

## Working with the classpath in Eclipse

To access a class from your Java code requires that class be on the classpath. There are two set of classes that are always on the classpath:

- Classes within the same package
- Classes in the java.lang package

There are four steps to working with external classes within an Eclipse project:

- The jar file must be accessible by Eclipse, it is recommended a Jars project be created within Eclipse to hold all jars used. This makes it easy to put the jars into your source code control system.
- The jar file must be added to the project's build path (classpath). Add it to the Project's Java Build Path property, via the Libraries tab.
- The class must be imported into the code that will use it. Eclipse offers a number of shortcuts to make this easier, including the Source → Organize Imports menu.
- A method of the imported class must be called.

Eclipse offers several hints to let you know if you have classpath issues.

- If you have a red underline on an import statement, that means the jar file cannot be found in the classpath.
- If you have a red underline on the external class name, that means you are missing an import statement.
- If you have a red underline on the method you are calling, that means you have the signature (parameters/return type) wrong.

It is recommended that single jar files be added to the Project's Java Build Path individually, however if a set of jar files (i.e. EasyMock requires 3 jar files) is required, a User Library, holding the set of jar files is recommended.

User Libraries are managed via the Window → Preferences → Java → Build Path → User Libraries menu option.