# CocoonVendorBranch

### **Abstract**

A variation of the Subversion "vendor branch" strategy for maintaining Cocoon releases under local version control.

#### Motivation

- 1. Control of local modifications (patches, additions to lib/local)
- 2. Being able to easily see what changed from release to release
- 3. Import into the repository only what we need for production builds:
  - · selected blocks only
  - · no samples
  - no documentation

#### Overview

This is a variation on the Subversion "vendor branch" strategy. It differs from the technique outlined in the book, in that we don't have a single main branch into which we merge vendor drops. We want to be able to assimilate a new Cocoon release without forcing all of our Cocoon application to upgrade, so we have a main branch for each Cocoon release that we bring in. That way, we can port each of our applications to a new Cocoon only if and when we need to do so.

I started using this system with the Cocoon 2.1.6 release and have been using it through 2.1.8.

#### **Details**

- 1. mkdir /usr/local/Cocoon/drop
- 2. cd /usr/local/Cocoon/drop
- 3. download and explode the Cocoon release tarball for the current release into this directory.
- 4. Prune the source tree! Delete the following files and directories in src/ (note, this is right for 2.1.6, check it for each version; also, YMMV e.g. you might need to keep something in deprecated/. Also, if you want to delete sources for blocks you know you will never, ever, need, you can do that here as well):
  - documentation/

  - samples/
     resources/javadoc/
  - in webapp/ everything except WEB-INF/ and sitemap.xmap
  - blocks/\*/samples/
  - deprecated/
- 5. Build it (to make sure it builds)
  - build.sh webapp
- 6. Clean it. We only want to control sources not derived objects (e.g. class files) build.sh clean
- 7. Import the drop.
  - For the initial drop (e.g., 2.1.6):

```
setenv cocoon_svn file:///var/svn/repos/cocoon
                                                  # (example
setenv cocoon_new_release cocoon-2.1.6 # (example)
cd /usr/local/Cocoon/drop
svn import -m "Initial drop (cocoon-2.1.6)" $cocoon_svn/dist/$cocoon_new_release
```

For subsequent drops:

```
setenv cocoon_svn file:///var/svn/repos/cocoon
                                                  # (example)
setenv cocoon_new_release cocoon-2.1.7
                                        # (example)
setenv cocoon_prev_release cocoon-2.1.6
                                           # (example)
cd /usr/local/Cocoon/drop
svn_load_dirs.pl -t $cocoon-new-release \
   $cocoon svn/dist \
   current \
   ./$cocoon-new-release
```

8. Create the new main branch and check it out:

```
svn copy -m "Create main branch" \
    $cocoon_svn/dist/$cocoon-new-release \
    $cocoon_svn/main/$cocoon-new-release

cd /usr/local/Cocoon/builds # (create this if it's the first time)
svn checkout $cocoon_svn/main/$cocoon-new-release
cd $cocoon-new-release
```

9. Merge in changes from the previous main branch (only if this isn't the initial drop!  $\stackrel{ }{ \bigcirc }$ 

```
svn merge
    $cocoon_svn/dist/$cocoon-prev-release
    $cocoon_svn/main/$cocoon-prev-release
```

- 10. Resolve any conflicts and commmit;
- 11. Build Cocoon
- 12. Delete /usr/local/Cocoon/drop

## Reader comments

Add 'em here, or email me --ML