**Introduction to the Standard Directory Layout**

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.

The next section documents 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 can't these settings can be overridden via the project descriptor.

|  |  |
| --- | --- |
| src/main/java | Application/Library sources |
| src/main/resources | Application/Library resources |
| src/main/filters | Resource filter files |
| src/main/webapp | Web application sources |
| src/test/java | Test sources |
| src/test/resources | Test resources |
| src/test/filters | Test resource filter files |
| src/it | Integration Tests (primarily for plugins) |
| src/assembly | Assembly descriptors |
| src/site | Site |
| LICENSE.txt | Project's license |
| NOTICE.txt | Notices and attributions required by libraries that the project depends on |
| README.txt | Project's readme |

At the top level files descriptive of the project: a pom.xml file In addition, there are textual documents meant for the user to be able to read immediately on receiving the source: README.txt, LICENSE.txt, etc.

There are just two subdirectories of this structure: src and target. The only other directories that would be expected here are metadata like CVS, .git or .svn, and any subprojects in a multiproject build (each of which would be laid out as above).

The target directory is used to house all output of the build.

The src directory contains all of the source material for building the project, its site and so on. It contains a subdirectory for each type: main for the main build artifact, test for the unit test code and resources, site and so on.

Within artifact producing source directories (ie. main and test), there is one directory for the language **java** (under which the normal package hierarchy exists), and one for resources (the structure which is copied to the target classpath given the default resource definition).

If there are other contributing sources to the artifact build, they would be under other subdirectories: for example src/main/antlr would contain Antlr grammar definition files.

<http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>