

TDK MacOSX

I'm reasonably sure this won't be of any use to anybody... but since we've got a nice wiki scratchpad, there seems no harm in scribbling down some stuff just in case. If anyone should find this useful, feel free to edit this comment and generally go wiki-nuts 😊

I played with the 2.2 version of the TDK earlier today, and it worked fine on my Debian system. That machine's CPU is really weak so I'm installing it to my G4 Powerbook. I've already got Tomcat 4.x running fine (and use [RobBook](#) as my primary webdev machine for my Turbine 2.1 app, which I need to port to 2.2). Also I've got both MySQL and PostgreSQL running fine on [RobBook](#).

So that's what these steps refer to. See why I don't think they'll be especially useful to anybody else? If you're starting where I started, you might just have a similar experience. But what are the odds your system matches mine? Shrug.

Here's what I did:

0. Ant 1.5, Tomcat 4.0.4, MySQL 3.23.53 all are good to go. Also Java 1.3.1, from Apple's OS X 10.2.3 distro I should think.

1. Aquired and unpacked tdk-2.2_01.tar.gz to /Users/rob/source/tdk-2.2

2. Updated build.properties for my test. I changed "newapp" to "test", and added a line "tdk.home=/Users/rob/source/tdk-2.2", and then pulled the ant trigger.

3. No love. Got error message "XML parser factory has not been configured correctly: Provider for javax.xml.parsers.SAXParserFactory cannot be found". I think I see why, my first go-around with ant on this box also stumbled on javax.xml stuff... and I ended up pointing my /usr/local/ant/lib/xerces-1.3.0.jar to the 2.1 TDK version, which I just removed before doing this test! Well, there's no shortage of other candidates returned by `locate xerces` so I adjusted my symbolic link to point to, um, how about the xerces-1.3.0.jar from my 2.1 app? Sure, that's the ticket. I should probably just copy that jar to /usr/local/ant/lib but I didn't.

4. Try again, worked! The test app generated fine, to webapps/test, on to the database settings in ~/source/tdk-2.2/webapps/test/WEB-INF/build/build.properties

5. Ratfarts. When I was resolving the issues of step 3 I accidentally did an `ant` on the Deb box, wiping out the critical DB parms. I had to figure them out again as I redo these steps on [RobBook](#). Blast. Would be a lot better if the job had stopped or not overwritten my first build or something. Mutter.

6. While I'm grumbling... I added `tdk.home=/Users/rob/source/tdk-2.2` to my generated build.properties file, same as was required to generate that file. The initial ant task of making my app's build.properties file should bring this over, agreed? Thought so. ha ha.

7. Here's the database config I came up with. I did not use "test" but instead "testapp" because the standard MySQL setup handles "test" as a special case, as I recall:

```
target.database = mysql

database.name = testapp
databaseHost = localhost
database = ${target.database}
databaseUrl = jdbc:mysql://${databaseHost}/${database.name}
createDatabaseUrl = ${databaseUrl}
buildDatabaseUrl = ${databaseUrl}
databaseDriver = org.gjt.mm.mysql.Driver
databaseUser = rob
databasePassword = smirk
```

8. Proceeded with `ant init` and... nope! Got error "java.lang.VerifyError: (class: org/apache/torque/task/TorqueJDBCTransformTask, method: execute signature: ()V) Incompatible object argument for function call", another peach of an error like one I seem to recall seeing before while hacking on the Turbine/Servlet/Java thing. Oh look, it's our friend xerces again! Things got better when I readjusted that ant library link, to point to the xerces supplied by TDK 2.2, good grief.

9. `ant init` got a bit farther, this time it stopped with "Unknown database 'testapp'". Time was, it would create the database if it didn't exist. No biggie, I created it manually. And then did the init thing again. And it worked!

10. Indeed, things seem good here, <http://localhost:8080/test/servlet/test> served fine and user turbine with password turbine got me in, I'm having big RDF fun now!

— [RobLeachman](#)