

PATH VS CLASSPATH

The **PATH** environment variable contains a set of directories within the file system that the operating system uses to find executable files. When you type a command in command prompt, operating system will look for the corresponding executable in the current folder and then it will look inside every folder mentioned in the PATH environment variable in order. For instance, when you type javac, OS will search for java.exe within the current folder and then every folder mentioned inside in the PATH environment variable in order. If it can't find one, you'll get "Command not found" error message. The "bin" directory of your jdk contains the javac.exe file. Therefore, if you type "javac" in command prompt, you should have the "bin" directory in the PATH or you should execute from the bin directory.

If your path to jdk bin folder (where javac is there) is 'C:\Program Files\Java\jdk1.6.0_13\bin', then you can add it to the PATH environment variable in windows as:

```
set PATH=C:\Program Files\Java\jdk1.6.0_13\bin;%PATH%
```

%PATH% will be replaced by existing path PATH environment variable contents.

[N] Directory paths in an environment variable is separated using semicolon (;) in windows and colon (:) in Unix.

The **CLASSPATH** environment variable contains a set of directories within the file system or JAR file locations that JDK/JVM will search for Java class files while compiling/executing a Java program. Setting a class path can be done similar to setting PATH, just use CLASSPATH instead of PATH. You can also specify classpath along with java command using classpath or cp options as:

```
java -classpath path HelloWorld
```

or

```
java -cp path HelloWorld
```

[N] -classpath will take precedence over any CLASSPATH environment variable set.

[N] When working in the command prompt, the current directory is denoted by dot (.) and parent directory is denoted by dot dot (..)

[N] If CLASSPATH variable is not set, java will add the current directory to CLASSPATH, but if you provide CLASSPATH, then you need to explicitly add the current directory (.) to class path.

```
java -cp . HelloWorld
```

To add current directory with other folders in a classpath, separate it with a semicolon or colon as per the OS you use as:

```
...path;.;path...
```

Source : <http://javajee.com/path-vs-classpath>