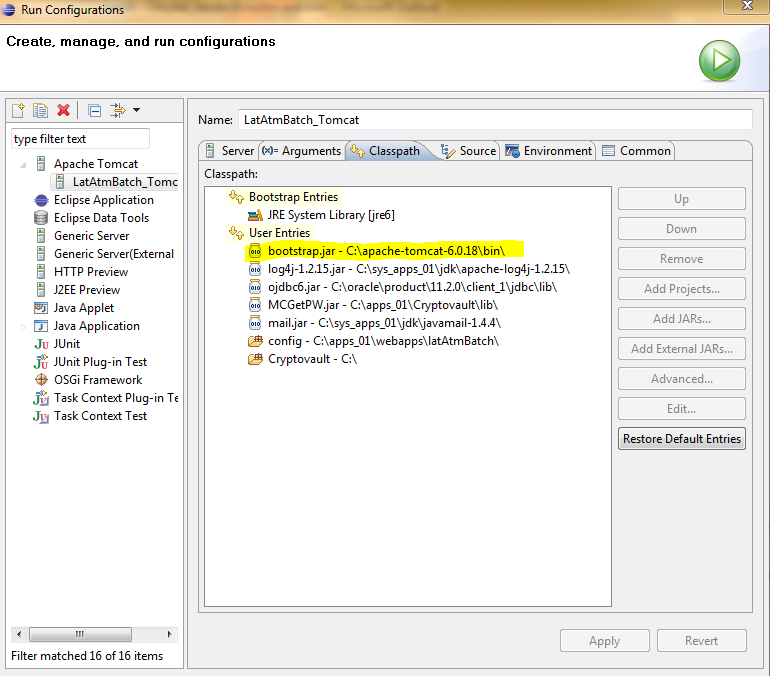
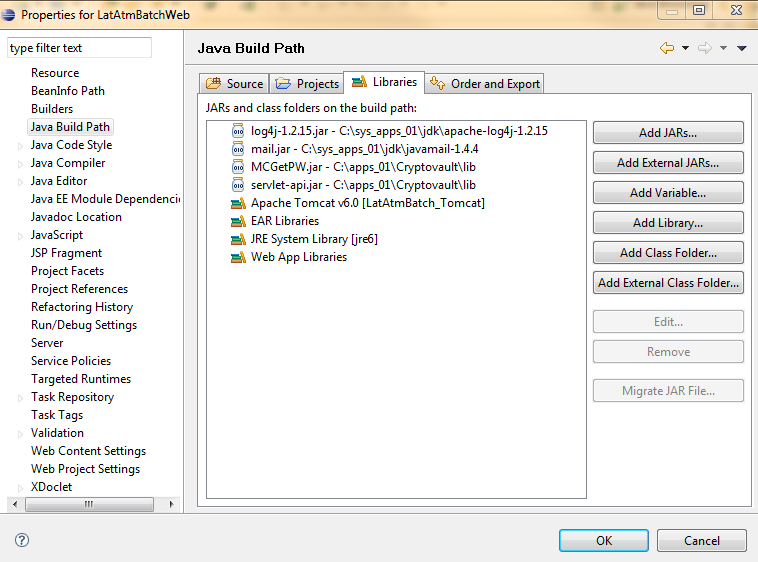
1) Run Configurations

* Right click on your project
  + then click "Run" -> "Run Configurations..."
  + there check your settings for you project (e. g. in my case it was the Apache Tomcat)
  + here look into you tab "Classpath" under "User Entries"



2) Project classpath

* Right click on your project -> "Properties" -> "Java Build Path"
  + now check the "Source" Tab as well as your "Libraries" Tab
  + Problems should be marked red in the "Libraries" tab
* 

<http://java67.blogspot.com/2012/08/what-is-path-and-classpath-in-java-difference.html>

What is PATH and CLASSPATH in Java - Path vs ClassPath

**What is PATH and CLASSPATH in Java**  
PATH and CLASSPATH are two most important environment variable of Java environment which is used to find JDK binaries used to compile and run Java in windows and Linux and class files which are compile Java byte codes. From my personal experience I can say that PATH and CLASSPATH are two most problematic thing for beginners in Java programming language due to two reasons; first because in most of Java courses nobody tell details of What is PATH and CLASSPATH, What does PATH and CLASSPATH do, What is meaning of setting PATH and CLASSPATH, What happens if we do not set them, Difference between PATH vs CLASSPATH in Java or simply [How Classpath works in Java](http://javarevisited.blogspot.sg/2011/01/how-classpath-work-in-java.html) etc. These basic question which answers most of details about PATH and CLASSPATH in Java are mostly not answered until Java programmer itself acquire these knowledge, Things may been changed now days but important of PATH and CLASSPATH is still high. Most common cause of dreaded error like [java.lang.NoClassDefFoundError](http://javarevisited.blogspot.sg/2011/06/noclassdeffounderror-exception-in.html) and [java.lang.ClassNotFoundException](http://javarevisited.blogspot.sg/2011/08/classnotfoundexception-in-java-example.html) are incorrect or misconfigured CLASSPATH in Java.

**Difference between PATH and CLASSPATH in Java**

Here are some of the common difference between PATH vs CLASSPATH in Java :  
  
1)Main difference between PATH and CLASSPATH is that  PATH is an environment variable which is used to locate JDK binaries like "java" or "javac" command used to run java program and compile java source file. On the other hand CLASSPATH environment variable is used by System or Application ClassLoader to locate and load compile Java bytecodes stored in .class file.  
  
2) In order to set PATH in Java you need to include JDK\_HOME/bin directory in PATH environment variable while in order to set CLASSPATH in Java you need to include all those directory where you have put either your .class file or JAR file which is required by your Java application.  
  
3) Another significant difference between PATH and CLASSPATH is that PATH can not be overridden by any Java settingsbut CLASSPATH can be overridden by providing command line option -classpath or -cp to both "java" and "javac" commands or by using Class-Path attribute in Manifest file inside JAR archive.  
  
4) PATH environment variable is used by operating system to find any binary or command typed in shell, this is true for both Windows and Linux environment while CLASSPATH is only used by Java ClassLoaders to load class files.  
  
These were some notable difference between PATH vs CLASSPATH in Java and they are worth remembering to debug and troubleshoot Java related issues.   
  
**How to set PATH and CLASSPATH in Windows and Unix**  
[What is PATH and CLASSPATH in Java , Difference between them](http://javarevisited.blogspot.sg/2011/06/comparator-and-comparable-in-java.html)If you are familiar with DOS operating system and how to use command prompt in Windows or shell in Linux setting PATH and CLASSPATH is trivial exercise. Both PATH and CLASSPATH are environment variable and can be set using export in Linux and using set keyword in DOS and Windows as shown below:  
  
Command to set PATH in Windows  
  
set PATH=%PATH%;C:\Program Files\Java\JDK1.6.20\bin  
  
Command to set PATH in UNIX/Linux  
  
export PATH = ${PATH}:/opt/Java/JDK1.6.18/bin  
  
Look at the difference between two commands, in Linux use colon(:) as separator and in Windows use semi-colon(;) as separator.  
  
Command to set CLASSPATH in windows  
  
set CLASSPATH=%CLASSPATH%;C:\Program Files\Java\JDK1.6.20\lib  
  
Command to set CLASSPATH in Unix/Linux  
  
export CLASSPATH= ${CLASSPATH}:/opt/Java/JDK1.6.18/lib  
  
Also don't forget to include current directory, denoted by dot(.) to include in CLASSPATH, this will ensure that it will look first on current directory and if it found the class it will use that even if that class also exists in another directory which exists in CLASSPATH.