# Working With Eclipse

You need to setup these tools.

## Java SE JDK

Download and run the installer executable from Oracle. Once finished, you will have installed both the SDK and JRE (Java Runtime Environment) components.

The installation of Eclipse depends on the creation of an environment variable ‘JAVA\_HOME’ which references the JDK installation directory. This environment variable should be installed by the JDK installation application.

## Eclipse IDE

Eclipse is a FOS (free open source) product that is distributed by the Eclipse Foundation and sponsored largely by IBM. Eclipse is the most widely used Java IDE. Many employers use Eclipse in their organizations and thus look for experience with Eclipse when making their hiring decision.

Eclipse is distributed as a zip file that needs to be extracted onto you PC. I recommend extracting into the directory ‘C:\’ i.e. the C-Drive root. Note that the zip file extracts into the ‘eclipse’ directory, so when the extraction is completed, you should find the directory ‘C:\eclipse’ on your PC (or the eclipse directory at the location selected by you).

To launch the Eclipse IDE, navigate to the installation directory e.g. C:\eclipse, and launch the ‘eclipse.exe’ executable found there.

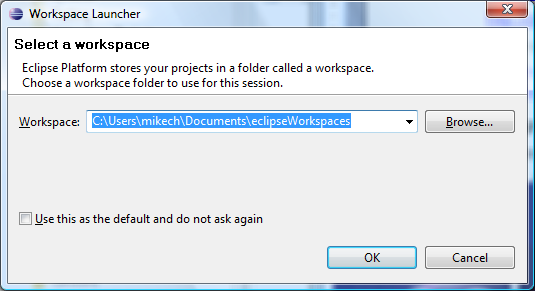
There are several Eclipse tutorials that can be found by Google’ing ‘eclipse tutorial’.

# Creating an Eclipse Workspace

The Eclipse IDE stores your work in the following hierarchy:

Workspace Directories contain Project Directory contain Project Files (java, classes, libraries, etc). You started an Eclipse workspace when you started Eclipse for the first time. You are welcome to use that workspace or start in a new workspace directory.

You select your current workspace, or create a new workspace, from the “Workspace Launcher” that comes up when you start Eclipse.

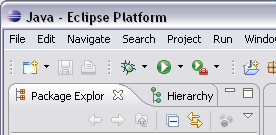


A second method of switching workspaces can be found in the “File” menu with the “Switch Workspace” option.

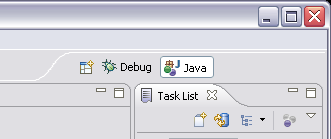
**Eclipse Perspectives**

After starting Eclipse in your target workspace, it is important that the IDE be in the “**Java (default)”** perspective. The Eclipse “perspective” could be considered the IDE’s “mode”. Eclipse provides several perspectives including Java, Debugging, Resource, and others. **Note** that Eclipse offers 2-3 Java perspectives but that you want to use the Java (default) perspective.

You can tell the IDE’s current perspective by looking at the title in the upper left hand corner of the window. The following graphic is the Java perspective (which is what you need).

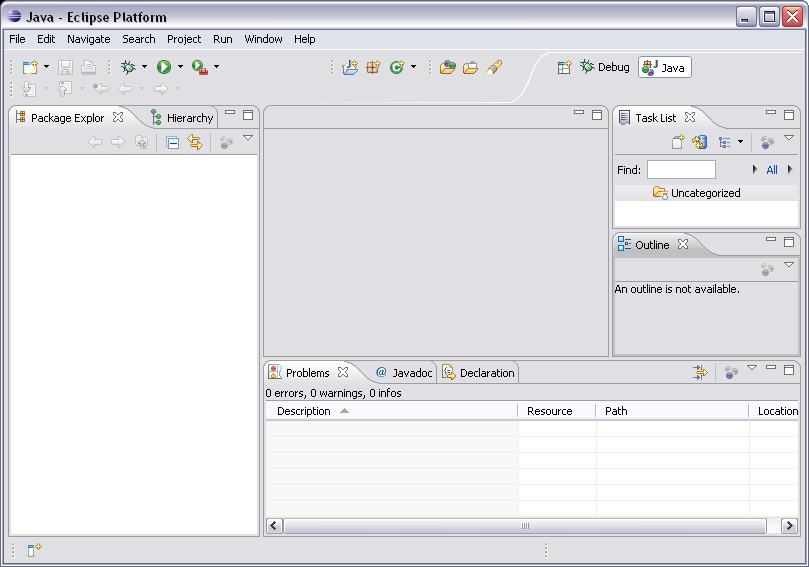


You switch perspectives by using the controls shown in the following graphic. Notice that there are two perspectives active: Debug and Java, but that the Java is active. You open new perspectives by selecting the symbol with the plus (+) or by using the Window > Open Perspective menu.



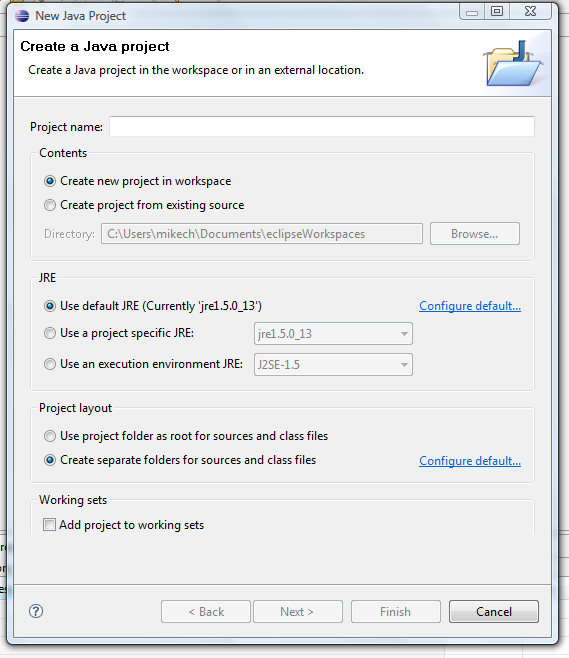
# Creating an Eclipse Project

Each workspace contains one or more projects. The projects are listed in the Java perspective in “Package Explorer” view shown on the LHS in the following graphic. NOTE: This graphic is of a newly created workspace and so does not contain any projects.

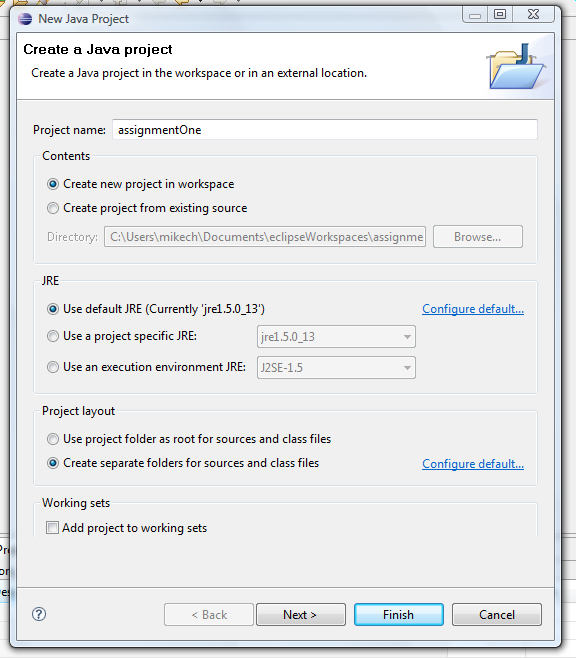


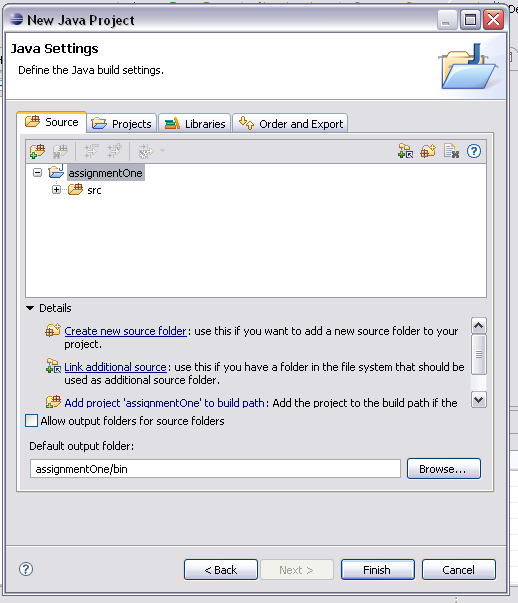
**Create a Java Project**

You create a project by selecting the File > Java Project menu. This action bring up the “New Java Project” dialog as shown in the following graphic.

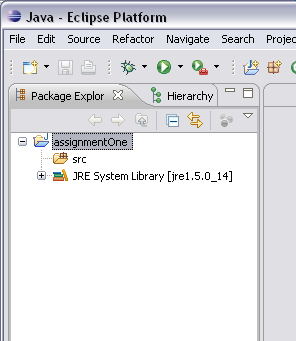


Into this dialog you will enter the new project’s name. Notice that I selected “assignmentOne” in this example. **Note**: You can not have duplicate project names. After a unique name is entered the “Next” button will enable as illustrated in the following graphic.

After entering a project name and pressing “Next” the following dialog is presented. You can accept the defaults project settings by pressing “Finished”.



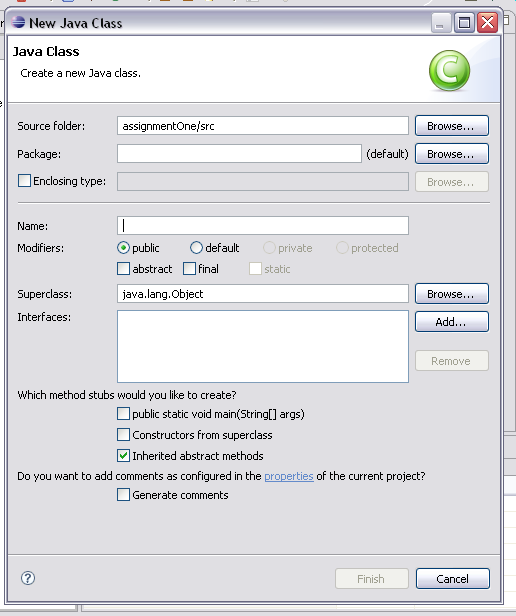
After you have finished with the New Project dialog, you will find your new project in the package explorer as shown in the following graphic



Notice that the project is presented as a tree and that in this graphic I have expanded to root project folder to show its contents which are the project’s source folder (src).

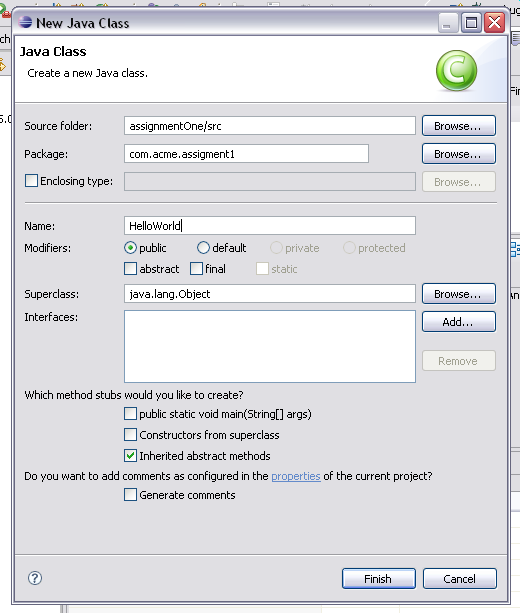
**Creating a New Class (.java file).**

You create a new class, and corresponding .java file, by selecting “File > New > Class” menu item while in the Java Perspective. This will cause the following “New Java Class” dialog to be presented.



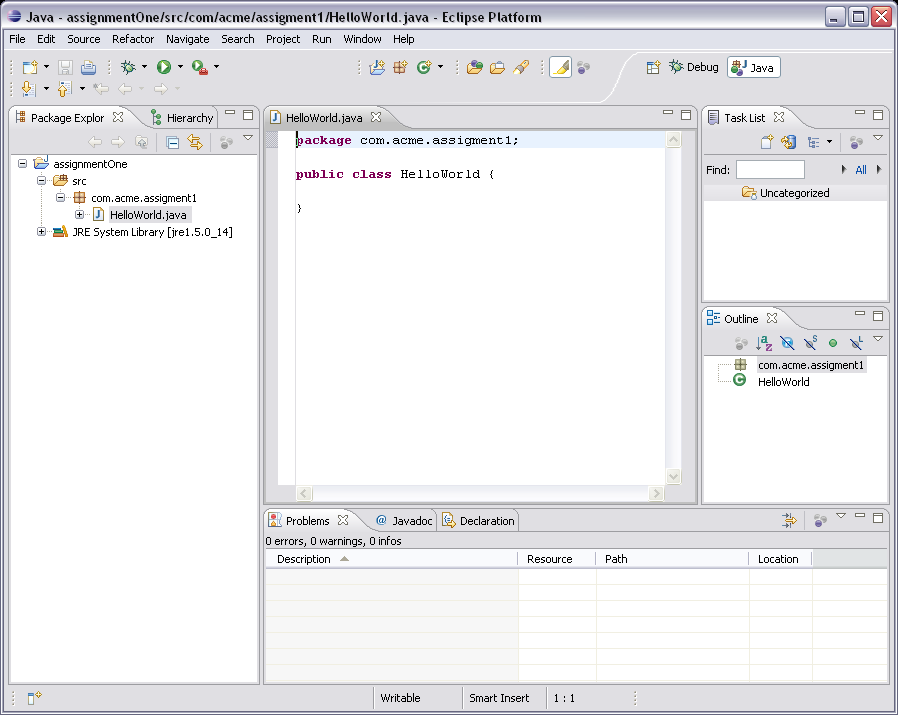
Into this dialog you will enter **both** a package name and a class name.

In the following graphic I have entered ‘com.acme.assignment1’ as the package name and ‘HelloWorld’ as the class name.



After you have entered the package and class names, the “Finished” button will enable. Pressing Finished will cause Eclipse to create this new class in the given package.

At this point the Java perspective should now look something like the following graphic with the new class in the Package Explorer (LHS) and the HelloWorld.java editor in the center window.

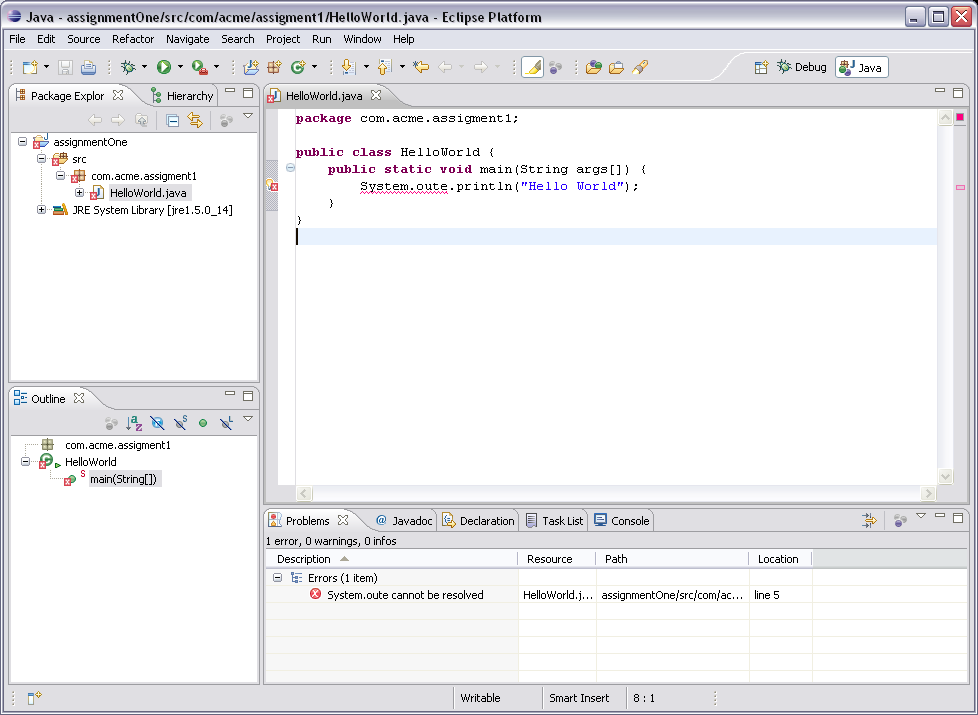


Congratulations, you are now ready to begin editing and testing you Java application.

# Code and Debug Your Assignment

You can complete the class started by Eclipse using the Java editor that is presented when you selected the class from the Package Explorer. When you save the class (and corresponding .java file), Eclipse will compile the class and flag any errors which you may have in the code.

The following figure provides an example of a class with and error. Notice that the error is presented in Package Explorer, in the Java editor itself, and in the “Problems” view at the bottom of the window (in a tab). You need to correct any errors found by the compiler before you can run the application.



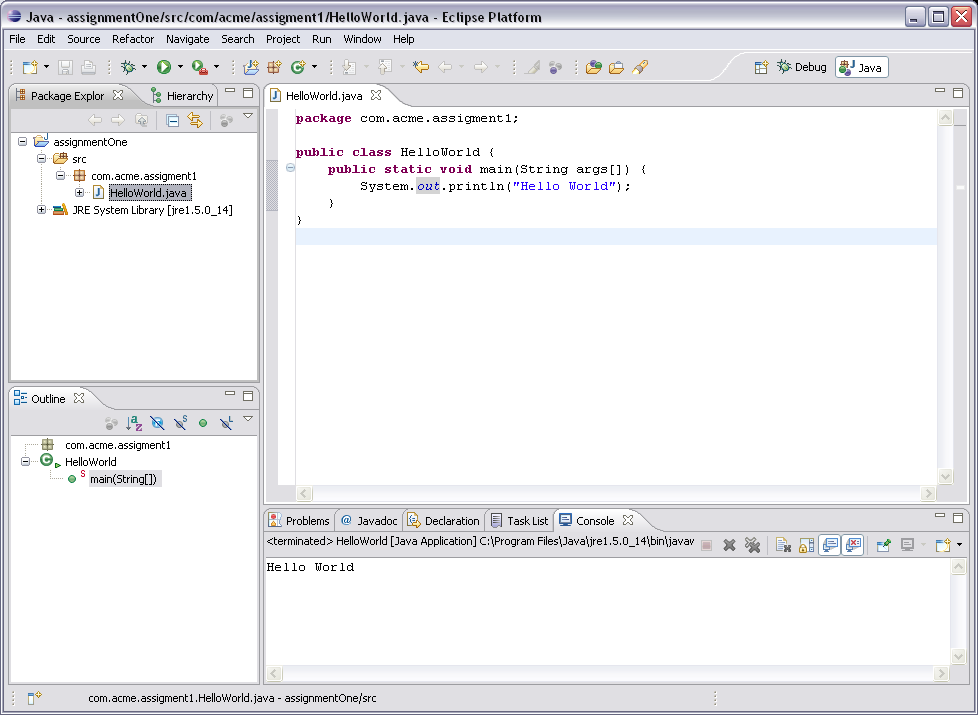
# Run / Test Your Assignment

You need to execute (run) your Java class and see it print “hello world” to the console this is done as follows.

After you have completed the needed class (HelloWorld in this case) you run the application using the Eclipse IDE. You do this by selecting the class in the Package Explorer and then selecting “Run > Run As > Application” menu item. This should cause your application to execute and present its text in the “Console” view in the tabs at the bottom of the window.

**NOTE**: If the menu item “Run > Run As” does not present you any options, this means that your class does not contain a valid “main()” method.

The following graphic presents a correctly executed application with the text printed to the Console view in the tabs at the bottom of the window.



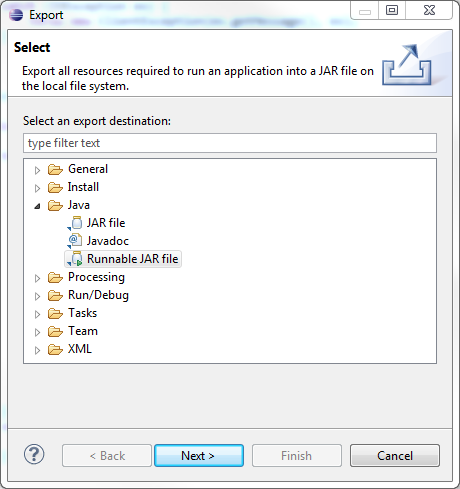
# Exporting an Executable Jar File

The Java Virtual Machine (java.exe) provides command line options that permit the executable of specially formatted jar files. The jar file will contain all of the classes composing your project plus any library jars needed by your design.

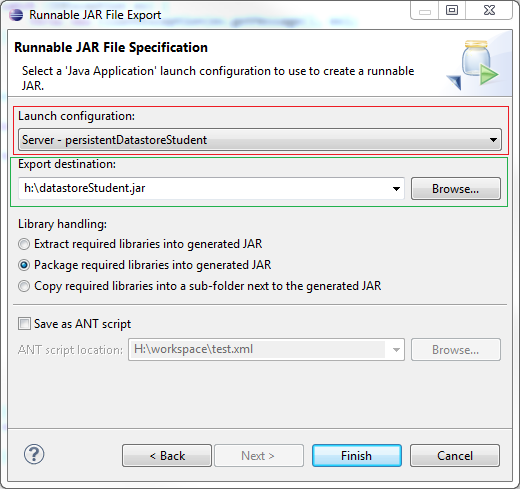
The command for executing an executable jar is:

Java –jar <jar-file-name>.jar <command line arguments> where any command line arguments to be passed to main() are provided after the jar file name.

Eclipse provides an option to easily build an executable jar. You need to construct you application so that it provides a main() class which serves as the application’s entry point. The process starts by selecting the “File>Export” menu item to bring up the Export dialog. Select the Java>Runnable JAR File option from the dialog. Press Next.



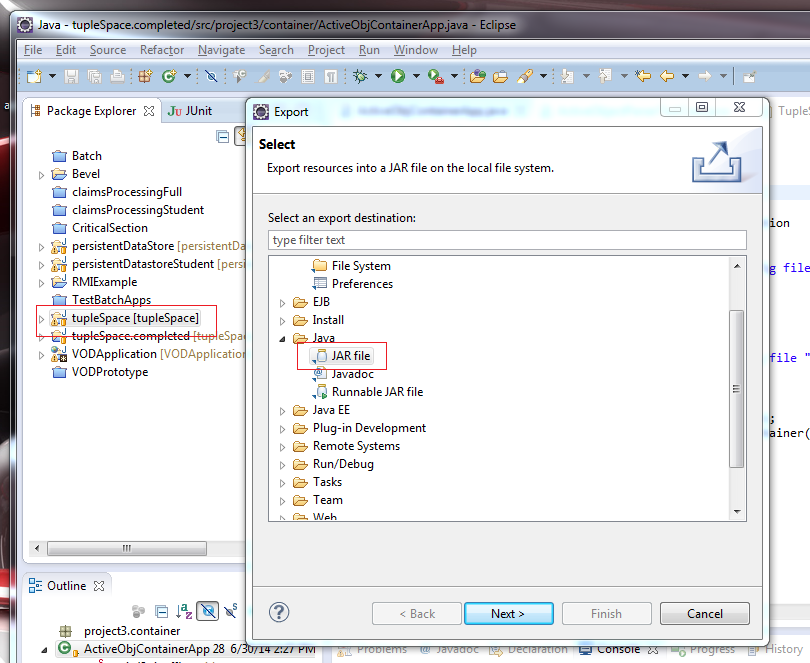
From the Runnable Jar File dialog (shown next) select the Launch Configuration that corresponds to the main() class in the project. For example in the following figure, we have selected the main() class Server contained in the project persistentDatastoreStudent. Next, you need to specify the file the executable jar file will be written to. This is done using the controls highlighted in green. Finally, press finish and the executable file will be written.



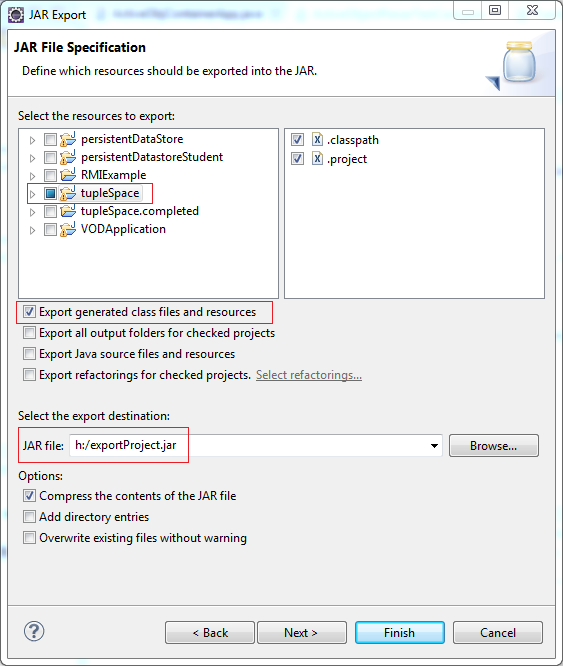
# Exporting a Library Jar File

An Eclipse project can be delivered for evaluation as a library jar file. This means that the exported jar file will contain the project’s classes and other resources in a format that can be linked with an Eclipse project or Java application.

Exporting a jar file starts selecting the project to be exported and selecting File>Export and select Java > JAR File from the Export Dialog.



Pressing ‘Next>” will produce the following dialog where the student enters the generated jar file path.

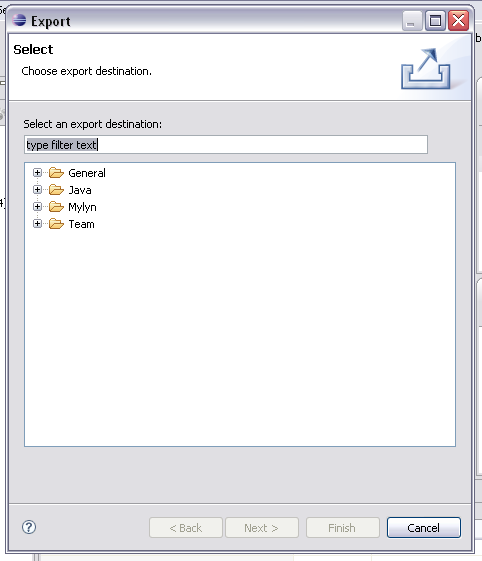


Press ‘Finished’ will generate the jar file at the given location / path.

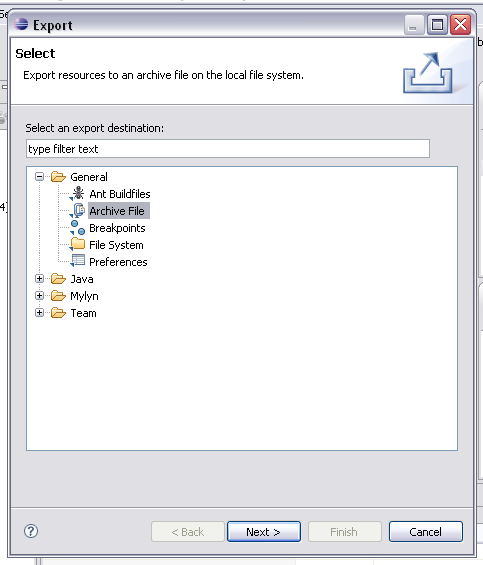
# Exporting a Project to a Zip File

After you have finished the assignment you need to export the project containing the assignment to a zip file. It is the zip file that needs to be submitted through WebCT. For example, you will not be able to submit just the Java files.

Exporting the project is accomplished using the “File > Export” menu item. Selecting this item brings up the following “Export” dialog.



Expand the “General” node, select “Archive File”, and press “Next” as shown in the following graphic.

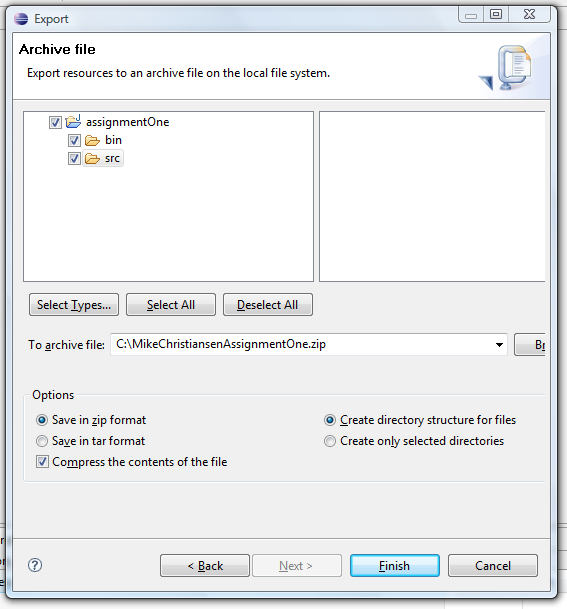


Select the project to export in the LHS view. Make sure that the classpath and project files are selected (which they are by default).

Next, enter the name and location of the exported project zip file. Make sure that the other options as shown in the following graphic are selected.

**NOTE**: You must use a file name that reflects your name and the name of the assignment. **Your project will not be graded if a filename of this format is not provided**. Notice in the following graphic the filename is “MikeChristiansenAssginmentOne.zip”. This will allow me and / or the TA to more easily grade the almost 50 assignments that will be turned in by this class.

After entering the exported project’s file name, press “Finish” to complete the process.



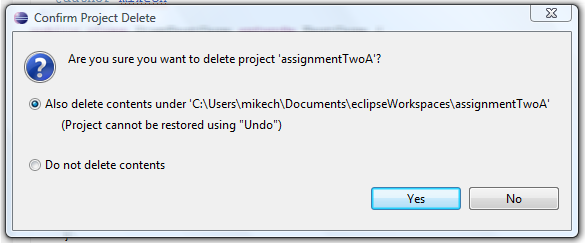
After pressing “Finished”, Eclipse will export all of the files in your project into a zip file at the specified location. You should open the zip file to be sure that it can be opened and contains your work. At this point you are ready to submit the exported project file for grading. This is done though WebCT as follows.

# Importing a Project from a Zip File

Many programming assignments will provide an exported project that you can import into you Eclipse workspace. These projects will be provided zip file archives that will be one of the files that can be downloaded from the WebCT assignment. The zip archive may contain sample code or a project template that can be used as a starting point for your efforts. You will be importing the project zip archive into your workspace.

**Optional: Removing existing projects with the same name from the workspace**

You can not import a project with the same name into the workspace. This means that if you import and try to re-import the project template you must first delete the old project from the workspace. This is accomplished by selecting the existing project from the package explorer and selecting the “Edit > Delete” menu item. This will bring up the dialog shown in the following graphic. Notice that the option “Also Delete Contents Under C:\...” is selected**. It is very important that this option is selected** so that the project files are removed from you workspace

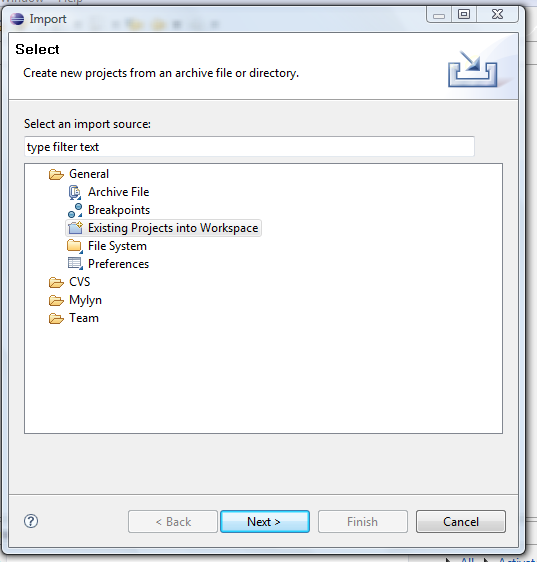


A this point the Old project will be have been removed from your workspace and you may begin importing the project template

## Importing the Project

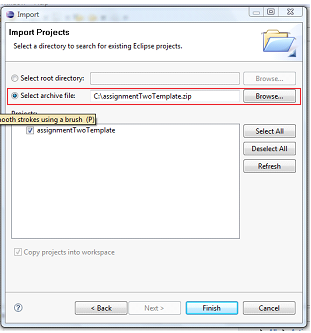
The process for importing the template project is a follows.

Open the import wizard using the “File > Import” menu item. This brings up the import dialog shown in the following graphic. Make sure to select the “Existing Projects into Workspace” option (under General) and press Next.



This brings up the following import dialog. There are a few import things to note:

1. You need to select the “Select archive file” option and then press browse to select the project template archive (zip) file.
2. When the file opens, you need to select the project.
3. Press Finish and the project will be imported into your workspace.



# Other Things to Note:

* You can use the same workspace for each of the class assignments. Simply create a new project with the process shown above and a new project will be presented in the Package Explorer view.
* Your projects should always contain a “src” folder into which your project class files will be placed.
* You can delete project, but if you do you should “delete the source” associated with the project or you will not be able to reuse the deleted project’s name.
* You can arrange the views in the default Java perspective however you please and Eclipse will retain those settings in the workspace. For example, the following graphic illustrates how I like to set up my Java perspective. That is, with the “Outline” view placed in the LLH side of the window and the “Tasks” view placed into the bottom tab bar.

