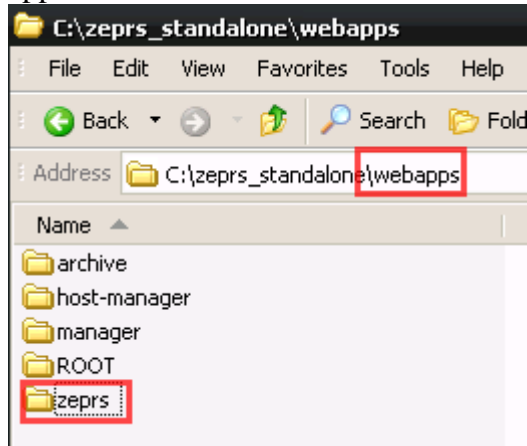


Creating a new Zcore-based Project

Chris Kelley
10/29/08

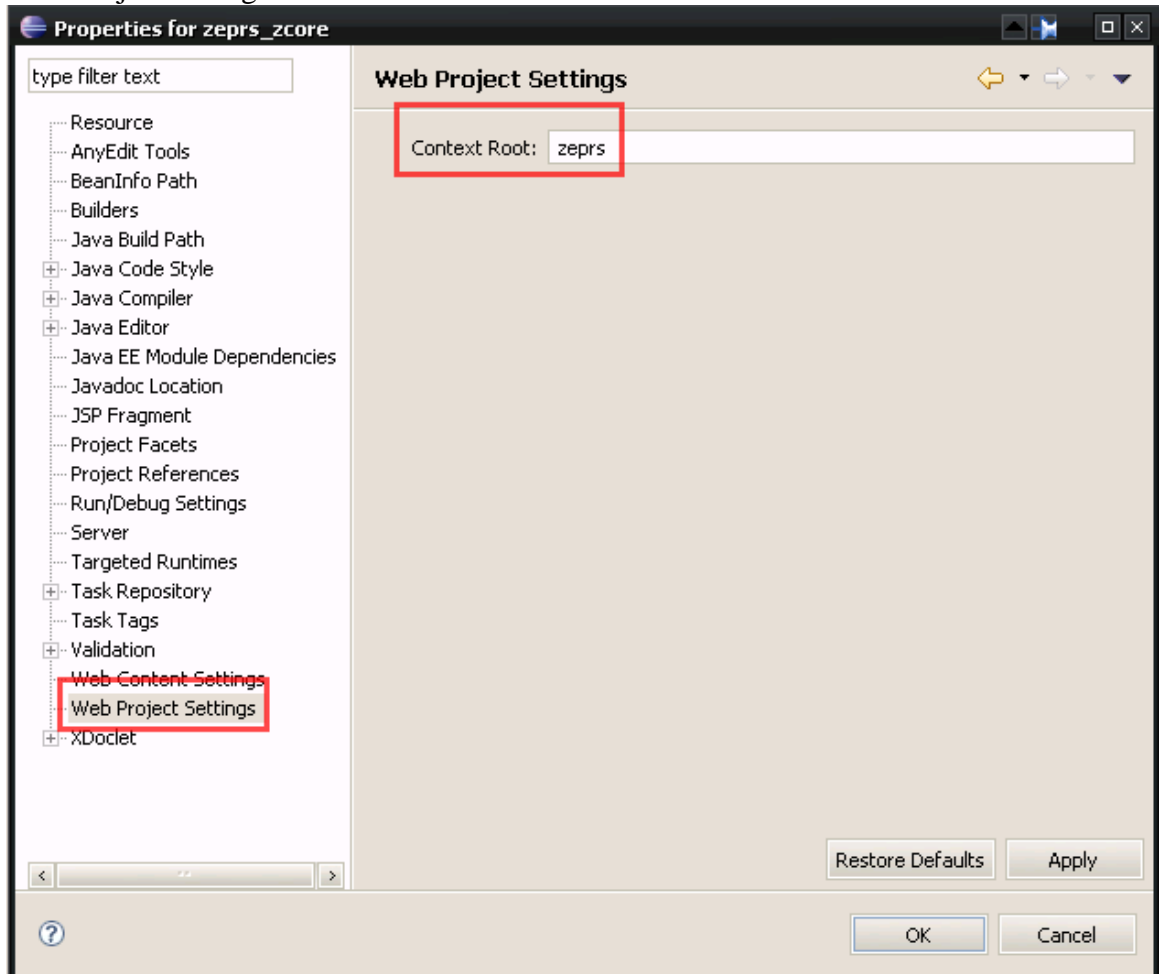
Here are the steps:

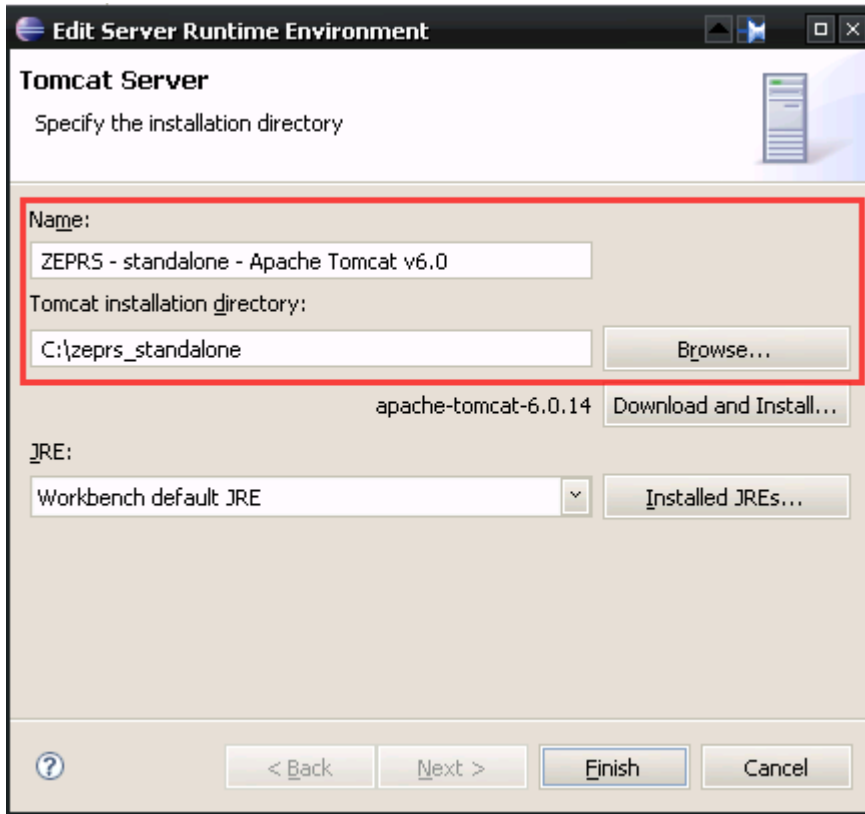
- In Eclipse, export zcore foundation to a new directory. Create a new directory in your source code dir. Right-click on the project name in Project Explorer ->Team ->"Export "... and browse to the new directory. The export process takes a while, but it is much better to use export than just copying the source folder because using "Export to..." removes all of the svn -related folders and files.
- Move this new project directory to the source code location of your choice.
- Change the name of the webapp to the deployment name, i.e., the name the web app when accessed from a web browser – <http://localhost/zeprs>:



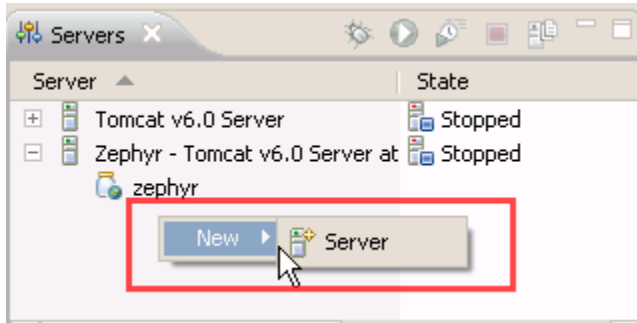
- Edit the .project file to the name of your project in a text editor.
- Edit .settings/ org.eclipse.wst.common.component – change the deploy-name: `<wb-module deploy-name="zeprs">`
This makes the correct name display when adding a web module.
- Reset buildNum=0 in version.properties
- Clear out all of the generated files from previous projects:
 - `src\zeprs\org\cidrz\project\zeprs\valueobject\gen`
 - `src\zeprs\org\cidrz\project\zeprs\valueobject\report\gen`
- Clear out all of the XML files in `src\zeprs\resources\xml\forms` except for:
 - Sites.xml
 - Clinics.xml
- Import the project into Eclipse.
- Copy the application directory, which contains the tomcat instance, eclipse rep code, database, and other useful files, to the deployment point (e.g. `C:\zeprs_standalone`).
- Setup the Eclipse server settings.

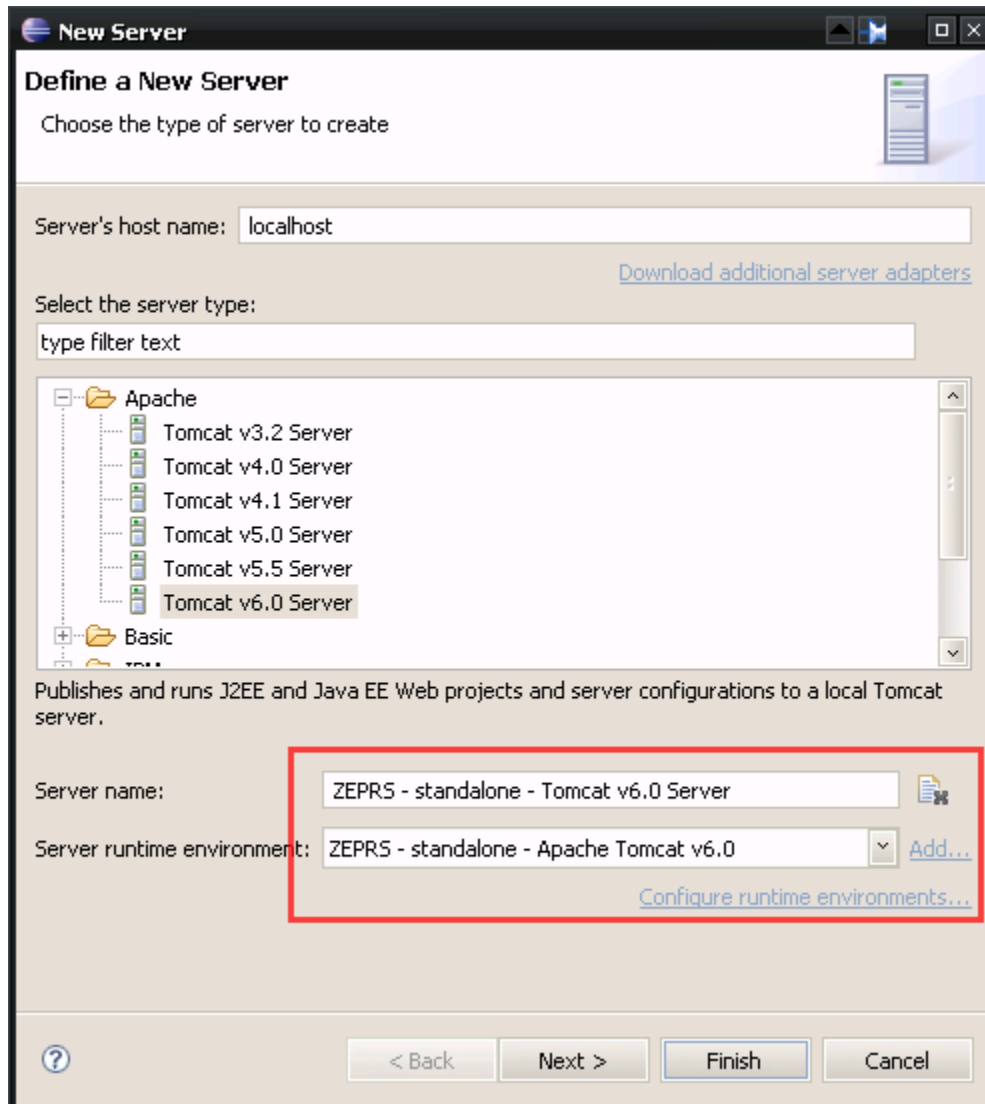
- Web Project settings:



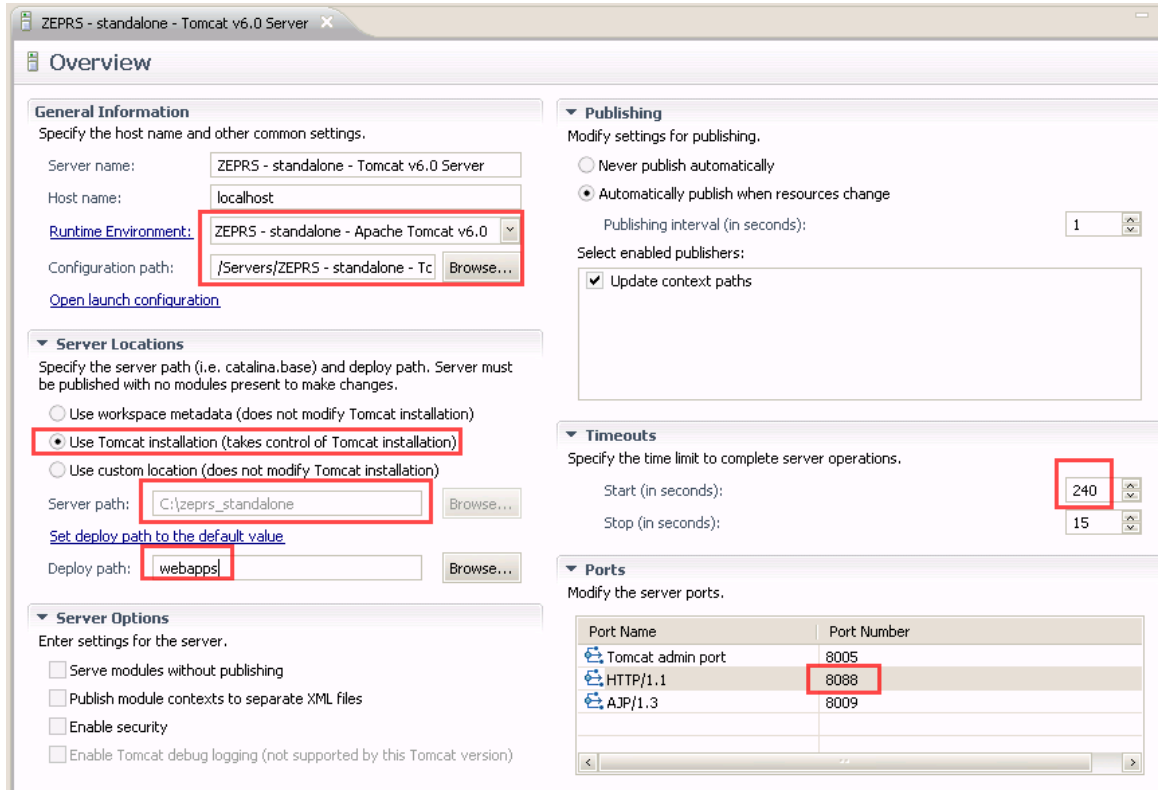


- Create a new Server in the Servers view:

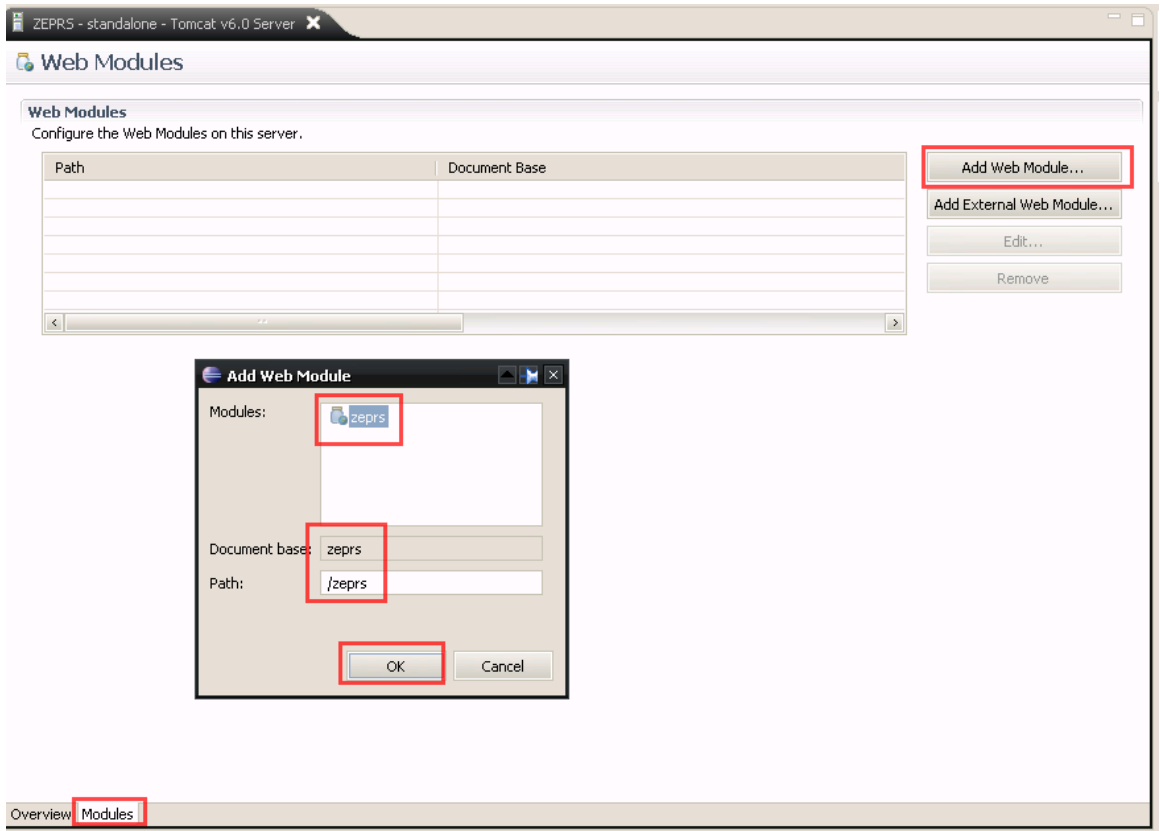




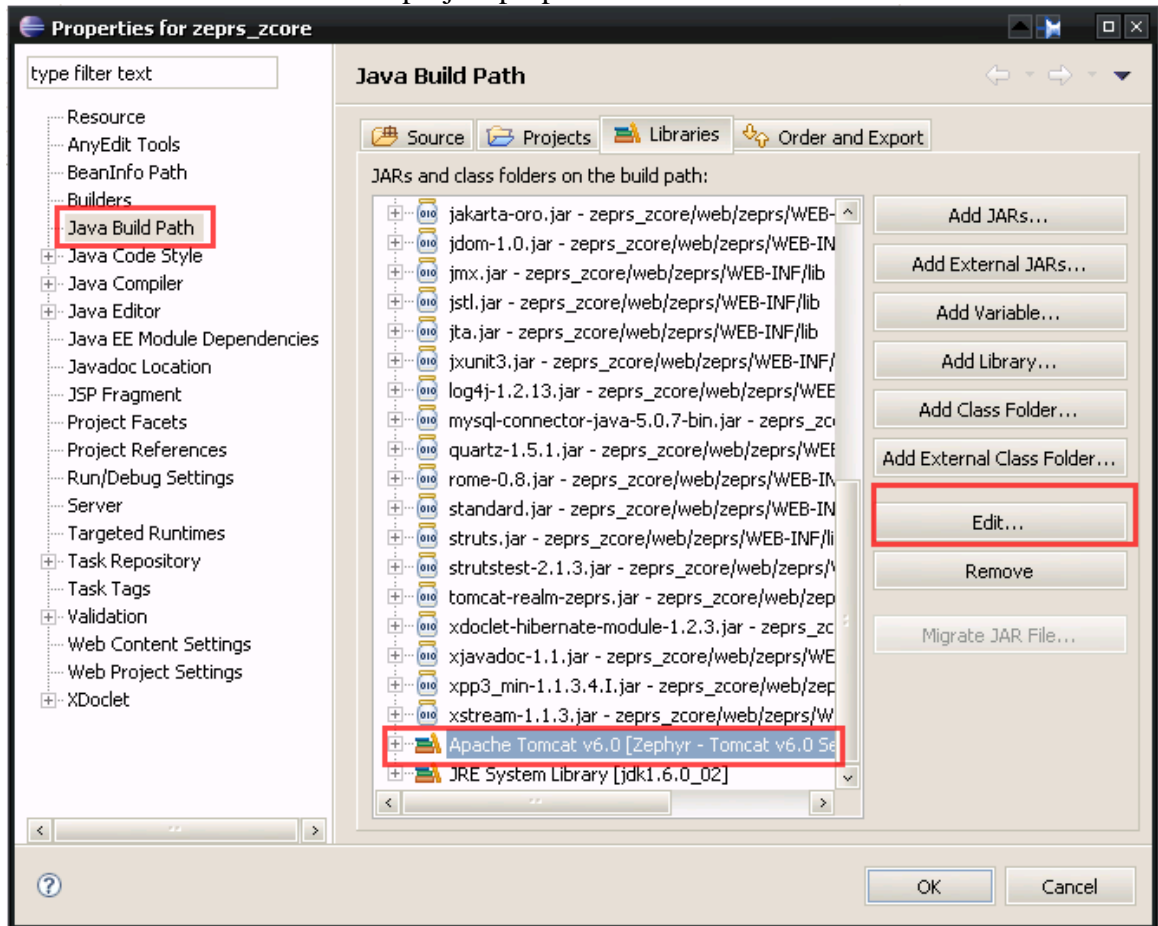
- Double-click on the new Server to edit its parameters:



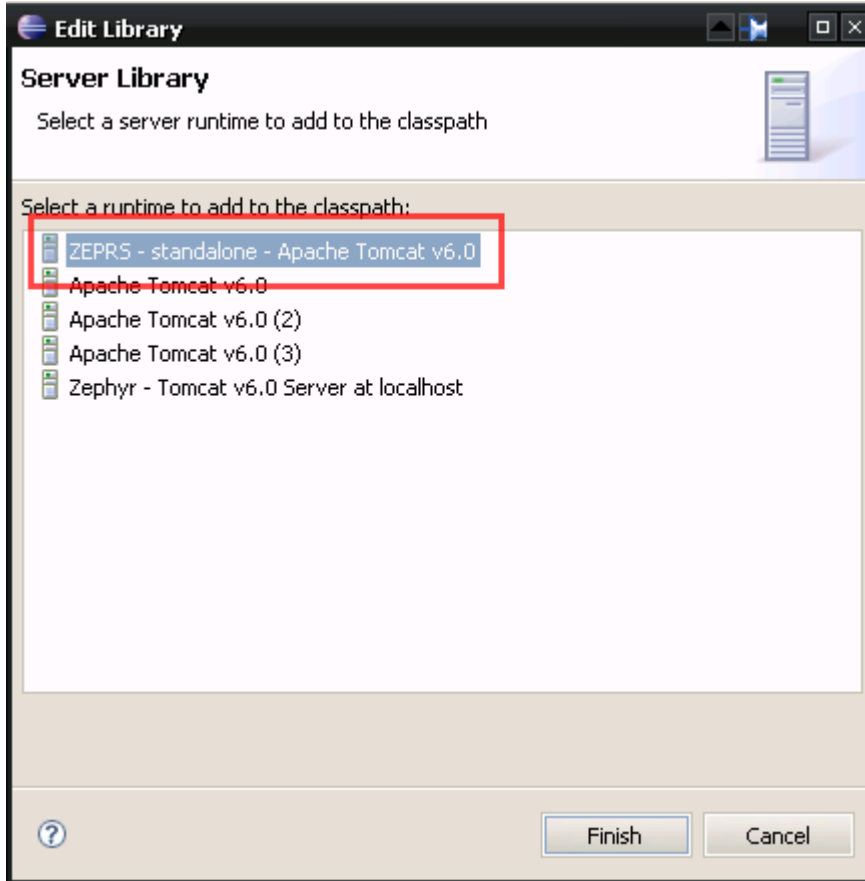
Click the Modules tab at the bottom of the window to Add New Module:

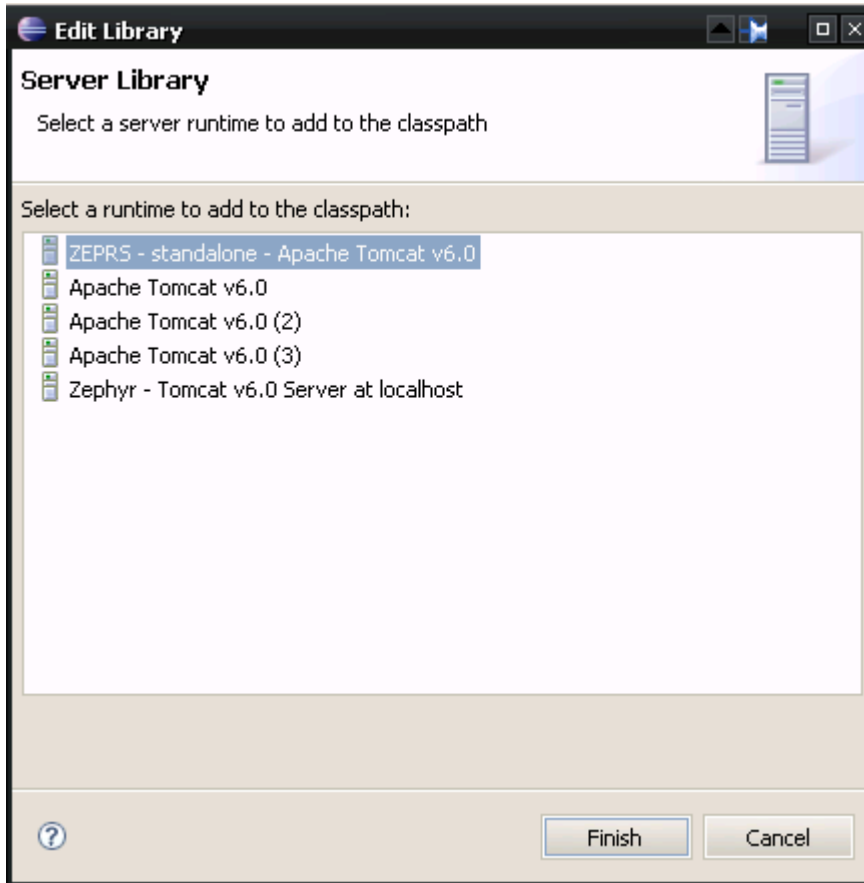


- Edit the Java Build Path in the project properties:



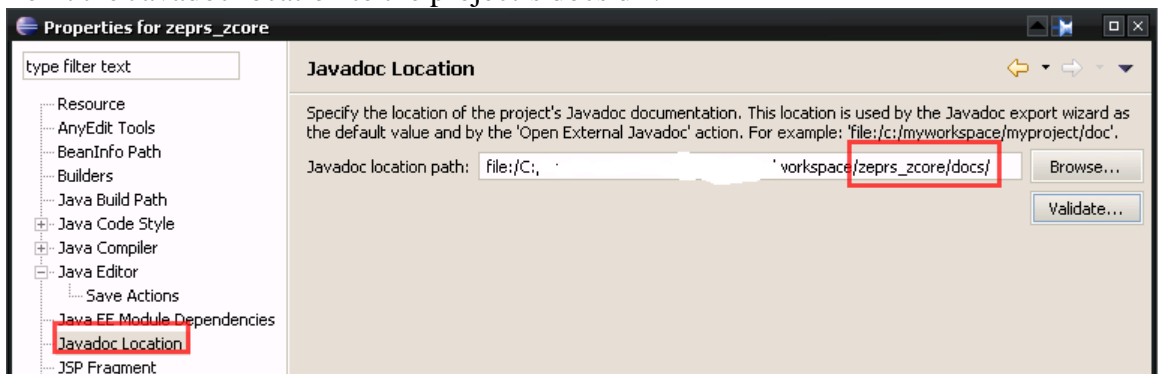
Choose the new server runtime you just setup:



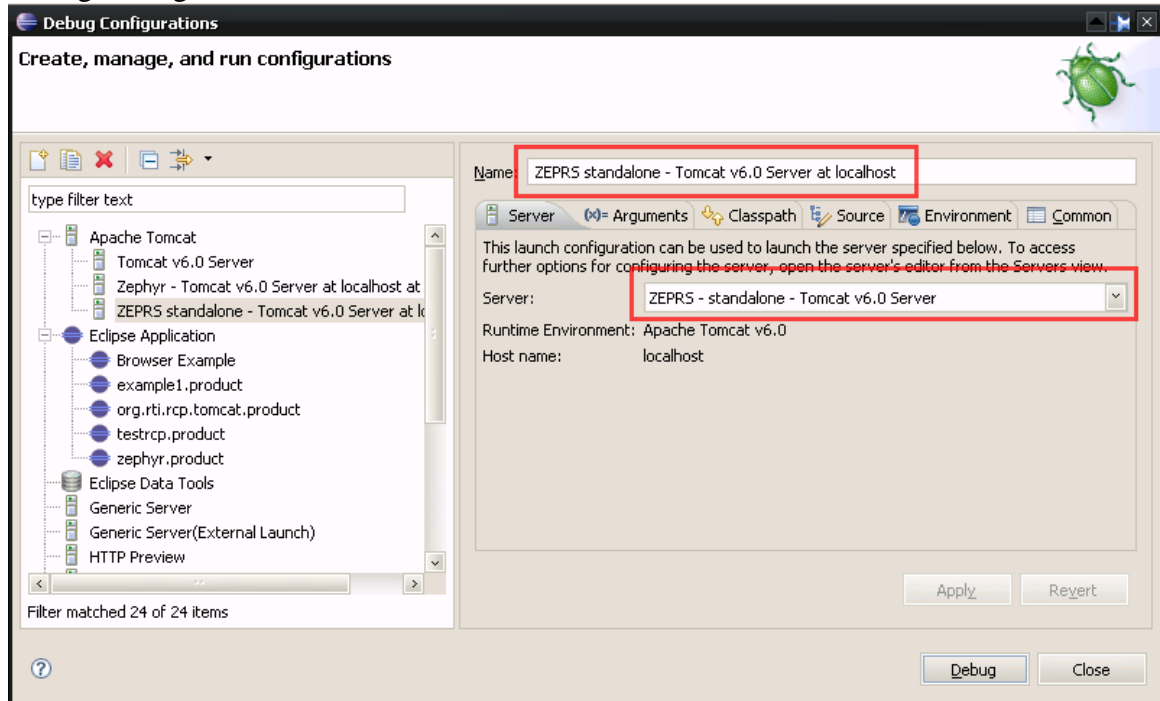


After you press Finish it will take a few minutes to compile the classes and display any errors about missing classes. The missing classes are likely due to the cleaning out the generated classes. This is OK. Your project probably will not call on the methods that use those classes.

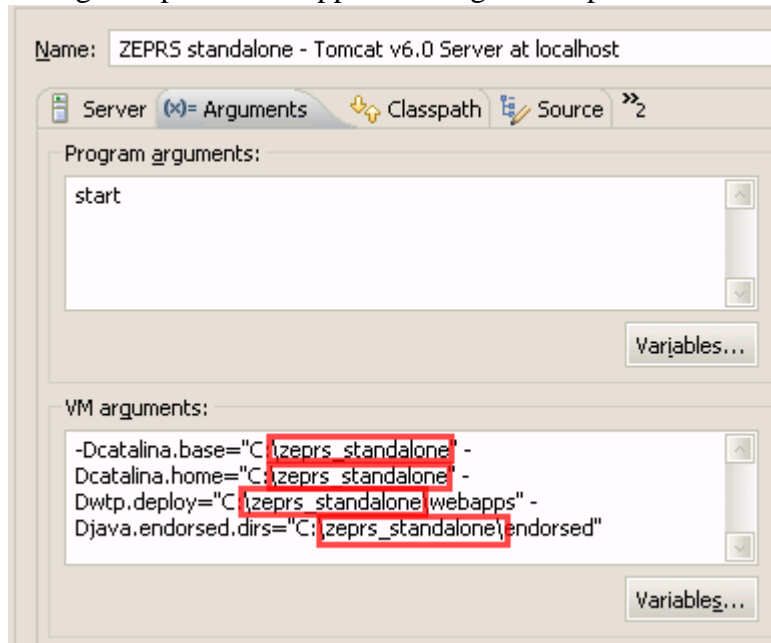
- Point the Javadoc location to the project's docs dir:



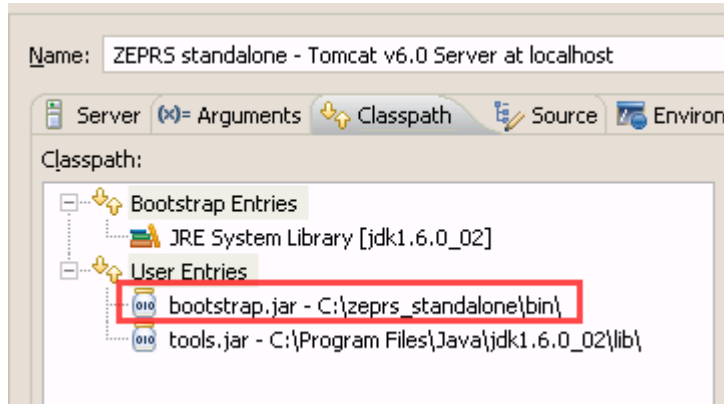
- Debug Configuration:



Change the path to the app in the Arguments pane:



Change the path to bootstrap.jar in the Classpath pane:



- Open web.xml and change the web-resource-name in the following section to the name of the deployed web app:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>zeprs</web-resource-name>
    <url-pattern>*.do</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>*</role-name>
  </auth-constraint>
</security-constraint>
```

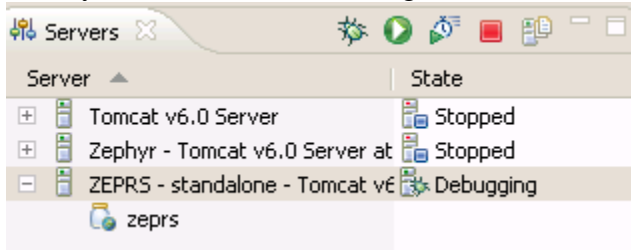
Also here:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>zeprs/admin</web-resource-name>
    <url-pattern>/admin/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>ALTER_PROGRAMS_AND_SCREEN_APPEARANCE</role-name>
  </auth-constraint>
</security-constraint>
```

- Open src/resources/application.properties and edit the first 3 properties: app.name, app.template, and app.title.
- Open src/resources/dev.properties and edit source and install properties
- Open build.properties (use for building wars and other automation with ant) and edit the first 2 properties:
 - project=zeprs
 - install.dir = C:\\zeprs_standalone\\

- Open web/zeprs/META-INF/context.xml and change path and docBase in the second line:
`<Context path="/zeprs" docBase="zeprs"`
- Edit web/zeprs/WEB-INF/pages/version.html. This file, which displays application data in the title bar, is normally built automatically when you run the build ant target; however, when you're starting out, it's nice to see the correct project name in the browser's title bar. Change the project name, build date, etc.
- ZEPRS 1.0 - buildDate: 2008/10/29 20:15 - buildNum: 1

Once you have all of these things sorted, launch the app in Debug:



You should be able to login and run the site config. Form-related functionality, such as creating a new patient, won't work yet. We first need to import the forms we'll use.

Importing forms

<p>From the left nav strip click Admin and then click "Form Import" under "Form Administration.</p>	<p>Administration</p> <p>Application Administration</p> <ul style="list-style-type: none"> • Form Administration <ul style="list-style-type: none"> ○ Create a new form ○ Generate Form Changes ○ Form Import ○ View Field List ○ View List of Rules ○ Manage Flows • SQL console • Application Update Job Log • Reload ZEPRS web application
---	---

Note the location of where you must place the forms.

<p>Form Import</p> <p>Please place the forms you wish to import in C:\zeprs_standalone\webapps\archive\forms\import\new\</p>
--

Place the PatientRegistration.xml form into the import/new directory and refresh the form import page. The system will now list the new form:

Form Import

Please place the forms you wish to import in C:\zeprs_standalone\webapps\archive\forms\import\new\

Here is a list of the forms available for import into the system.
Click on the desired name to import.

Form label
PatientRegistration.xml

Click on the link to import the form.