

Configuring the Site Descriptor

You can create your own site descriptor for your project when you want to override the navigation tree for the site. For example, aside from the generated reports you want to add additional content to your site. In order for it to be accessible in the generated site, you must configure your site descriptor. You create the site descriptor in a file called `site.xml` which should be located in your `src/site` directory.

If you are migrating from Maven 1.x you will be interested to know that the `navigation.xml` file has been replaced by the site descriptor in Maven 2. The format is nearly the same, so you can probably reuse most of your old file.

There is a [reference documentation for the site descriptor](#) available.

The XML Schema for the site descriptor is published at <http://maven.apache.org/xsd/decoration-1.6.0.xsd>.

Have a look at the [site descriptor for Maven Site Plugin](#) for a real life example.

- [Title](#)
- [Banner](#)
- [Publish Date](#)
- [Version](#)
- ["Powered by" Logo](#)
- [Inheritance](#)
- [Including Generated Content](#)
- [Inject xhtml into <head>](#)
- [Links](#)
- [Breadcrumbs](#)
- [Custom footer](#)
- [Custom content](#)
- [Skinning](#)
- [Custom Properties](#)
- [Expressions](#)

Title

The title of each generated page will be a combination of the *site title* and the title of the current page. By default the Site Plugin will use the value of the `<name>` element from your `pom.xml` file as the site title. The complete title for this page is "Maven Site Plugin - Configuring the Site Descriptor".

If you want to use a different site title, but do not want to change the `<name>` element in your `pom.xml`, you can configure this in your `site.xml`, like this:

```
<project name="My Site Title">
  ...
</project>
```

Banner

You can include some logos on top of your site, using the following syntax:

```
<project>
  ...
  <bannerLeft>
    <name>Project Name</name>
    <src>http://maven.apache.org/images/apache-maven-project-2.png</src>
    <href>http://maven.apache.org/</href>
  </bannerLeft>

  <bannerRight>
    <src>http://maven.apache.org/images/maven-logo-2.gif</src>
  </bannerRight>
  ...
```

```
</project>
```

Refer to the site descriptor [reference](#)  for a complete tag description.

Publish Date


With the out-of-the-box Velocity template, the position of the "Last Published" date is configurable. By default, the position is on the left but you can change it. To do this, you can add a `<publishDate>` element to your `site.xml` like the following:

```
<project>
...
<publishDate position="right"/>
...
</project>
```

The `position` attribute can have one of these values: `left`, `right`, `navigation-top`, `navigation-bottom`, `bottom`.

If you want hide the publish date, you can use this in your `site.xml`:

```
<project>
...
<publishDate position="none"/>
...
</project>
```

The format of the "Last Published" date is the ISO date format that is [recommended by the W3C](#) . Because the web has an international, cross-cultural audience it is recommended to *not* change the date format. There is however a `format` attribute to the `<publishDate>` element that can be used if you really need a different date format.

Version

You can show the "Version" of your project on the site, by adding a `<version>` element to your `site.xml` like this:

```
<project>
...
<version position="right"/>
...
</project>
```

The `position` attribute can have the same values as the `publishDate` attribute, see above. If the `position` attribute is omitted, the default value is `left`.

If you want hide the version, you can use this in your `site.xml`:

```
<project>
...
<version position="none"/>
...
</project>
```

"Powered by" Logo

You can add your own "Powered by" logo to your site. To do this, you add a `<poweredBy>` element in your `site.xml`

like this:

```
<project>
...
<poweredBy>
  <logo name="Maven" href="http://maven.apache.org/"
    img="http://maven.apache.org/images/logos/maven-feather.png"/>
</poweredBy>
...
</project>
```

➔ Inheritance

See [Building multi-module sites](#).

➔ Including Generated Content

Files in the format-based directory structure can be linked to by their target HTML filename, e.g. `$basedir/src/site/apt/foo.apt` and `$basedir/src/site/fml/faq.fml` can be linked to via:

```
<project>
...
<body>
...
  <menu name="Overview">
    <item name="Foo" href="foo.html" />
    <item name="FAQ" href="faq.html" />
  </menu>
...
</body>
...
</project>
```

There are also several preset menus that can be used in the site descriptor to include generated content from your project. These are linked via the `ref` attribute, like so:

```
<project>
...
<body>
...
  <menu ref="modules" />
...
</body>
...
</project>
```

Note: The old syntax using `${reports}`, `${parent}` and `${modules}` has been deprecated and you are encouraged to use the new syntax instead. The support for the old syntax will be removed in a future version of the Site Plugin.

The currently available preset menus are:

- `reports` - a menu with links to all the generated reports for your project
- `parent` - a menu with a link to the parent project's site
- `modules` - a menu containing the links to the sites of the submodules of this project

➔ Inject xhtml into <head>

You can inject some xhtml into the generated `<head>` element of each page by adding a head element to the body element of your project's site descriptor. The following example adds some javascript:

```
<project>
...
<body>
...
<head>
  <script src="http://www.google-analytics.com/urchin.js" type="text/javascript" />
</head>
...
</body>
...
</project>
```

➔ Links

To add links below your site logo, just add a links element to the `<body>` element of the site descriptor. Each item in the links element will be rendered as a link in a bar directly below your project's logo.

```
<project>
...
<body>
...
<links>
  <item name="Apache" href="http://www.apache.org"/>
  <item name="Maven" href="http://maven.apache.org"/>
</links>
...
</body>
...
</project>
```

➔ Breadcrumbs

If there exists a logical hierarchy within your site modules, you may want to generate a series of breadcrumbs to give a way to easily navigate the project tree.

To configure breadcrumbs, add a `<breadcrumbs>` element to the `<body>` element in the site descriptor. Each item element will render a link, and the items in the `<breadcrumbs>` element will be rendered in order. The breadcrumb items should be listed from highest level to lowest level.

```
<project>
...
<body>
...
<breadcrumbs>
  <item name="Doxia" href="http://maven.apache.org/doxia/index.html"/>
  <item name="Trunk" href="http://maven.apache.org/doxia/doxia/index.html"/>
</breadcrumbs>
...
</body>
...
</project>
```

Note that in a multi-module build, if the parent contains a breadcrumb and the inheriting site descriptor doesn't, a breadcrumb with the current site descriptor's name will be added automatically. See the notes on [building a multi-module site](#) for more details.

Custom footer

You can replace the auto-generated footer content by specifying a custom `<footer>` element:

```
<project>
...
<body>
...
<footer>All rights reserved.</footer>
...
</body>
...
</project>
```

Custom content

There is also a dummy `<custom>` element then can be used to specify some arbitrary content. Note that you need to write your own velocity template to make use of this element, it is ignored by the default Velocity template used by the Site Plugin.

```
<project>
...
<custom>Custom content</custom>
...
</project>
```

Skinning

Skins can be created to customize the look and feel of a site in a consistent way. For more information on creating a skin, see [Creating a Skin](#). To use a specific skin in your project, you use the `<skin>` element of the site descriptor. This is a regular artifact or dependency-like element. For example, to use the [Maven Classic Skin](#), you would include:

```
<project>
...
<skin>
  <groupId>org.apache.maven.skins</groupId>
  <artifactId>maven-classic-skin</artifactId>
  <version>1.0</version>
</skin>
...
</project>
```

Note: The `<version>` element is optional and, like plugins, if omitted the latest version available will be used. It is recommended that you always specify a version so that your site is reproducible over time.

This skin will copy the necessary resources including CSS and if necessary use the included alternate Velocity template to render the site.

If you don't specify a skin, the Site Plugin will use [Maven Default Skin](#).

Custom Properties

The authors of skins have the option to use custom properties that are unique to their skin. The users of the skin can then specify their own values for these properties in their `site.xml` using the `<custom>` element.

One skin that uses this is the Maven Fluid Skin. There are many examples on [their site](#) showing what is

possible to do with custom properties. Here is one example:

```
<project>
...
<custom>
  <fluidoSkin>
    <topBarEnabled>true</topBarEnabled>
    <sideBarEnabled>>false</sideBarEnabled>
  </fluidoSkin>
</custom>
...
</project>
```

➔ Expressions

The `site.xml` can contain some expressions, like `${project.name}`. Each expression will be evaluated when the site is rendered.

Expressions can be:

- `${project.*}`, for instance `${project.organization.name}` referenced in `<project><organization><name>`
- `${project.properties}`, for instance `${myProperty}` referenced in `<project><properties><myProperty>`
- `${environmentVariable}`, for instance `${JAVA_HOME}`

Note: Support for some expressions, like `${project.name}` is present in version 2.0-beta-5 of this plugin. Full support is available since version 2.0-beta-6.