

# Using VanillaDB

ctasi@DataLab  
Cloud Database  
Spring, 2017

# TODAY'S AGENDA

- VanillaDB Core
  - Server Properties
  - Starting Up VanillaDB
  - Console SQL Interpreter
- Benchmark Project
  - Starting Up Server for Benchmarking
  - Setting Benchmark Configurations
  - Running Benchmark Client

# TODAY'S AGENDA

- VanillaDB Core
  - Server Properties
  - Starting Up VanillaDB
  - Console SQL Interpreter
- Benchmark Project
  - Starting Up Server for Benchmarking
  - Setting Benchmark Configurations
  - Running Benchmark Client

# VanillaDB Properties File

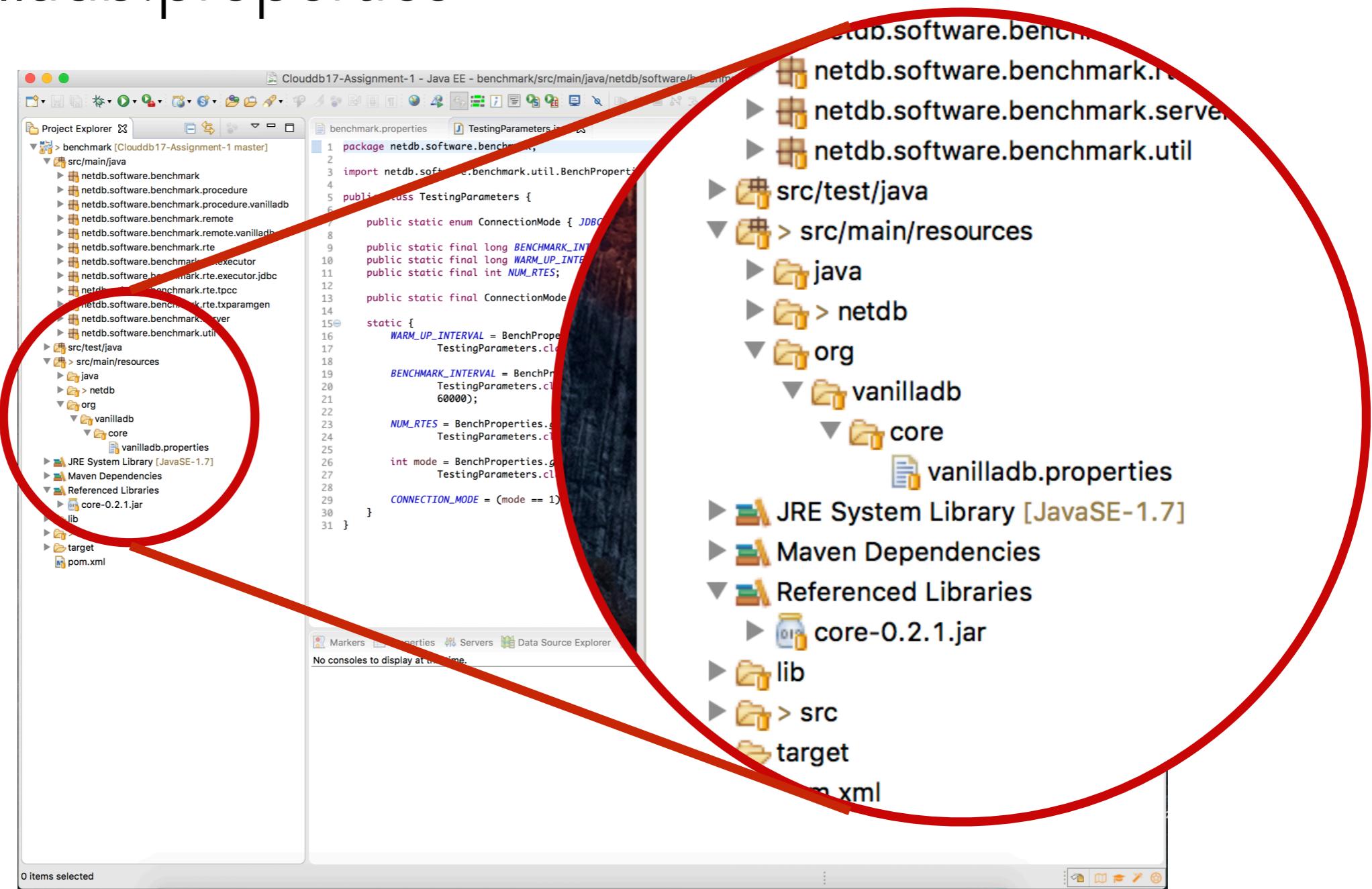


- Configurations for VanillaDB are all stored in a properties file

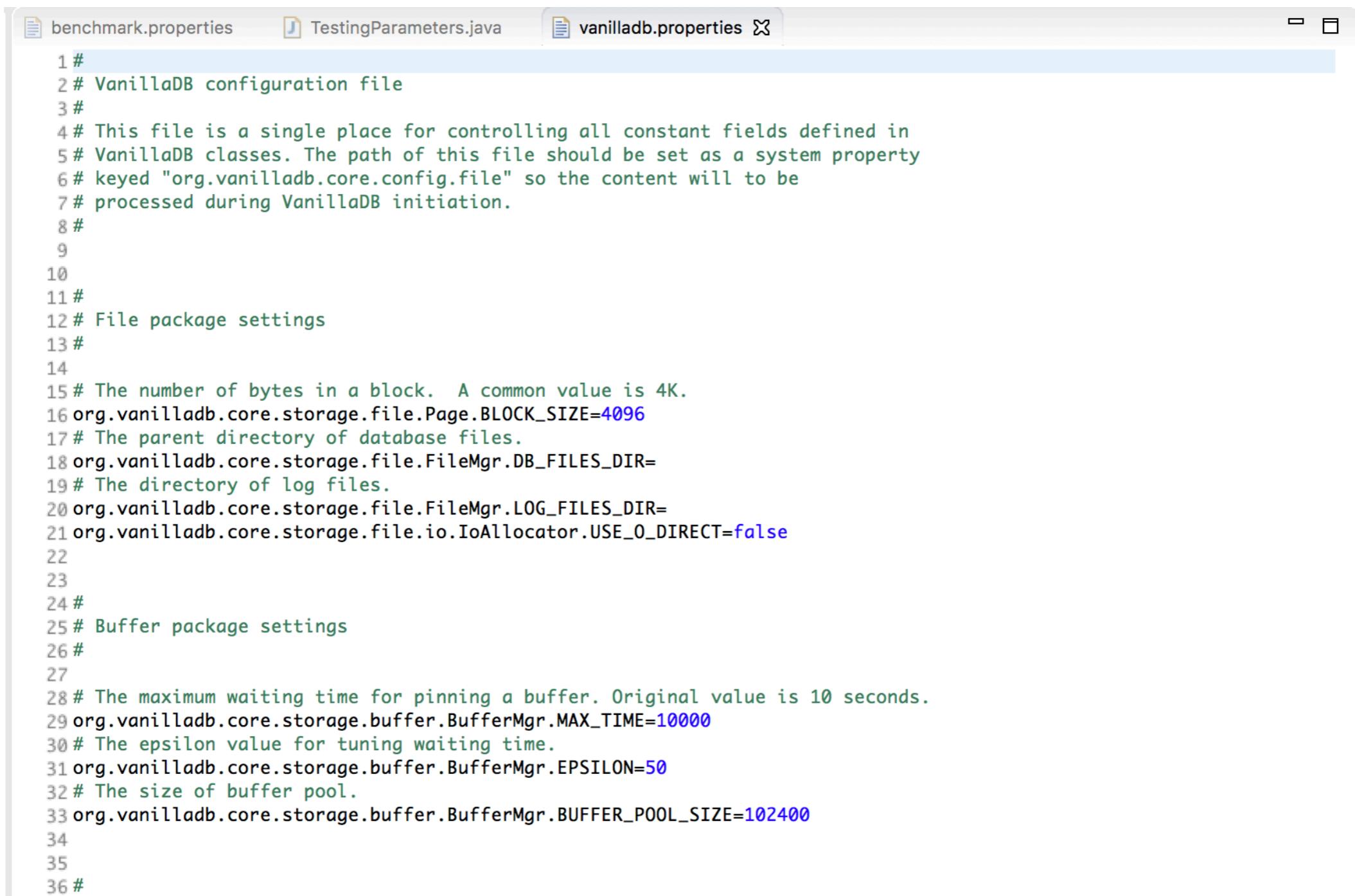
[Icon Source]

# 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

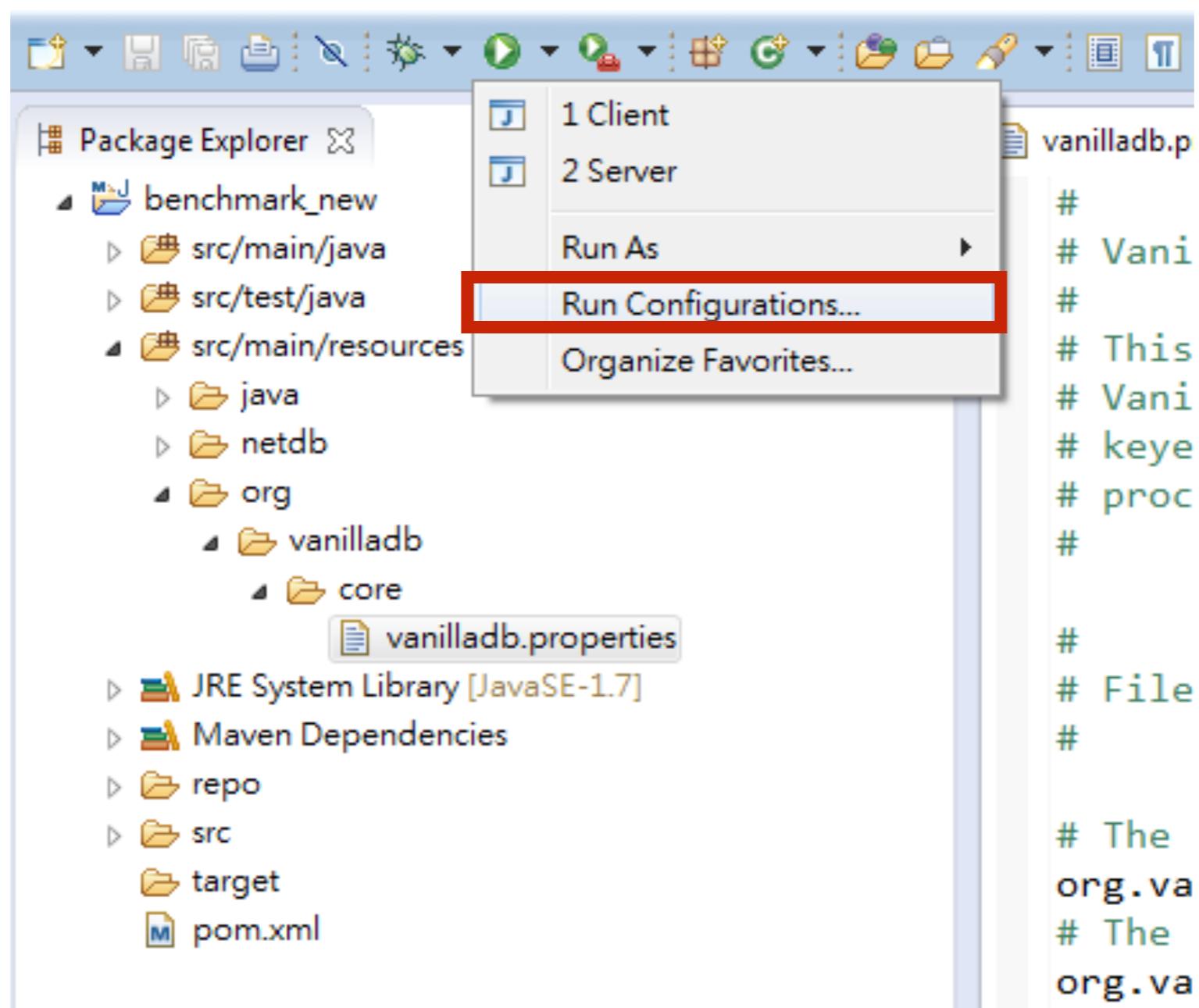
# Starting Up VanillaDB

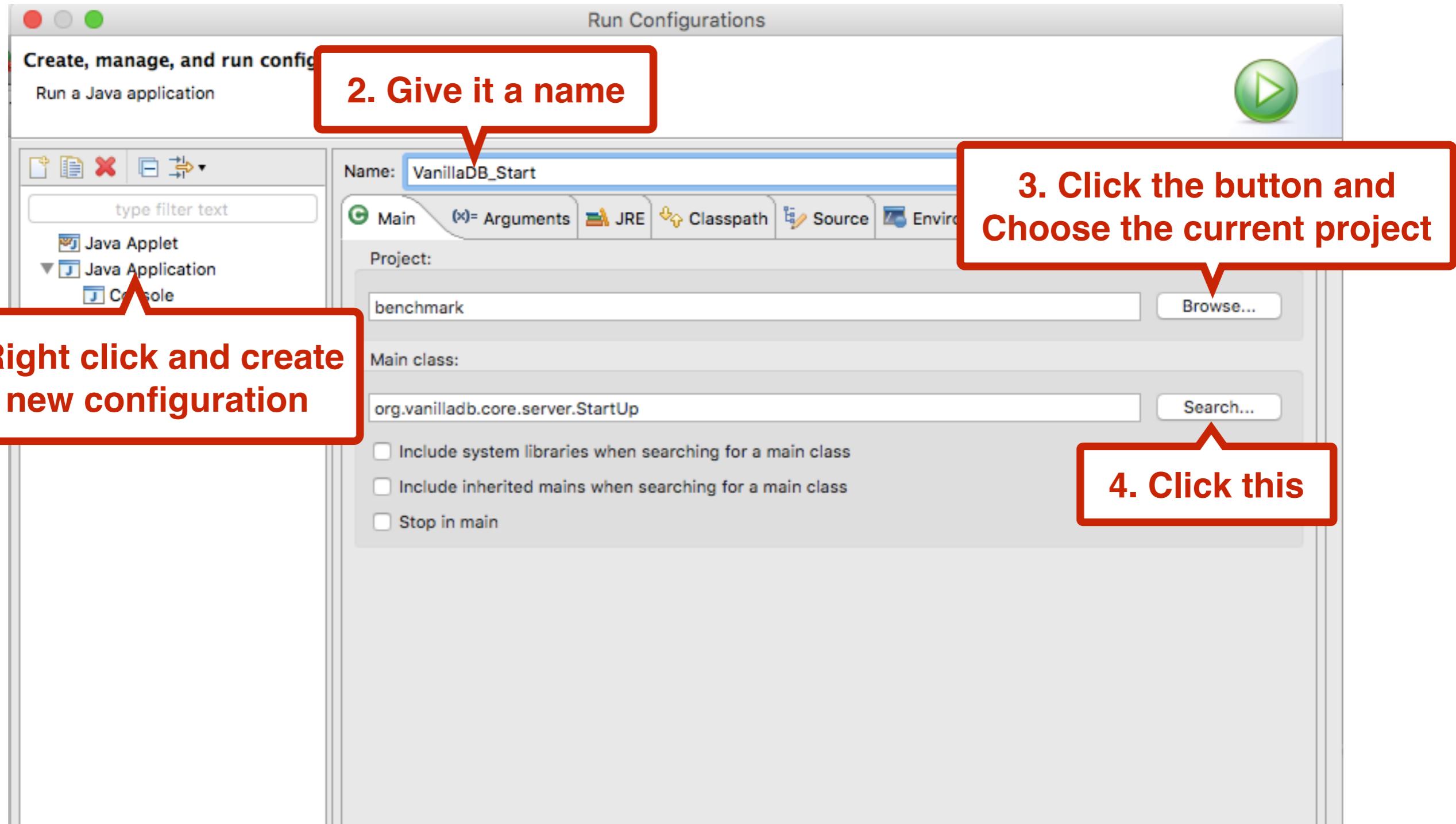


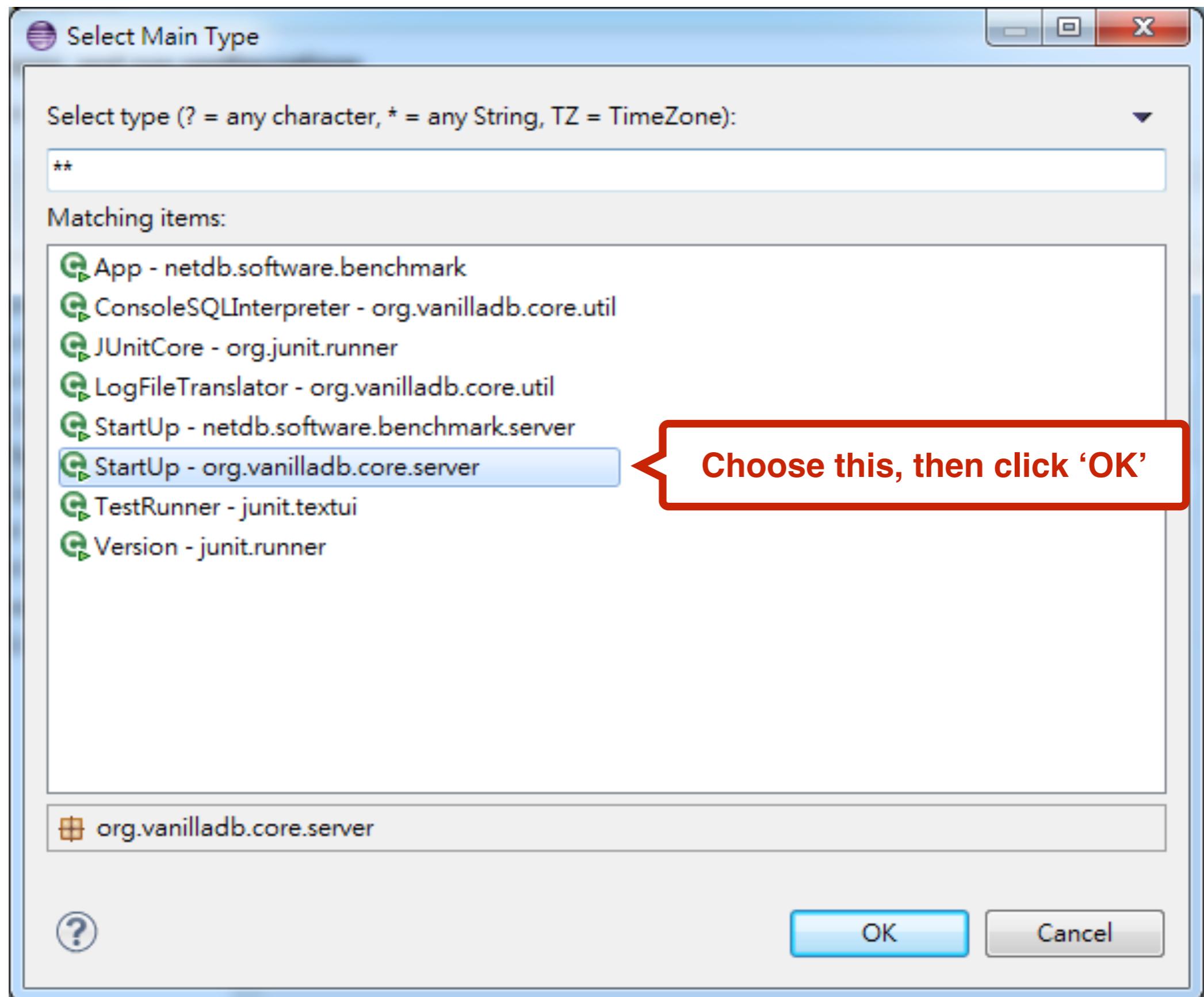
- To start up a VanillaDB server, we have to give it the following arguments
  - Database Directory Name
  - The locations of properties files

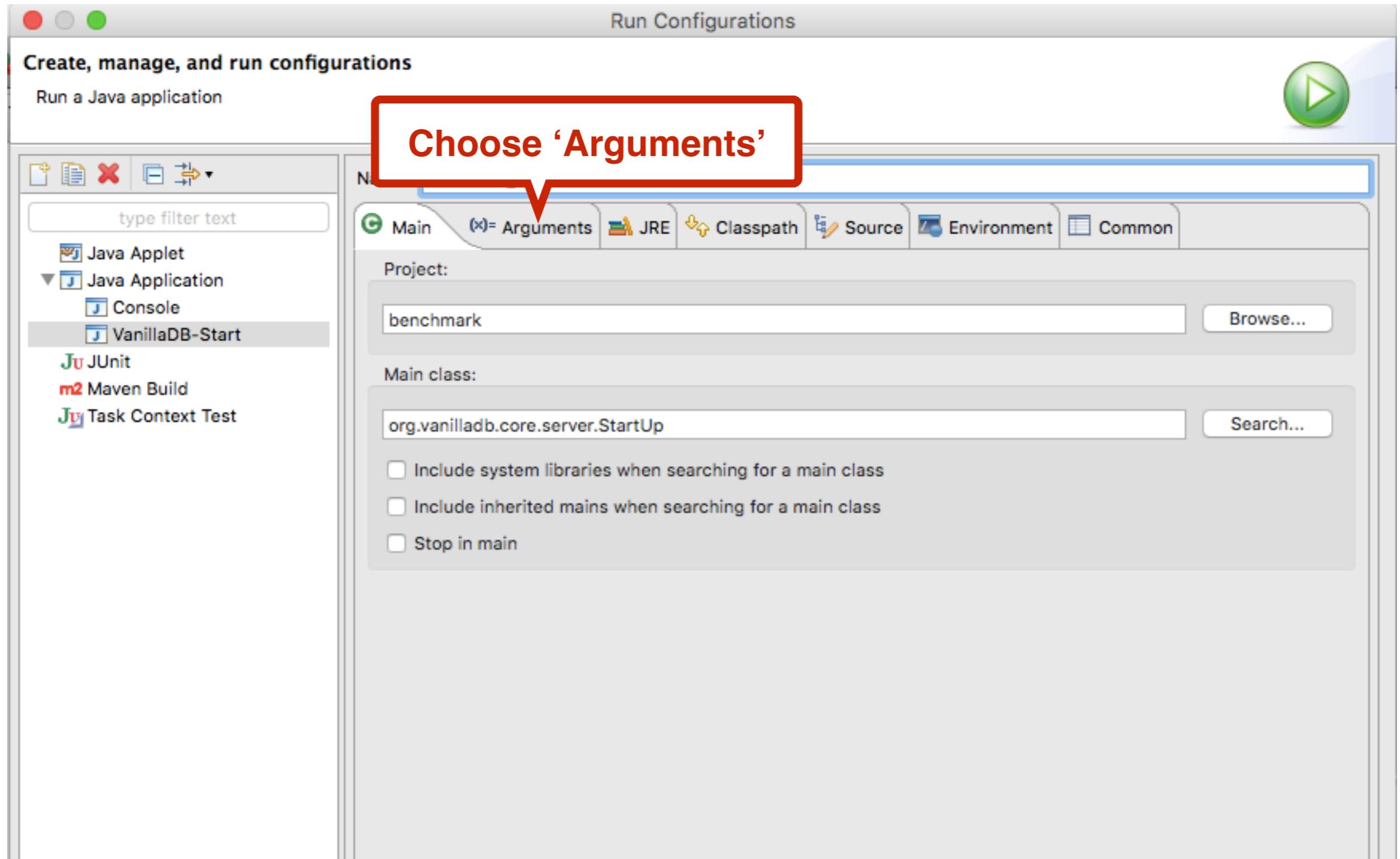
[Icon Source]

# Setting Run Configuration









# Arguments (1/2)

- Program Arguments

- Format

[Database Directory Name]

- Example

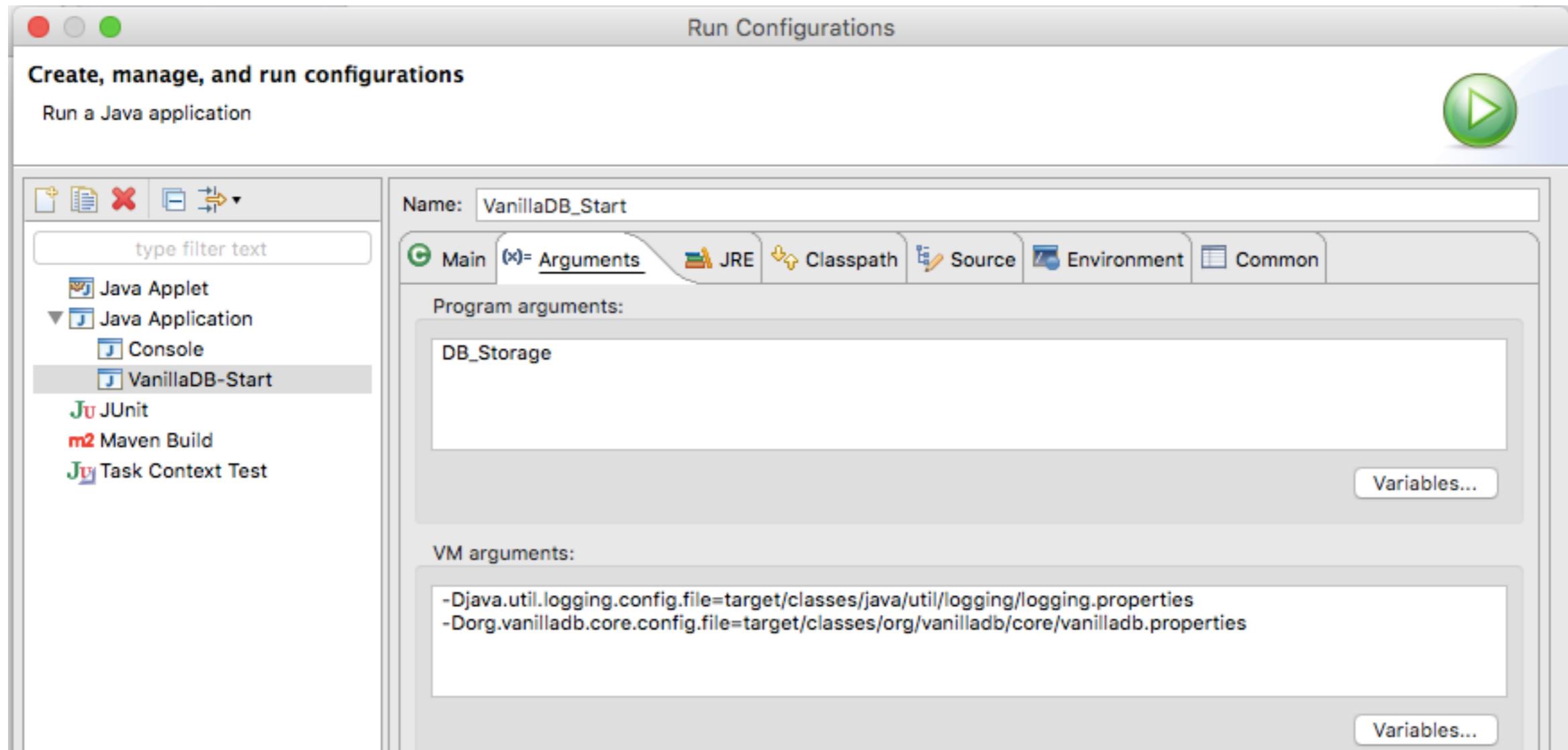
DB\_Storage

# Arguments (2/2)

- VM Arguments
  - For logging properties

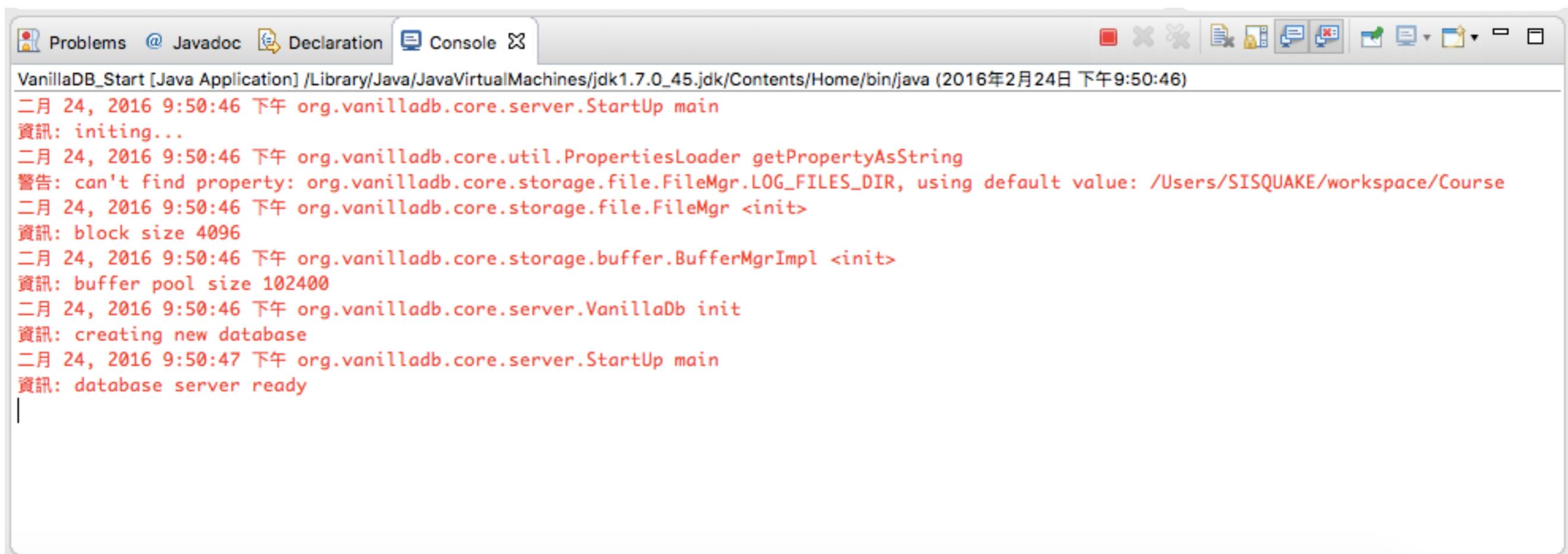
```
-Djava.util.logging.config.file=target/classes/java/util/logging/logging.properties
```
  - For VanillaDB properties

```
-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)

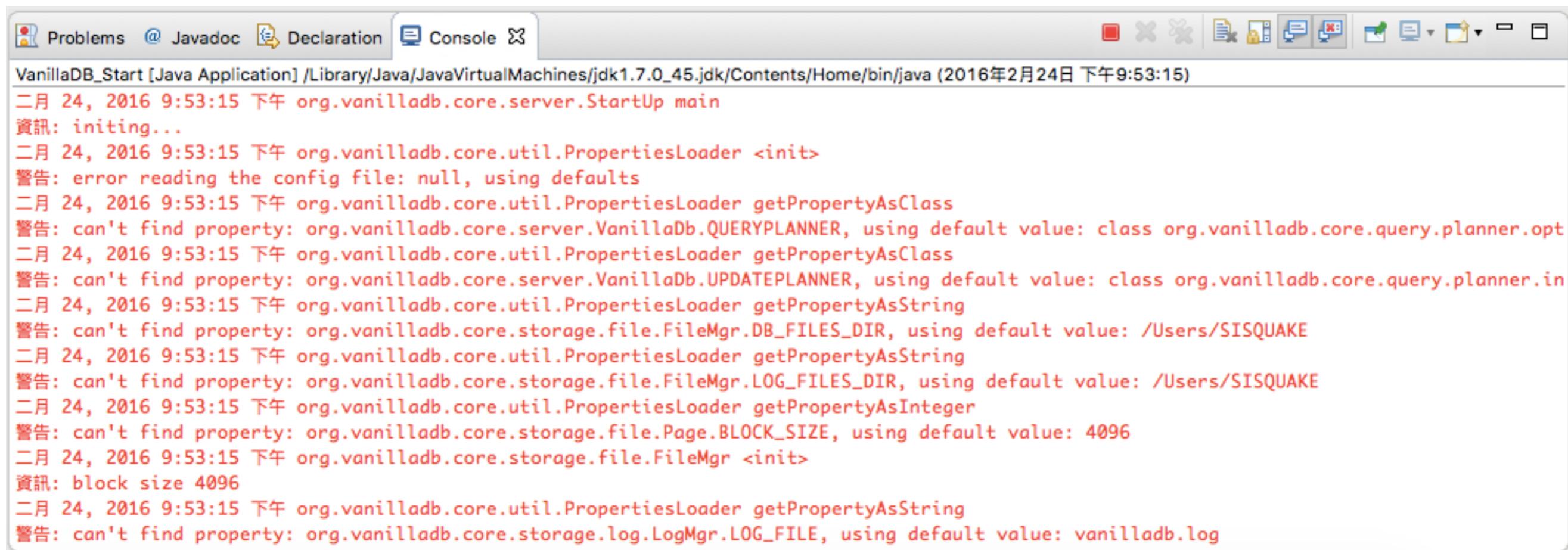


The screenshot shows a Java application named "VanillaDB\_Start" running in an IDE. The "Console" tab is selected, displaying the following log output:

```
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 in Chinese and indicate the application is initializing and loading properties from a configuration file. Some properties are found and used, while others are not found and default values are used instead. The log entries include:

```
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

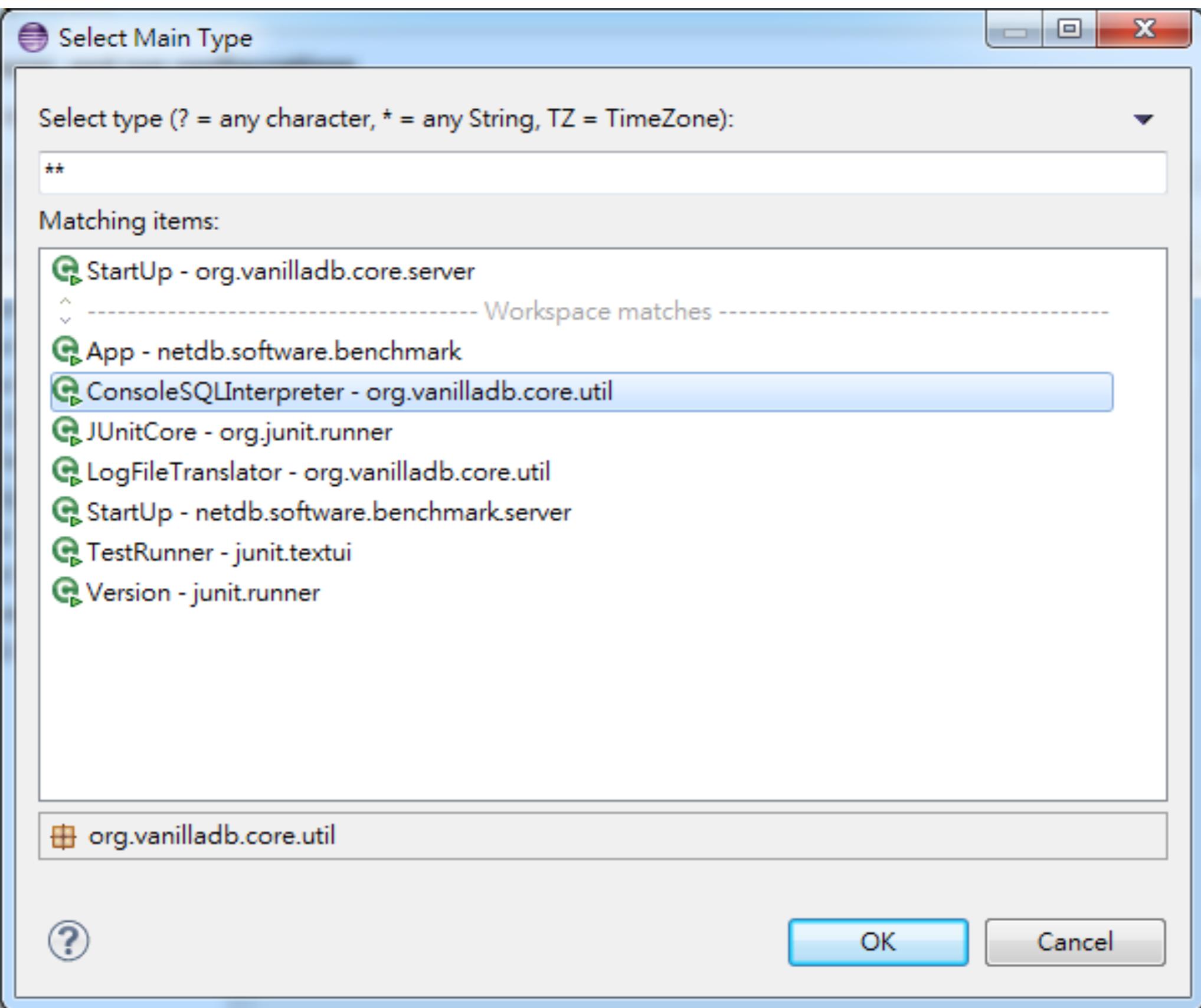


- You can use Console SQL Interpreter we provided in VanillaDB-Core to connect with server

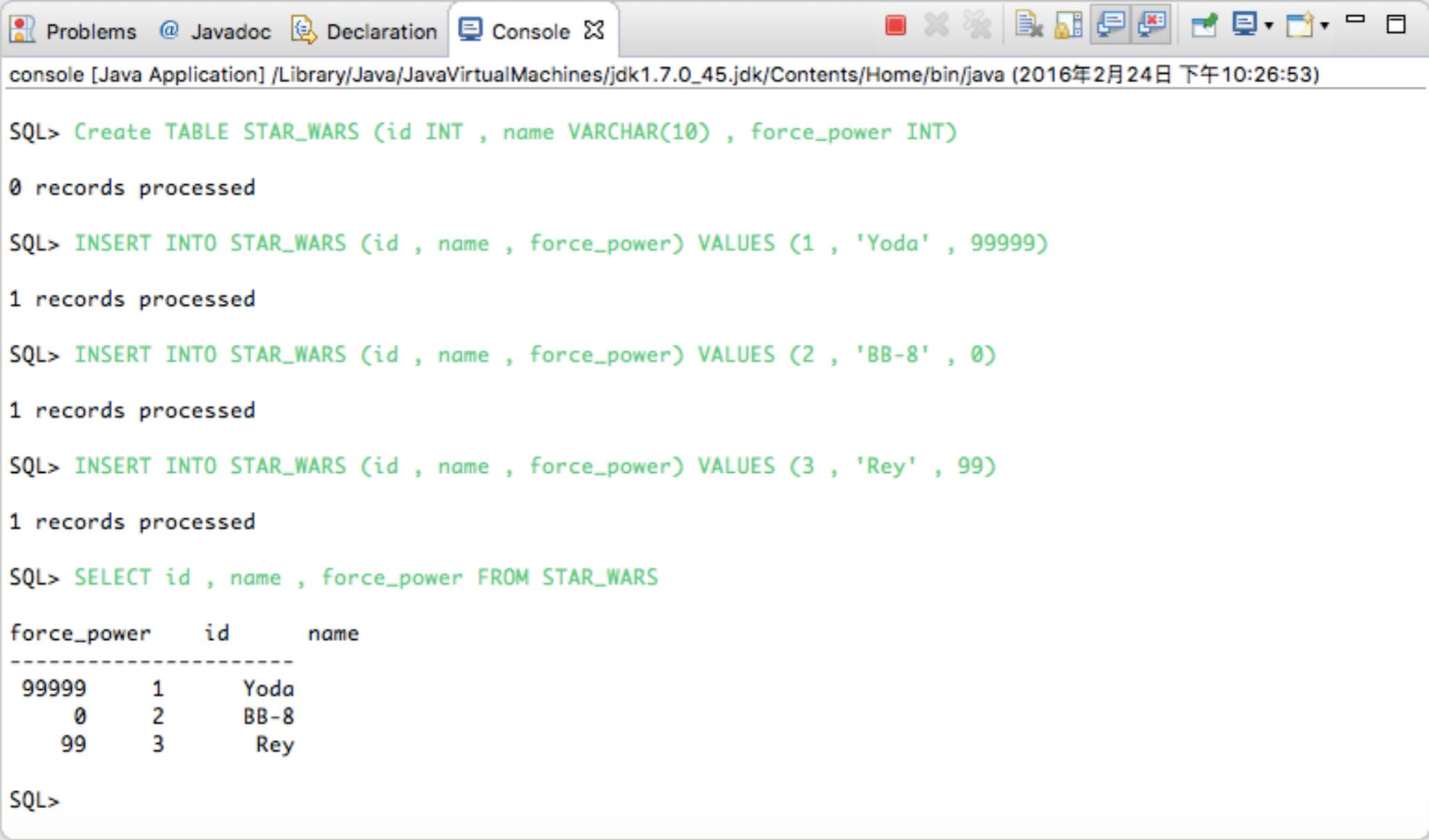
[Icon Source]

# 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 columns id (INT), name (VARCHAR(10)), and force\_power (INT). Three rows are inserted into the table: (1, 'Yoda', 99999), (2, 'BB-8', 0), and (3, 'Rey', 99). A final SELECT statement retrieves all data from the table.

# TODAY'S AGENDA

- VanillaDB Core
  - Server Properties
  - Starting Up VanillaDB
  - Console SQL Interpreter
- Benchmark Project
  - Starting Up Server for Benchmarking
  - Setting Benchmark Configurations
  - Running Benchmark Client

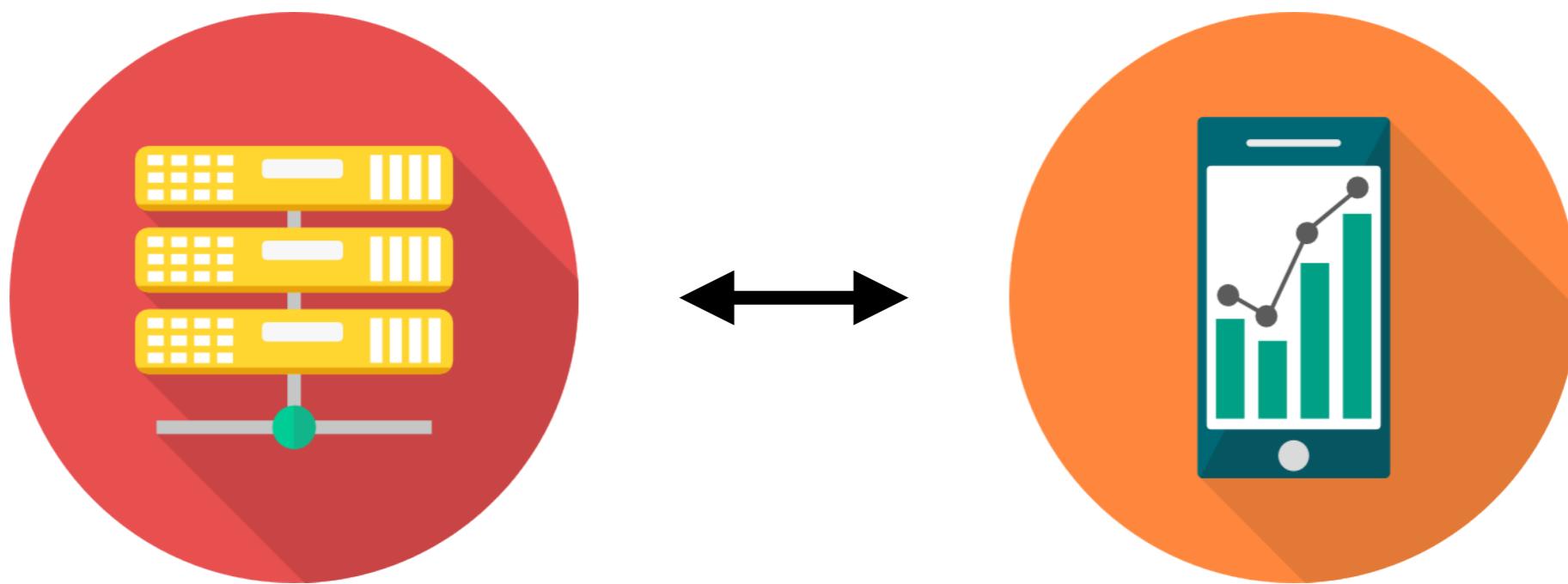
# Benchmark Project



- Benchmark project is a special client project for automatically benchmarking
- It contains a testbed loader and a testing process
- It also has a lot of adjustable testing parameters

[Icon Source]

# Server & Client

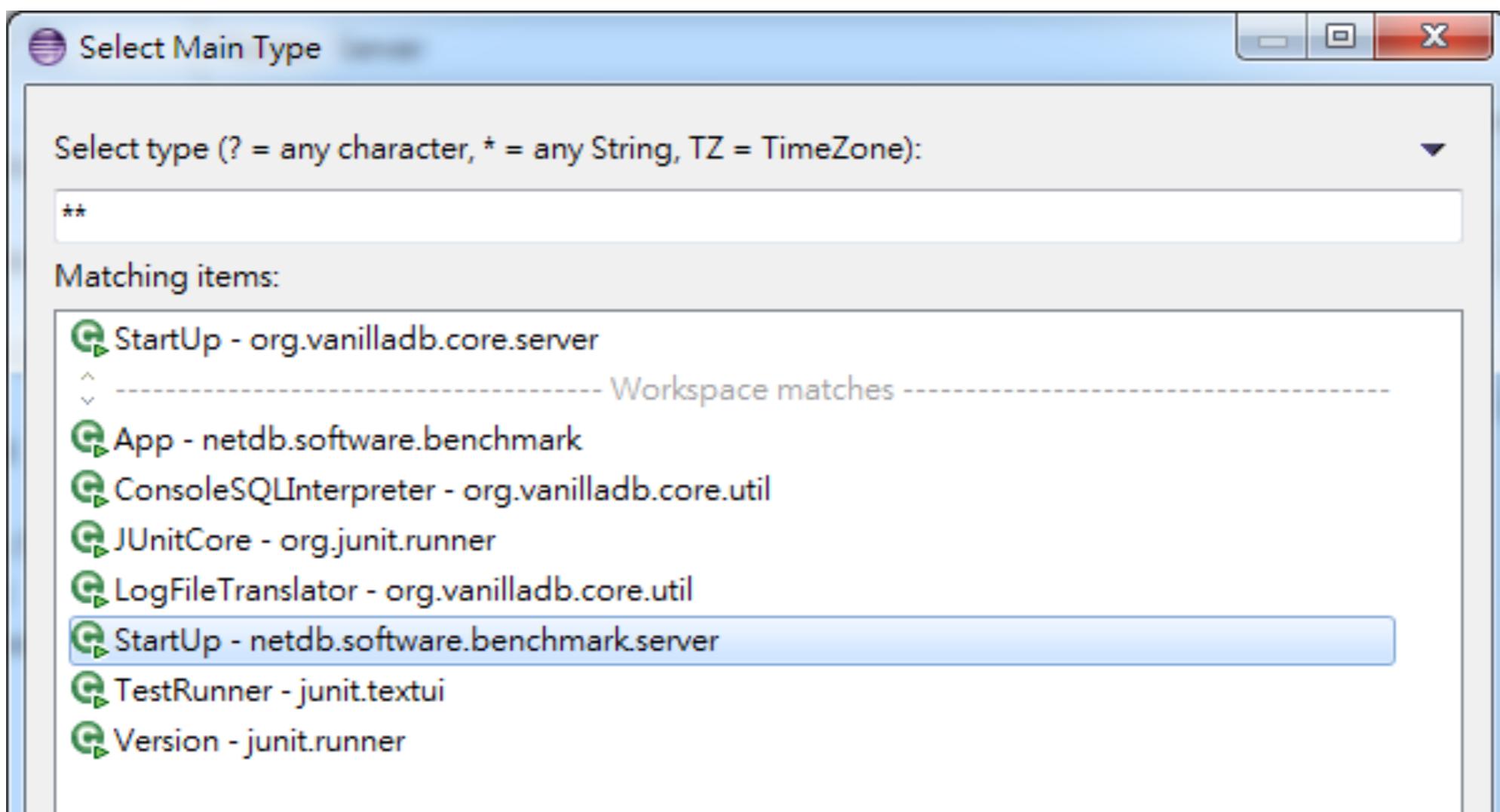


Benchmark  
Server

Benchmark  
Client

# Starting Up Server (1/2)

- To benchmark a VanillaDB server, you need to start up the server in another entry point



# Starting Up Server (2/2)

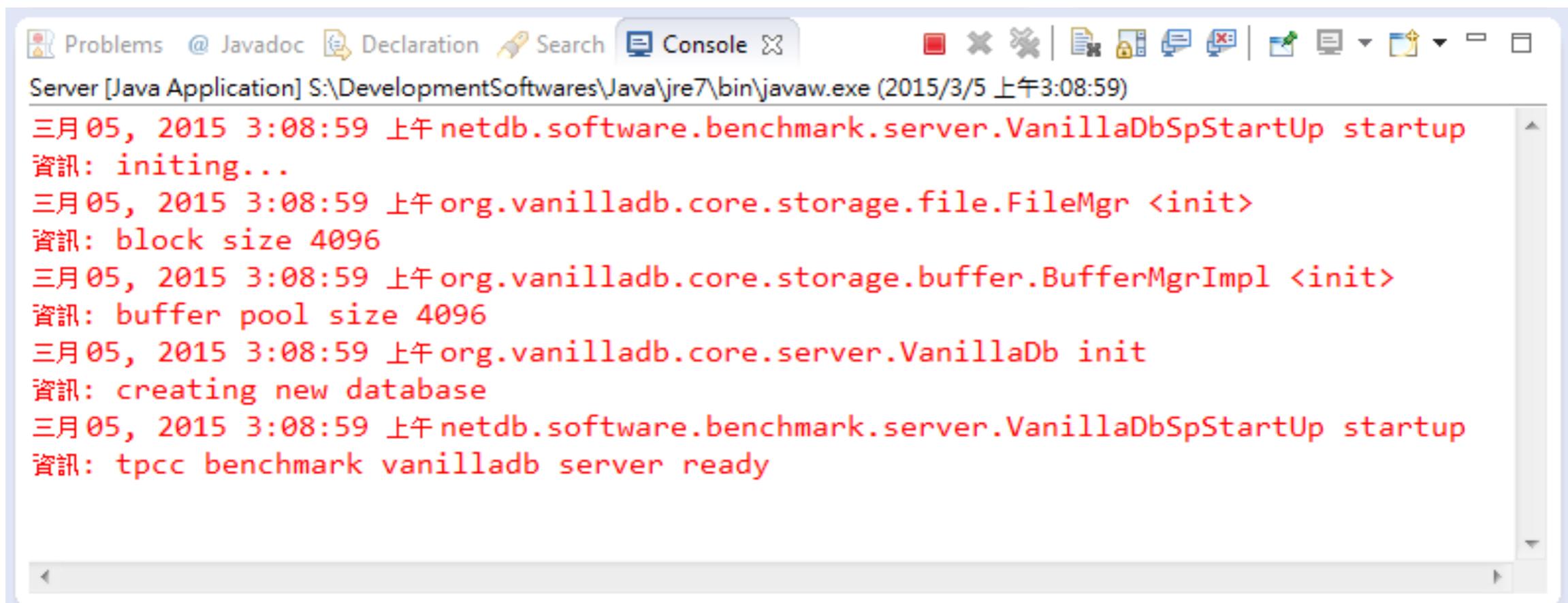
- You also need to add one more VM argument for benchmarking.
- Don't forget to add the Database Directory Name

```
-Dnetdb.software.benchmark.config.file=target/classes/netdb/  
software/benchmark/benchmark.properties
```



You can copy those arguments from [here](#)

# Server Messages



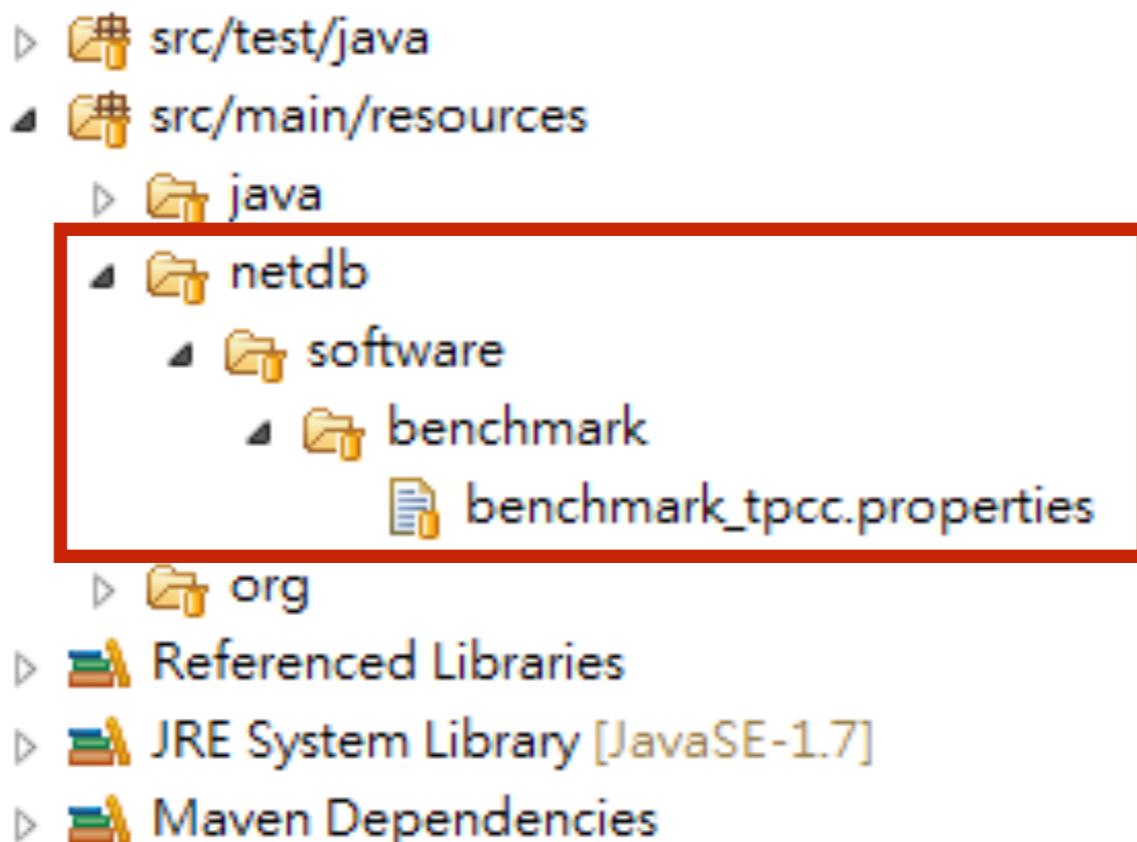
The screenshot shows a Java application console window titled "Server [Java Application] S:\DevelopmentSoftwares\Java\jre7\bin\javaw.exe (2015/3/5 上午3:08:59)". The window displays the following log output:

```
三月 05, 2015 3:08:59 上午 netdb.software.benchmark.server.VanillaDbSpStartUp startup  
資訊: initing...  
三月 05, 2015 3:08:59 上午 org.vanilladb.core.storage.file.FileMgr <init>  
資訊: block size 4096  
三月 05, 2015 3:08:59 上午 org.vanilladb.core.storage.buffer.BufferMgrImpl <init>  
資訊: buffer pool size 4096  
三月 05, 2015 3:08:59 上午 org.vanilladb.core.server.VanillaDb init  
資訊: creating new database  
三月 05, 2015 3:08:59 上午 netdb.software.benchmark.server.VanillaDbSpStartUp startup  
資訊: tpcc benchmark vanilladb server ready
```

You should see similar messages like before,  
if there is nothing wrong.

# Setting Benchmark

- Benchmark project also has a properties file for configurations
  - `benchmark_tpcc.properties`



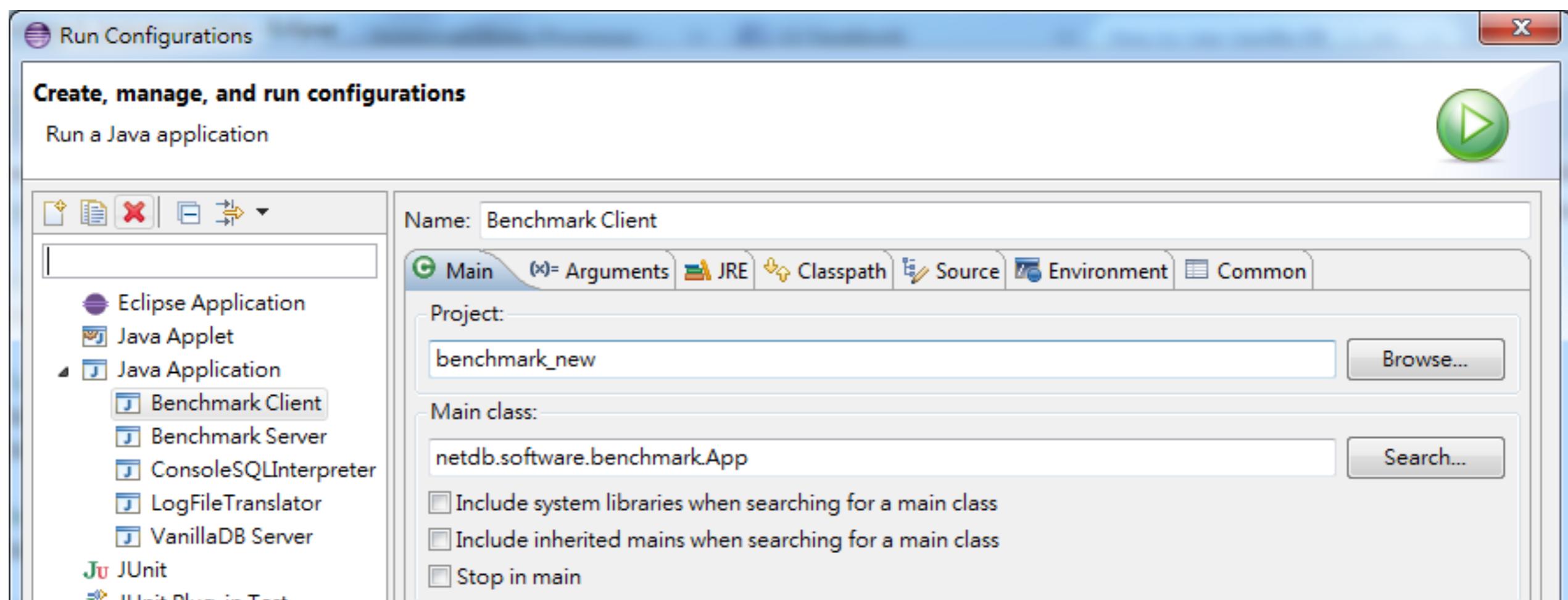
```
benchmark.properties  TestingParameters.java  vanilladb.properties  SampleTxnProc.java  SampleTxnProcPara
1 # Benchmark Parameters
2
3 # Number of items in the test-bed
4 netdb.software.benchmark.TpccConstants.NUM_ITEMS=100000
5
6 # Testing parameters
7 netdb.software.benchmark.TestingParameters.WARM_UP_INTERVAL=60000
8 netdb.software.benchmark.TestingParameters.BENCHMARK_INTERVAL=60000
9
10 # Decide the connection mode is JDBC=1 or Stored Procedures=2
11 netdb.software.benchmark.TestingParameters.CONNECTION_MODE=2
12
13 # The number of remote terminal executors of this bench-marker
14 netdb.software.benchmark.TestingParameters.NUM_RTES=20
15
16 # Statistic manager's parameters, the default is the home directory of current user
17 netdb.software.benchmark.StatisticMgr.OUTPUT_DIR=
18
19 # Remote server IP of vanilla-core. The default server IP is local host.
20 # If you want to use local host, just make this value empty.
21 netdb.software.benchmark.remote.vanilladb.VanillaDbDriver.SERVER_IP=127.0.0.1
22
23 # Choose the action the bench-marker takes (0: do nothing, 1: load test-bed, 2: do benchmarking)
24 netdb.software.benchmark.App.ACTION=2
25
```

# Important Parameters

- CONNECTION\_MODE
  - It means whether it should use JDBC or stored procedures
- App.ACTION
  - 0 : Do Nothing
  - 1 : Load Test-Bed
  - 2 : Lunch Benchmark

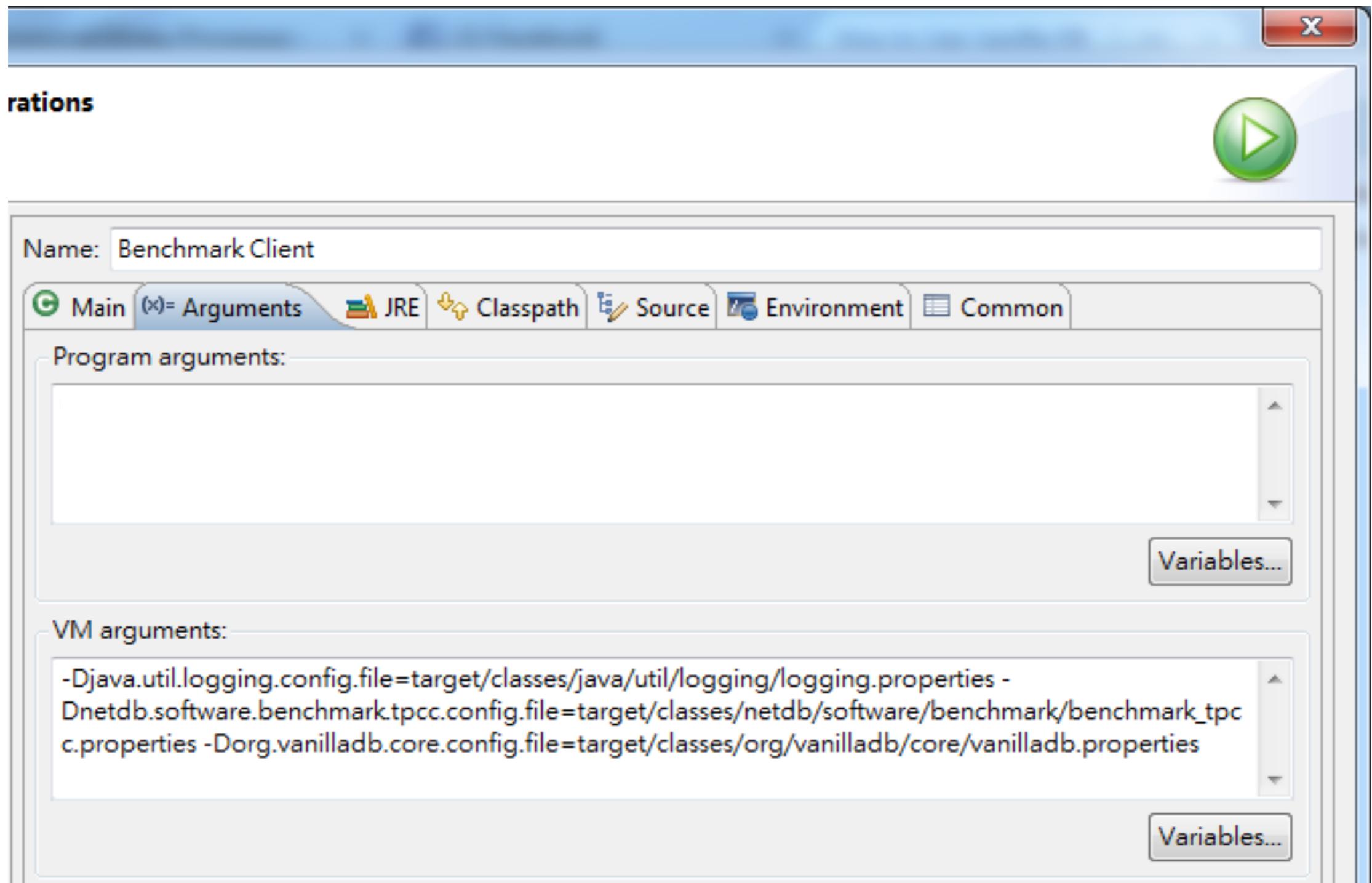
# Running Client

- To run clients, create a run configuration for it



# Arguments

- We also need to set some arguments
- Program Arguments
  - No need
- VM Arguments
  - Use the same as the server



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

# Notes

- Before running testing, please **load testbed** first
- Loading testbed only works when you use stored procedures as interface
- Check if the output directory path that show in properties file exists
  - If not, create one or change the path

# Run VanillaDB in Command Line

- Export client and server in .jar format
- Copy three property files into the folder where you export .jar file
- Type in the command as follow:

```
java { VM Arguments } -jar { server/client }.jar { Program Arguments }
```

# Q&A

- If you got any problem, you can check here first
- <https://shwu10.cs.nthu.edu.tw/courses-cloud-databases-2017-spring/FAQ>
- If your problem was very unique, just send a email let us know