

override web.xml

The override web.xml file

The [webdefault.xml](#) file supplies configuration of web.xml file elements that would otherwise be repetitive and onerous to set up for every web application, such as setting up mime-type mappings, JSP servlet-mappings etc. It is applied **BEFORE** the webapp's own `WEB-INF/web.xml` file. Generally, it is convenient for all webapps deployed in a jetty instance to share the same webdefault.xml file, however, it is certainly possible to provide differentiated webdefault.xml files for individual web applications.

A common need is to be able to deploy the same *unchanged* webapp/war into different environments, but also to have that webapp customized for that environment. To some extent, changing configuration in the [jetty.xml](#) will help with this, as that file does not form part of the webapp, and so therefore can be changed per deployment.

However, there are some things you might want to change that [jetty.xml](#) cannot help you with: for example, servlet init-params and context init-params. Using webdefault.xml won't help you either, because it is applied *first* so cannot override values inside the webapps web.xml.

The solution is to use an **override web.xml** file. This is another *whole or partial* web.xml file that resides *externally* to the webapp, and is applied **AFTER** the webapp's `WEB-INF/web.xml` file and can therefore override or add new elements.

Configuring an override web.xml file for jetty standalone

There is an example of applying an override web.xml file in `$jetty.home/contexts/test.xml`. To apply an override to a single webapp:

```
<Configure
class="org.mortbay.jetty.webapp.WebAppContext"
...
  <Set
name="overrideDescriptor"><SystemProperty
name="jetty.home"
default="." />/my/path/to/override-web.xml</Set>
...
</Configure>
```

In code this is:

```
import
org.mortbay.jetty.webapp.WebAppContext;

...
WebAppContext wac = new WebAppContext();
...

//Set the absolute path to the override
descriptor.
//Alternatively, use the classloader to get
the path to the
//override descriptor as a resource.
wac.setOverrideDescriptor(System.getProperty
("jetty.home")+"/my/path/to/override-web.xml
");
```

Configuring an override web.xml file for the jetty maven plugin

Use the `<overrideWebXml>` element as follows:

```
<project>
  ...
  <plugins>
    ...
    <plugin>
      <groupId>org.mortbay.jetty</groupId>

      <artifactId>maven-jetty-plugin</artifactId>
      <configuration>
        ...

        <overrideWebXml>src/main/resources/override-
web.xml</overrideWebXml>
        ...
      </configuration>
    </plugin>
    ...
  </plugins>
</project>
```

There is more information on configuring the [jetty maven plugin here](#).