Draft: Proposal for ASF-wide documentation staging and publishing

Table of contents

Warning:

This is a draft proposal document. It is not yet the consensus of ASF nor of the Infrastructure committee. This proposal is a summary of various email discussions held over the past years, especially on infrastructure@a.o around 2004-07-29 which expanded on previous discussions.

Note:

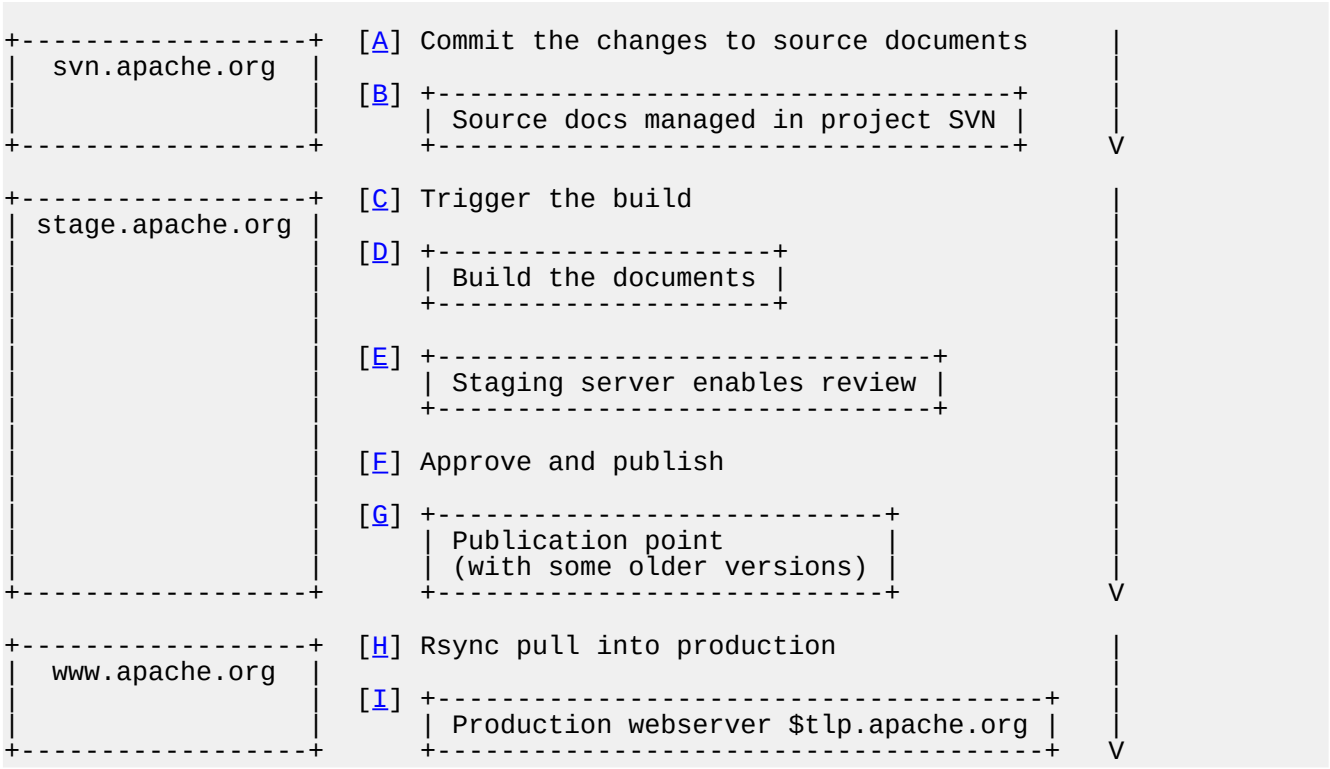
There is nothing specific to Apache Forrest in this proposal.

1. Overview

All ASF projects need to be able to concentrate on their projects and the content of their websites, rather than get tangled up in arcane website publication procedures.

There would be a "staging and publishing server" which is separate from the live production web server. The project committers would commit their source changes, then trigger a "documentation build", then review the staging website. When satisfied, they "approve and publish" so as to copy the stage to the "publication point". There are a number of rotated older versions of the publication point. A deliberate action on the live webserver causes rsync to pull the current publication point into production.

2. Publication infrastructure and actions



2.1. [A] Commit the changes to source documents

The content changes are committed to the project's source repository. The committer might have already built and tested the documents with their local documentation build system. On other occasions, they might commit changes without building locally. Some committers might not even have installed a local build system, they might just edit or patch the content.

2.2. [B] Source docs are managed in project SVN

The source files for the project's website are held in an SVN repository. These might be XML source for some projects, while others might have simple HTML docs.

2.3. [C] Trigger the build

Via a secure https web interface, or via ssh to the server and use command-line.

2.4. [D] Build the documents

The build system on the server will generate the project documents and deploy them to the staging server website.

Projects can use various documentation tools: Anakia, Forrest, Maven, raw html, etc. Each system would have its own ways to report build problems to the committer (e.g. xml validation, broken links, content and spelling errors, configuration errors).

2.5. [E] Staging server enables review

A pre-release website. Anyone can review online. Some projects might want to password-protect.

2.6. [F] Approve and publish

When satisfied, they "approve and publish" so as to copy the stage to the "publication point". Via a secure https web interface, or via ssh to the server and use command-line.

2.7. [G] Publication point

A holding area, from which the production website can be recreated as required. Keep a number of rotating versions, i.e.

```
rm -rf ${publish_dir}.3
mv ${publish_dir}.2 ${publish_dir}.3
mv ${publish_dir}.1 ${publish_dir}.2
mv ${publish_dir} ${publish_dir}.1
mv $staging_dir $publish_dir
```

2.8. [H] Rsync pull from publication point into production

Someone with commit access for the project would issue a command on the live web server to synchronise with the current contents of the publication server via rsync pull. This would either be executed by ssh and command-line, or via a secure https web interface.

We want the final rsync to be independent, so that it can also be executed by infrastructure people in the event that the web sites need to be recreated.

```
The rsync would be manual.

The old way ...
   cd /www/$tlp.apache.org; cvs up -Pd
 or
   cd /www/jakarta.apache.org/$proj; cvs up -Pd
```

```
The new way ...

   rsync -avz -e ssh --delete stage.apache.org:/www/$tlp.apache.org/ \
                                                /www/$tlp.apache.org/
 or
   rsync -avz -e ssh --delete stage.apache.org:/www/jakarta.apache.org/$proj/ \
                                                /www/jakarta.apache.org/$proj/
```

2.9. [I] Production webserver for the project

The live production website for the project at $tlp.apache.org

3. Other notes

- The actions (A and C and F and H) are completely independent manual steps and are deliberate accountable acts. This ensures human oversight in the deployment process.
- The actions should not be automated, especially action H. If someone did manage to break in to the publishing server, then their changes would be automatically published.
- Some people would like action C and action F to be automated (say every 30 hours). Committers can still trigger it manually at other times.
- The actions F and H could be combined. For example, we could have a script on the production server that contacted the publishing server to perform action F and then performed the rsync (action H).
- The proposal from Apache Forrest to have an ASF Forrestbot as one method for projects to handle the "staging server" (item C through to item G). This does not preclude other mechanisms.
- The Doco concept adds interactive and workflow capabilities to this publication infrastructure.

4. Background and impediments

This is a collection of notes about the past impediments which have hindered the publishing process ...

- The generated project sites were maintained in source control, primarily to enable the infrastructure team to restore the live web server in case of emergency. That added one more level of complexity for the projects.
- When people wanted to work on projects docs, they were hindered by needing to install the document generation system locally. That was too onerous for some projects and caused delays with website maintenance.

5. Demonstration

See demonstration on brutus. Some sites are generated by a forrestbot while others are generated by Maven and Anakia (all local to brutus).

We know that no publishing will be done using brutus - this is just a demonstration until a suitable Apache machine is available.

6. Scratch notes

Some notes which have not yet been incorporated ...

```
* How would we log the actions?

* Noel: We need to accomodate sites that come from a single source,
and sites that come from multiple sources,
e.g. Jakarta or the XML Federation.
```