

Using VanillaDB

Database Systems
DataLab, CS, NTHU
Spring, 2018

Outline

- VanillaDB Core
 - Prepare Everything You Need
 - Server Properties
 - Starting Up VanillaDB
 - Console SQL Interpreter

Outline

- VanillaDB Core
 - Prepare Everything You Need
 - Server Properties
 - Starting Up VanillaDB
 - Console SQL Interpreter



VanillaDB

Simple, fast, and extensible database system prototypes.

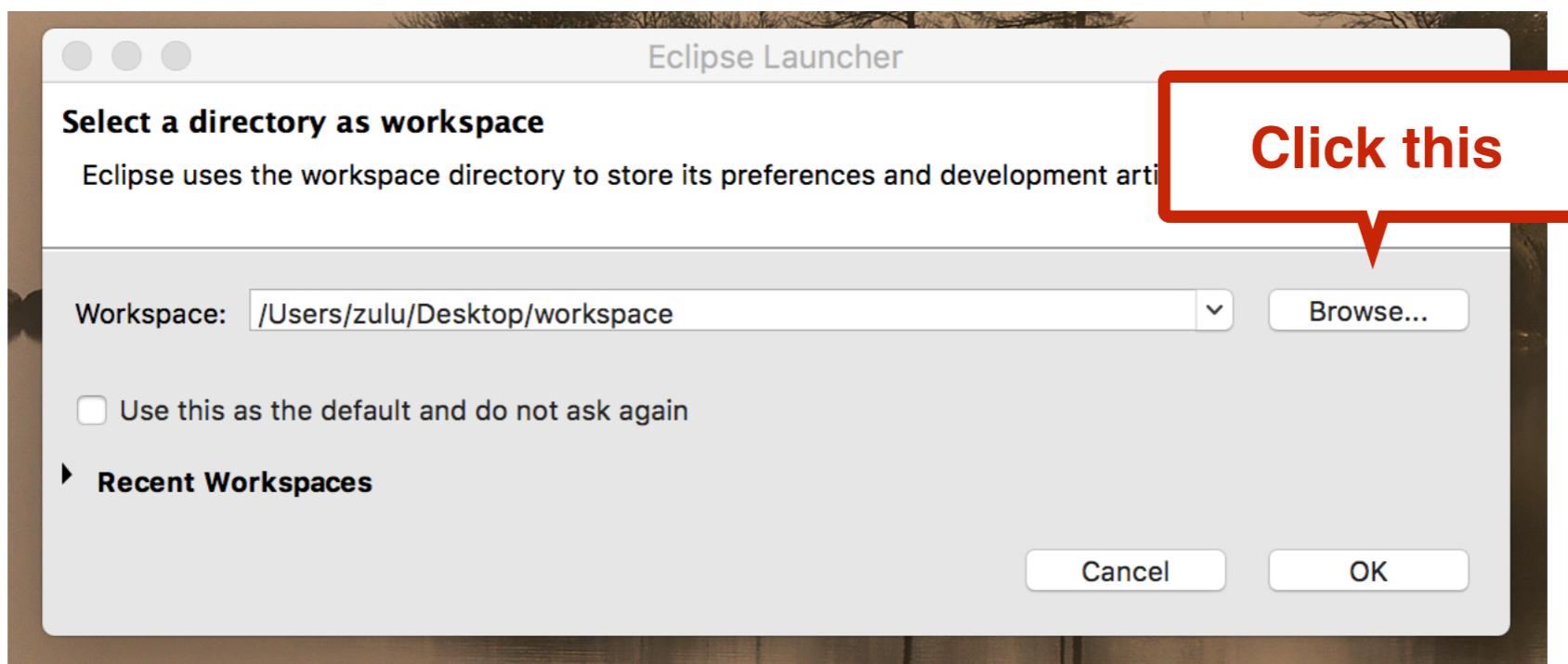
Installation

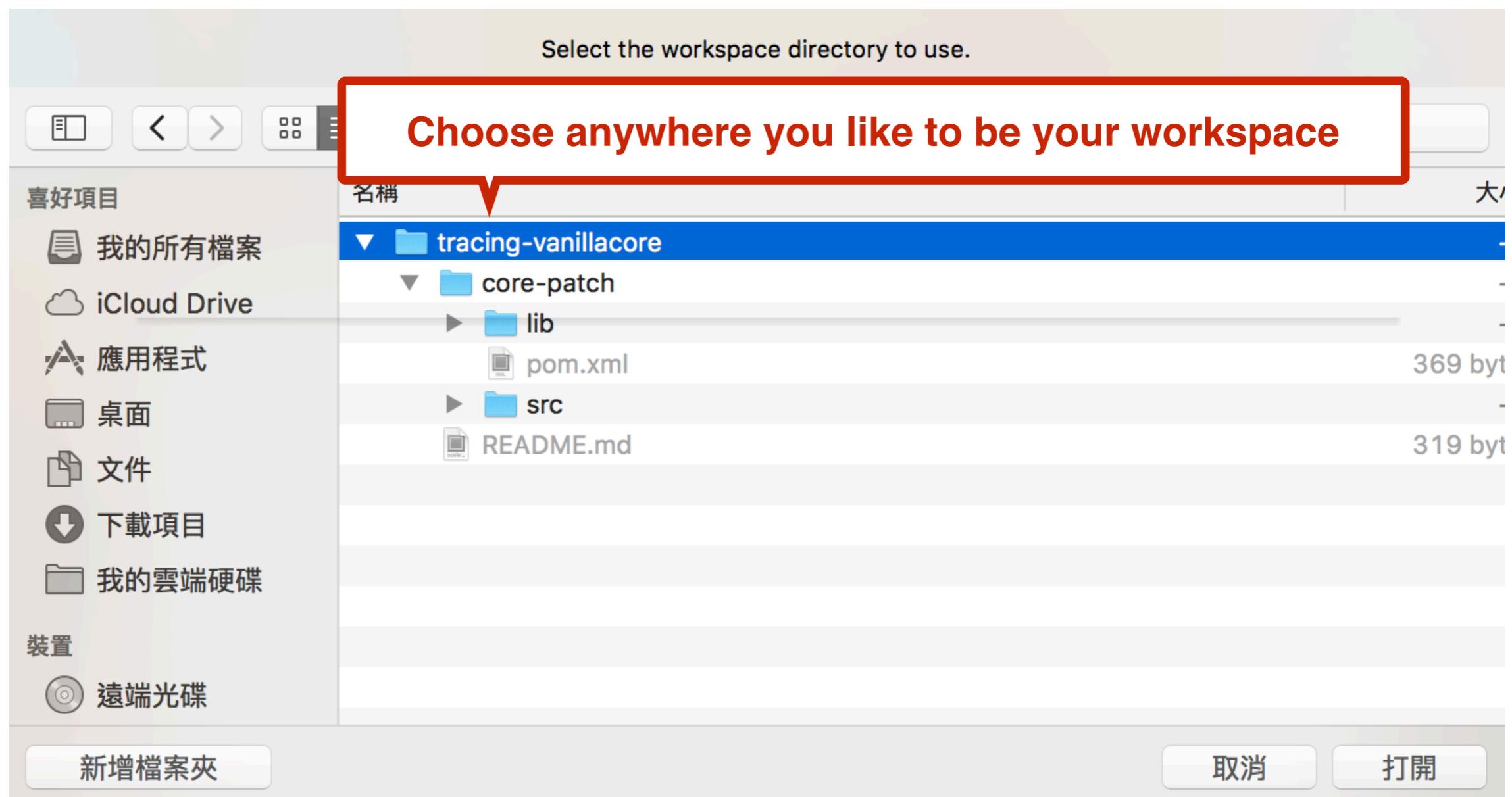
- JDK 8
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Eclipse
 - <http://www.eclipse.org/downloads/packages/>
 - Choose Eclipse IDE for Java [EE] Developers

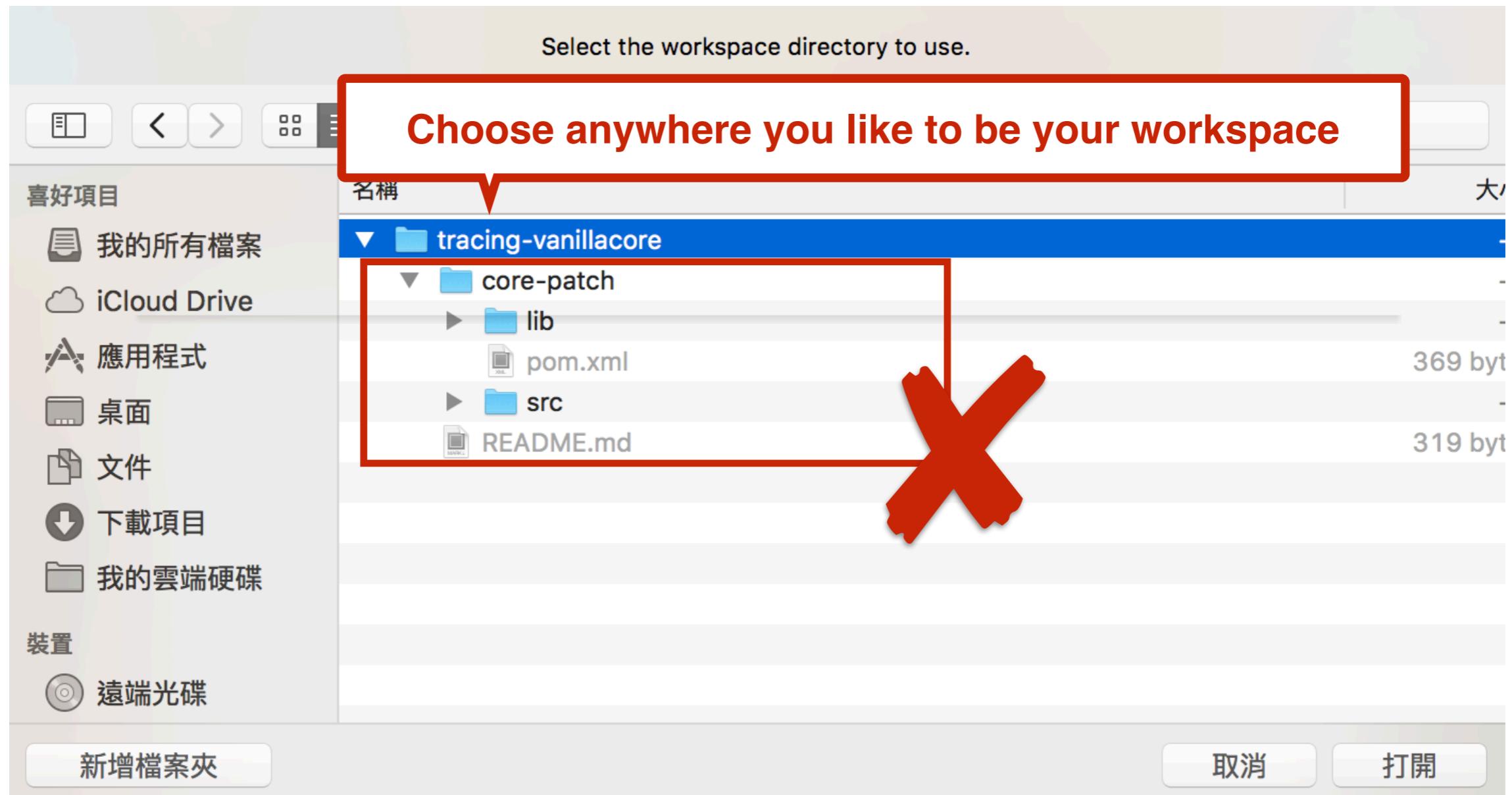
VanillaDB Project

- Clone VanillaDB here
 - <https://shwu10.cs.nthu.edu.tw/courses-databases-2018-spring/tracing-vanillacore.git>

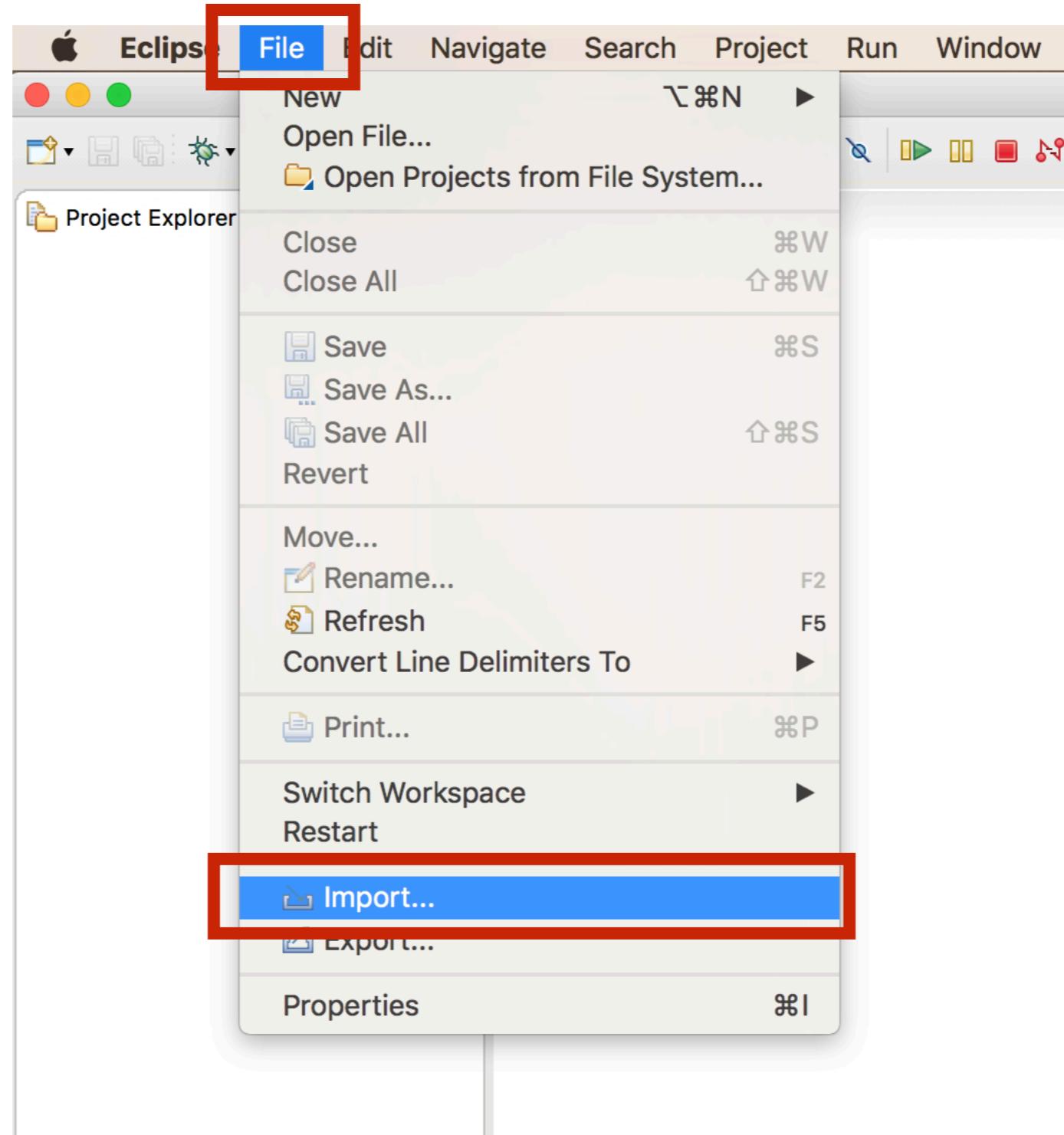
How to Import VanillaDB

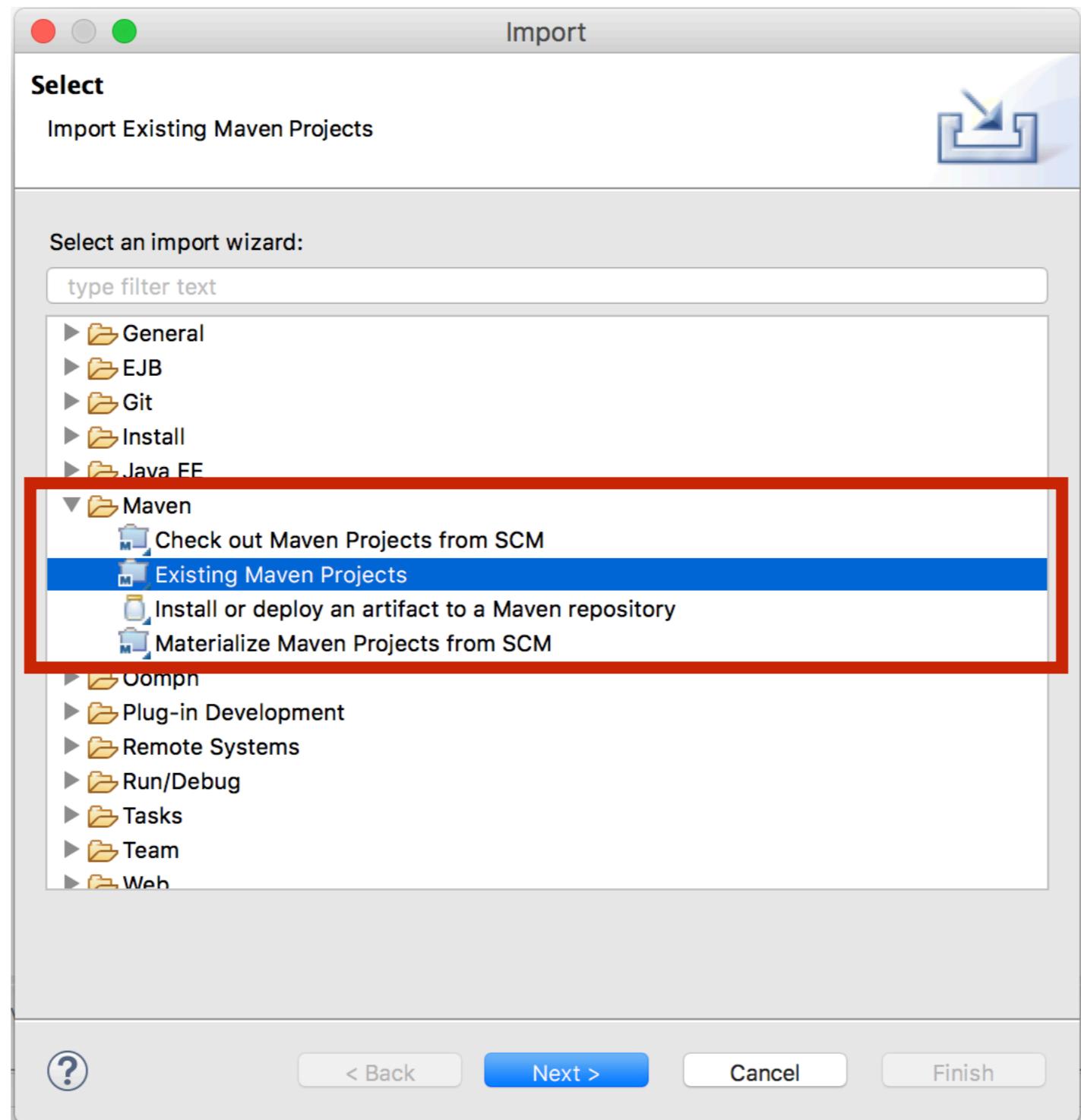


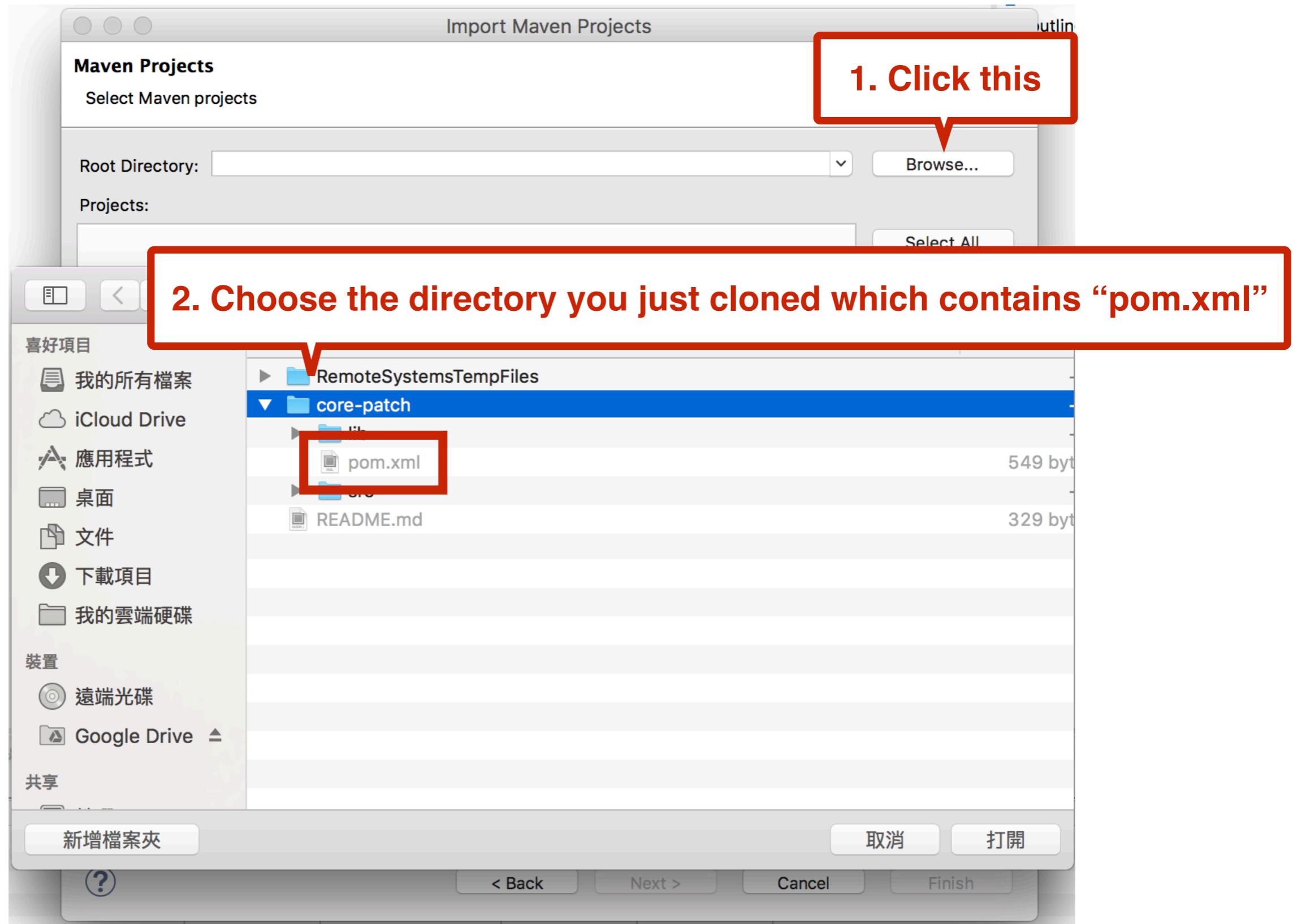


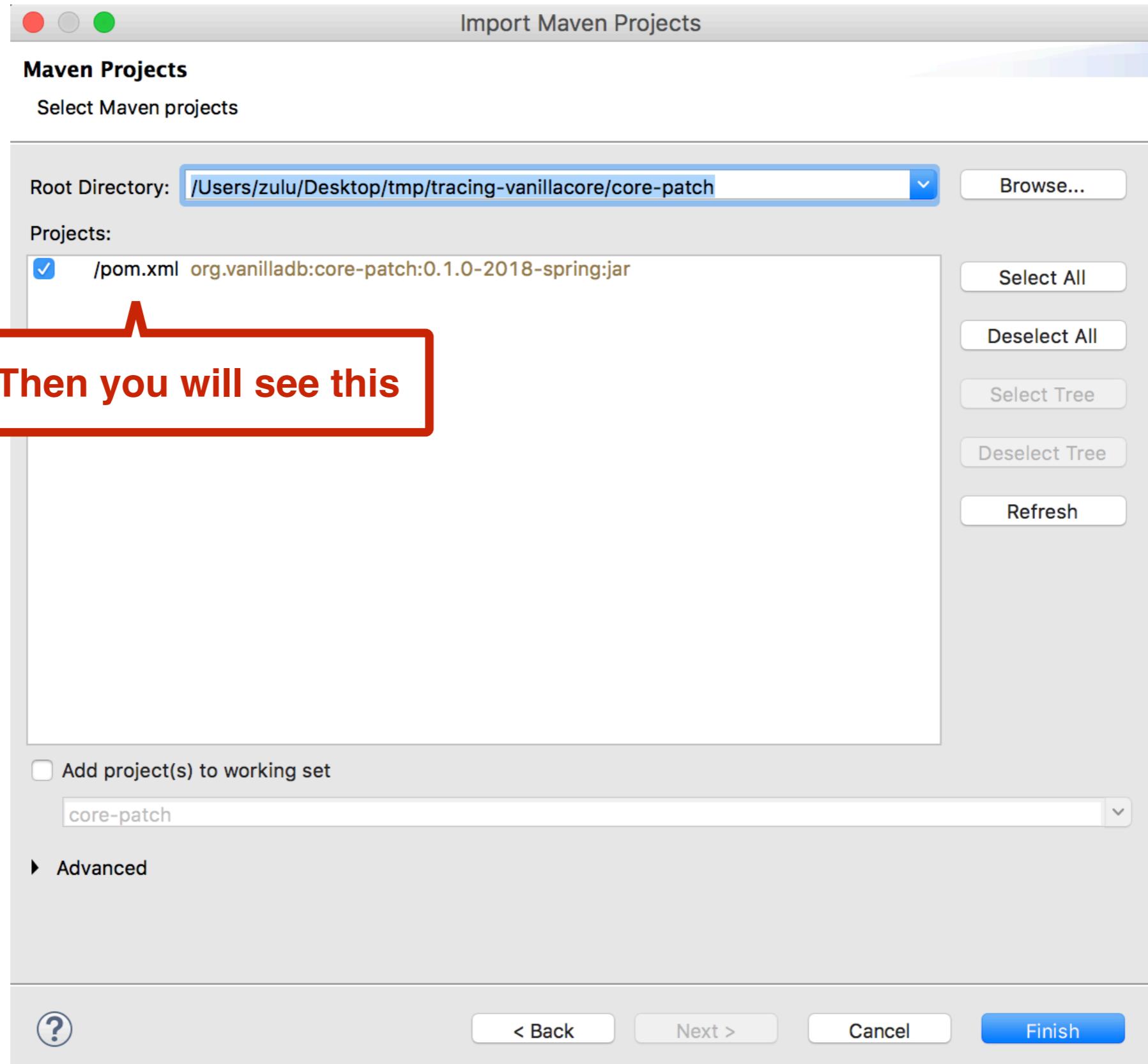


**But be careful not choose the directory which contains
“pom.xml” !!**

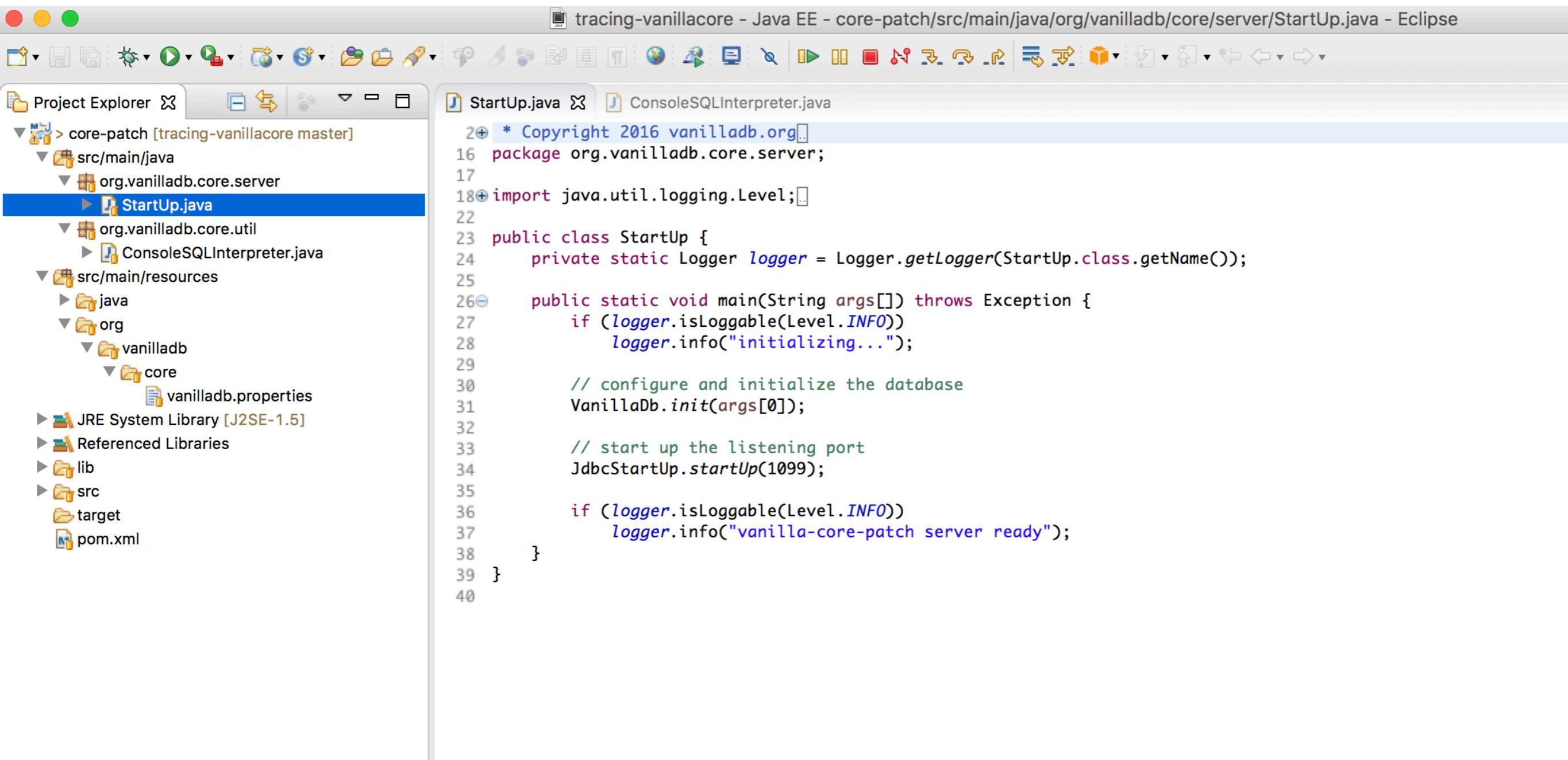








Done



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** tracing-vanillacore - Java EE - core-patch/src/main/java/org/vanilladb/core/server/StartUp.java - Eclipse
- Toolbar:** Standard Eclipse toolbar icons.
- Project Explorer:** Shows the project structure:
 - core-patch [tracing-vanillacore master]
 - src/main/java
 - org.vanilladb.core.server (selected)
 - StartUp.java
 - org.vanilladb.core.util
 - ConsoleSQLInterpreter.java
 - src/main/resources
 - java
 - org
 - vanilladb
 - core
 - vanilladb.properties
 - JRE System Library [J2SE-1.5]
 - Referenced Libraries
 - lib
 - src
 - target
 - pom.xml
- Editor Area:** Displays the content of StartUp.java:

```
2+ * Copyright 2016 vanilladb.org
16 package org.vanilladb.core.server;
17
18+ import java.util.logging.Level;
22
23 public class StartUp {
24     private static Logger logger = Logger.getLogger(StartUp.class.getName());
25
26+     public static void main(String args[]) throws Exception {
27         if (logger.isLoggable(Level.INFO))
28             logger.info("initializing...");
29
30         // configure and initialize the database
31         VanillaDb.init(args[0]);
32
33         // start up the listening port
34         JdbcStartUp.startUp(1099);
35
36         if (logger.isLoggable(Level.INFO))
37             logger.info("vanilla-core-patch server ready");
38     }
39 }
40 }
```

Outline

- VanillaDB Core
 - Prepare Everything You Need
 - Server Properties
 - Starting Up VanillaDB
 - Console SQL Interpreter

VanillaDB Properties File

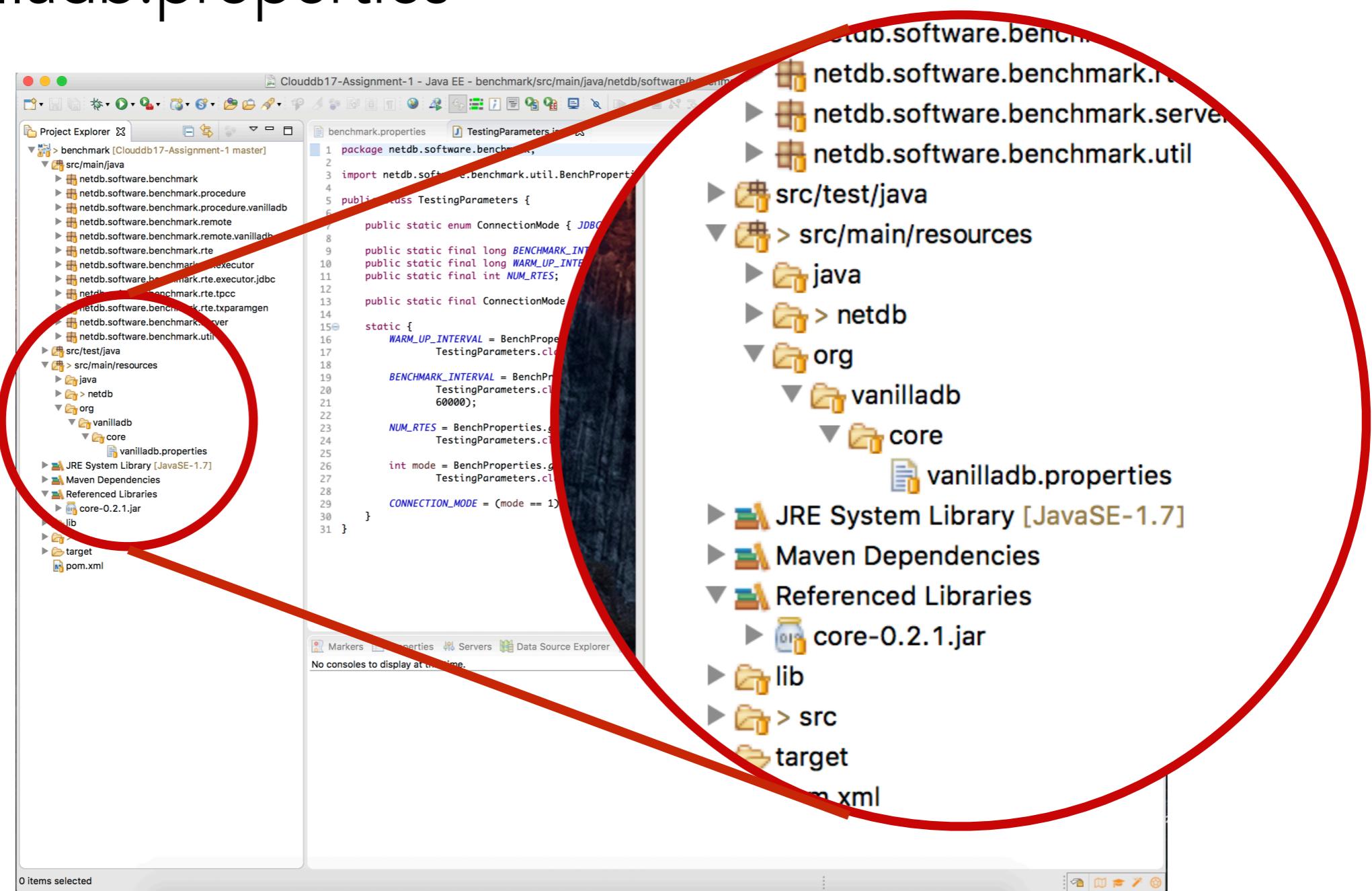


- Configurations for VanillaDB are all stored in a properties file

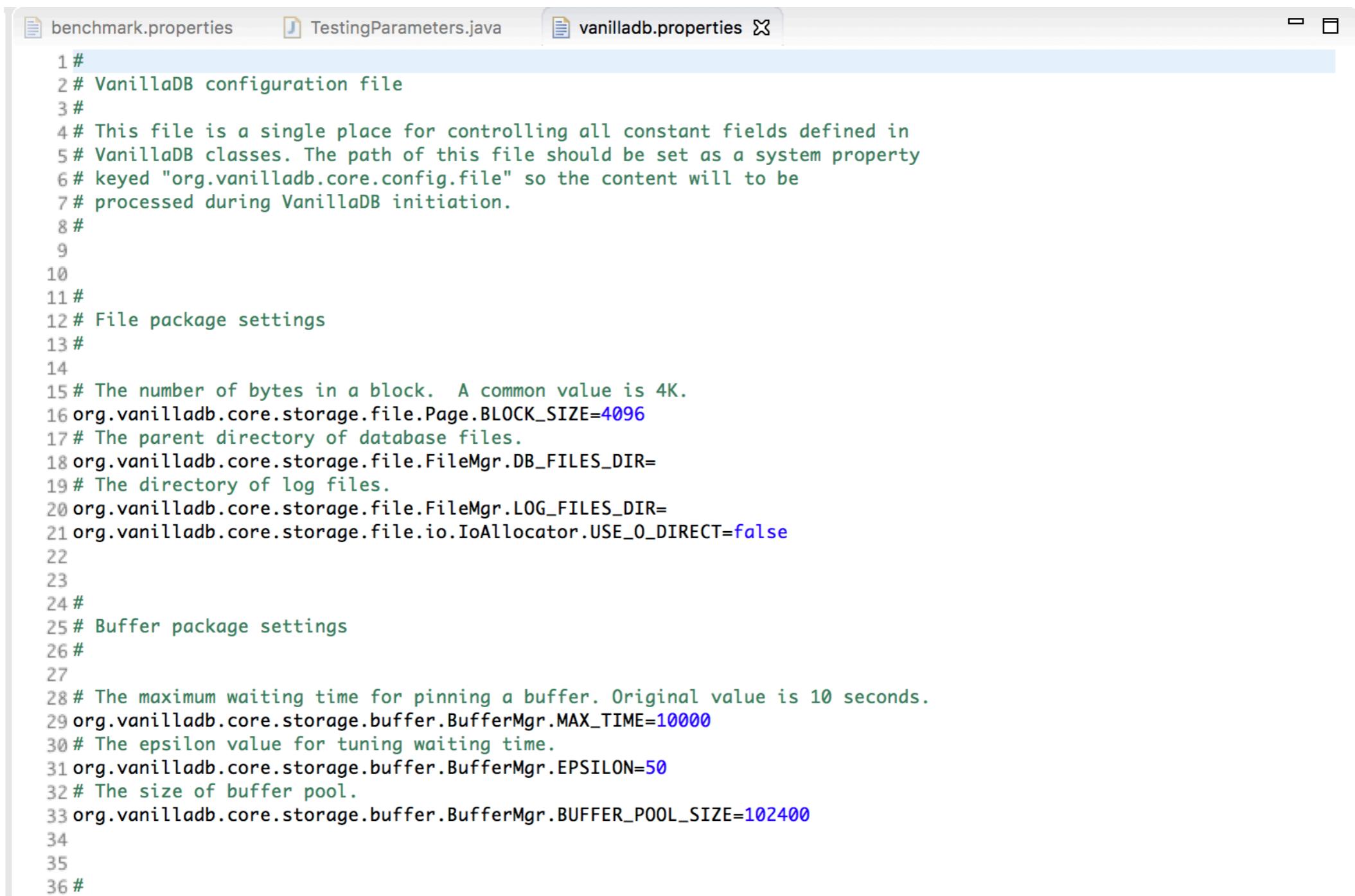
[Icon Source]

VanillaDB Properties File

- vanilladb.properties



vanilladb.properties



The screenshot shows a code editor window with three tabs at the top: 'benchmark.properties', 'TestingParameters.java', and 'vanilladb.properties'. The 'vanilladb.properties' tab is active and contains the following configuration file content:

```
1 #
2 # VanillaDB configuration file
3 #
4 # This file is a single place for controlling all constant fields defined in
5 # VanillaDB classes. The path of this file should be set as a system property
6 # keyed "org.vanilladb.core.config.file" so the content will be
7 # processed during VanillaDB initiation.
8 #
9
10
11 #
12 # File package settings
13 #
14
15 # The number of bytes in a block. A common value is 4K.
16 org.vanilladb.core.storage.file.Page.BLOCK_SIZE=4096
17 # The parent directory of database files.
18 org.vanilladb.core.storage.file.FileMgr.DB_FILES_DIR=
19 # The directory of log files.
20 org.vanilladb.core.storage.file.FileMgr.LOG_FILES_DIR=
21 org.vanilladb.core.storage.file.io.IoAllocator.USE_O_DIRECT=false
22
23
24 #
25 # Buffer package settings
26 #
27
28 # The maximum waiting time for pinning a buffer. Original value is 10 seconds.
29 org.vanilladb.core.storage.buffer.BufferMgr.MAX_TIME=10000
30 # The epsilon value for tuning waiting time.
31 org.vanilladb.core.storage.buffer.BufferMgr.EPSILON=50
32 # The size of buffer pool.
33 org.vanilladb.core.storage.buffer.BufferMgr.BUFFER_POOL_SIZE=102400
34
35
36 #
```

vanilladb.properties

```
10
11 #
12 # File package settings
13 #
14
15 # The number of bytes in a block. A common value is 4K.
16 org.vanilladb.core.storage.file.Page.BLOCK_SIZE=4096
17 # The parent directory of database files
18 org.vanilladb.core.storage.file.FileMgr.DB_FILES_DIR=
19 # The directory of log files.
20 org.vanilladb.core.storage.file.FileMgr.LOG_FILES_DIR=
21 org.vanilladb.core.storage.file.io.IoAllocator.USE_O_DIRECT=false
22
23
```

- Your DataBase files will be stored in this directory
- If it is empty, the Default directory would be your User directory

Outline

- VanillaDB Core
 - Prepare Everything You Need
 - Server Properties
 - Starting Up VanillaDB
 - Console SQL Interpreter

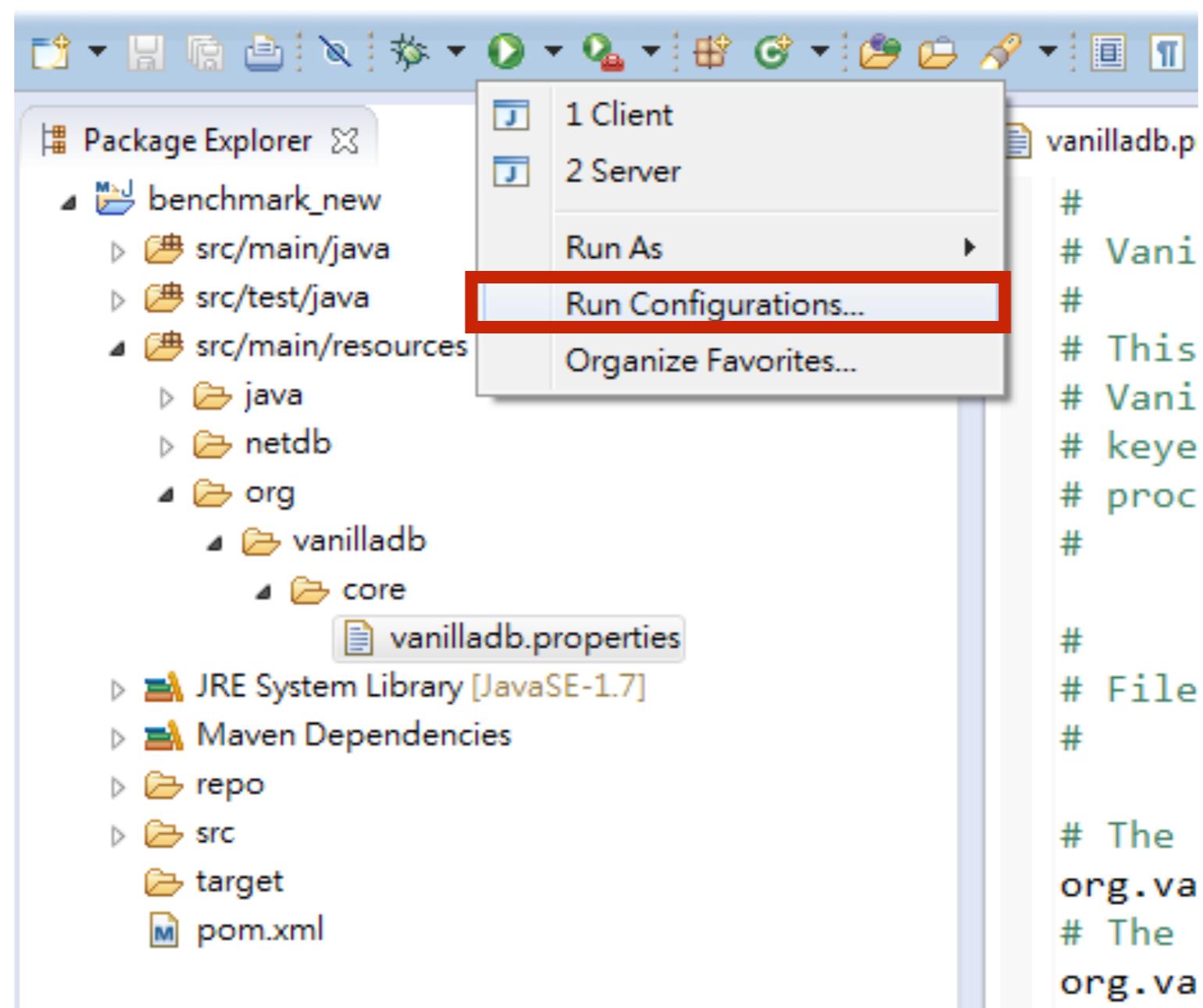
Starting Up VanillaDB

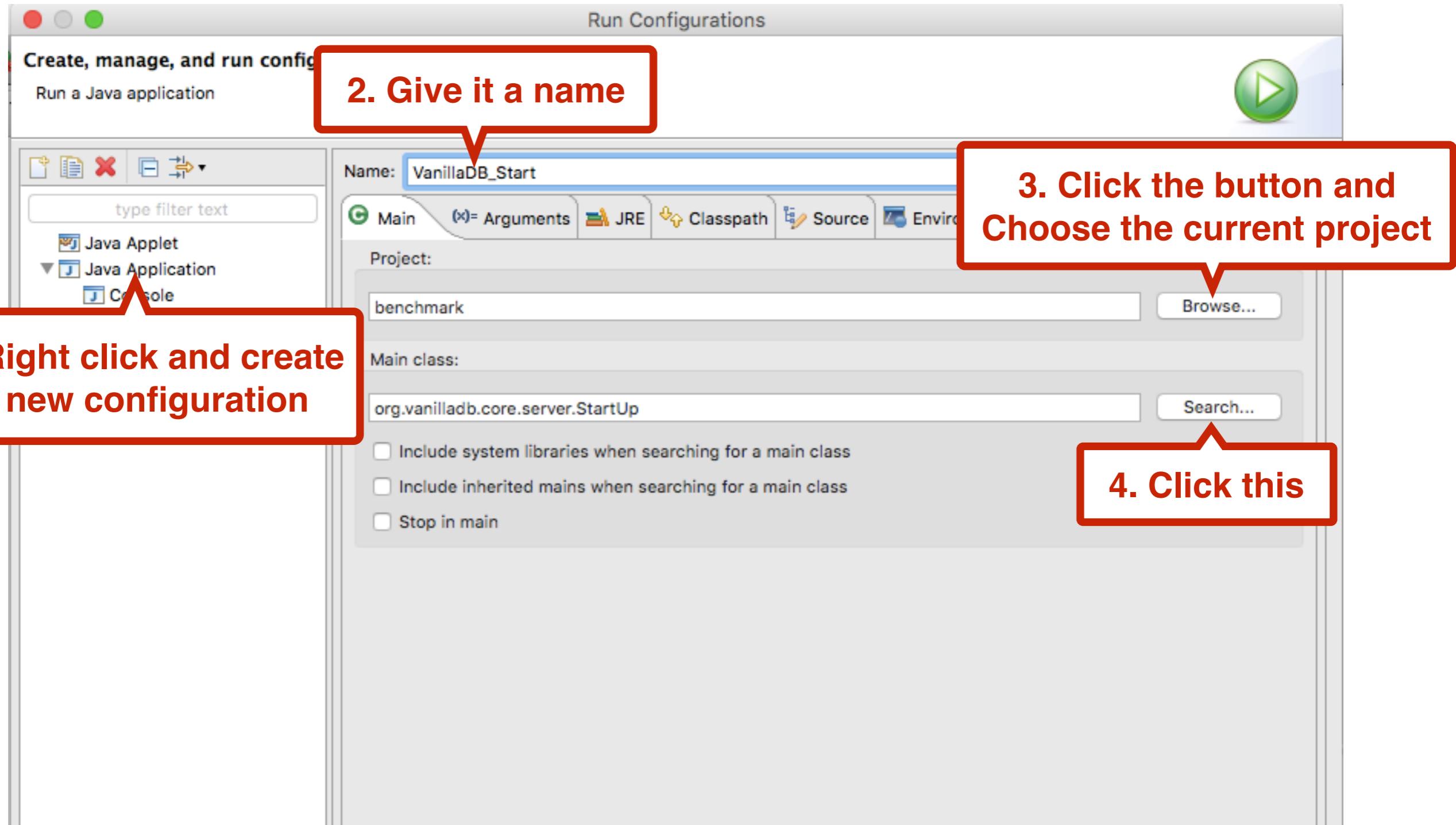


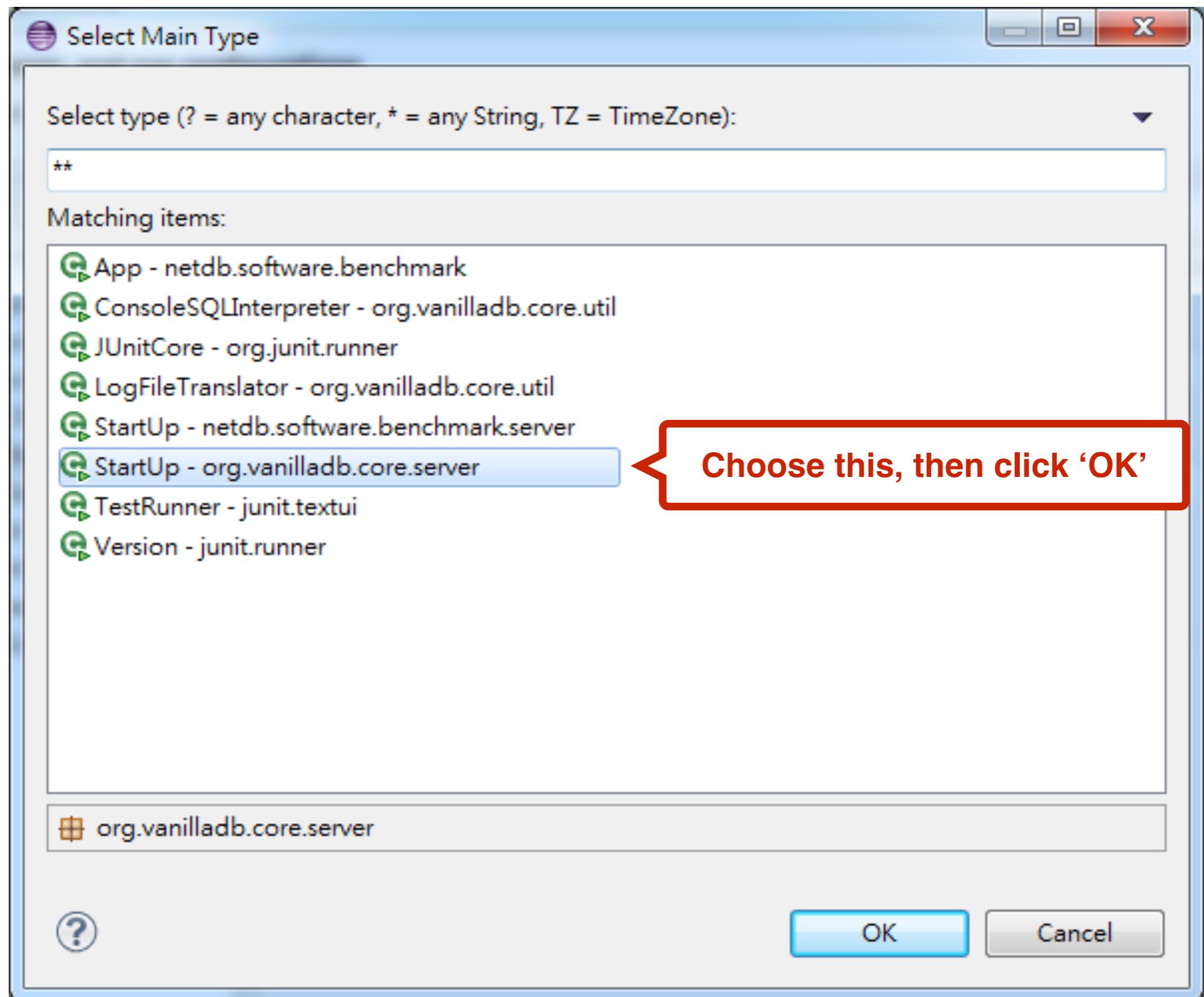
- To start up a VanillaDB server, we have to give it the following arguments
 - Database Directory Name
 - The locations of properties files

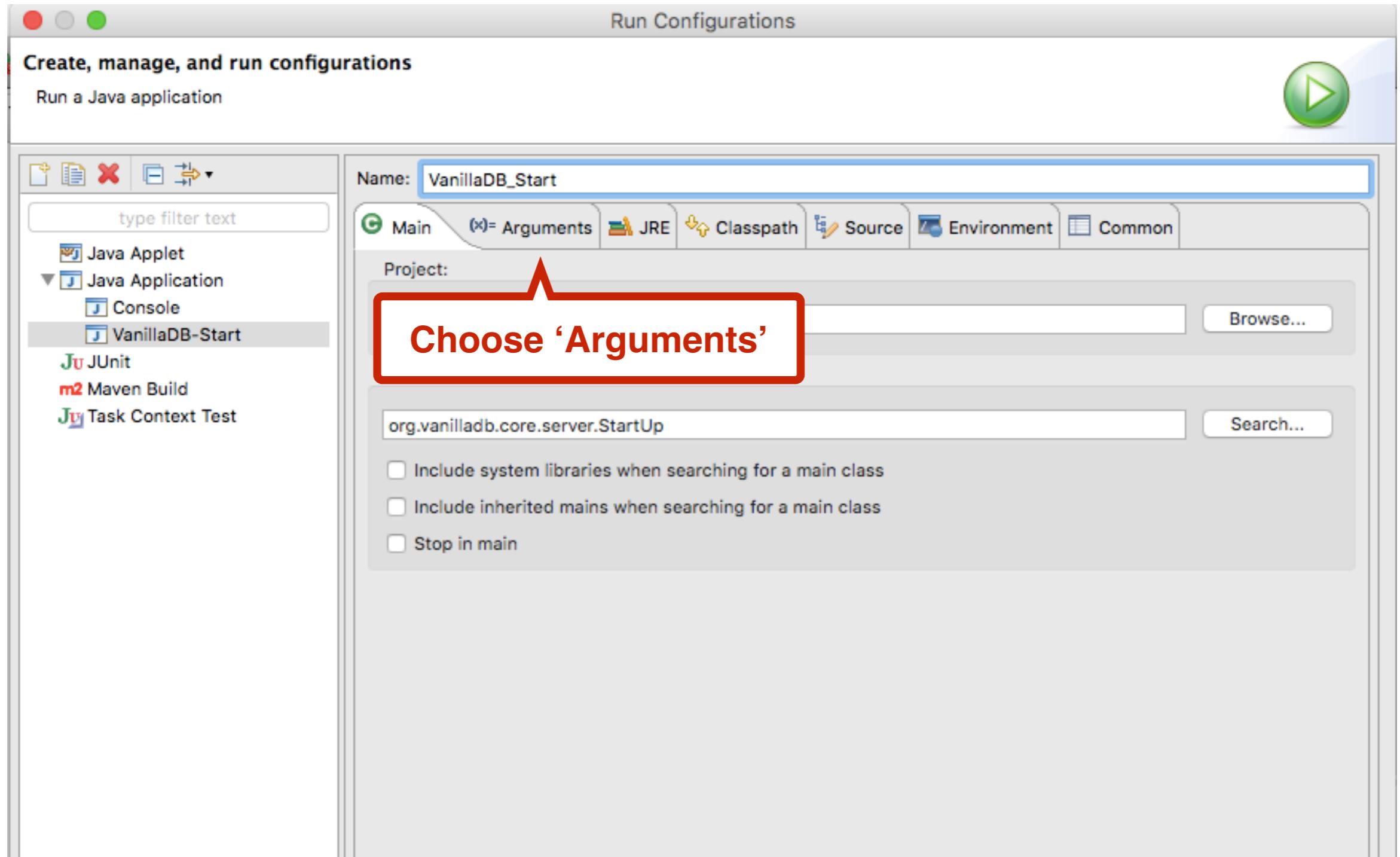
[Icon Source]

Setting Run Configuration









Arguments (1/2)

- Program Arguments

- Format

[Database Directory Name]

- Example

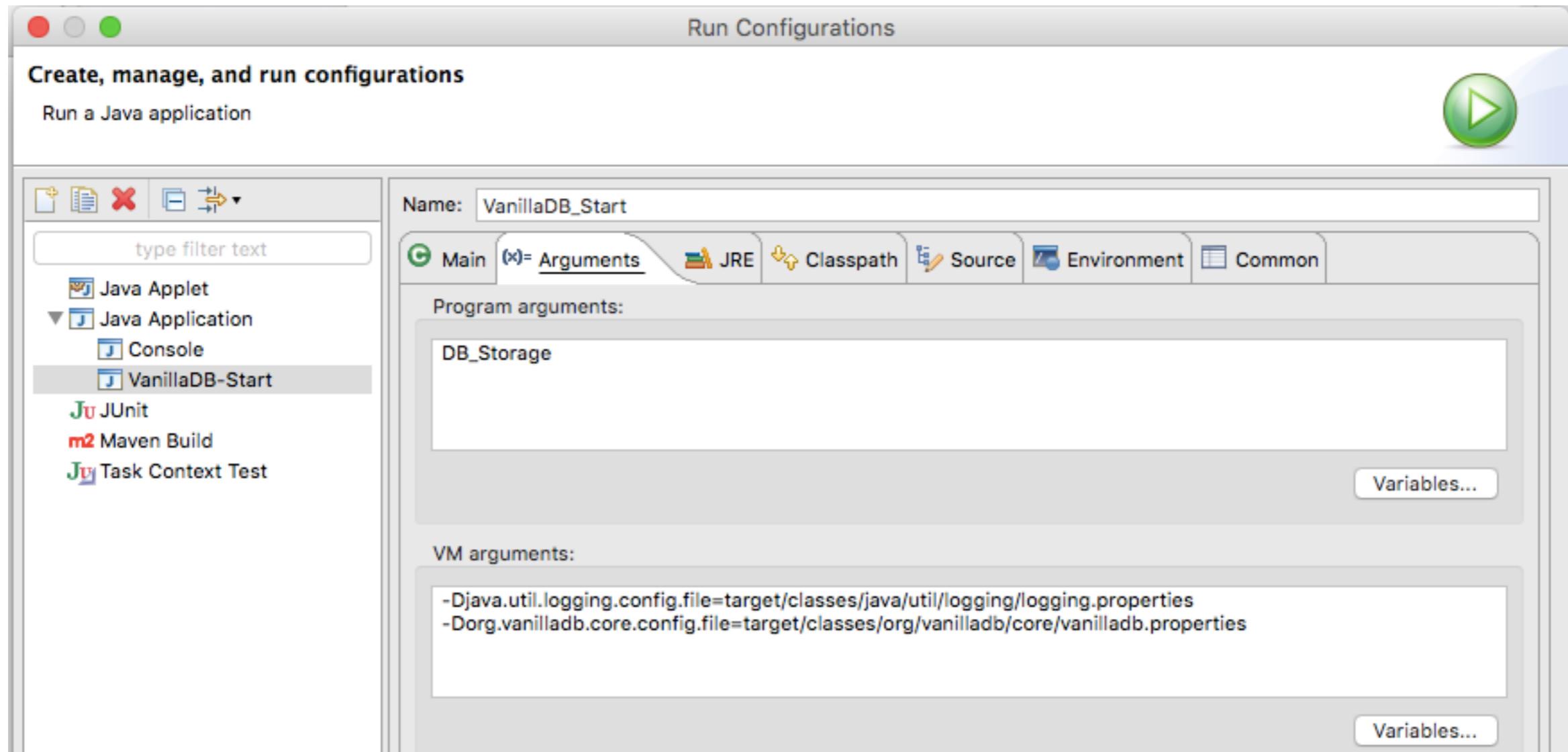
DB_Storage

Arguments (2/2)

- VM Arguments
 - For logging properties

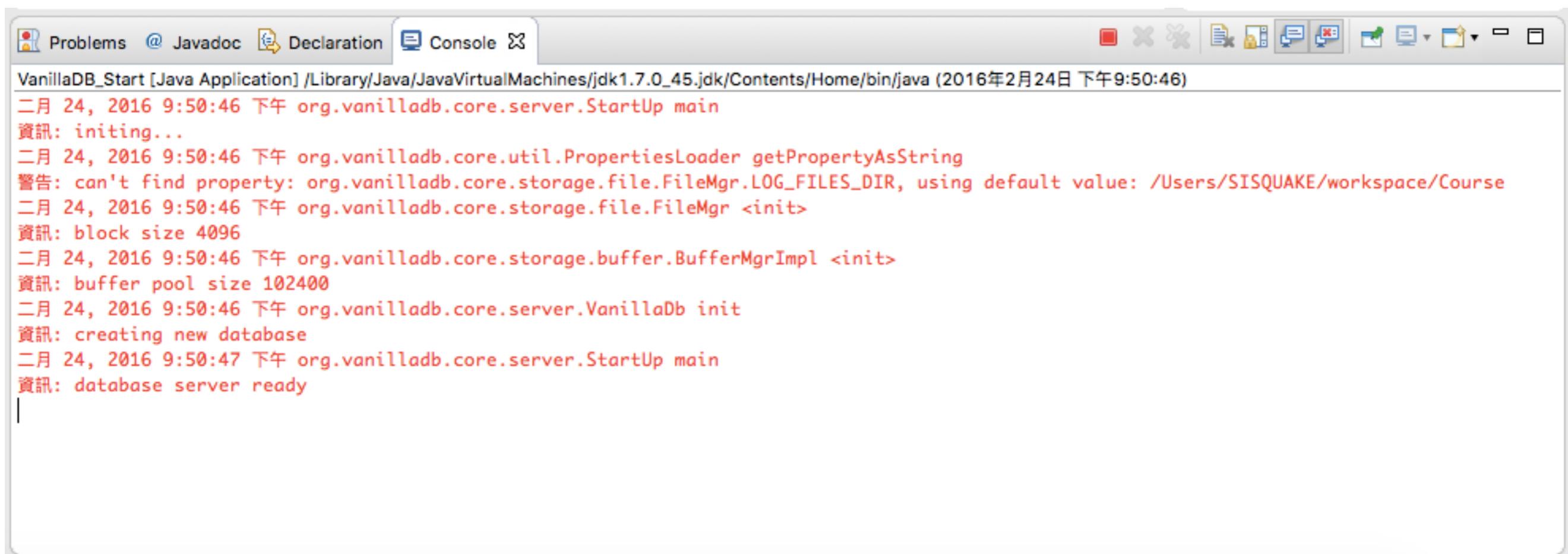
```
-Djava.util.logging.config.file=target/classes/java/util/logging/logging.properties
```
 - For VanillaDB properties

```
-Dorg.vanilladb.core.config.file=target/classes/org/vanilladb/core/vanilladb.properties
```



You can copy those arguments from [here](#),
then click 'Apply' and 'Run'

Server Messages (1/3)

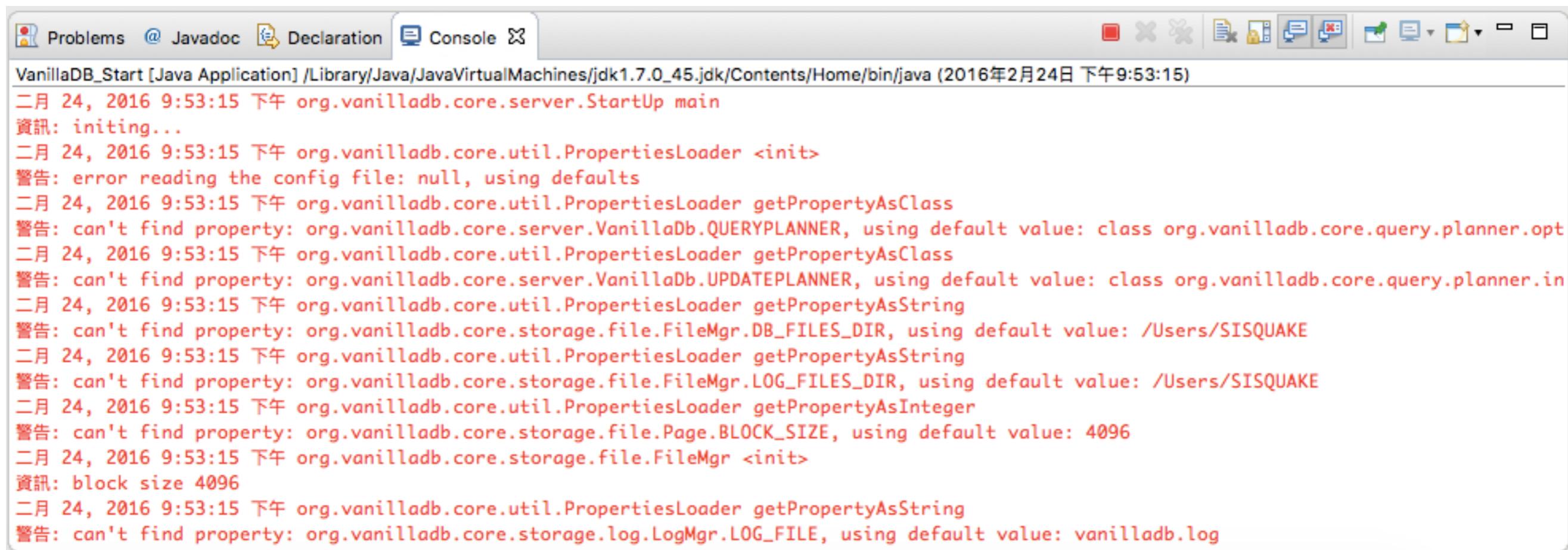


The screenshot shows a Java application named "VanillaDB_Start" running in an IDE. The "Console" tab is selected, displaying the following log output:

```
二月 24, 2016 9:50:46 下午 org.vanilladb.core.server.StartUp main
資訊: initting...
二月 24, 2016 9:50:46 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsString
警告: can't find property: org.vanilladb.core.storage.file.FileMgr.LOG_FILES_DIR, using default value: /Users/SISQUAKE/workspace/Course
二月 24, 2016 9:50:46 下午 org.vanilladb.core.storage.file.FileMgr <init>
資訊: block size 4096
二月 24, 2016 9:50:46 下午 org.vanilladb.core.storage.buffer.BufferMgrImpl <init>
資訊: buffer pool size 102400
二月 24, 2016 9:50:46 下午 org.vanilladb.core.server.VanillaDb init
資訊: creating new database
二月 24, 2016 9:50:47 下午 org.vanilladb.core.server.StartUp main
資訊: database server ready
```

You should see this if there is nothing wrong.

Server Messages (2/3)



The screenshot shows a Java application named "VanillaDB_Start" running in an IDE. The "Console" tab is selected, displaying log messages from the application's startup. The messages are primarily in Chinese, with some English words like "warning" and "info". The log output includes several "warning" messages about properties not being found and using default values, such as "can't find property: org.vanilladb.core.server.VanillaDb.QUERYPLANNER" and "can't find property: org.vanilladb.core.storage.file.FileMgr.DB_FILES_DIR". There are also "info" messages like "資訊: initing..." and "資訊: block size 4096". The IDE interface includes tabs for "Problems", "@ Javadoc", "Declaration", and "Console".

```
VanillaDB_Start [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_45.jdk/Contents/Home/bin/java (2016年2月24日 下午9:53:15)
二月 24, 2016 9:53:15 下午 org.vanilladb.core.server.StartUp main
資訊: initing...
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader <init>
警告: error reading the config file: null, using defaults
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsClass
警告: can't find property: org.vanilladb.core.server.VanillaDb.QUERYPLANNER, using default value: class org.vanilladb.core.query.planner.opt
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsClass
警告: can't find property: org.vanilladb.core.server.VanillaDb.UPDATEPLANNER, using default value: class org.vanilladb.core.query.planner.in
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsString
警告: can't find property: org.vanilladb.core.storage.file.FileMgr.DB_FILES_DIR, using default value: /Users/SISQUAKE
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsString
警告: can't find property: org.vanilladb.core.storage.file.FileMgr.LOG_FILES_DIR, using default value: /Users/SISQUAKE
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsInteger
警告: can't find property: org.vanilladb.core.storage.file.Page.BLOCK_SIZE, using default value: 4096
二月 24, 2016 9:53:15 下午 org.vanilladb.core.storage.file.FileMgr <init>
資訊: block size 4096
二月 24, 2016 9:53:15 下午 org.vanilladb.core.util.PropertiesLoader getPropertyAsString
警告: can't find property: org.vanilladb.core.storage.log.LogMgr.LOG_FILE, using default value: vanilladb.log
```

If you saw any ‘Warning’ message,
you should check it carefully.

Server Messages (3/3)

- “error reading config file, using default”
 - It usually happens when you give a wrong location for a properties file
- “can't find property:, using default: ...”
 - It means that there is a property missing in your properties file

Console SQL Interpreter

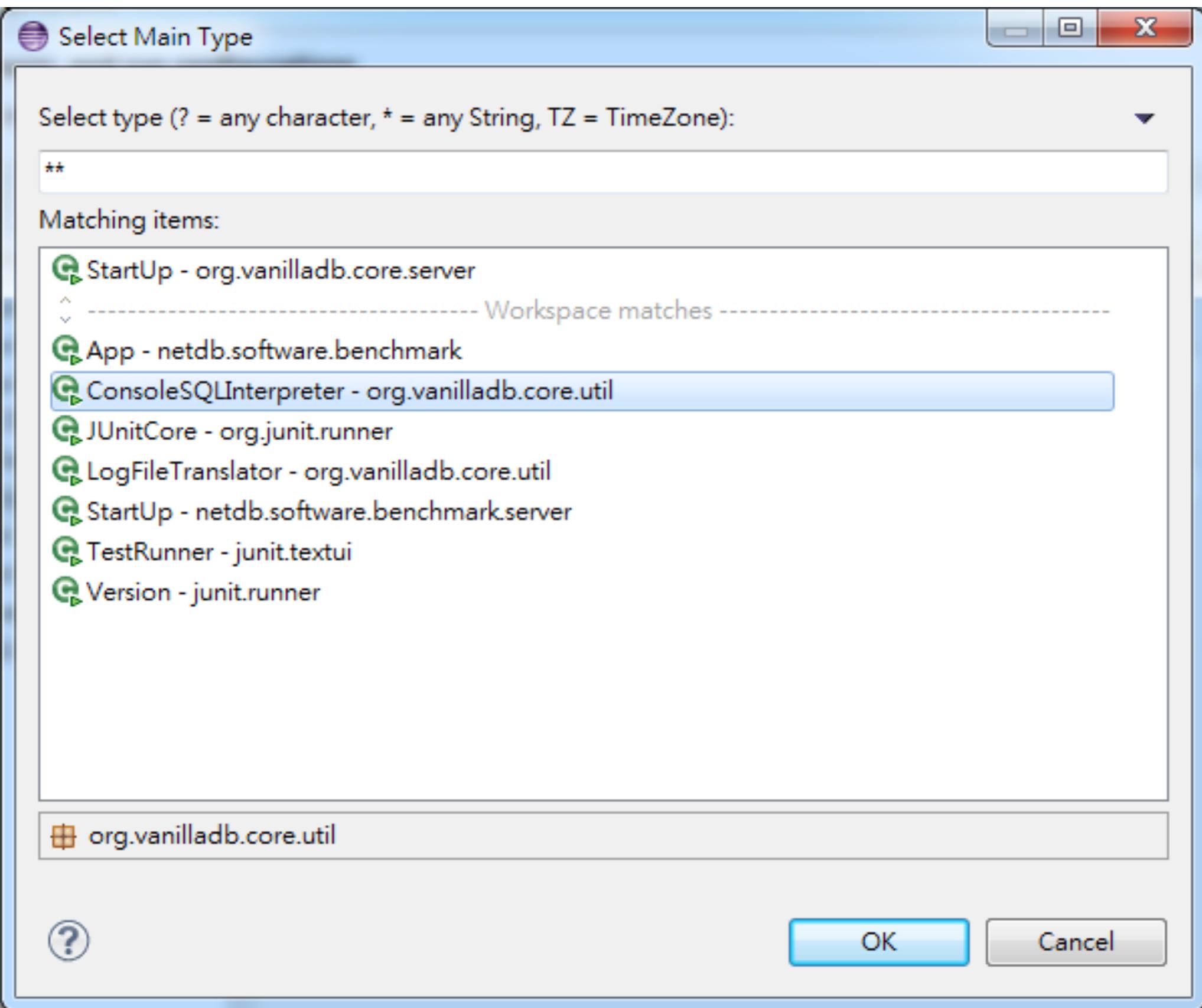


- You can use Console SQL Interpreter we provided in VanillaDB-Core to connect with server

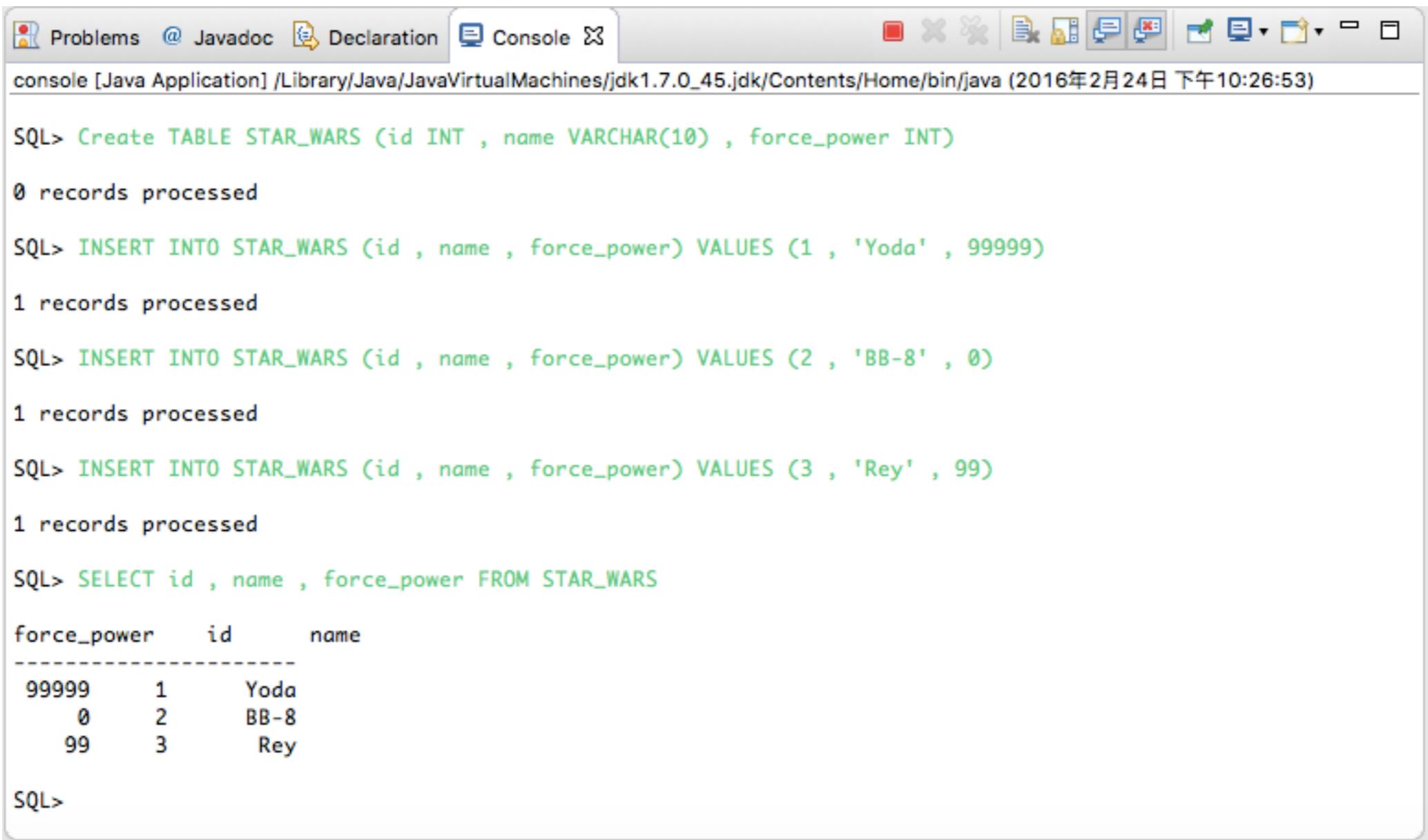
[Icon Source]

Console SQL Interpreter

- To use Console SQL Interpreter, just follow these steps
 1. Create a new run configuration
 2. Give it a name and choose your project
 3. Choose “ConsoleSQLInterpreter” for “Main Class”
 4. No VM Argument is required
 5. Run it



Try it !



The screenshot shows a Java application console window with the following SQL session:

```
SQL> Create TABLE STAR_WARS (id INT , name VARCHAR(10) , force_power INT)
0 records processed

SQL> INSERT INTO STAR_WARS (id , name , force_power) VALUES (1 , 'Yoda' , 99999)
1 records processed

SQL> INSERT INTO STAR_WARS (id , name , force_power) VALUES (2 , 'BB-8' , 0)
1 records processed

SQL> INSERT INTO STAR_WARS (id , name , force_power) VALUES (3 , 'Rey' , 99)
1 records processed

SQL> SELECT id , name , force_power FROM STAR_WARS
force_power      id      name
-----
 99999         1      Yoda
     0         2      BB-8
    99         3      Rey

SQL>
```

The console tab is selected in the top bar. The output shows the creation of a table named STAR_WARS with three columns: id (INT), name (VARCHAR(10)), and force_power (INT). Three rows are inserted into the table: (id=1, name='Yoda', force_power=99999), (id=2, name='BB-8', force_power=0), and (id=3, name='Rey', force_power=99). Finally, a SELECT statement retrieves all data from the table.

Q&A

- To see what exactly queries you can use, please check here
 - https://shwu10.cs.nthu.edu.tw/courses-databases-2018-spring/FAQ/blob/master/Vanilladb_Sql.md
- If you got any problem, you can check here first
 - <https://shwu10.cs.nthu.edu.tw/courses-databases-2018-spring/FAQ>
- If your problem was very unique, just send a email let us know