# README:

The goal of this document is to provide the reader with technical and functional information regarding the **ship code verifier** application.

## Technical Documentation:

Git repository:

<https://github.com/raffivegas/WillSon>

## Versions of important things:

java version "11.0.8" 2020-07-14 LTS

Java(TM) SE Runtime Environment 18.9 (build 11.0.8+10-LTS)

Java 8.0.2610.12

Eclipse IDE for Java Developers

Version: 2020-09 (4.17.0)

Build id: 20200910-1200

JUnit 5 Jupiter 5.7.0

## What does it do?

This application is intended to be run as an excecutable jar file which can run on most computers which have the latest JVM installed (or the Java version listed above in the 'Versions of important things' section of this document).

The application reads a series of zip code ranges from a file and outputs a condensed set of ranges. The acception criteria (of the story or epic) is as follows: when given a collection of 5-digit ZIP code ranges (each range including both their upper and lower bounds), the application must produce the minimum number of ranges required to represent the same restrictions as the input.

## How does it work?

The input file format is a tab delimited text file (.txt) with zipcodes within brackets and the number pair separated by a comma. For example, a valid file may contain the following:

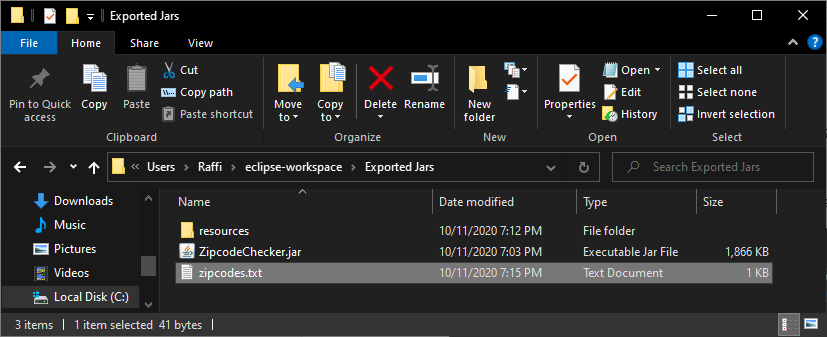
[94133, 94133] [94200,94299] [94226,94399]

As there is no file or input formatting validation implemented yet at this time, please ensure the file is formatted correctly with valid zipcode ranges.

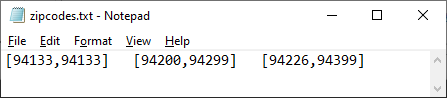
### Exporting and running the application:

After pulling the remote repository to local, open the project in Eclipse and export as an excecutable jar:

1. Right-click on ship code verifier project folder.
2. Left-click on Export...
3. Select the Runnable JAR File export wizard (you can search for it in the textbox).
4. Choose an available launch configuration.
5. Select and export destination and jar name (i.e. C:\Users\Raffi\eclipse-workspace\Exported Jars\ZipcodeChecker.jar)
6. Leave Library handling default (Extract required libraries...)
7. Click Finish
8. In the folder of the jar file, create a new text file. In this example, I will name the file zipcodes.txt.



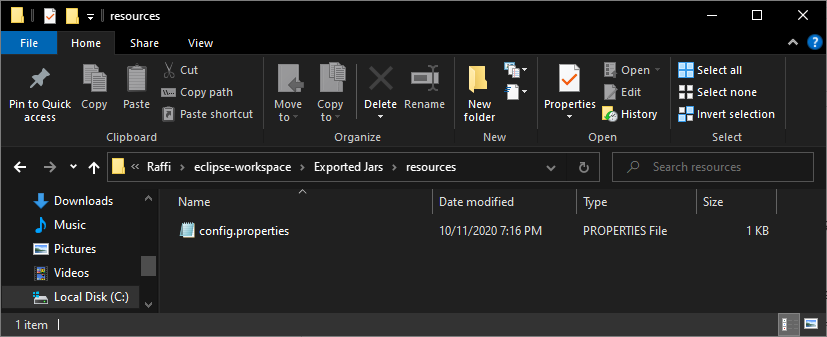
1. Edit the zipcodes.txt file and enter a tab delimited range of zipcodes as described earlier in this document.



1. In the folder of the jar file, create a new folder named 'resources', for example:

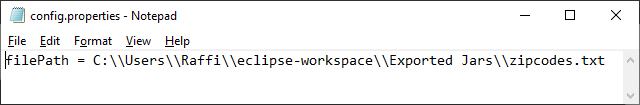
C:\Users\Raffi\eclipse-workspace\Exported Jars\resources

1. Inside the resources folder, either create or copy a config.properties file (copy from the Eclipse project's resources folder).



1. Edit the config.properties file and add a key value pair for the path to the input file (in this case zipcodes.txt) as such:

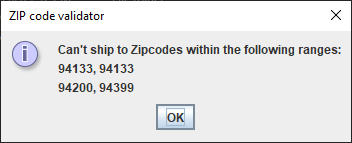
filePath = C:\\Users\\Raffi\\eclipse-workspace\\Exported Jars\\zipcodes.txt



replacing the folder names with your own.

To run the application and process the zipcodes, double-click on the jar file in the root folder. The application will display a window containing the results of the processing. The application may also be run within Eclipse, with the output displaying in the console. **IF RUNNING WITHIN ECLIPSE**, edit the config.properties file and update it with the correct path to your zipcode input file. If the application does not display anything or displays something incorrectly, please verify the input file format looks correct.

Application Output:



In the IDE/project, the application starting point is:

/src/logic/ShipCheckerMain.java

Unit tests are in /src/tests

README v0.0.1 - Raffi Tchakmakjian, 10/11/2020