

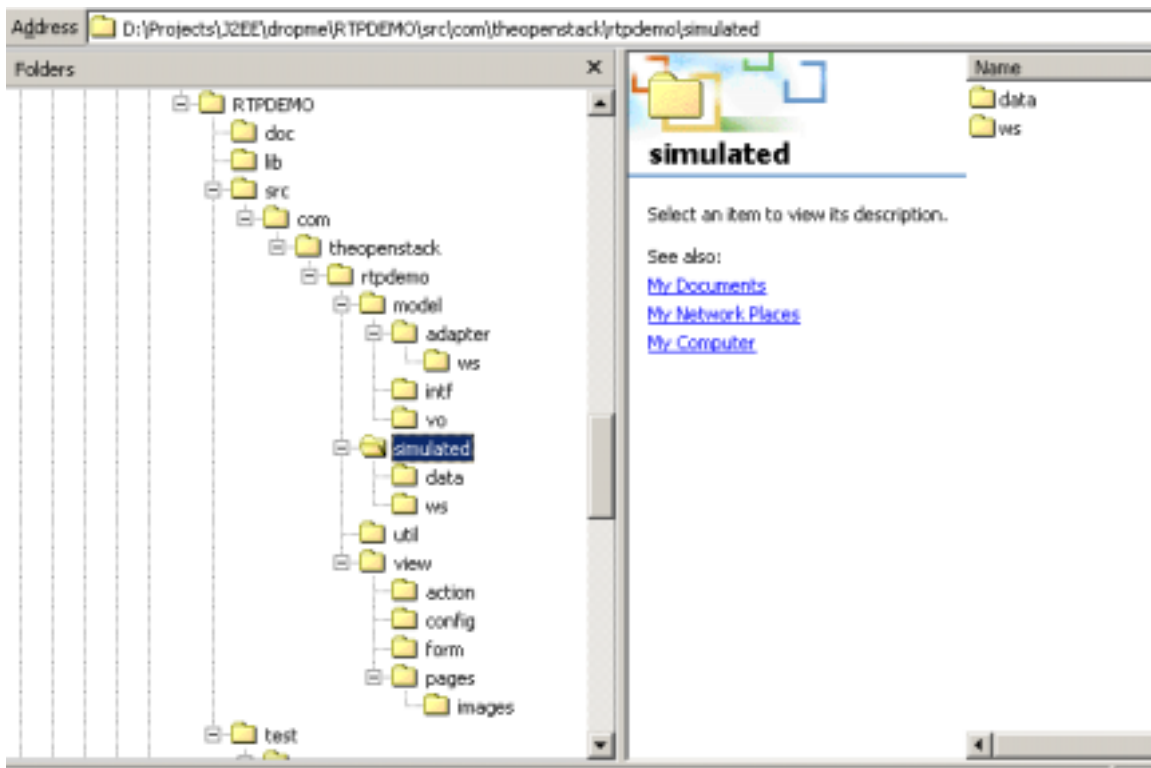
Directory structure and development environment set up

1. Install ANT:

- Download & unzip (or untar) the ant zip file - jakarta-ant-1.5.1-bin.zip to a directory say ANT_HOME (any directory is fine)
- Add the following to the PATH: (ANT_HOME)\jakarta-ant-1.5.1\bin for example
 - On Linux: lets say you downloaded the tar to /usr/anthome then Set to be - \$PATH;/usr/ anthome/jakarta-ant-1.5.1\bin
 - On Windows. Lets say you downloaded the ant zip file to C:/anthome then set path using: Set PATH=%PATH%; C:\anthome\jakarta-ant-1.5.1\bin
- Type ant- help on the command prompt – if you see the help message from ant with different options – ant is installed fine.

2. **Unzip the RTPDEMO.zip file** – this will be downloaded as a CVS refresh from sourceforge in future.

It will create the following directory structure



This directory structure along with the build file should make development as well as deployment easier.

Lets spend a minute discussing the directory structure under 'src' (source):

Under our project directory (com.theopenstack.rtpdemo) we have two important sub directories 'model' and 'view'.

MODEL:

The model directory contains the value objects (or business objects) under 'vo' directory, the interfaces for our model under 'intf' directory.

We have defined the ProductPriceIF interface that has two methods one returns the list of products and other returns price of a product. You will see the interface java file under the 'intf' directory.

The third important part of our 'model' is our 'thin' adapter layer. There are two sample adapters defined under the 'adapter' sub directory – JDBCAdapter and WSAdapter. Both these adapters implement the ProductPriceIF interface. Thus both the adapters return the list of Product value object (defined under the 'vo' directory).

The JDBCAdapter connects to a MySQL database using JDBC and retrieves products based on the search string.

The WSAdapter invokes a web service to do the same. The 'ws' directory under the adapter directory stores the 'generated' client classes for the web service.

VIEW:

The action and form directories under view are self-explanatory. They store the action classes and form beans of STRUTS, respectively.

The pages directory stores the JSP and HTML pages.

The config directory stores the config files (such as struts config) the properties files (for Messages etc) and the Tag Libraries for STRUTS.

Simulated Back End Systems:

In the real time pricing application the product list and price actually comes from a variety of back end systems as described in the specs. We have demonstrated integration with a back end database using JDBC and a back end Application that exposes a Web Service. For our demo application to work we have simulated these back end systems. These simulated back end system – code will reside in the simulated directory.

Rest of the directory structure is standard. The util directory under source contains utility code like service locator. The lib directory under RTPDEMO – base directory has all the jar files of the base components such as STRUTS, Log4J, Axis and so on. The doc directory will contain the documents such as JavaDoc and other docs.

The test directory contains the Unit tests (JUnit).

3. Build and deploy the demo:

After unzipping, open a command prompt and go to the RTPDEMO directory.

Update the build.properties file with your environment.

Development mode: If you plan to use IDE like NetBeans, ideally you should be able to work on this exact same directory structure, add / modify files, compile and run the JSPs – all of this without leaving the IDE.

For development mode use build_dev.xml build file while doing the build. Run “ant - buildfile build_dev.xml build” at the command prompt. Follow the instructions under development mode in the next section. (ant command arguments and the build files are explained later).

If you just want to deploy / test on Tomcat, use the build.xml file. Run Run “ant deploy” at the command prompt. If you are using TomCat directly - just go to <http://localhost:8080/rtpdemo/login.jsp> And test the app*.

Use JDBC or WS as user id and password as password. Any search string for product. Currently the UserVO hard codes the value for BU to be equal to the user id. Hence with JDBC as the user id you are searching for products using the MySQL DB and with WS as the user id – web service is being invoked.

* For now you need to manually create the Database and test data for testing. On MySQL prompt execute the script using ‘source’ command to do this.

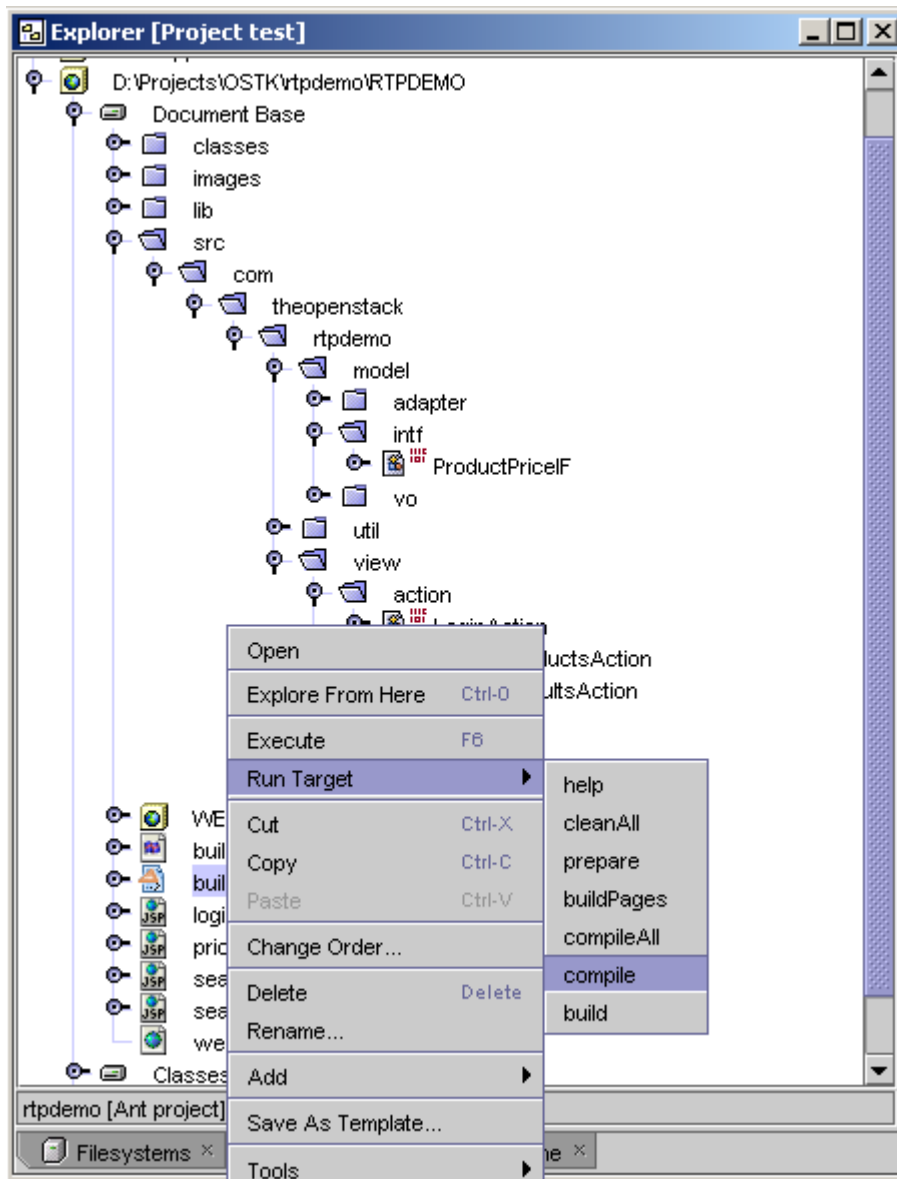
4. Working in the development mode:

Using NetBeans and the ant build in development mode:

Now you can open Netbeans and mount the top level directory (RTPDEMO in my case). You are ready for your development work. (If NetBeans is set up properly, it will load all the jars and the build file, else you will need to load those separately).

Compiling a Java File:

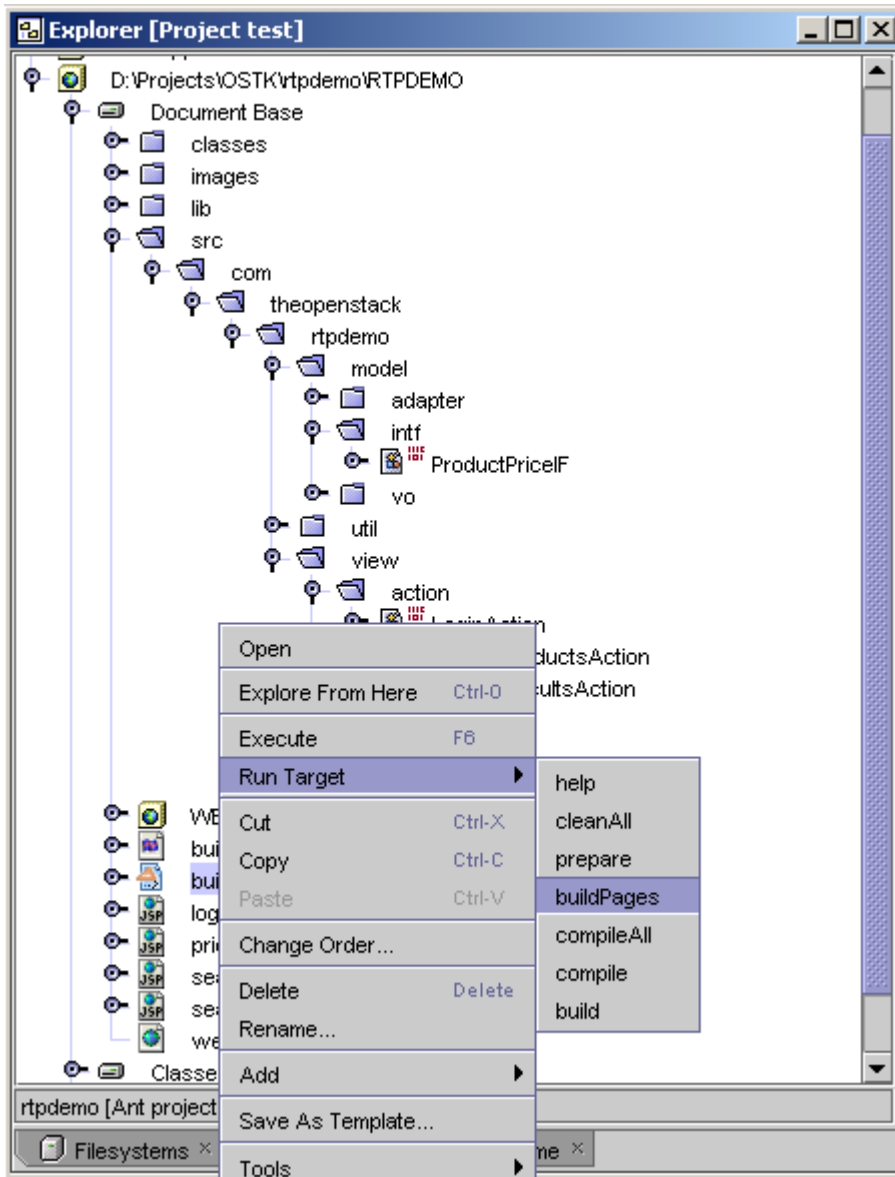
Any time you change or add java files, run the ‘compile’ target as part of build_dev as shown:



This will compile changed java files and put the classes into the right directory.

Modifying a JSP or config files:

If you change or add a JSP / HTML / image file, run the buildPages target as shown, at this point of time all it does is copies the JSP /HTML or image files to the correct location for tomcat.



You can just right click on the JSP and click execute to test. Click execute (force reload) if you added a JSP.