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 CVS 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

inking 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 source repository (cvs) at <a href="http://icarus.apache.org/home/cvs/xml-fop">icarus.apache.org/home/cvs/xml-fop</a>	Raw XML and other content	<a href="#">ViewCVS</a>
<a href="#">Forrestbot "refresh"</a> . Automatically done every six hours. Can be manually refreshed by authorized users at <a href="#">the Forrestbot Web Interface</a> .	Physical location unknown and unimportant to us	web-ready	Content can be viewed <a href="#">here</a> .
<a href="#">Forrestbot "publish"</a> . Always done manually at <a href="#">the Forrestbot Web Interface</a> .	FOP web repository (cvs) at <a href="http://icarus.apache.org/home/cvs/xml-site/targets/fop">icarus.apache.org/home/cvs/xml-site/targets/fop</a>	web-ready	<a href="#">ViewCVS</a>
Automatic Live Site Update every six hours (midnight, 6am, noon, 6pm Pacific time).	FOP live web site, somewhere on <a href="http://daedalus.apache.org">daedalus.apache.org</a>	web-ready	<a href="#">FOP Web Site</a>

#### 3.2. Forrestbot "refresh" Step-by-Step

### Note:

The Forrestbot "refresh" is automatically run on the server every six hours. Only follow the steps below if you need to check the results more promptly than that allows, or if you need to "refresh" in preparation for a "publish".

To manually use the Forrestbot "refresh", simply follow these steps:

- Make sure your changes are committed to the FOP source repository.
- Open <http://forrestbot.cocoondev.org> in your browser.
- Login.
- In the "Select a module" box, select "xml-fop".
- Click the "refresh" button to have the interim site built. On-screen instructions tell you how to view the log as the build progresses.

### 3.3. Forrestbot "publish" Step-by-Step

To "publish" the Forrestbot output to the FOP Web repository:

- Make sure you are satisfied with the "refreshed" site.
- Open <http://forrestbot.cocoondev.org> in your browser.
- Login.
- In the "Select a module" box, select "xml-fop".
- Click the "publish" button. On-screen instructions tell you how to view the log as the build progresses.
- Wait for the next 6-hour live-site update cycle and check your changes.

### 3.4. Live Site Update

If there are problems with the live site update (the process of copying the web site contents from the FOP web repository to the live site):

- The CVS update logs can be viewed at <http://www.apache.org/~rubys/updatesite.logx>, where "x" is a number from 1-4, these four files being the updates done over the past 24 hours. Review the most recent log file for clues.
- The script is maintained by Sam Ruby ([rubys@apache.org](mailto:rubys@apache.org)). Contact him for further help.

### 3.5. Using a Local Forrest

#### Note:

Most documentation content changes do not require a local copy of Forrest. In general, use the Forrestbot instead. Forrestbot is easy to use, always uses the "approved" methodology, and has fewer error-prone manual steps.

There are some situations where you may want to have a local Forrest installation. For example, you do not want to tie up server resources testing major changes, such as sitemap

building, that may require many edit/build/test/debug cycles. After you are done testing, use the forrestbot to "refresh" and "publish" the site.

**Note:**

Forrest needs to be run on a machine with a graphical environment. It will fail in a headless environment when it tries to use FOP to generate the PDF files.

To use a local Forrest:

- checkout the xml-forrest module (same repository as xml-fop)
- checkout the xml-site/targets/fop module (same repository as xml-fop)
- you will also need access to a current xml-fop sandbox (you probably already have one)
- cd to xml-forrest
- run: build.sh(bat) dist to build forrest
- set environment variable FORREST\_HOME=~/.xml-forrest/build/dist/shbat where ~ is the directory in which xml-forrest is installed (see <http://xml.apache.org/forrest/your-project.html> for details)
- set environment variable PATH=\$PATH:\$FORREST\_HOME/bin
- cd to xml-fop directory
- run forrest(.bat), which will build the web-site documents in xml-fop/build/site.

### 3.6. Updating the FOP Web Repository Manually (Deprecated)

**Warning:**

The steps in this section should not ordinarily be used. They are documented here for historical reasons, and for emergencies. See [manually deleting retired files](#) for an exception to this rule.

- Copy (or sym-link) the documents generated by Forrest (in xml-fop/build/site) to xml-site/targets/fop on your local machine.
- Commit xml-site/targets/fop.

### 3.7. Deleting Documentation Files

The one place where manual updates of the web cvs repository are required is when a document is retired. At this point, it will no longer be generated. However, it will still exist in the web cvs repository. You will need to use a cvs client to remove the files, then commit the changes to keep them from continuing to exist on the live site.