

Documentation Only

Just the docs, please.

By default, Enunciate compiles, packages and otherwise integrates itself with your Web service application. The reason for this is to make life easier for the basic developer who doesn't want to have to care about deployment descriptors and configuration.

But there are often cases where you're already compiling and packaging up your own endpoints, and you'd just like to have Enunciate generate it's documentation and client-side libraries. No problem.

Maven

As of Enunciate 1.10, you can use "docs" goal of the the Enunciate Maven Plugin. For example, the following POM snippet will generate all documentation (including client-side libraries) and put them in the "target/docs" directory.

pom.xml

```
<project ...>
  ...
  <build>
    <plugins>
      ...
      <plugin>

<groupId>org.codehaus.enunciate</groupId>

<artifactId>maven-enunciate-plugin</artifact
Id>
  <!-- check for the latest version
-->
  <version>1.17</version>
  <executions>
    <execution>
      <goals>
        <goal>docs</goal>
```

```
        </goals>
        <configuration>

            <!-- the directory where to
put the docs -->

<docsDir>${project.build.directory}/docs<doc
sDir>

            </configuration>
        </execution>
    </executions>
</plugin>
...
</plugins>
</build>
```

```
...  
</project>
```

Maven "Site" Inclusion

As of Enunciate 1.17, you can also use the Maven plugin to include the generated documentation as part of the generated Maven site:

pom.xml

```
<project ...>  
  ...  
  <reporting>  
    <plugins>  
      <plugin>  
  
<groupId>org.codehaus.enunciate</groupId>  
  
<artifactId>maven-enunciate-plugin</artifact  
Id>  
  <version>1.17</version>  
  <configuration>  
  
    <!-- The subdirectory of the site  
where the documentation will be put. -->  
    <!-- Note that if this isn't set,  
the documentation will overwrite the site.  
-->  
  
    <docsSubdir>apidocs</docsSubdir>  
  
    <!-- Alternatively, you can just
```

rename the Enunciate index page -->

```
<indexPageName>apidocs.html</indexPageName>  
  </configuration>
```

```
</plugin>
```

```
</plugins>
```

```
</reporting>
...
</project>
```

Non-Maven Users

The simple answer is that you export the "docs" artifact to a specified (existing) directory or to a file. (If you export it to a file, it will be exported as the zipped-up directory.)

Ant

If you using Ant, this exports the "docs" artifact to the `target/docs` directory.

build.xml

```
<enunciate basedir="src/main/java">
  <include name="**/*.java"/>
  <classpath refid="enunciate.classpath"/>
  <export artifactId="docs"
destination="target/docs"/>
</enunciate>
```