This content came from

<http://www.jajakarta.org/turbine/en/turbine/maven/reference/dirlayout.html>

**Rationale**

Having a common directory layout would allow for users familiar with one Maven project to immediately feel at home in another Maven project. The advantages are analogous to adopting a site-wide look-and-feel. Common Maven procedures strengthen the Maven brand.

The directory structure proposed here is based on best-practices as suggested by Jakarta committers. We recognize that the cost of changing an existing directory structure can be high. As such, this proposal is not binding. However, integrating Maven into a project that does not follow the recommendations in this proposal will require additional integration work.

The next two sections document the directory layout expected by Maven and the directory layout created by Maven. Please try to conform to this structure as much as possible; however, if you must these settings can be overridden via properties as documented in the [Integration](http://www.jajakarta.org/turbine/en/turbine/maven/start/integrate.html#Properties) document.

<section containing directory layout expected by Maven omitted here>

**Maven-Generated Layout**

| **Directory name** | **Content** | **Comment** |
| --- | --- | --- |
| target/ | Contains compiled classes and JARs. | The contents of the target/ directory should be enough to use the project. This directory contains the final JARs that are generated. |
| target/classes | Contains compiled classes and JARs. | The target/classes directory contains all compiled classes. This directory is used when packaging the final JAR for a project. |
| target/test-classes | Contains compiled unit test classes. | The target/test-classes directory contains all compiled unit test classes. This directory is used when executing the unit tests. |
| target/iutest-classes | Contains compiled integration unit test classes. | The target/iutest-classes directory contains all compiled integration unit test classes. This directory is used during the execution of the integration unit tests. |
| target/generated-docs | Contains Maven generated xdocs. | The target/generated-docs directory contains all of the Maven-generated xdocs. All content generated by Maven is first converted to xdoc format, so the same stylesheet used to transform the rest of the site, can be used on generated content. The contents of this directory are transformed and stored in the docs/ directory. |
| target/test-reports | Contains the individual unit test results. | The test-reports/ directory contains individual unit test reports in both XML and plain text formats. The reports in this directory is used when Maven creates the final unit test report. |
| target/docs | Documentation files intended for the website publication. | The docs/ directory contains only generated documentation that is intended to be published as the project's website. This directory includes the Velocity/DVSL generated HTML files, JavaDocs, cross-referenced sources, and various generated reports. Generally, all documentation is stored in the xdocs/ directory and then "transformed" into this directory. The specific documents that Maven generates are described below. |
| target/docs/index.html | Starting point for browsing the documentation. | Browsing the documentation locally should yield the same results as browsing the documentation on the project's home page. If a project does not provide an index.xml in the xdocs/ directory, Maven will automatically generate a simple front page based on the [description](http://www.jajakarta.org/turbine/en/turbine/maven/reference/project-descriptor.html#project) element in the project descriptor. |
| target/docs/apidocs | API documentation. | Maven automatically generates JavaDocs for projects using the JavaDoc utility. Placing the API documentation under docs/apidocs/ makes it slightly easier for other documentation files under docs/ to reference API documentation and vice versa. |
| target/docs/xref | Cross-referenced source code. | Maven automatically generates cross-referenced source code that enables easy browsing of an entire source tree. Placing the cross-referenced sources under docs/xref/ makes it slightly easier for other documentation files under docs/ to reference API documentation and vice versa. |
| target/docs/mail-lists.html | Mailing list documentation. | Maven automatically generates a list of mailing lists based on the information provided in the [project descriptor](http://www.jajakarta.org/turbine/en/turbine/maven/reference/project-descriptor.html#mailingLists). |
| target/docs/team-list.html | The list of project team members. | Maven automatically generates a list of project team members based on the information provided in the [project descriptor](http://www.jajakarta.org/turbine/en/turbine/maven/reference/project-descriptor.html#developers). |
| target/docs/dependencies.html | The list of dependencies. | Maven automatically generates a list of dependencies based on the information provided in the [project descriptor](http://www.jajakarta.org/turbine/en/turbine/maven/reference/project-descriptor.html#dependencies). |
| target/docs/changelog.html | The CVS change log. | Maven automatically generates a change log from CVS log messages. This log is currently limited to the past 5 days (but will be configurable in the future). |
| target/docs/jdepend-report.html | Metric report. | Maven automatically generates a report on various [source code metrics](http://www.jajakarta.org/turbine/en/turbine/maven/reference/metrics.html). This report can provide further insight into a project. |
| target/docs/junit-report.html | Unit test report. | Maven automatically generates a report on the results of unit testing. This report provides a confidence level for users of your project. |
| target/docs/checkstyle-report.html | Checkstyle report. | Maven automatically generates a report on the results of Checkstyle. This report provides assurance that the coding conventions for your project are being followed. |