NetBeans Primer v8.0

Using an IDE for Web Development

Using a simple text editor to create source code and compiling from the command line enables the programmer to concentrate on the code and not be encumbered by the complexity of an IDE. However, with web development, there are a number of features provided by an IDE that make using an advanced tool advantageous.

For example: automatic creation of the web directory structure, package creation and management, automatic .war file creation and updates, code completion, inclusion and easy management of an integrated web server, and debugging, all provide considerable benefits for the web developer. Furthermore, the overwhelming majority of professional web developers utilize IDE's so becoming proficient with their use is recommended. That said, all of the JSP development assignments can be completed with a simple text editor and use of the javac command line compiler.

Installing the JDK, NetBeans, and a Web Server

The steps below include installation of the following 3 components: the Java Development Kit (which also installs a Java Runtime Environment (JRE)), NetBeans, and a web server (Tomcat) that contains a servlet container (Catalina). The Java EE version of NetBeans (see below) contains 2 web servers with servlet containers (Tomcat and Glassfish). Therefore, additional web server installation is not required. However, for INEW 2338, we will install the XAMPP stack for other purposes but for the Tomcat servlet container included in XAMPP. Furthermore, NetBeans starts the web server automatically when a web project is started. And, the web server can be stopped from within NetBeans.

There are a number of ways to accomplish NetBeans, JDK/JRE, and web server platform installation and configuration. The steps below outline a path which places the entire development environment on a flash drive. This method is recommended due to its portability. A professional developer would very likely choose to develop from a local hard drive for performance and other considerations. However, as a student, there are advantages to having the ability to use your entire development on any machine.

As with any set of progressive computer configuration instructions, it is imperative that the directions be implemented as depicted. **Take your time and pay attention to detail**, the steps can be performed in a few minutes and will become second nature after practice. Making a small configuration mistake can derail the entire sequence and starting over may be required.

I recommend installing all software development tools to your local drive. However, the installation described in the document below is installed to a flash drive. If you plan to maintian multiple installations on flash drives then I recommend using the same drive letter for each installation. I change the drive letter to N: on the flash drive using the Disk Management program in Windows. R-click the drive | Change Drive Letter and Paths… I chose N: to remind me of NetBeans and to not conflict with lower letters already in use. Again, I recommend that all development tools be installed to C: (but not Program Files due to UAC restrictions).
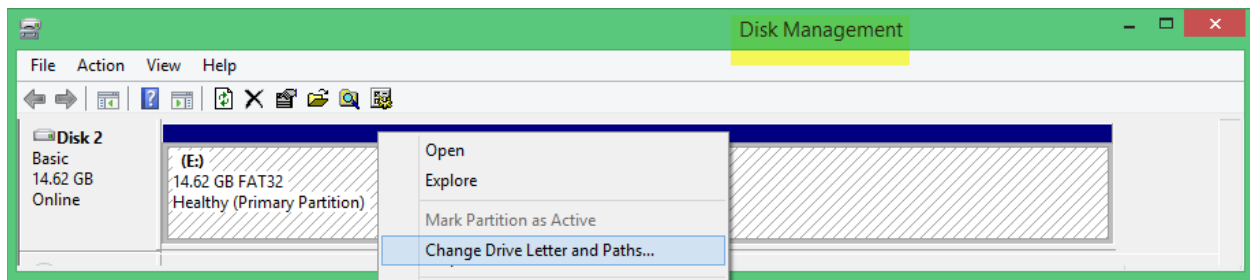
Install the JDK first. If the JDK is not installed you will receive this error when attempting to install NetBeans.
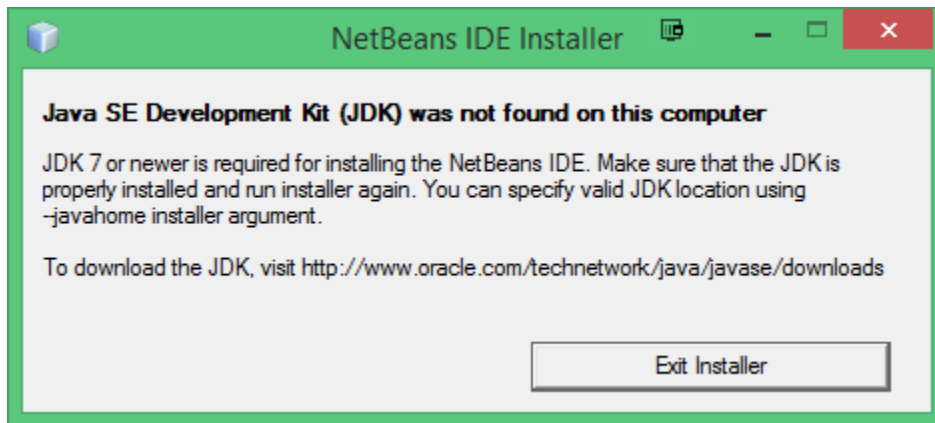
Download the latest JDK, samples, and JavaFX Demos from:

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Java SE Development Kit 7u51

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	67.7 MB	jdk-7u51-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	67.68 MB	jdk-7u51-linux-arm-vfp-sflt.tar.gz
Linux x86	115.65 MB	jdk-7u51-linux-i586.rpm
Solaris SPARC 64-bit (SVR4 package)	23.92 MB	jdk-7u51-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.33 MB	jdk-7u51-solaris-sparcv9.tar.gz
Windows x86	123.64 MB	jdk-7u51-windows-i586.exe
Windows x64	125.46 MB	jdk-7u51-windows-x64.exe

Java SE Development Kit 7u51 Demos and Samples Downloads

Java SE Development Kit 7u51 Demos and Samples Downloads are released under the Oracle BSD License

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	9.89 MB	jdk-7u51-linux-arm-vfp-hflt-demos.tar.gz
Linux ARM v6/v7 Soft Float ABI	9.78 MB	jdk-7u51-linux-arm-vfp-sflt-demos.tar.gz
Linux x86	16.6 MB	jdk-7u51-linux-i586-demos.rpm
Solaris SPARC 64-bit	0.86 MB	jdk-7u51-solaris-sparcv9-demos.tar.gz
Windows x86	17.41 MB	jdk-7u51-windows-i586-demos.zip
Windows x64	17.51 MB	jdk-7u51-windows-x64-demos.zip

JavaFX Demos and Samples Downloads

JavaFX Demos and Samples Downloads are released under the Oracle BSD License

Product / File Description	File Size	Download
Linux	20.24 MB	javafx_samples-2_2_51-linux.zip
Mac OS X	20.24 MB	javafx_samples-2_2_51-macosx-universal.zip
Windows	20.24 MB	javafx_samples-2_2_51-windows.zip

Create a folder C\:Java and move the files highlighted to that folder. We will only use the jdk-7u51-windows-x64.exe file for now but you may want to use the samples and demos later.

Close all programs when installing this or any software. Double-click jdk-7u51-windows-x64.exe to begin the installation process and make the selections as shown below.
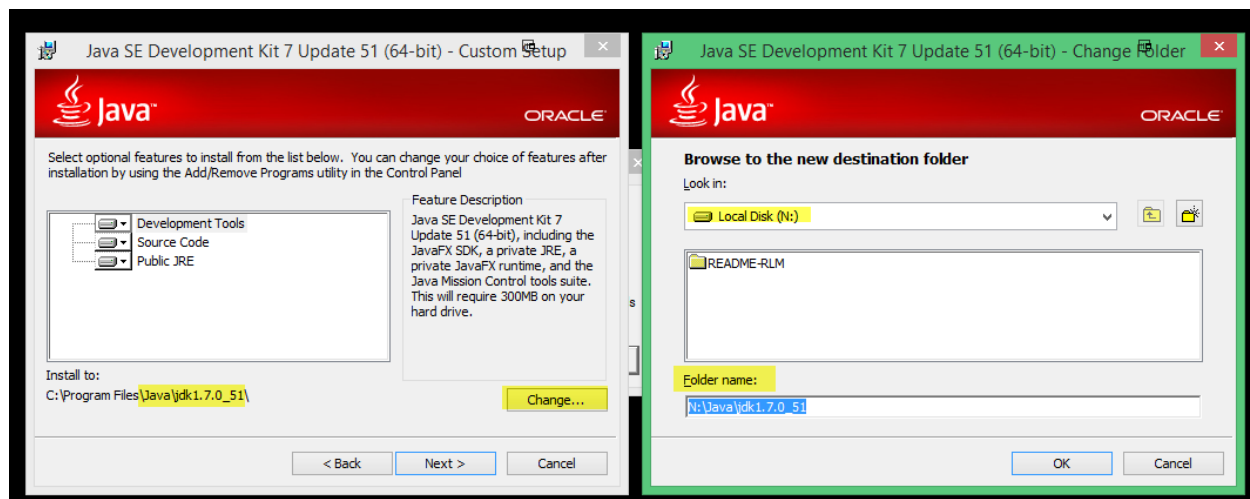
This screen may be displayed for a few minutes as the installation progresses. Personally, I have had installations stall at the Java Setup – Progress dialog below. In those cases, I waited approximatedly 20 minutes and used Task Manager end the installation process. On the few occassions when this has happened, the installation was successful. However, if unsure, you should end the process,

unistall the JDK via control panel and try again. Remember to close all programs when installing software.

After the JDK has successfully installed, go to C: and double-click the NetBeans file to start that installation. Select to install Glassfish. Apache can be installed later if desired.

Now we are ready to download and install NetBeans. Navigate to the URL below and select the download button for NetBeans 8.0 Java EE version as shown. Download to C:.
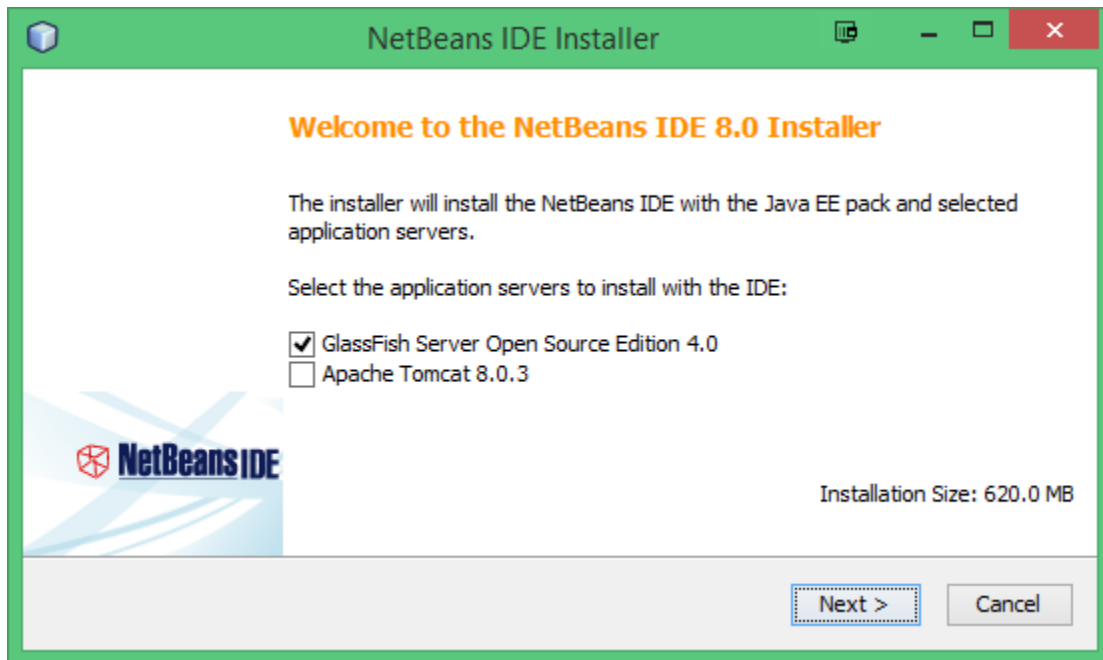
NetBeans IDE 8.0 Download

| Supported technologies *	Java SE	Java EE
NetBeans Platform SDK	●	●
Java SE	●	●
Java FX	●	●
Java EE		●
Java ME		
HTML5		●
Java Card™ 3 Connected		
C/C++		
Groovy		
PHP		
Bundled servers		
GlassFish Server Open Source Edition 4.0		●
Apache Tomcat 8.0.3		●
	Download	Download
	Free, 90 MB	Free, 191 MB

To install NetBeans, double-click on the file netbeans-8.0-javaee-windows.exe.
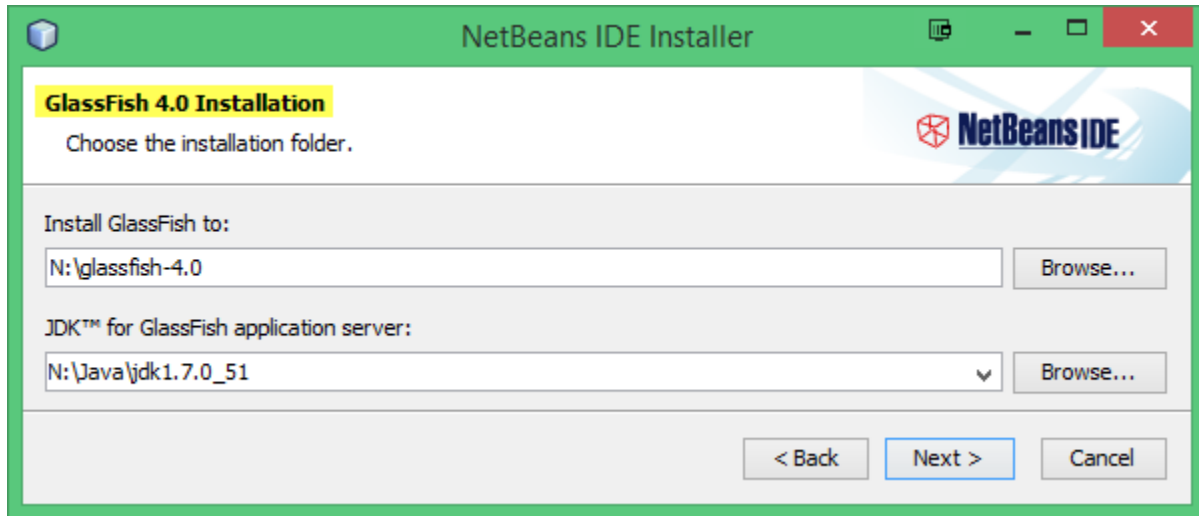
Make the appropriate selections for NetBeans 8.0. Install to C:.
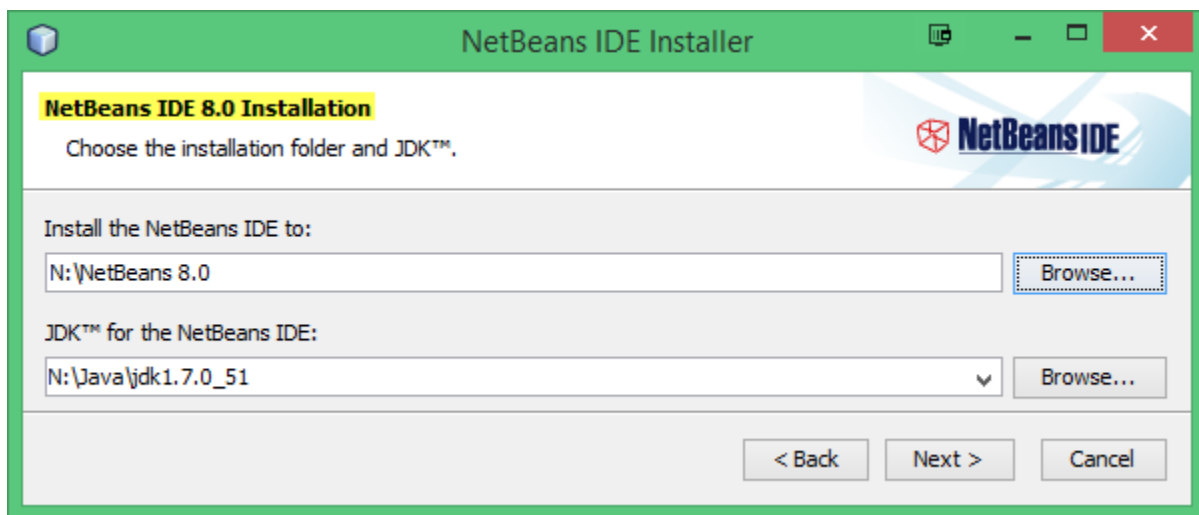
Make the appropriate selections for Glassfish 4.0 and click the Next button. Choose to automatically check for updates in the next dialog box (if that dialog box appears). After a few minutes NetBeans and Glassfish will install.

NetBeans and Glassfish should now be installed. If there is not a NetBeans shortcut on the desktop, you can create one by navigating to the netbeans64.exe location and R-clicking on the netbeans64.exe and selecting Create shortcut. Place the shortcut on the desktop or another location of your choosing.

Double-click your netbeans64.exe shortcut to start NetBeans.

To prevent the default Java DB server (know as Derby) from automatically starting each time Glassfish is started, select the Services group | Servers | R-click Glassfish | Properties and Servers dialog will appear.
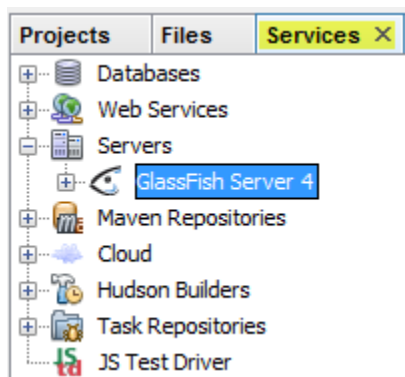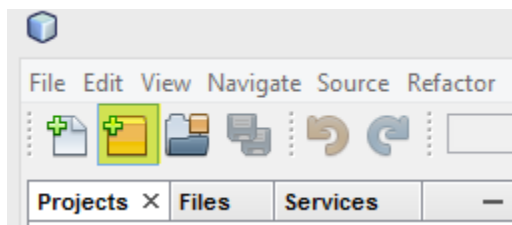
Then uncheck the "Start Registered Derby Server" checkbox as shown. In this course, we will utilize MySQL and not the Apache Derby database that installs by default with NetBeans. MySQL will be installed as part of the XAMPP stack with is covered in a later tutorial.
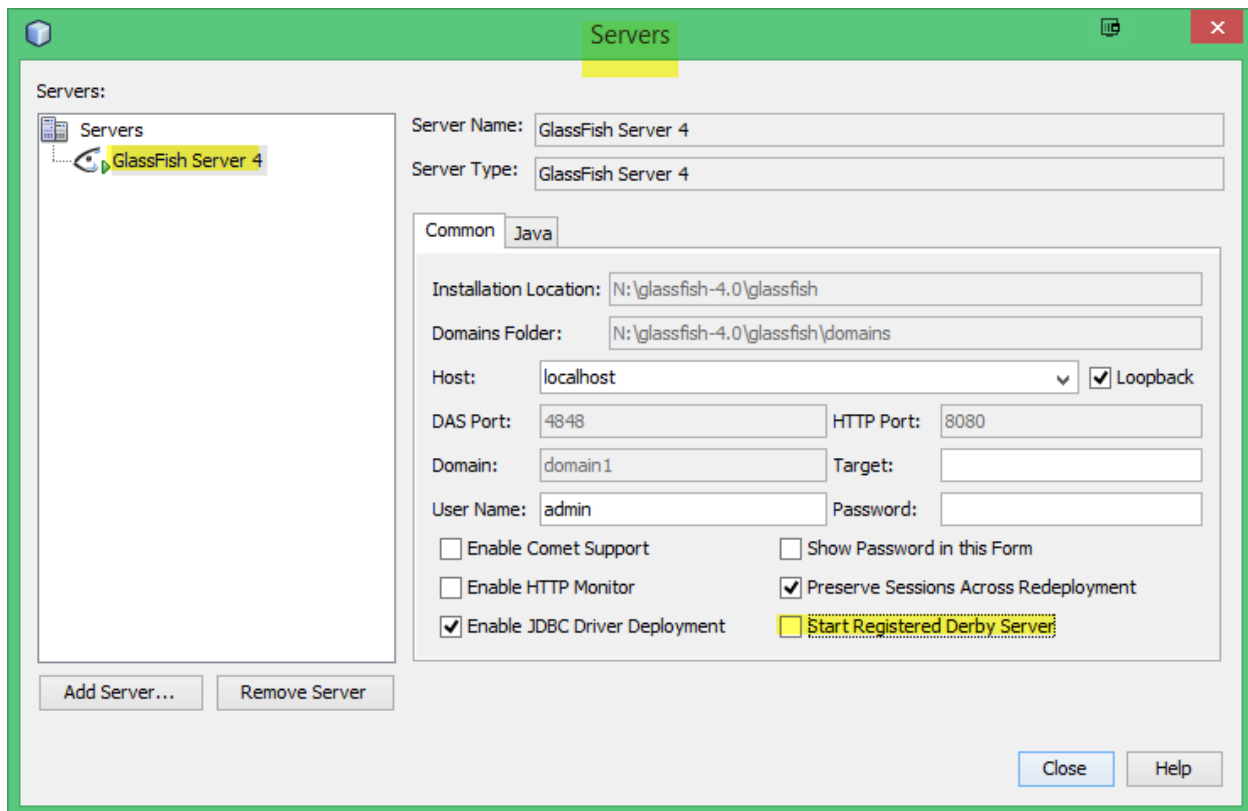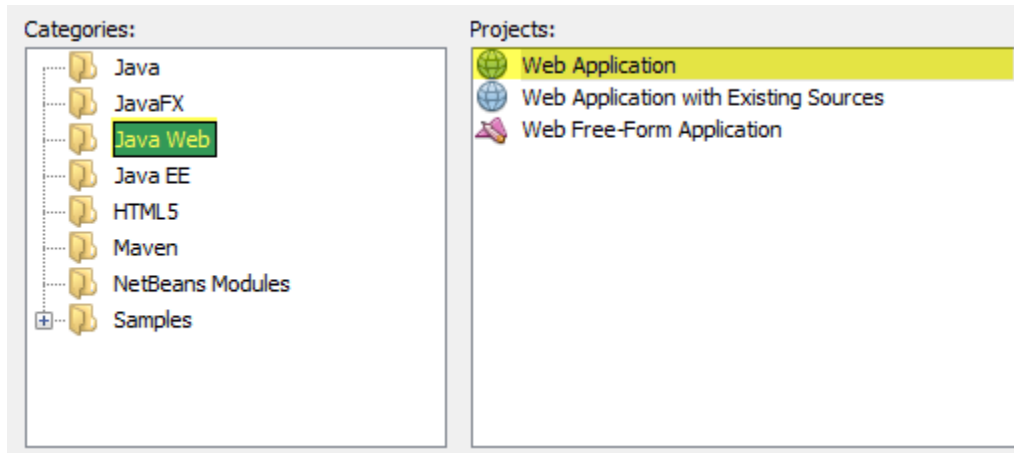
We are now ready to create a new NetBeans project. Click the Create a New project button.
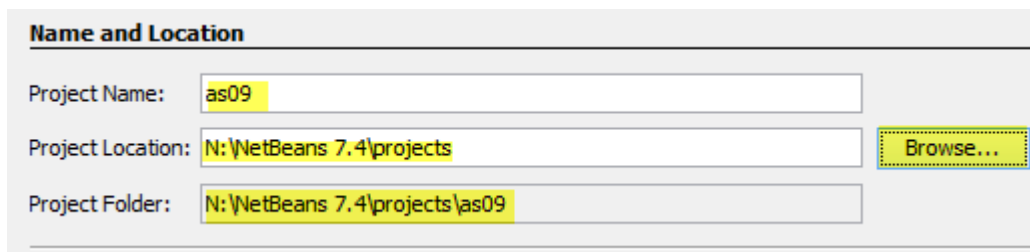
In the next dialog box make the following selections.

Categories:	Projects:
Java	Web Application
JavaFX	Web Application with Existing Sources
Java Web	Web Free-Form Application
Java EE	
HTML5	
Maven	
NetBeans Modules	
Samples	

Set the name and location of your web application.

Name and Location

Project Name:	as09
Project Location:	N:\NetBeans 7.4\projects	Browse...
Project Folder:	N:\NetBeans 7.4\projects\as09

If this is the first time creating a web application in this installation of NetBeans, you will receive a warning at the bottom of the next dialog box.

No servers are registered in the IDE. To register a server, click the "Add..." button.
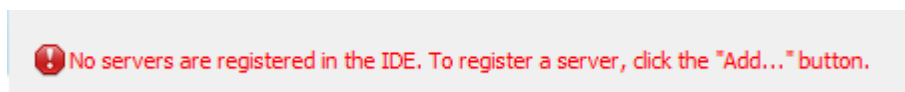
Click the Add button to register a Tomcat server using the appropriate settings. The username admin and password admin are defaults and will work. You can set new usernames and password by modifying the tomcat_users.xml in the Apache conf directory.

Installation and Login Details

Specify the Server Location (Catalina Home) and login details

Server Location: N:\Apache Tomcat 7.0.41   Browse ...

☐ Use Private Configuration Folder (Catalina Base)

Catalina Base: Browse ...

Enter the credentials of an existing user in the manager or manager-script role
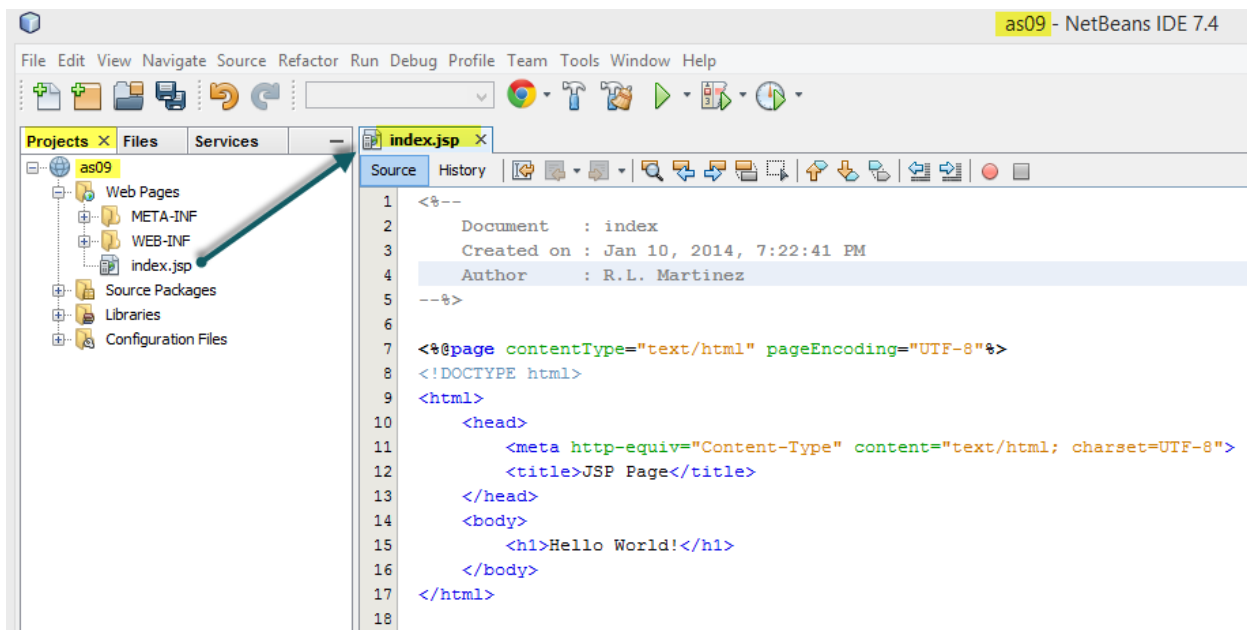
Username: admin

Password: •••••

☑ Create user if it does not exist

Accept the default settings in the next window and select Next. No frameworks are required for this course so none should be selected in the next window. Select Finish. A Hello World (index.jsp) is created like below. With the Projects tab selected a logical view of the project is displayed.

Looking at the results I see I made a mistake naming the project. No problem. R-click the project name and select Rename. When renaming a project, it is usually advisable to rename the project folder also.
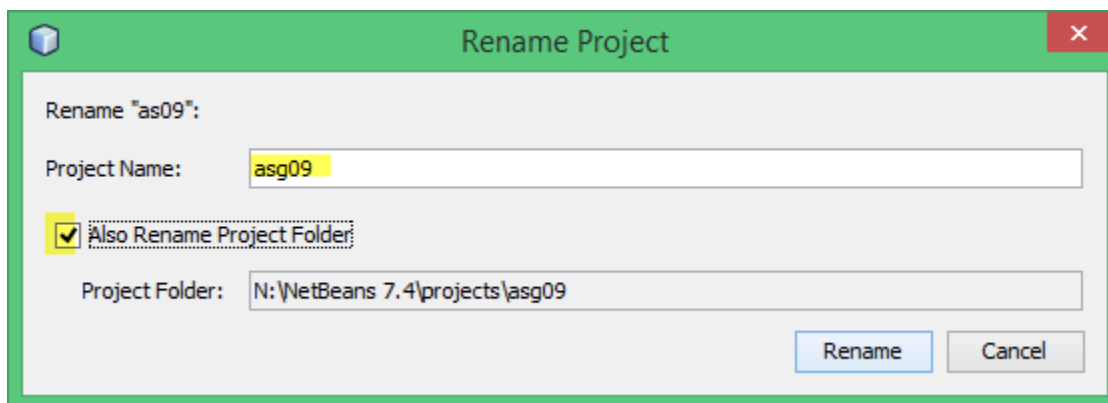

With the project open, select the Run Project ▷ button in the top menu to test the project. Tomcat will automatically be started when the run project button is selected provided you have specified a web server with your project which was done above. If not, R-click the project name to specify the server and select Specify Web Server at the bottom of the menu or in Properties.
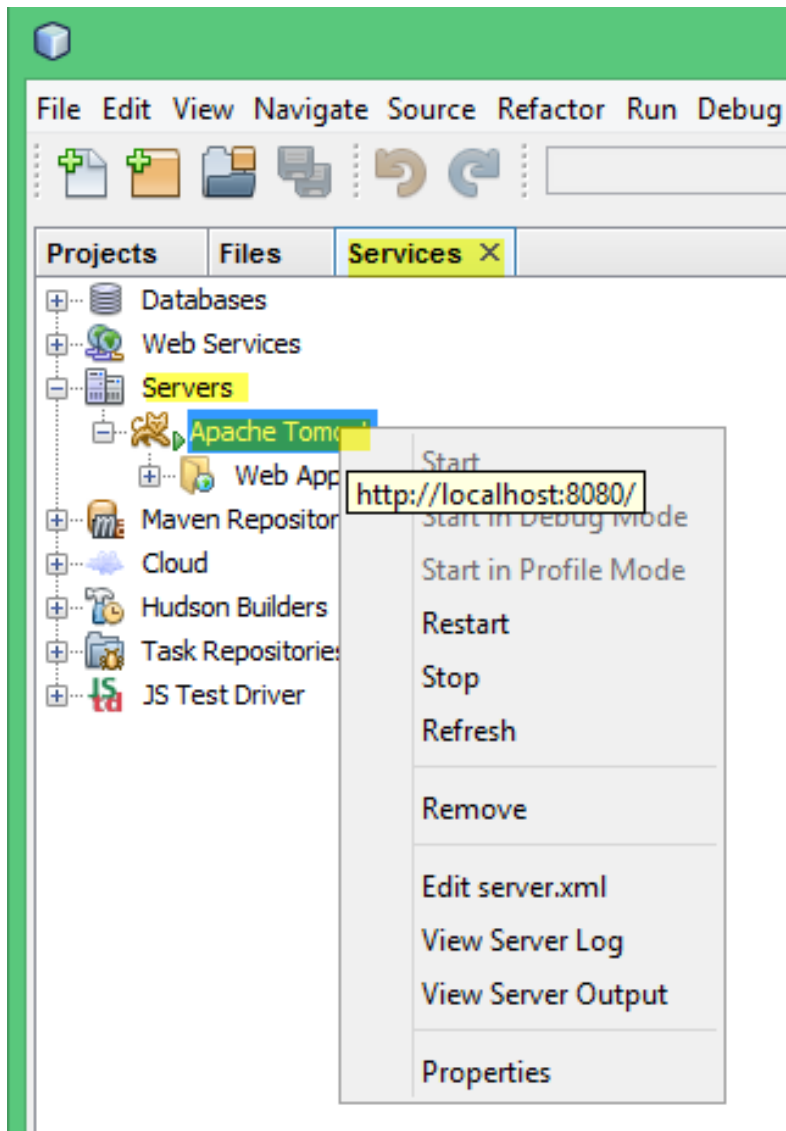
Tomcat can also be started and stopped independent of a project start by the alternatives shown below.

Select the Apache Tomcat tab:
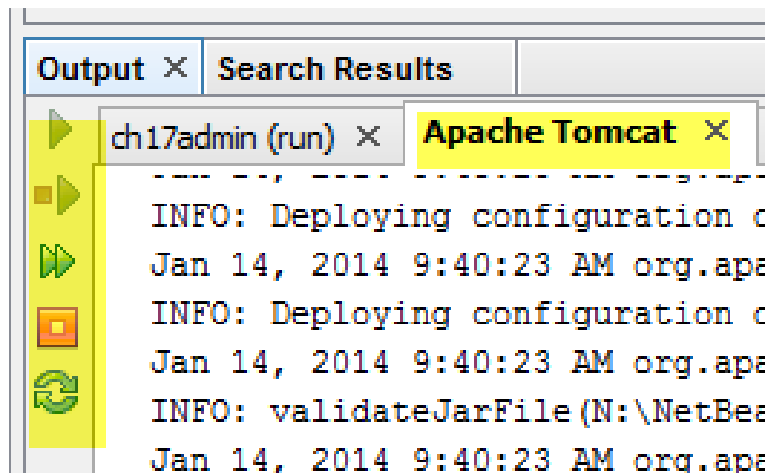
R-click the Servers icon under the Services tab.

When the project is run the default page of the project (index.jsp) will open which appears below. Notice the URL that appears in the address box. That contains the default name of the server (localhost), the default port that Tomcat listens on (8080), and the path to the project which is in the asg09 folder.
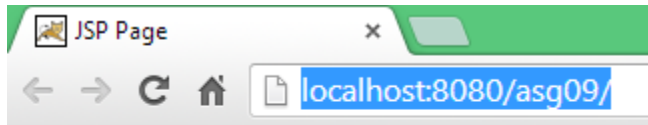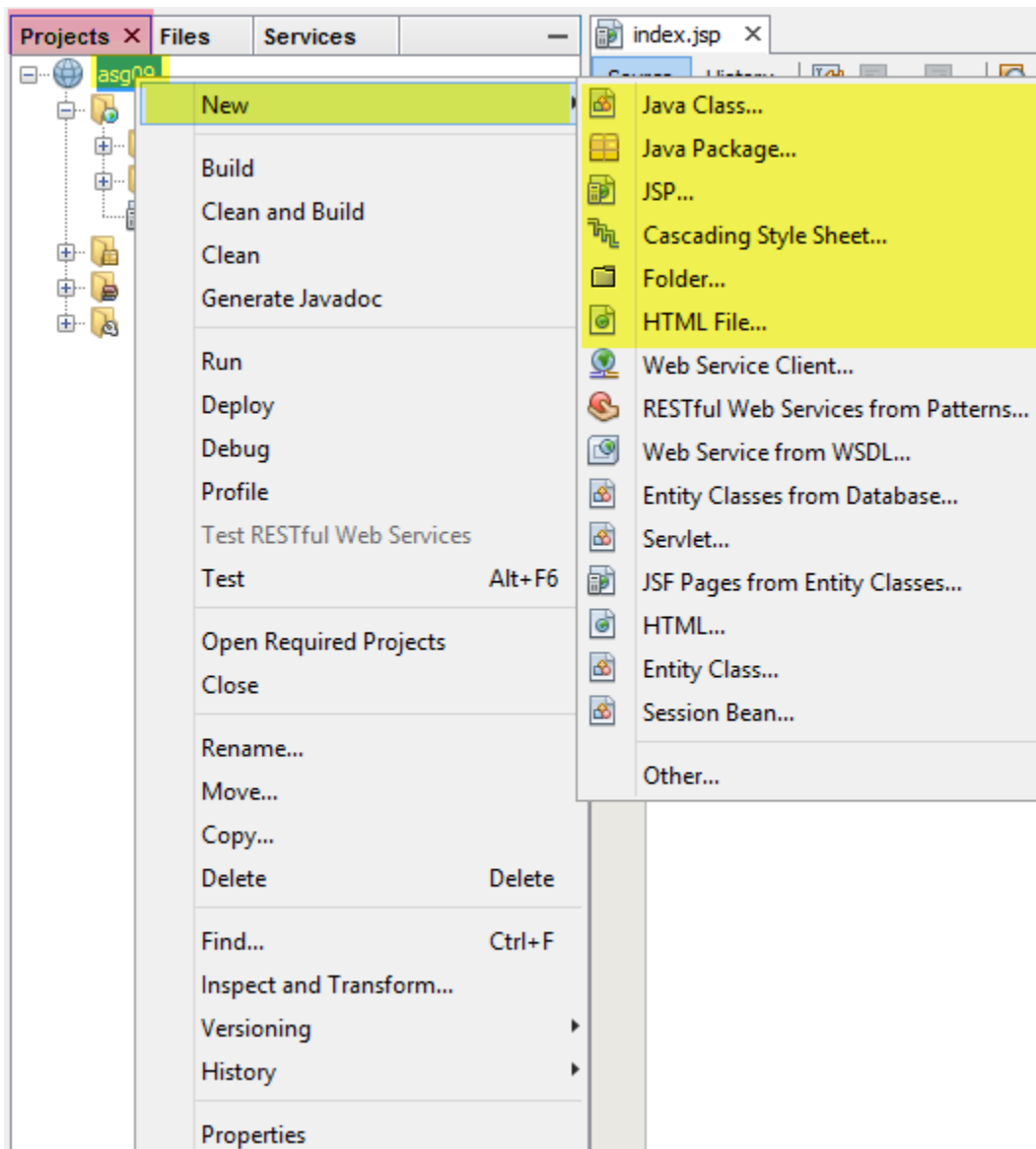
To add files to your project, select the Projects tab, R-click the project name (asg09 in the example), select New, and then the new file type. The types hightlighted are those most often used in this course.

Creating packages: Packages provide a convenient means to organize Java classes and namespaces. NetBeans simplifies the creation and use of packages. To create a package to contain some Java class files, R-click on the project name, select New, and then Java Package… The dialog box below appears. We will use the name college. Notice that NetBeans places the package in the standard location of src\java\{packagename}.
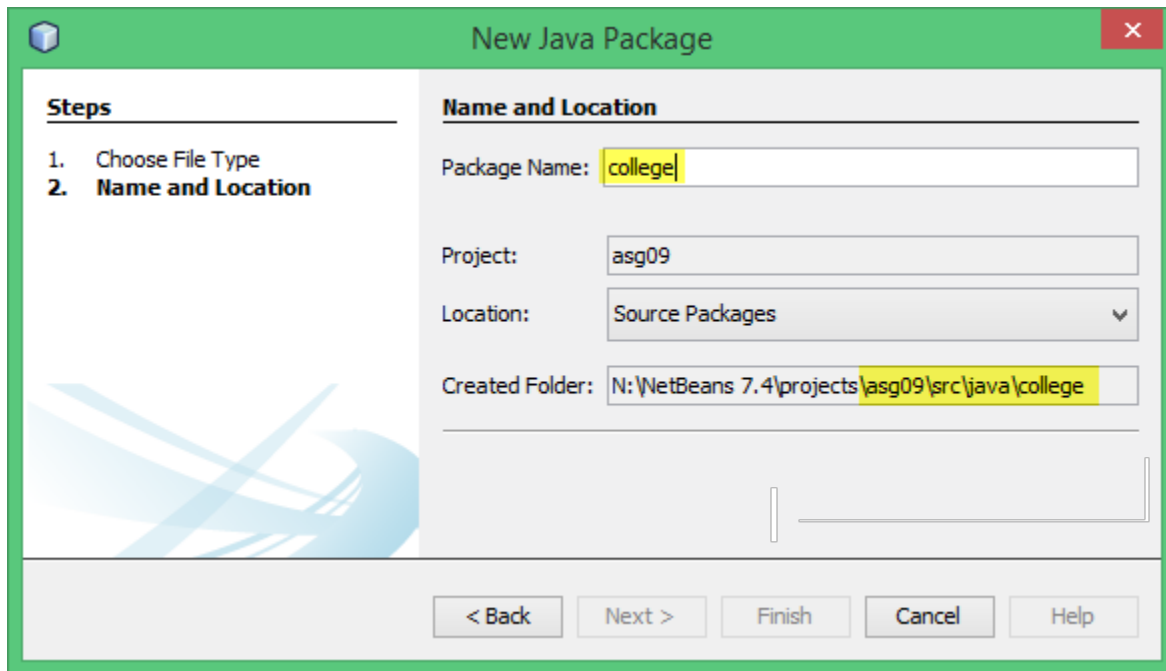
The package is created in the location shown in the different views. Recall that the Projects tab on the left shows a logical configuration of the project while the Files tab displays the actual file and folder structure.
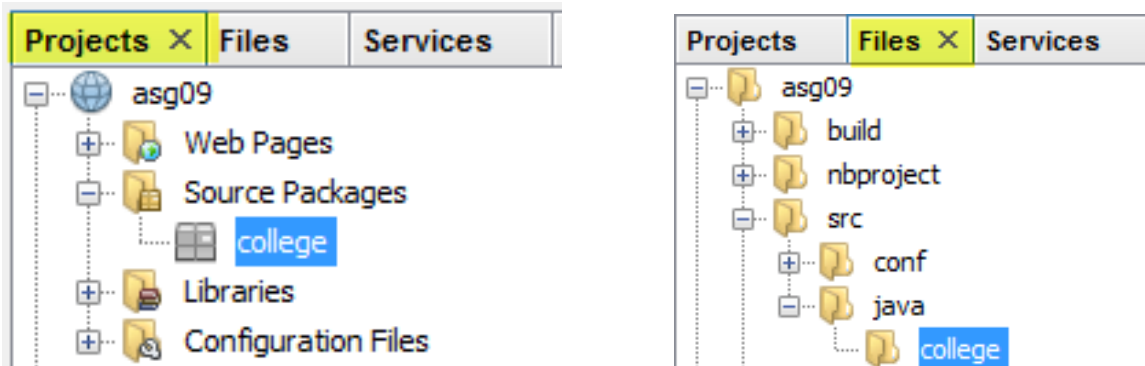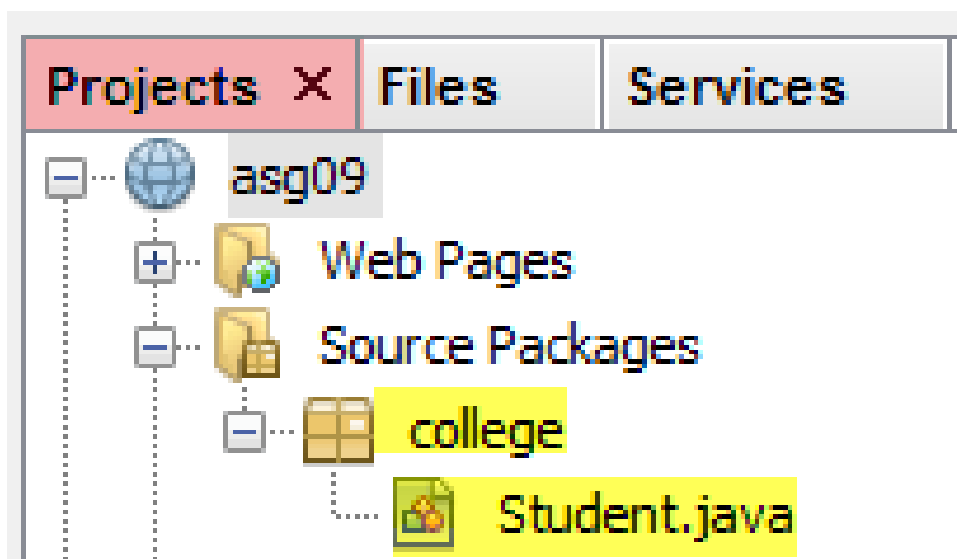

After the package is created, you can add a Java class file to the package by R-clicking the package name and selecting Java class… The class file is named Student in the example, placed in the college package, and in the path shown.
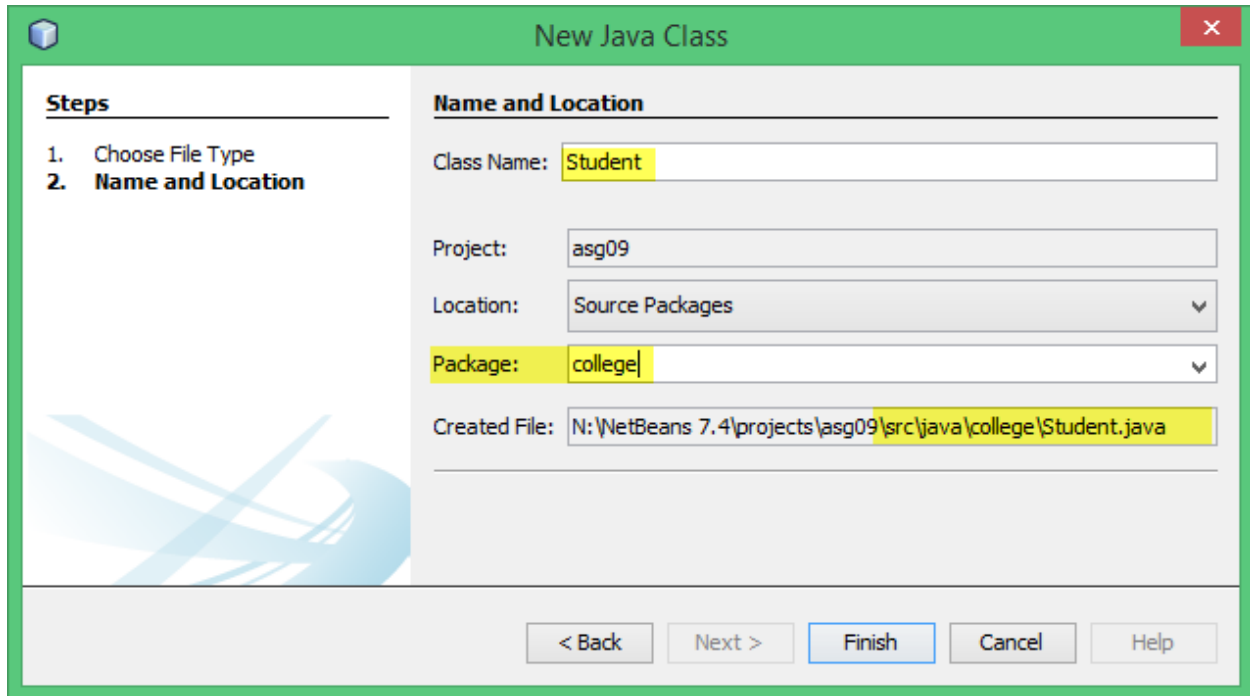
The college package now contains the Student.java class file. By the way, the globe icon identifies asg09 as a web project.

Tomcat creates the following starter file Student.java from the .java template.
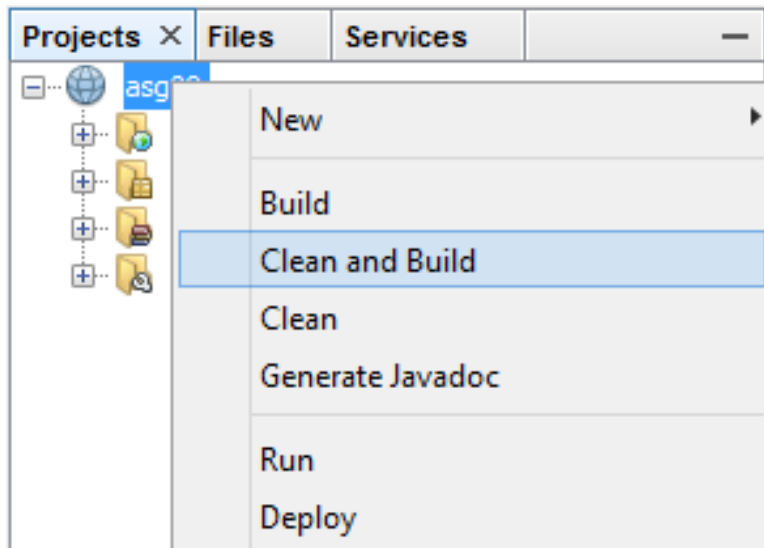
```
1   /*
2    * To change this license header, choose
3    * License Headers in Project Properties.
4    * To change this template file, choose Tools |
5    * Templates and open the template in the editor.
6    */
7
8   package college;
9
10  /**
11   *
12   * @author Rudy
13   */
14  public class Student {
15
16  }
```

Recall from the course material that an import statement like that below is used to import the package and its class files for use in a .jsp.
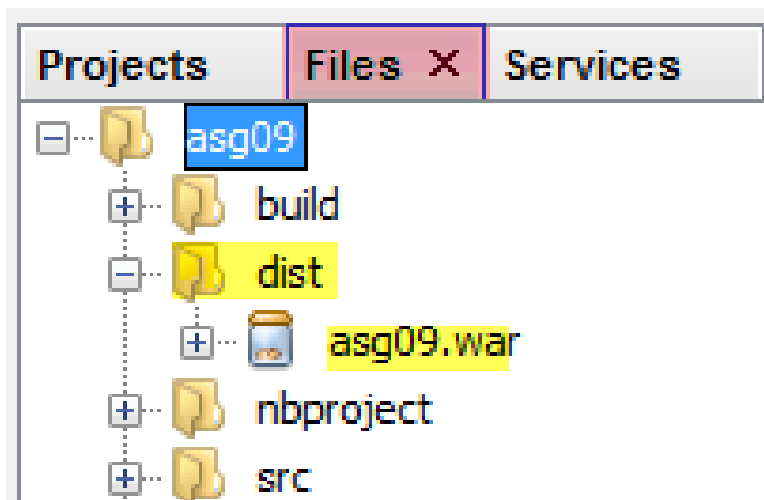
```
<%@ page import="college.Student, data.*, java.io.*, java.util.*" %>
```

After testing the project and when ready to upload to the student website, R-click the project and select Clean and Build.

After a successful Clean and Build, NetBeans automatically creates the dist (for distribution) folder and places the asg09.war file in the asg09/dist folder.

The asg.war file contains all files, in their proper locations, required for a servlet container to run the website. It is this file that should be uploaded to the student website per the instructions located at:
http://www.austincc.edu/rmartin6/websitesetupandpublishing.pdf

This link can also be located in Blackboard in the JSP Assignments section.

You are now ready to code your projects asg09-asg12. Good luck with your efforts and enjoy the process.