

# Getting set up to debug EJS programs with Eclipse

© August 2011 by Larry Engelhardt

1. Make sure that you have the most recent Java JDK installed on your computer. On Macs this should already be installed. For PCs and linux machines, this can be downloaded from <http://java.sun.com/javase/downloads/index.jsp>.
2. Download the “Eclipse IDE for Java Developers” from [www.eclipse.org](http://www.eclipse.org). Save the installation file (ending in .zip) on your desktop. Unpack the .zip file. The .zip file will contain a folder named “eclipse”. Copy (unpack) the eclipse folder somewhere on your computer (e.g., C:\Program Files\eclipse or C:\eclipse). For convenience, create a shortcut to the application named “eclipse” in the eclipse folder.
3. Create a folder inside of your EJS folder with a name such as “eclipse\_workspace”. Note, this location for the folder is not necessary, but it will be convenient for copying files.
4. Open Eclipse. When prompted to enter the workspace, select the folder that you just created in the previous step. Once Eclipse has opened, proceed to the workbench. (There are also tours/tutorials available, and once you have gotten started debugging, you can access these resources using the “Help” menu if you want more help.)
5. Under the “File” menu, select “New” then “Java Project”.
6. Give the project a name such as “ejs\_project”. Then select “Finish” or “Next” then “Finish”.

7. Look inside of the `eclipse_workspace` folder, and you should see a new folder with a name that matches the name that you gave to the project (e.g., “`ejs_project`”). Within the `ejs_project` folder, you should see two new folders named “`src`” and “`bin`”. Add another (third) folder named “`jars`” inside of the `ejs_project` folder.
8. Minimize Eclipse, navigate to the `EJS/bin` folder, and copy the JAR files from the `EJS/bin` folder into your `eclipse_workspace/ejs_project/jars` folder. Also copy all JAR files from `EJS/bin/extensions` and `EJS/bin/extensions/model_elements` directly into your `eclipse_workspace/ejs_project/jars` folder. (Do not create subfolders under `eclipse_workspace/ejs_project/jars`.)
9. Now you will need to tell Eclipse where to find the JAR files that are used by EJS. In Eclipse, right click on the newly created `ejs_project`, select “Properties”; click on “Java Build Path”; and select the “Libraries” tab. Click the “Add External Jars” button; select all of the JAR files that you copied into the `eclipse_workspace/ejs_project/jars` folder; and click “Okay”. (Note, when new versions of EJS are released, you should replace these JAR files with the updated JAR files that are distributed with the newer version of EJS.)
10. Now you are ready to copy an EJS program into your Eclipse workspace. EJS generates a `.java` file containing the Java code for each EJS program that you create, and within Eclipse these `.java` files can be used to debug your EJS programs. To do this, minimize Eclipse, and copy the `.java` file(s) from your `EJS/workspace/output` folder into your `eclipse_workspace/ejs_project/src` folder. For example, if you create an EJS program named `testProgram.ejs`, you will find a folder named `testProgram_pkg` located inside your `EJS/workspace/output` folder. Copy the entire `testProgram_pkg` folder (not just the contents of the folder!) into your `eclipse_workspace/ejs_project/src` folder.  
When copying these files, be aware of the following pitfall: If the `.ejs` file for your simulation was saved in a *subfolder* of your `EJS/workspace/source` folder (e.g., `EJS/workspace/source/folderName/simName.ejs`) then the `.java` file will also be saved into a *subfolder* of your `EJS/workspace/output` folder (e.g., `EJS/workspace/output/folderName/simName.java`). When copying the `.java` file into the Eclipse workspace, this directory structure must be copied also. A simple way to avoid this confusion is

to save a copy of the .ejs file that is to be debugged directly in your `EJS/workspace/source` folder—not in a subfolder of `source`.

11. In Eclipse, right click on `ejs_project`, and select “Refresh” (or press F5 to refresh). Now within Eclipse you should be able to navigate to the `testProgram_pkg` folder and open the file `testProgram.java`.
12. To run a program in the debugger, press F11. To add a breakpoint at a certain line of code, right click to the left of the line of code, and select “Toggle Breakpoint”.