

CocoonReleaseHowTo

This info is for Cocoon committers who need to prepare a new release of Cocoon.

See also

- [TestingBeforeRelease](#)
- [CocoonWebsiteUpdate](#)
- [LatestRelease](#) - please update that page when a release is done
- To update the live demo on cocoons.apache.org, see the instructions at <https://svn.apache.org/repos/private/pmc/cocoon/cocoon.zones.apache.org/admin-notes/howto-upgrade-the-release-demo.txt>

HOW TO DISTRIBUTE AN APACHE COCOON RELEASE

Notes on how to prepare a Cocoon release, by [PierFumagalli](#) on cocoon-dev, extended with a step by step procedure by [CarstenZiegeler](#).

A) The Release Process

Code freeze

Prior to the release day, a code freeze should be announced approx. 7 days in advance.

Announce the release process

Send a simple mail to the dev list. This is in order to ensure that no one will check in during the release process. If someone checks in during the building, testing and tagging, the release process has to be started at the beginning again.

Get the version

Check-out a fresh copy from the svn on a Unix platform (this is important, do not use windows!)

Set correct version information

Change the version information in `src/java/org/apache/cocoon/cocoon.properties` by removing "-dev" and eventually add new release information like m1, b1 etc. If this release is a final version, change the "released.version" info to the new version as well. Change `status.xml` by adding the release with proper version and date. Also change the version accordingly in `forrest.properties`. But don't check-in this yet!

Exclude unstable blocks from the default build

Edit the `blocks.properties` file and exclude all unstable blocks. Since it's a release they should not get compiled by default.

Build the distribution

Make sure that you make a clean build and that you are really not using windows

```
# ./build.sh clean-dist
# ./build.sh dist
```

Test the distribution

Uncompress the build distribution and test it. These tests should include different platforms (windows and Unix) and different JDKs (1.4 and 1.5). Testing includes building Cocoon from the release version and trying out the samples. Please also run 'build test'.

Decide what to do next

If any problem occurs during building and/or testing, you have to decide whether this is a blocker and has to be fixed or if this problem can be ignored. If it is a blocker, the problem must be fixed and after that you have to restart at the beginning. This decision is a from case to case decision and it's the release manager who decides if it's a blocker or not 😊

Continue with the release

Now check-in the changed version from the first step and tag the release in svn - if svn externals are used (for example in `src/blocks`) make sure to change the references and let them point to a specific revision.

Sign the release

Therefore you have to put your public key in the KEYS file before! In addition create a md5 file for the distribution

Start the formal vote

On the dev@c.a.o public list. Wait for 72 hours and 3 bindings +1

Start next version

Check the version in `src/java/org/apache/cocoon/cocoon.properties` by increasing the version information and adding "-dev" at the end, for example m3-dev. Change `status.xml` by starting a new release section with the placeholders. Add a dummy action here. Also change the version in `forrest.properties`

Bug Tracking

Enter the new version into Jira, so users can file entries against it.

Upload the release

Upload the release and the signatures to `www.apache.org` and put it under `/www/www.apache.org/dist/cocoon` into the correct directories (sources and binaries). Make sure that you set the permissions correctly. It's important to give the group write access! Add/change the links from the cocoon directory, and update `HEADER/README` in subversion as described below in **Publishing and Linking**.

End code freeze

Announce to the dev list that the release process is finished and end a possible code freeze.

Update the site docs

Change the `mirror.html` in the `cocoon-site` module and update this single file on the website. Update rest of the site as described in the [CocoonWebsiteUpdate](#)

Take a break...

Now wait for 24h so the mirror sites can pick up the new version. You can have a look at <http://www.apache.org/mirrors/> to see the status of the mirrors.

Create the announcement mail

Currently this is hand-typed (or copied) - we have to reinstall the announcement target. Send this email to wherever appropriate. Currently it goes to (dev at cocoon.apache.org, users at cocoon.apache.org, general at xml.apache.org, announce at apache.org, and announcements at xml.apache.org)

Register final version

Enter a final version (no betas or milestones) into freshmeat.

Enter new version [LatestRelease](#)

Update the **DOAP project file** at <http://svn.apache.org/repos/asf/cocoon/site/site/doap.rdf> to reflect the latest version, so that <http://projects.apache.org/> is up to date

😊 Naming Conventions:

The name of each downloadable archive should be named after the following regular expression:

```
PROJECT(-SUBPROJECT)?-VERSION(-VARIANT)?-(src|bin).(zip|tar.gz)
```

Where:

- PROJECT is the name of the top-level project,
- SUBPROJECT is the name of the eventual subproject (optional)
- VERSION is a version string (without - dashes)
- VARIANT identifies the "type of distribution" (ex. win32, jdk12, linux, jdk14...) (optional)

So, in Cocoon's case, all our archives shall be called something like:

```
cocoon-2.0.4-jdk14-bin.tar.gz
cocoon-2.1m1-src.zip
```

and so on...

If one day we ever released some subproject code (for example Lenya), the name should be along the lines of

```
cocoon-lenya-1.0-bin.tar.gz
```

You get the point.

C) Directories:

The only (ONLY) place where distributions shall reside is inside the /www/www.apache.org/dist/cocoon on people.apache.org.

This directory contains three subdirectories:

- BINARIES: this is where the binary distribution tar/zipballs are located.
- SOURCES: this is where the source distribution tar/zipballs are located.
- events: materials from the events.

In the future, when other subprojects of Cocoon will start putting out their own releases, new directories will be created under the top level directory, with the same structure. For example the final directory will look something like:

```
{{{www/www.apache.org/dist/cocoon/
```

+ - BINARIES/

+ - cocoon-2.0.3-bin.tar.gz
- cocoon-2.0.4-bin.tar.gz

+ - SOURCES/

+ - cocoon-2.0.3-src.tar.gz
+ - cocoon-2.0.4-src.tar.gz
- cocoon-2.1.0-src.tar.gz

+ - cocoon-latest-bin.tar.gz -> BINARIES/cocoon-2.0.4-bin.tar.gz

+ - cocoon-latest-src.tar.gz -> SOURCES/cocoon-2.0.4-src.tar.gz

+ - lenya/

+ - BINARIES/	
	- cocoon-lenya-1.0-bin.tar.gz
+ - SOURCES/	

	- cocoon-lenya-1.0-src.tar.gz
+ cocoon-lenya-latest-bin.tar.gz -> BINARIES/cocoon-lenya-1.0-bin.tar.gz	
- cocoon-lenya-latest-src.tar.gz -> SOURCES/cocoon-lenya-1.0-src.tar.gz	

- license.txt
 }}}

D) Publishing and linking:

Once the distribution ball is rolled following the naming convention and copied in the appropriate directory as outlined above, make sure that the following links are present (and only those links) in the root directory for the project/subproject:

- A link to the latest released stable version for each tar/zipball: for example, if the latest release is 2.0.4, and this release includes 3 different balls, 2 for binaries and 1 for sources, the following links must be done:

```
# cd /www/www.apache.org/dist/cocoon
# ln -s BINARIES/cocoon-2.0.4-vm14-bin.tar.gz cocoon-2.0.4-vm14-bin.tar.gz
# ln -s BINARIES/cocoon-2.0.4-vm14-bin.zip cocoon-2.0.4-vm14-bin.zip
# ln -s BINARIES/cocoon-2.0.4-bin.tar.gz cocoon-2.0.4-bin.tar.gz
# ln -s BINARIES/cocoon-2.0.4-bin.zip cocoon-2.0.4-bin.zip
# ln -s SOURCES/cocoon-2.0.4-src.tar.gz cocoon-2.0.4-src.tar.gz
# ln -s SOURCES/cocoon-2.0.4-src.zip cocoon-2.0.4-src.zip
```

- A link to the latest milestone version for each tar/zipball (if present): for example, if 2.1m1 publishes only one ball, the source one:

```
# cd /www/www.apache.org/dist/cocoon
# ln -s SOURCES/cocoon-2.1-src.tar.gz cocoon-2.1-src.tar.gz
# ln -s SOURCES/cocoon-2.1-src.zip cocoon-2.1-src.zip
```

- A link to the LATEST STABLE DOWNLOADABLE with the "version" string replaced by the "latest" keyword. If the above example of 2.0.4 is still valid:

```
# cd /www/www.apache.org/dist/cocoon
# ln -s BINARIES/cocoon-2.0.4-vm14-bin.tar.gz cocoon-latest-vm14-bin.tar.gz
# ln -s BINARIES/cocoon-2.0.4-vm14-bin.zip cocoon-latest-vm14-bin.zip
# ln -s BINARIES/cocoon-2.0.4-bin.tar.gz cocoon-latest-bin.tar.gz
# ln -s BINARIES/cocoon-2.0.4-bin.zip cocoon-latest-bin.zip
# ln -s SOURCES/cocoon-2.0.4-src.tar.gz cocoon-latest-src.tar.gz
# ln -s SOURCES/cocoon-2.0.4-src.zip cocoon-latest-src.zip
```

Make sure to change the latest version strings and URLs in the README.html and HEADER.html files in the <https://svn.apache.org/repos/asf/cocoon/site/mirrors/> and use 'svn up' command in /www/www.apache.org/dist/cocoon/ directory to update.

Update the mirror.html file in subversion to link to latest files, and update the /www/cocoon.apache.org/ directory with these changes.

E) Mirroring and announcing:

Once the release is published **wait at least 24 hours before announcing it** to the mailing lists, so that mirror sites will have the opportunity to pick the changes up and you won't get bugged by people unable to download the distributions.

Remember that the locations to mention in any announcements when downloads are involved is

```
http://cocoon.apache.org/mirror.cgi
```

So that people will actually **use** the mirrors and not peg the Foundation's bandwidth (which is quite expensive).

Have fun...

Pier