YourCocoonBasedProjectAnt16

This page is a HOW-TO for setting up your own Cocoon Based project with Ant 1.6

Credits go to MarcPortier for the content and the Ant build script (and to StenAkselHeien for small page and script updates).

See also:

- YourCocoonBasedProject, works in a similar way but uses ant 1.5.
- BricksCms which uses a slightly modified version of this for its build

One important reason for using this script is that it completely separates your project code from the Cocoon distribution. This allows you to update the Cocoon distribution without running the risk of deleting your own project or of overwriting modifications of Cocoon specific files such as the root sitemap or cocoon.xconf.

1. get the Cocoon distribution (if you don't already have it)
2. make project directory and download build-cocoon-targets.xml in the directory
3. edit build-cocoon-targets.xml and replace project name "Your{{`Cocoon}}Based`Project" with your project name
4. 'seed' the project
   a. `ant -f build-cocoon-targets.xml -Dcocoon.distro.home=/path/to/your/cocoon/distribution`
   - build.xml have been generated for you with your project name
   - Note: the cocoon.distro.home is now set in user.properties
   - Note for windows users: set the slashes as forward slashes
5. edit build-cocoon-targets.xml (again) and add "Targets" to your project name
   - build-cocoon-targets.xml is imported by the build.xml file and the two projects can't have the same name.
6. customize the src/cocoon/local.build.properties
   a. **important: remove the path variables in this file** (ant cannot substitute variables in property files, so ${...} variables would cause build errors). Remove sections build, build webapp, src, standalone demo, dir layout, tools, deprecated, ide, lib, dist, site, legal, gump, so that there are no more ${...} and variables in this file.
   b. set include to false for the blocks you don't need in the top part of local.build.properties (remove # in front of line)
   c. exclude the samples and documentation too if you don't need them (time consuming build process)(remove # in front of line).
7. `ant cocoon:get`
   - -> repeat this step later when changing the distribution or the settings in local.build.properties
8. `ant webapp`
   - -> builds the Java classes and copies the entire project to the tools tree
   - Note: classes are built with debug flag on.
9. `ant cocoon:run`
   - starts Jetty
   - (you need to restart if you have done any of the following: changing the libs, java compile)
10. http://localhost:8888/
11. cvs check-in to check your (now still empty project) in the CVS
12. add your project to Eclipse
    a. set src to src/java
    b. set output folder to build/classes
13. use XPatchTaskUsage to modify the Cocoon configuration files.
    - -> Note: you can use cocoon:unpatch to revert the configuration files back to the original versions.

Once this is setup use the src/java tree for your own customized Java components and src/cocoon/webapp/Your{{`Project}}`Name for your Cocoon related files (sitemap, XSL and XML files etc.). Ant webapp will compile your classes and copy the files in the src tree to the appropriate places in the tools tree. Jetty (ant cocoon:run) will use the files in the tools tree.

You can reach your project under ht{{`tp://localhost:8888/Your}}`Project`Name

Attachment: build-cocoon-targets.xml