Using SimpleDB with Eclipse

Eclipse is a fantastic tool for working with the SimpleDB code. This document describes an easy way to import your downloaded SimpleDB files into Eclipse. The basic idea is to create two projects: one project to hold the SimpleDB source files, and one project to hold the client files. There are other possibilities—for example, you can create a separate project for each client program—but the strategy described here is perhaps the simplest.

1. Creating the SimpleDB Source Project

A. In Eclipse, create a new Java project named *SimpleDBServer*.

- You will need to specify the location. I recommend using the default location, which tells Eclipse to create a folder named *SimpleDBServer* within its workspace.
- You should specify "Create separate folders for sources and class files". Eclipse will create folders named *src* and *bin* within the *SimpleDBServer* folder.
- Click on the Finish button.

B. From the finder, copy the entire downloaded *simpledb* folder (including the *simpledb* folder itself) to the folder *SimpleDBServer*/*src* folder in the Eclipse workspace.

- When you are done, the *src* folder should have one child folder, namely *simpledb*. The *simpledb* folder should have the child folders *buffer*, *file*, etc.

C. In Eclipse, execute Project/refresh (F5) to compile all of the source files.

- The *bin* folder will then contain a *class* file for each source file.

2. Creating the SimpleDB Client Project

A. In Eclipse, create a new Java project named *SimpleDBClients*.

- Use the same configuration as above to specify the location and to use separate src and bin folders.
- Instead of clicking "Finish", click "Next" to get to the Java Settings window.
- Click on the Projects tab. Then click Add, and click the box for the *SimpleDBServer* project. (Doing so adds the SimpleDB source code to the project's class path. Otherwise, the client code will not be able to resolve references to the SimpleDB classes.)
- Now you can click the Finish button.

B. From the finder, copy the six Java files from the downloaded *studentClient/simpledb* folder into the folder *SimpleDBClients*/*src* folder in the Eclipse workspace. Note that unlike the previous step B, you do not copy the enclosing simpledb folder.

C. In Eclipse, execute Project/refresh as in the previous step C.

D. Whenever you write a client program, you should save it in the same place as the demo programs from step B.

3. Running SimpleDB as a Server

A. From the *SimpleDBServer* project, run the class *server.Startup.* You will need to supply an argument that is the name of the database you want to use. The job of this code is to accept connections from JDBC client programs and to service their database requests. It will run forever until you explicitly terminate it. The code writes status information to its console. It you use print statements to debug the behavior of the server, that output will appear in this console.

B. From the *SimpleDBClients* project, run one of the JDBC programs. (The demo programs are all JDBC except for *StudentMajorNoServer*.) This code will terminate, writing output to its console.

C. You can run the client and server programs in the same Eclipse window. The console frame has a button that lets you toggle between the two consoles.

D. It is perfectly fine to have multiple JDBC clients running simultaneously. In fact, the server is designed to support such concurrent activity. However, it makes no sense to run multiple copies of the server concurrently (unless you change the server code to have each server run on a different port).

4. Running SimpleDB as a Standalone Program

It is also possible to write a program that calls the SimpleDB source code directly, instead of calling *server.Startup*. The demo program *StudentMajorNoServer* is an example. In this case, the entire database source code is exclusively available to the program. The advantage of such a program is that you can use it to test your changes to the source code without having to run the server and a client. These programs can be placed in the *SimpleDBClients* project. (Or if you like, you can place them in another project configured similarly.)