

# WebsiteManagement

## Overview

The following explains how to update and maintain the main Apache Excalibur website at <http://excalibur.apache.org>. If you have further questions, contact the developers mailing list

## Updating the website

Is done roughly like this. Should probably automate more.

- maybe edit trunk/buildsystem/project-common.xml
- maybe edit contents of trunk/site/xdocs/
- generate site:

```
cd trunk/site
maven site
```

- copy to site location in svn

```
cp -r target/docs/* ../../site
```

- add and commit svn files

```
cd ../../site
# this can probably be done cleaner using import...
find | xargs svn add -q
svn commit -m "New version of generated site"
```

- optionally update files on the server (or wait 4 hours)

```
ssh minotaur.apache.org
~leosimons/bin/excalibur-site-up.sh
```

- maybe check contents of

```
http://excalibur.apache.org/svnuplog.txt
```

to make sure nothing went wrong, and check the website.

## Please, keep valid urls valid

IMNHSO its quite important that URLs keep working once you've created them. This doesn't mean stuff can't move around. Rather, add Redirect or [RedirectMatch](#) (see [http://httpd.apache.org/docs-2.0/mod/mod\\_alias.html](http://httpd.apache.org/docs-2.0/mod/mod_alias.html)) directives to the file site/.htaccess in subversion as you move files around. It's real easy to do, just follow along with the samples. In particular, the file

```
/www/jakarta.apache.org/.htaccess
```

on minotaur.apache.org should be a good source for inspiration. Some extensive redirecting is going on there!

## XDoc format

Maven uses the "xdoc" format first promoted by anakia. It's pretty easy to figure out what to do from looking at the existing content (which is probably how most people learn it).

## Look and feel

We currently are just using the maven defaults. If someone feels like changing those, take a look at

- <http://maven.apache.org/reference/plugins/xdoc/properties.html>
- [http://maven.apache.org/reference/user-guide.html#Site\\_Generation](http://maven.apache.org/reference/user-guide.html#Site_Generation)

## Javadocs

```
export RSYNC_RSH=ssh
cd ~/svn/excalibur/trunk/site
# aggregate all src trees, then run "maven site" on the aggregation
sh scripts/gen-site.sh
# make sure to manually check the generated content!
# copy the generated javadocs to their location on minotaur
rsync -azv --delete-after \
  target/base/target/docs/apidocs \
  minotaur.apache.org:/www/excalibur.apache.org/
```

We don't have javadocs generation yet. Does anyone know a good way to make that work for our multiple source directories using maven?

## Clover reports

I've asked The Cortex for a free clover license for excalibur. Once I get a reply we can look into getting these up.

## Autogenerated site?

Once we get these basics figured out, I'd like to have cronjob set up somewhere which automatically updates the site each night. Anyone have experience with that?

## So how does it work?

The content of

<http://excalibur.apache.org/>

is a checkout of the SVN materials at

<https://svn.apache.org/repos/asf/excalibur/site/>

that is refreshed every 4 hours using a simple shell script that runs in my account. To change the contents of the site, commit changes to the svn location and wait 4 hours. Alternatively, log in to the machine using SSH and run

```
sh ~/lsimons/bin/excalibur-site-up.sh
```

(for which you need to be in the excalibur unix group.)

The log of the latest invocation of that script is made available at

```
http://excalibur.apache.org/svnuplog.txt
```

Manually change the contents of the

/www/excalibur.apache.org

directory (where the live site lives) only with much caution.

The xdoc generation process in maven is handled by the xdoc plugin,

<http://maven.apache.org/reference/plugins/xdoc/>

which itself uses Anakia,

<http://jakarta.apache.org/velocity/anakia.html>

which is a thin wrapper (ant task) around velocity:

<http://jakarta.apache.org/velocity/>