

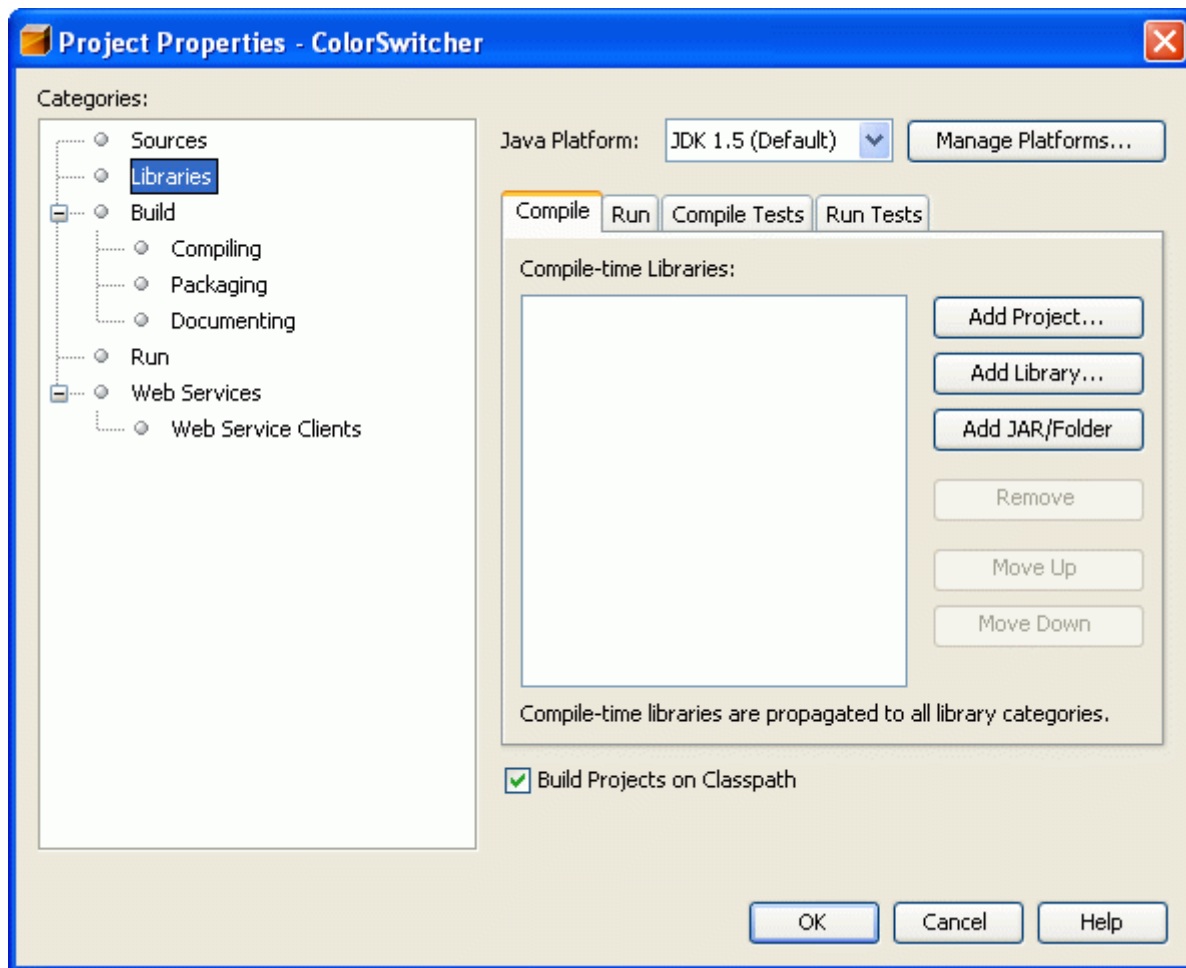
Managing Project's Classpath in NetBeans

Source http://www.netbeans.org/kb/50/using-netbeans/project_setup.html#96961

Adding a group of class files to a project's classpath tells the IDE which classes the project should have access to during compilation and execution. The IDE also uses classpath settings to enable code completion, automatic highlighting of compilation errors, and refactoring. You can edit the classpath declarations for an existing project in the Project Properties dialog box.

- **Standard projects.** In standard projects, the IDE maintains separate classpaths for compiling and running your project, as well as compiling and running JUnit tests (for J2SE applications). The IDE automatically adds everything on your project's compilation classpath to the project's runtime classpath. You can add JAR files, libraries, and dependent projects to the project's compilation classpath in the Compile tab of the Project Properties dialog box. You can also right-click the Libraries node in the Projects window and add JAR files, libraries and projects to your project.
- **Free-form projects.** In free-form projects, your Ant script handles the classpath for all of your source folders. You declare the classpath in the New Project wizard when you set up your free-form project. The classpath settings for free-form projects only tell the IDE what classes to make available for code completion and refactoring. You can declare the classpath for free-form projects using the Java Sources Classpath panel in the Project Properties dialog box. For more, see [Managing the Classpath in Free-form Projects](#) below.

If you have attached Javadoc and source files to a JAR file in the Library Manager, the IDE automatically adds the Javadoc and source files to the project when you register the JAR file on a project's classpath. You can step into classes and look up Javadoc pages for the classes without configuring anything else.



Project Properties dialog box

~~~ End of Article ~~~