

# INB370 Assignment 2 Submission Procedure

---

This document outlines the steps you need to follow to submit INB370 Assignment 2. As with Assignment 1, a number of automated tools will be used to assist with the marking of part 1. To this end, please follow these instructions carefully so that your files are in the appropriate folders to be detected by our tools. **In particular, check that you have spelled the names of your Java packages, classes and methods correctly and that they are capitalised exactly as specified.**

## Exporting Your Eclipse Project

Your entire submission will be contained in a single zip archive exported from within Eclipse. From the initial release, the project was named `CarParkSimulator` and you should make sure that you maintain this name. (To change the project's name, right-click on it and select `Refactor > Rename`.) The overall project file structure will look something like the one below:

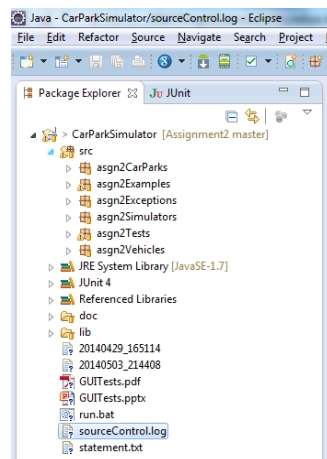


Figure 1 – Right-click on the project folder and choose the `Export...` menu entry.

In the top directory, you will see the packages that you have been working with. Not all of these need be submitted. You will also see some files which we list in turn below.

**statement.txt** – a TEXT file, not a word doc or docx, which contains the following information about you and about your assignment:

- The name and student number of both partners.
- A statement of contribution from each partner and a note that you *both* agree with what is written
- Any known bugs or incomplete features in the program
- A clear statement of the location of class which holds the main method for your GUI and the instructions needed to run it, whether or not you have provided us with a `run.bat` method (which of course you should – see below)

- Any email from me giving permission to do the assignment alone, or any extenuating circumstances or extensions granted
- Anything else you think we need to know to mark your assignment – this includes any details of approved non-standard library use.

*run.bat* - a windows batch file that specifies the command to run the simulator with the defaults i.e. no arguments. If you specify it as shown below, adjusting as needed if your project uses a different main class from GUISimulator. We will use this to run things conveniently.

```
java -cp ".;bin" asgn2Simulators.GUISimulator
```

You may jar the application binaries if you wish (see the build prac) but you will have to specify the jar explicitly on the classpath.

*sourceControl.log* – This will be a general name used for source control (usually git) logs. For most people, this means capturing the results of the `git log` command in a file of this name. If we don't believe your logs, we reserve the right to demand a look at your repo, but it is too cumbersome to have access to all of them by default.

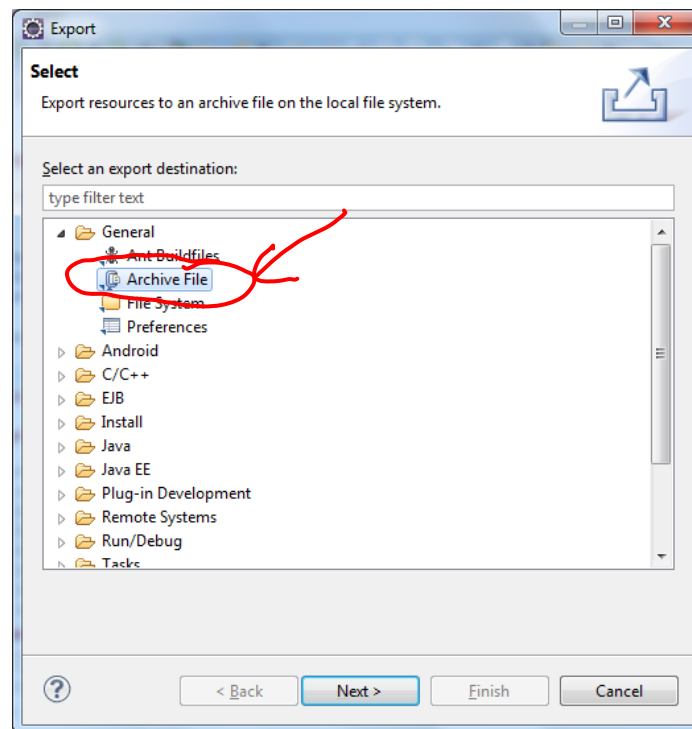
*GUITests.pptx* or *GUITests.pdf* – NOT *GUITests.docx* or *GUITests.doc* – this is the file containing your GUI test screenshots. Please don't use a word doc. Each test case should show screenshots for:

- The GUI state before pressing start (showing the numbers)
- The TextArea in its final form
- The line chart at the end of the simulation

Please see the Notes on Testing at the end of the document. For the sake of clarity, you may at your discretion change the scale of these line plots before capture, even if the total cars and number archived go off the scale. We would prefer to see the trends. [Just tell us in *statement.txt*]. Various log files are also included above – these aren't required but if you want to submit logs corresponding to your tests you may do so.

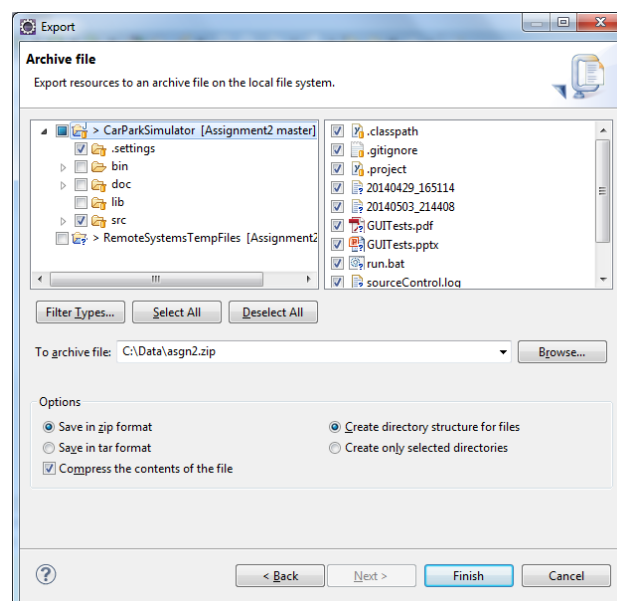
### ***Organising the archive:***

Once you are ready to submit, open the Eclipse workspace containing your assignment. Right-click on the project folder in Figure 1 above and select the **Export...** entry in the pop-up menu. **It is important that you select the *project* folder, as shown in the figure, otherwise not all of your files will be included in the zip archive.** Once you have done this an Export Wizard dialogue box like that shown in Figure 2 will open. Select the **Archive File** option under the **General** heading



**Figure 2** – The first step of the Export Wizard, select the *Archive File* option and click *Next >*

The next step of the wizard is shown in Figure 3. In the top left list of projects ensure that there is a check mark next to the name of the project containing your assignment. Do not include any other project (your assignment submission must be contained in a single project). Your submission need not include the lib folder – unless an additional library is used, in which case you should tell us, the examples directory or the RemoteSystemsTempFiles. For most people, the set up as shown below should be fine:



**Figure 3** – Ensure that the project containing your assignment has a check next to it, that the key files at right are checked and that the src directory is included. (doc and bin can be regenerated). Select a save location; your zip file must be named *asn2.zip*. Once you are ready select *Finish* to export your work.

Make sure in the Options list that the radio button next to 'Save in zip format' is selected as are the 'Create directory structure for files' and 'Compress the contents of the file' options. Once this is done, press Finish.

You can check that the zip archive is correctly structured by unzipping it and confirming that this produces (at least) a CarParkSimulator folder containing the src directory and the key files in the top directory. Note that if you make any changes to your assignment *after* performing the above steps, you need to redo them to create a new zip archive before submitting it again through BB.

Part 1 of submitted assignments will be tested automatically, so you must adhere precisely to the specifications in these instructions. Your program code classes will be unit tested against our own test suite to ensure that they have the necessary functionality. Your unit test classes will be exercised on defective programs to ensure that they adequately detect programming errors. The GUI in part 2 will be manually assessed, but we will attempt to run your GUI against our Vehicle and CarPark classes. So if you break the spec, you will still have issues.

### Submitting Your Eclipse Project

When you have successfully created your zip archive `asgn2.zip` you should submit via the link on BB in the INB370>Assessment>Assignment 2 area. You must submit your solution before midnight (actually 11:59) on the due date to ensure that your assignment is accepted. You should take into account the fact that the network might be slow or temporarily unavailable when you try to submit. **Network problems near the deadline will not be accepted as an excuse for late assignments, and QUT's assignment submission policy is now very strict.**

## Some Notes on Testing

We promised to provide a small set of logs and parameters for you to use in exploring the system. Some guidance has been given above on the screenshots to be used in the GUI tests. Here we are focused on two things: the parameter settings, and the unit tests to be provided to you for you to check your work.

### *Provided Unit Tests:*

These are in a file called UnitTests.txt. It contains fragments of java code clearly labelled as to the class in which they should be inserted – CarTests.java, MotorCycleTests.java and CarParkTests.java. Recall that these tell you if you have a problem – passing them doesn't guarantee that you have it right.

Note that I said in the earlier email that we would provide you with a test for the getStatus method. In looking at it, I can't actually do this without releasing a whole bunch of helper methods as well, which would be unfair to those who have already written their own. I will thus release an archive of 4 logs (see below).

### *Log Files and Test Conditions:*

*Defaults.log – the standard conditions*

*NoQueue.log – as above, but no queue*

*SmallCars.log – 80% of car spaces are for small cars; we see dominance of small cars and motorcycles. Have a mean stay of 180mins*

*LongStay.log – longer stay. Mean of 300 mins, sd of 100.*

maxCarSpaces	maxSmallCarSpaces	maxMotorCycleSpaces	maxQueueSize	Seed	carProb	smallCarProb	mcProb	mean	sd	Log
100	30	20	10	100	1	0.2	0.05	120	39.6	defaults.log
100	30	20	0	100	1	0.2	0.05	120	39.6	noqueue.log
100	80	30	10	100	0.75	0.5	0.2	180	60	smallCars.log
100	30	30	10	100	0.75	0.2	0.1	300	100	longStay.log