

## ***Introduction***

## ***Code Repository***

The source code for the Whispir project is available at <https://github.com/johnfarnell/whispir>. Checkout the *Master* branch to a local folder which will be referred to as <ROOT\_FOLDER>

## ***Prerequisites***

To build and run the software the following is required

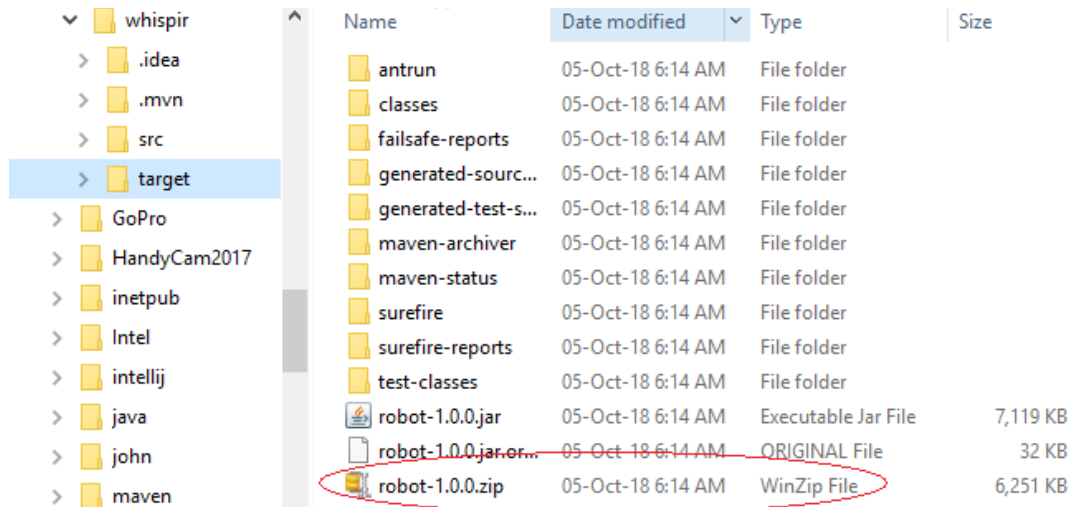
1. Maven 3
2. Java 1.8

## ***Build the software***

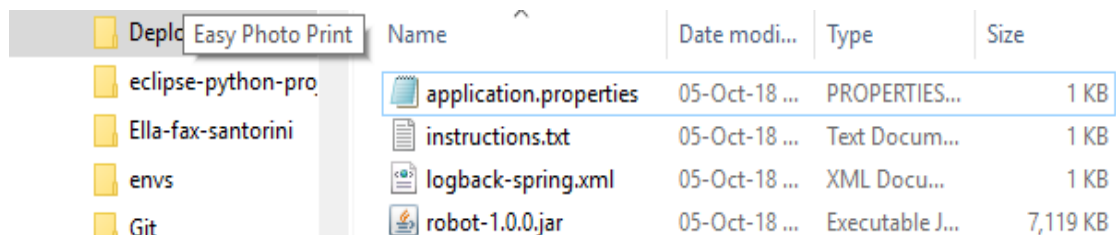
From the command line at the <ROOT\_FOLDER>, enter *mvn clean install*

## Install and run the software

After successfully building the software, navigate to <ROOT\_FOLDER>/target and locate the file **robot-1.0.0.zip**

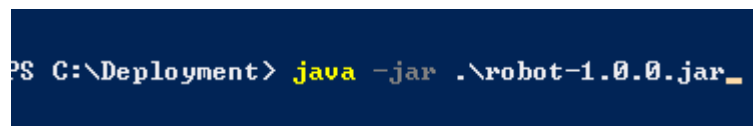


and extract the contents of this folder to a location e.g (C:\\Deployment)of your choosing (referred to later as the <DEPLOYMENT\_FOLDER> – below is an image of what the zip file contains



Navigate to the folder and, from the command line run

```
java -jar robot-1.0.0.jar
```



The console log will show the following, and you will be invited interactively to move the robot onto the table and manoeuvre accordingly :-

```
PS C:\Deployment> java -jar .\robot-1.0.0.jar

=====
:: Spring Boot ::                (v2.0.5.RELEASE)

2018-10-05 06:37:12.722 INFO 26292 --- [
ID 26292 <C:\Deployment\robot-1.0.0.jar started by C
2018-10-05 06:37:12.727 INFO 26292 --- [
efault
2018-10-05 06:37:12.798 INFO 26292 --- [
onConfigApplicationContext@27abe2cd: startup date [F
2018-10-05 06:37:13.679 INFO 26292 --- [
2018-10-05 06:37:13.700 INFO 26292 --- [
 2.203>
2018-10-05 06:37:13.703 INFO 26292 --- [

Enter an instruction
e.g. "PLACE 0,0,NORTH"
or "LEFT"
or "RIGHT"
or "MOVE"
or "REPORT"
or just press ENTER to exit

---> _
```

Note, you must PLACE the robot on the table first, before performing any other instructions. Entering REPORT at the command prompt will provide details of the robots orientation (coordinates/direction). Invalid commands will be reported but ignored, the robot will not be able to leave the table, any attempt to do this will be reported and ignored. To end the application, just press ENTER and confirm.

## Alternate processing?

It is possible to batch a series of commands into a file. Looking back at the <DEPLOYMENT\_FOLDER> , you will see 2 files *application.properties* and *instructions.txt*. The text file contains a list of instructions. This file can be manipulated in any way to provide a list of instructions for the ROBOT. To switch the input source from the console to the file open and edit the following line in *application.properties*

```
# Used by FileCommandDao to identify the file containing the instructions on the root of the classpath
command.file=instructions.txt
# Used in ApplicationConfiguration to determine which implementation of CommandDao it to be used.
commandDao=scanner
# The following table based properties define the boundaries of the table within which the Robot must remain
table.bottomLeftX=0
table.bottomRightY=0
table.topRightX=2
table.topRightY=1
```

Change *commandDao=scanner* to *commandDao=file*

```
# Used by FileCommandDao to identify the file containing the instructions on the root of the classpath
command.file=instructions.txt
# Used in ApplicationConfiguration to determine which implementation of CommandDao it to be used.
commandDao=file
# The following table based properties define the boundaries of the table within which the Robot must remain
table.bottomLeftX=0
table.bottomRightY=0
table.topRightX=2
table.topRightY=1
```

NOTE: You can also edit the bottom left and top right coordinates of the Table.

Return to the <DEPLOYMENT\_FOLDER> and rerun

```
2018-10-05 06:55:34.853 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction PLACE 0,0,EAST
2018-10-05 06:55:34.856 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction REPORT
Robot Robot 1
(0,0) EAST
2018-10-05 06:55:34.857 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction MOVE
2018-10-05 06:55:34.858 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction REPORT
Robot Robot 1
(1,0) EAST
2018-10-05 06:55:34.858 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction LEFT
2018-10-05 06:55:34.858 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction REPORT
Robot Robot 1
(1,0) NORTH
2018-10-05 06:55:34.859 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction MOVE
2018-10-05 06:55:34.860 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction REPORT
Robot Robot 1
(1,1) NORTH
2018-10-05 06:55:34.861 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction RIGHT
2018-10-05 06:55:34.862 INFO 16608 --- [main] a.com.farnellj.robot.dao.FileCommandDao : instruction REPORT
Robot Robot 1
```

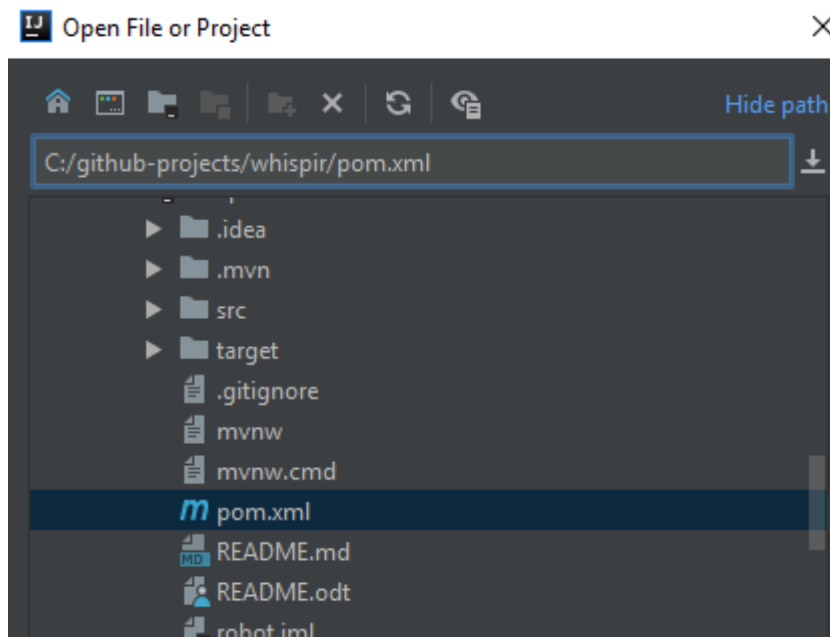
This time the application ran through the commands from start to finish without any console interaction

## Design Process

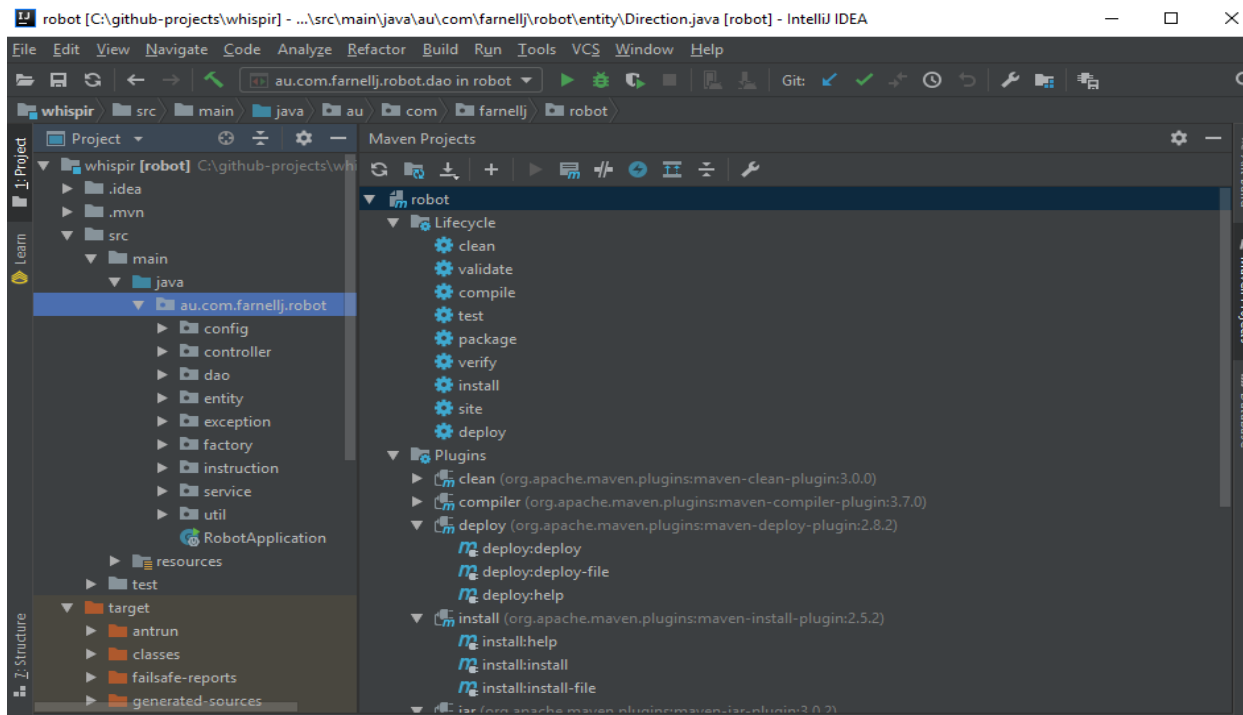
The application uses Spring BOOT and was developed in IntelliJ.



To view the source code open IntelliJ and then, within the IDE, open pom.xml at <ROOT\_FOLDER>, select *Open as Project*



It is a standard maven project with a main application with the entry point at *RobotApplication* which when run starts the spring boot based application



## Testing

Mockito and junit have been used to test the individual classes.

