Usage

You can put additional content (e.g. documentation, resources, etc.) in your site. See Creating Content for more information on this. If you want to change the menus, breadcrumbs, links or logos on your pages you need to add and configure a site descriptor. If you like, you also can let Maven generate some reports for you, based on the contents of your POM.

- Generating a Site
- Deploying a Site
- Staging a Site
- Running a Site

Generating a Site

To generate the project's site and reports, execute:

```
mvn site
```

By default, the resulting site will be in the `target/site/` directory.

Note: If you have a multi module project, then the links between the parent and child modules will *not* work when you use '`mvn site`' or '`mvn site:site`'. If you want to use those links, you should use '`mvn site:stage`' instead. You can read more about that goal further down on this page in the section called '*Staging a Site*'.

Note: For performance reasons, Maven compares the timestamps of generated files and corresponding source documents, and only regenerates documents that have changed since the last build. However, this only applies to documentation source documents (apt, xdoc,...). If you change anything in your `site.xml`, any relevant sections in your pom, or any relevant properties or resource files, you should generate the site from scratch to make sure all references and links are correct.

Deploying a Site

To be able to deploy the site, you must first specify where the site will be deployed. This is set in the `<distributionManagement>` element of the POM as shown below.

```
<project>
  ...
  <distributionManagement>
    <site>
      <id>www.yourcompany.com</id>
      <url>scp://www.yourcompany.com/www/docs/project/</url>
    </site>
  </distributionManagement>
  ...
</project>
```

The `<id>` element identifies the repository, so that you can attach credentials to it in your `settings.xml` file using the `<servers>` element as you would for any other repository.

The `<url>` gives the location to deploy to. In the example above we copy to the host `www.mycompany.com` using the path `/www/docs/project/` over the `scp` protocol. You can read more about which protocols are supported on this page. If subprojects inherit the site URL from a parent POM, they will automatically append their `<artifactId>` to form their effective deployment location.

Now you can execute the `site:deploy` goal from your project directory.

Note: A site must be generated first before executing `site:deploy`.

```
mvn site:deploy
```

If you want to generate the site *and* deploy it in one go, you can utilize the `site-deploy` phase of the site lifecycle. To do this, just execute:

```
mvn site-deploy
```

Staging a Site

Note: This goal is available in version 2.0-beta-5 or later of the Site Plugin.

To review/test the generated web site before an official deploy, you can stage the site in a specific directory. It will use the `<distributionManagement>` element or the project hierarchy to link the project and its modules.

Just execute the `site:stage` goal from your project

```
mvn site:stage
```

Note: Since version 2.3, a site must be generated first before executing `site:stage`.

By default, the site will be staged in a directory `target/staging/`. A different staging location can be chosen with the `stagingDirectory` parameter as shown below:

```
mvn site:stage -DstagingDirectory=C:\fullsite
```

Note: `stagingDirectory` cannot be dynamic, i.e. `stagingDirectory=${basedir}\fullsite`

To stage a site and to deploy it, just execute the `site:stage-deploy` goal from your project with the required parameters. The `site:stage-deploy` goal will use the value of `distributionManagement.site.id` as default id to lookup the server section in your `settings.xml`; unless this is not defined, then the String `stagingSite` will be used as id. So if you need to add your username or password separately for stage-deploy in `settings.xml`, you should use `<id>stagingSite</id>` for that `<server>` section. See the Guide to Deployment and Security Settings for more information on this.

By default, the site will be stage-deployed to `$distributionManagement.site.url/staging/`. A different location can be chosen with the `stagingSiteURL` parameter as shown below:

```
mvn site:stage-deploy -DstagingSiteURL=scp://www.mycompany.com/www/project/
```

Note: Since version 2.3, a site must be generated first before executing `site:stage-deploy`.

Note: Due to a bug in Wagon, the password is not always picked up when you run the `site:stage-deploy` goal. The bug has been fixed, but the version of Wagon that is used by the Site Plugin is determined by the version of Maven you use. The current 2.0.x releases of Maven use a version where this bug is still present.

Running a Site

The Site Plugin can also be used to start up the site in Jetty. To do this, execute:

```
mvn site:run
```

The server will, by default, be started on `http://localhost:8080/`. See http://jetty.mortbay.org/ for more information about the Jetty server.

Note: Running a site only works for single-module sites. To preview a multi-module site one should use site:stage.