

Detailed guide to run Demo/Callables on localhost

RUNNING CALLABLES

For Windows it is recommended to use some kind of bash-like environment like Git Bash (see <https://git-for-windows.github.io/>) or Ubuntu Bash on Windows 10. This guide will be using Git Bash on Windows and is compatible with Linux Bash. Be sure to run your chosen bash as an admin if on Windows. It is recommended that your Java runtime environment is 1.8.0_151 or higher (check with `'java -version'`). We will also need the Java SE Development Kit version 1.8.0_151 or higher (check with `'javac -version'`). See the Installing JDK section below for how to get the development kit. You can also run the demo without the JDK if you use the provided rmiregistry binary files, but you will not be able to recompile the Demos. There are some slight changes to the instructions for the non-JDK method that will be in **red**.

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ls
Detailed_Demo_Callables.pdf  Oracle  WorkNode.jar  rundemo.sh  src  startreg_unix.sh  startreg_win.sh
LICENSE                     README.md  bin           rundemo_nojdk.sh
```

Fig. 1: main directory

To begin, you must start the RMIregistry in the bin directory so it has access to the RemoteMethods class.

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ cd bin/

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService/bin (master)
$ rmiregistry 5555
```

Fig. 2

For non-JDK use the `startreg_win.sh` script for windows or the `startreg_unix.sh` script for unix. You can run these from the top-level directory (*DistributedExecutorService/*) with the port you want as an argument:

- `./startreg_win.sh 5555`

For a more practical purpose, one would background (&) this process. From here we'll start a work node, which is the actual program that handles the distributed executor service's work. There is a .jar file in the main directory you can run that starts a work node:

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
```

Fig 3.

This starts a work node and connects to an RMI registry on localhost at port 5555 with debug enabled. It is also possible to start a work node from the bin directory using the command

- `java worknode.WorkNode <host> <port> <debug>`

Passing more than two arguments will enable debugging, which outputs a constant tick every second of how many requests the node is currently handling.

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: 3d3b97f6:15f8aea7917:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

Fig. 4

The node takes some time to initialize and then constantly outputs 0 tasks. To run the demo, we will open another terminal to connect on localhost. You can either run the rundemo.sh script from the main directory which has the following format

- `./rundemo.sh <Demo_To_Run> <host> <port> <arg3> <arg4> ...`

Or run the java program directly from the bin/ directory using

- `java Demo.<Demo_To_run> <host> <port> <arg3> <arg4> ...`

The rundemo.sh script will rebuild the source files and create the bin directory if it does not exist. For each you specify the host and port the RMI registry is on and any arguments that a Demo might take. A screenshot of running the script for the Callables demo next to the work node terminal is shown below.

For non-JDK replace '`./rundemo.sh`' with '`./rundemo_nojdk`'. Keep in mind that this script does not rebuild the files, so no changes to the source files will be compiled, while '`./rundemo.sh`' does rebuild.

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh Callables localhost 5555
building Callables
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running Callables with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e

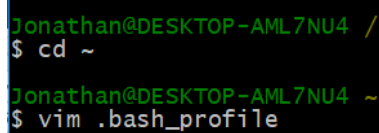
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: 2cabf1d5:15f8afe631a:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
Feeling Sleepy...
~I have 6 Tasks~
~I have 6 Tasks~
~I have 6 Tasks~
Hello World!
Feeling Sleepy...
~I have 5 Tasks~
~I have 5 Tasks~
```

Fig. 5

You have now successfully completed the demo! If there are exceptions thrown in regards to the connection to the RMI server, it is recommended to terminate all work nodes / running programs and restart the RMI registry. For a complete breakdown of what the Callables Demo does, see the Callables Breakdown section below

WINDOWS JAVA SE DEVELOPMENT KIT INSTALLATION

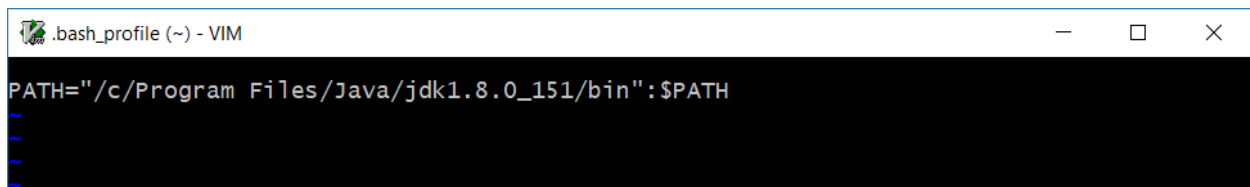
This package relies on the user to start the RMI registry themselves, so we also need to download the Java SDK which can be found on the Oracle site <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> (version 1.8.0_151 or higher is required). Once downloaded run the jdk-8u151-windows-x64.exe installer and let it complete. Keep track of where the jdk is installed, by default it should be in c:/Program Files/Java/. To quickly set up your path variables with the SDK programs, edit (or create and edit) the .bash_profile in your home directory:



```
Jonathan@DESKTOP-AML7NU4 /  
$ cd ~  
  
Jonathan@DESKTOP-AML7NU4 ~  
$ vim .bash_profile
```

Fig. 6

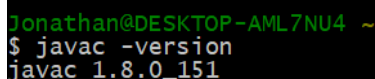
Then add the following line:



```
.bash_profile (~) - VIM  
PATH="/c/Program Files/Java/jdk1.8.0_151/bin":$PATH  
~  
~  
~
```

Fig. 7

But inside the quotes give the path to your jdk if it was placed in another folder. This places the Java SDK at the beginning of your path. From here you should be good to go. You can check by running SDK commands or checking javac with -version.



```
Jonathan@DESKTOP-AML7NU4 ~  
$ javac -version  
javac 1.8.0_151
```

Fig. 8

LINUX JAVA SE DEVELOPMENT KIT INSTALLATION

For an Ubuntu distribution you can get the java SDK using the command

- `sudo apt-get install openjdk-8-jdk`

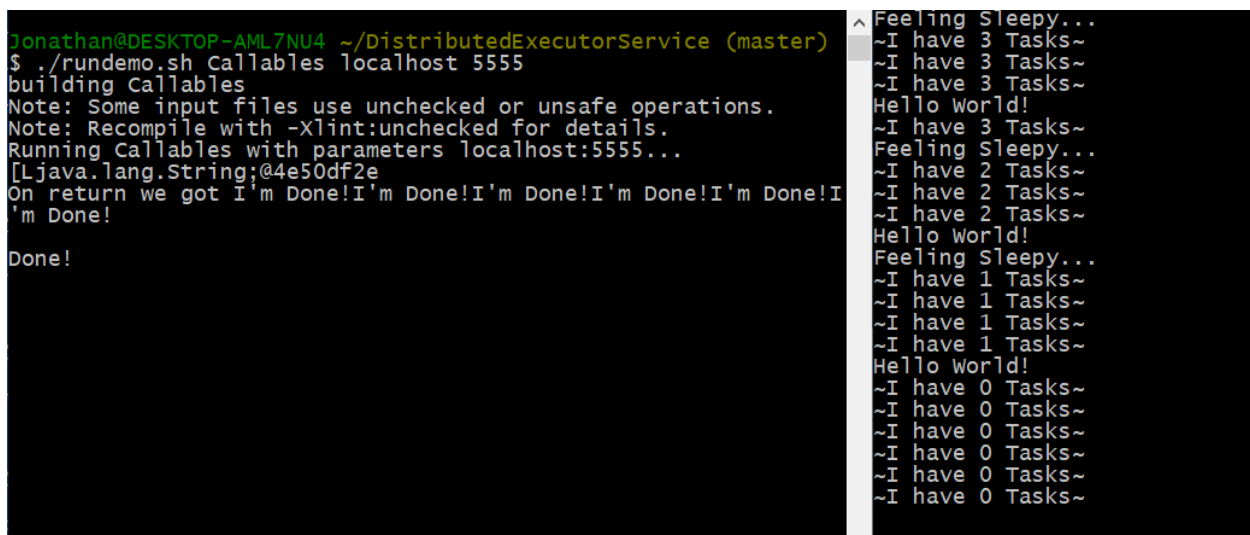
Then export the environment variable

- `export PATH=/usr/lib/jvm/java-8-openjdk/bin:$PATH`

To put it at the beginning of your path. Make sure to point it to your actual jdk if it is not located in /usr/lib/jvm. Test with javac -version or another SDK program to make sure everything is installed correctly.

CALLABLES BREAKDOWN

in src/demos/callables.java the program first creates a distributed executor service, then creates a new callable SleepyHelloWorldCall thread. What this thread will do is sleep for 7 seconds and then return the string "I'm Done!". We submit this thread to the executor service 6 times and save the returned futures, exactly like it was a normal executor service. We then call get() on the futures, where each call blocks until the thread that is running on another terminal finishes. This execution can be seen in figure 5, where the work node goes from 0 tasks to 6. It prints that it is sleeping, then when it wakes up it prints "Hello World!" and the work node moves to 5 tasks. Each thread is returning a string when it completes, which can be seen in Figure 10 below. This is the same execution as Figure 5 but later in time when all the threads finished. The Callables.java program prints out the return value of each future ("I'm Done!") before printing "Done!".



```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh Callables localhost 5555
building Callables
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running Callables with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
On return we got I'm Done!I'm Done!I'm Done!I'm Done!I'm Done!
I'm Done!
Done!
```

```
^ Feeling Sleepy...
~I have 3 Tasks~
~I have 3 Tasks~
~I have 3 Tasks~
Hello World!
~I have 3 Tasks~
Feeling Sleepy...
~I have 2 Tasks~
~I have 2 Tasks~
~I have 2 Tasks~
Hello World!
Feeling Sleepy...
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
Hello World!
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

Fig. 10

To get a larger picture of the distributed executor service, the figure on the next page shows the result of running the CallablesRNG Demo. This demo program starts a given number of threads that each return a random number from a work node. In this example there are 8 work nodes and the CallablesRNG is set to request 100 random numbers. These requests split as evenly as possible among the work nodes.

```
MINGW32/c/Users/Jonathan/DistributedEx...  -  □  ×
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecut
orService (master)
$ ./rundemo.sh CallablesRNG localhost 5555
100
building CallablesRNG
Note: Some input files use unchecked or unsa
fe operations.
Note: Recompile with -Xlint:unchecked for de
tails.
Running CallablesRNG with parameters localho
st:5555 ...
[Ljava.lang.String;@4e50df2e
Results:
933301299 -2034394421 -1165371974 -11923355
69 -721538402 -275264663 -1514567632 -191262
1542 1409478371 1459011932 -378744248 130169
1760 -2144287850 -1647558930 -1958045867 115
5088405 -390198506 -1572766933 1966034367 20
50813573 -908891976 1745350486 1593221845 -2
29104879 -287048032 -2063879226 -1372833352
-2084817344 -1945612106 -501282382 -19936299
24 1204264133 -776128375 -1307654382 4255676
52 1086133520 669303797 707073554 1936016104
2082521647 -1147033447 -1491789905 84484742
0 1496332335 -639616770 -1147592514 -1789418
935 1529564042 1461616092 -271018240 -191299
2130 1855478164 790651850 804435754 12493278
0 1187132998 994201260 -582221803 688044895
1970128545 614609285 1352006602 495690081 -1
088640903 -1530621650 -1353791675 -190792427
8 -714818159 -494857877 -1520833773 -4188792
3 1573129032

MINGW32/c/Users/Jonathan/Distribut...  -  □  ×
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
Your Random Number Is...
-I have 12 Tasks~
-1514567632!!!
-I have 12 Tasks~
Your Random Number Is...
1459011932!!!
-I have 11 Tasks~
Your Random Number Is...
-229104879!!!
-I have 10 Tasks~
Your Random Number Is...
-2063879226!!!
-I have 8 Tasks~
Your Random Number Is...
669303797!!!
-I have 8 Tasks~
Your Random Number Is...
1529564042!!!
-I have 7 Tasks~
Your Random Number Is...
-271018240!!!
-I have 6 Tasks~
Your Random Number Is...
614609285!!!
Your Random Number Is...
-I have 4 Tasks~
1573129032!!!
-I have 4 Tasks~
Your Random Number Is...

MINGW32/c/Users/Jonathan/Distribut...  -  □  ×
Node ready: 56229ea6:15f8b31e8f3:-7fff
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
Your Random Number Is...
933301299!!!
-I have 13 Tasks~
Your Random Number Is...
-1165371974!!!
-I have 12 Tasks~
Your Random Number Is...
-390198506!!!
Your Random Number Is...
-I have 10 Tasks~
-2084817344!!!
-I have 10 Tasks~
Your Random Number Is...
1936016104!!!
-I have 9 Tasks~
Your Random Number Is...
-1147033447!!!
-I have 8 Tasks~
Your Random Number Is...
1855478164!!!
Your Random Number Is...
-I have 6 Tasks~
495690081!!!
-I have 6 Tasks~
Your Random Number Is...
-1530621650!!!
-I have 5 Tasks~
Your Random Number Is...

MINGW32/c/Users/Jonathan/Distribut...  -  □  ×
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
-I have 0 Tasks~
Your Random Number Is...
1745350486!!!
Your Random Number Is...
-501282382!!!
-I have 9 Tasks~
Your Random Number Is...
707073554!!!
-I have 8 Tasks~
Your Random Number Is...
-1147592514!!!
-I have 7 Tasks~
Your Random Number Is...
804435754!!!
Your Random Number Is...
1352006602!!!
-I have 5 Tasks~
Your Random Number Is...
-1520833773!!!
-I have 4 Tasks~
Your Random Number Is...

MINGW32/c/Users/Jonathan/Distribut...  -  □  ×
-I have 0 Tasks~
Node ready: 35aaef3c:15f8b31f239:-7fff
-I have 0 Tasks~
-I have 0 Tasks~
Your Random Number Is...
-275264663!!!
-I have 12 Tasks~
Your Random Number Is...
1409478371!!!
-I have 11 Tasks~
Your Random Number Is...
1593221845!!!
-I have 10 Tasks~
Your Random Number Is...
-287048032!!!
Your Random Number Is...
-I have 8 Tasks~
1086133520!!!
-I have 8 Tasks~
Your Random Number Is...
-1789418935!!!
-I have 7 Tasks~
Your Random Number Is...
1461616092!!!
-I have 6 Tasks~
Your Random Number Is...
1970128545!!!
Your Random Number Is...
-I have 4 Tasks~
-41887923!!!
-I have 4 Tasks~
Your Random Number Is...

MINGW32/c/Users/Jonathan/Distribut...  -  □  ×
Debug Activated
-I have 0 Tasks~
Node ready: -29699207:15f8b31f6a9:-7fff
-I have 0 Tasks~
Your Random Number Is...
-2034394421!!!
-I have 12 Tasks~
Your Random Number Is...
-I have 11 Tasks~
-1192335569!!!
Your Random Number Is...
-I have 10 Tasks~
-908891976!!!
-I have 10 Tasks~
Your Random Number Is...
1204264133!!!
-I have 9 Tasks~
Your Random Number Is...
-1307654382!!!
-I have 8 Tasks~
Your Random Number Is...
-639616770!!!
-I have 7 Tasks~
Your Random Number Is...
1187132998!!!
-I have 6 Tasks~
Your Random Number Is...
-582221803!!!
-I have 5 Tasks~
Your Random Number Is...
-494857877!!!
-I have 4 Tasks~
Your Random Number Is...
```