

For those of us NOT using Eclipse for Java but for these evil languages called „script languages“ like PHP, Python, Perl, JavaScript etc. Eclipse nowadays still lacks some features, but also has some good advantages against using things like vi.

If you're like me using Eclipse because you're dealing with a lot of different languages and want to have features like code completion, SCM-integration and stuff (and are too lazy to do this via vi), there's one huge missing part in Eclipse: An easy method of transferring files to a remote location.

I'm currently in a progress of moving from the slowly disintegrating Aptana back to Eclipse. Aptana had a real nice feature of remote connections and copying and even syncing between the local copy and the remote one. The barenaked Eclipse lacks this feature.

For clearance: I know, that this can be achieved by a team provider copying the files. But I already have a SCM-team provider in use for my projects, so that's no solution. You may ask now, why I'm not using the SCM to do this. The answer is simple: Because I'm using a development server. I will not checkin every change I do while fiddling with certain development issues. I tend to check in clean code.

I tried the nice, but outdated [\[\[http://esftp.sourceforge.net\]\]](http://esftp.sourceforge.net), but this lacks the feature of sftp key authentication which I needed.

Next thing I tried was the remote systems/target manager-functions of eclipse, but I honestly don't get them. I can make a nice remote explorer with them, but it's hard to simply transfer some files from a project (especially when you like to use hotkeys).

Finally I found the builder-concept of Eclipse and the „Program builder“ (which should be bundled with your Eclipse installation). With this one you can simply call an external program and do something.

It's very easy to just set up a program builder to use bash and scp to copy files from the working directory set to „`${project_loc}`“ to the remote server- especially when you're developing under a *nix-like OS.

I used „`/bin/bash`“ as the location, „`${project_loc}`“ as the working directory and this as arguments:

```
-c "scp -r * user@remote:/remotepath"
```

Easy copying sourcecode to remote using Eclipse Program Builder

Just press „Ctrl-B“ (or Cmd-B on Mac) after saving and tadaaa – works like a charm.