Installation

Option 1: Install from Eclipse Marketplace (Easier, straightforward)

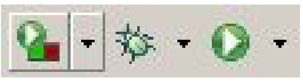
1. Help -> Eclipse Marketplace
2. Type “EclEmma” in the search box
3. If you didn’t install before, then press install button to install EclEmma, and restart Eclipse afterward. Else, you are all set and ready for testing.

Option 2: Manually install

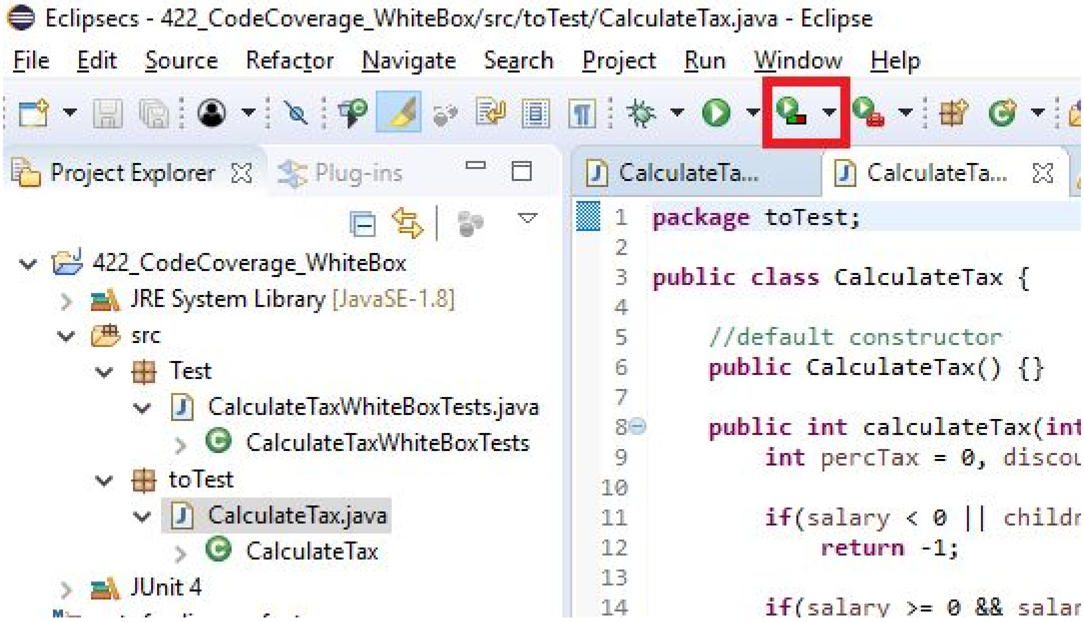
1. Download a version of EclEmma zip file from the following website: http://www.eclemma.org/download.html
2. Unzip the archive into dropins folder of your Eclipse installation, then restart Eclipse

User Guide

After one of the above installation, you should be able to see the EclEmma coverage button in the toolbar as following:



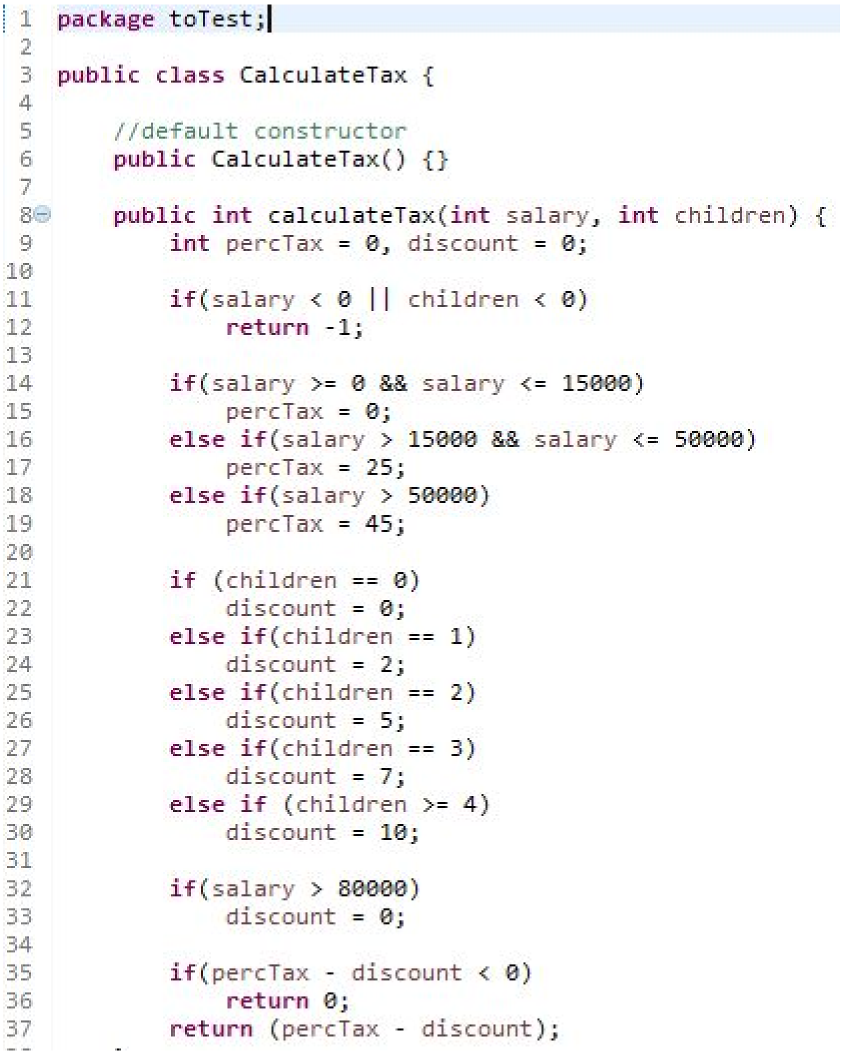
The most important button out of the toolbar is the one mark in the red box as below:


Eclipsecs - 422_CodeCoverage_WhiteBox/src/toTest/CalculateTax.java - Eclipse
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
Project Explorer
Plug-ins
CalculateTa...
CalculateTa...
422_CodeCoverage_WhiteBox
JRE System Library [JavaSE-1.8]
src
Test
CalculateTaxWhiteBoxTests.java
CalculateTaxWhiteBoxTests
toTest
CalculateTax.java
CalculateTax
JUnit 4

```
package toTest;

public class CalculateTax {

    //default constructor
    public CalculateTax() {}

    public int calculateTax(int
        int percTax = 0, discou

        if(salary < 0 || childr
            return -1;

        if(salary >= 0 && salar
```

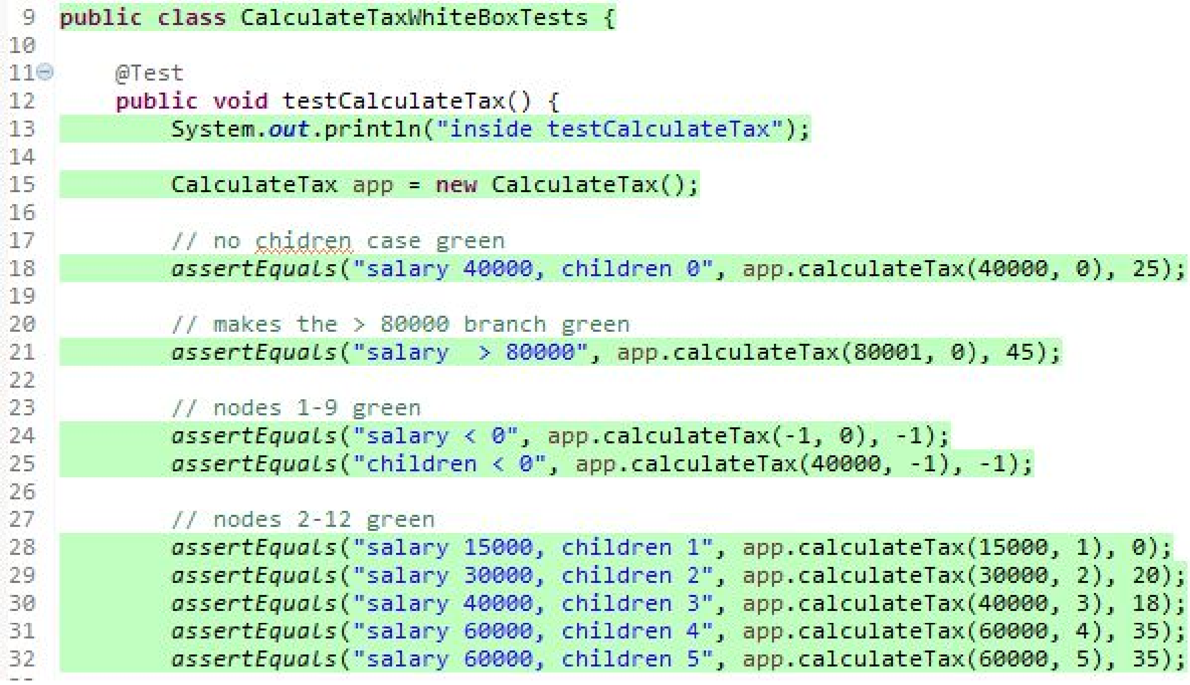
Example

Here is an example class CalculateTax. Inside the class there is a function/method having salary and children pass in as parameters.



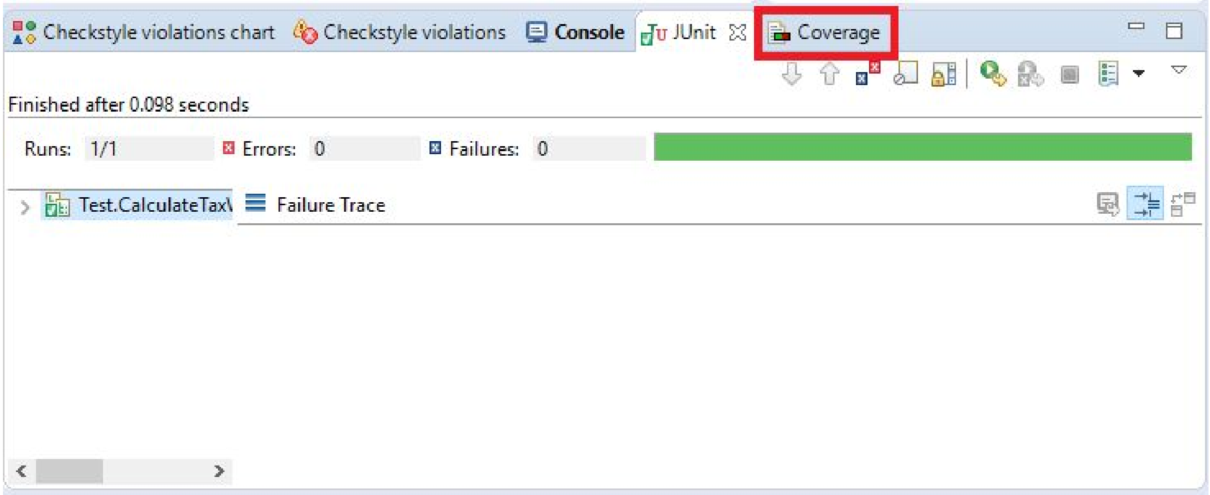
```java
package toTest;

public class CalculateTax {

    //default constructor
    public CalculateTax() {}

    public int calculateTax(int salary, int children) {
        int percTax = 0, discount = 0;

        if(salary < 0 || children < 0)
            return -1;

        if(salary >= 0 && salary <= 15000)
            percTax = 0;
        else if(salary > 15000 && salary <= 50000)
            percTax = 25;
        else if(salary > 50000)
            percTax = 45;

        if (children == 0)
            discount = 0;
        else if(children == 1)
            discount = 2;
        else if(children == 2)
            discount = 5;
        else if(children == 3)
            discount = 7;
        else if (children >= 4)
            discount = 10;

        if(salary > 80000)
            discount = 0;

        if(percTax - discount < 0)
            return 0;
        return (percTax - discount);
```

And the following is the test cases



```
public class CalculateTaxWhiteBoxTests {

    @Test
    public void testCalculateTax() {
        System.out.println("inside testCalculateTax");

        CalculateTax app = new CalculateTax();

        // no chidren case green
        assertEquals("salary 40000, children 0", app.calculateTax(40000, 0), 25);

        // makes the > 80000 branch green
        assertEquals("salary  > 80000", app.calculateTax(80001, 0), 45);

        // nodes 1-9 green
        assertEquals("salary < 0", app.calculateTax(-1, 0), -1);
        assertEquals("children < 0", app.calculateTax(40000, -1), -1);

        // nodes 2-12 green
        assertEquals("salary 15000, children 1", app.calculateTax(15000, 1), 0);
        assertEquals("salary 30000, children 2", app.calculateTax(30000, 2), 20);
        assertEquals("salary 40000, children 3", app.calculateTax(40000, 3), 18);
        assertEquals("salary 60000, children 4", app.calculateTax(60000, 4), 35);
        assertEquals("salary 60000, children 5", app.calculateTax(60000, 5), 35);
```

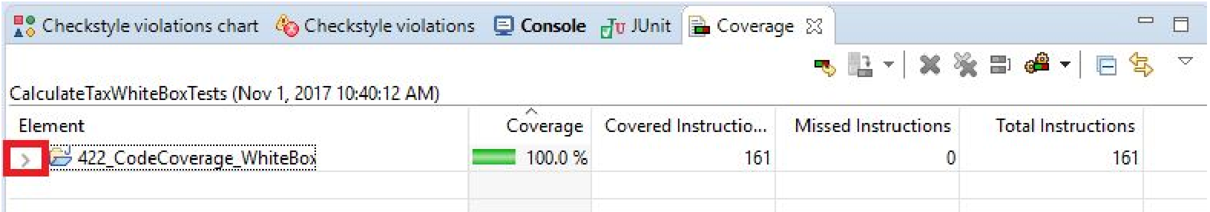
While you have your files open up, just simply press the button to test for the coverage. After finish loading, an additional block should appear on the bottom of Eclipse like below:



Finished after 0.098 seconds

Runs: 1/1 Errors: 0 Failures: 0

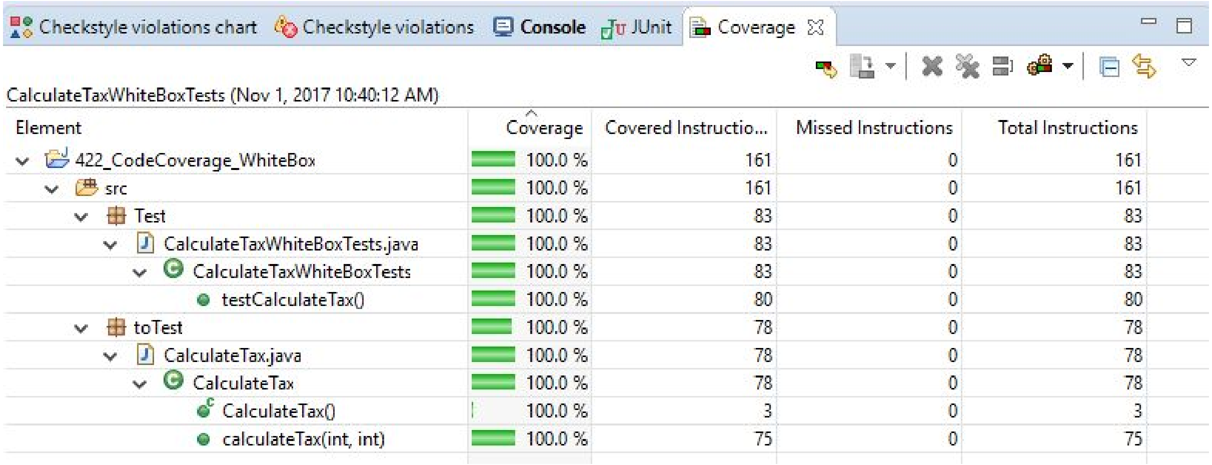
Tap on "Coverage". The Coverage column is the total coverage out of the entire package, and if you want to check on each individual class/function/method's coverage, click on the arrow mark as below, until you get to the part you want to review:



CalculateTaxWhiteBoxTests (Nov 1, 2017 10:40:12 AM)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
> 422_CodeCoverage_WhiteBox	100.0 %	161	0	161

Now you know all the coverage over the entire package with EclEmma.



CalculateTaxWhiteBoxTests (Nov 1, 2017 10:40:12 AM)

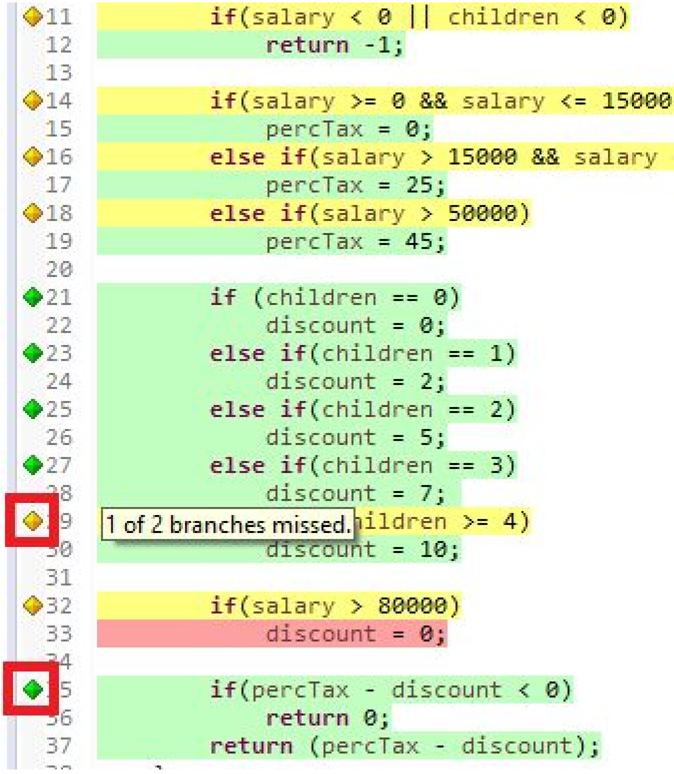
Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
422_CodeCoverage_WhiteBox	100.0 %	161	0	161
src	100.0 %	161	0	161
Test	100.0 %	83	0	83
CalculateTaxWhiteBoxTests.java	100.0 %	83	0	83
CalculateTaxWhiteBoxTests	100.0 %	83	0	83
testCalculateTax()	100.0 %	80	0	80
toTest	100.0 %	78	0	78
CalculateTax.java	100.0 %	78	0	78
CalculateTax	100.0 %	78	0	78
CalculateTax()	100.0 %	3	0	3
calculateTax(int, int)	100.0 %	75	0	75

Q: What to do if the coverage is not 100%?

There are more information can be helpful on the java page itself. Now when you get to the java file page, you realize the lines are either red/ yellow/ green. The meaning of each coloring as below:

red - The line is not coverage base on the test you wrote
yellow - There are path you didn't not cover in you test
green - This path is being cover

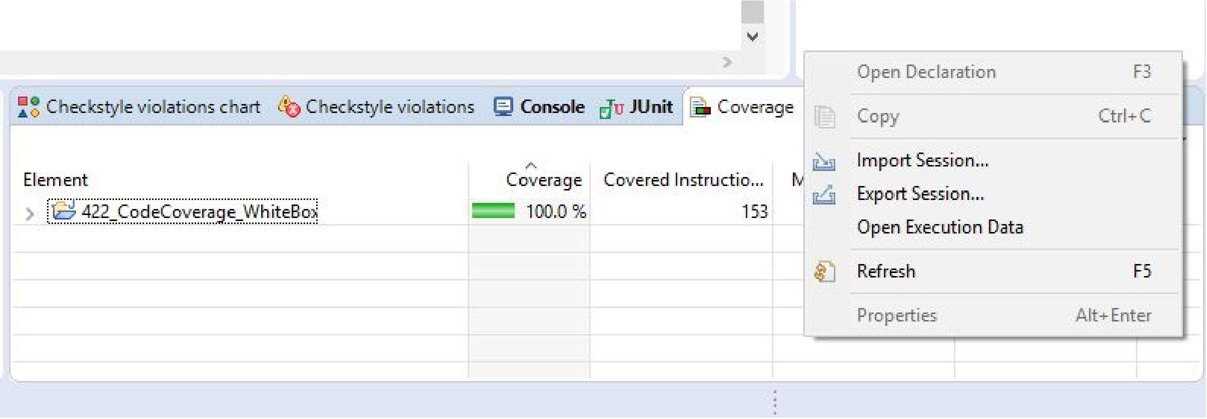
When your mouse hover on the green diamond shape, it will notice you all the branches are cover and the line will appear in green.
When your mouse hover on the yellow diamond shape, it will notice you have how many branches you are missing on coverage, and the line will appear in yellow.



```
        if(salary < 0 || children < 0)
            return -1;

        if(salary >= 0 && salary <= 15000)
            percTax = 0;
        else if(salary > 15000 && salary <= 50000)
            percTax = 25;
        else if(salary > 50000)
            percTax = 45;

        if (children == 0)
            discount = 0;
        else if(children == 1)
            discount = 2;
        else if(children == 2)
            discount = 5;
        else if(children == 3)
            discount = 7;
 1 of 2 branches missed.ildren >= 4)
            discount = 10;

        if(salary > 80000)
            discount = 0;

        if(percTax - discount < 0)
            return 0;
        return (percTax - discount);
```

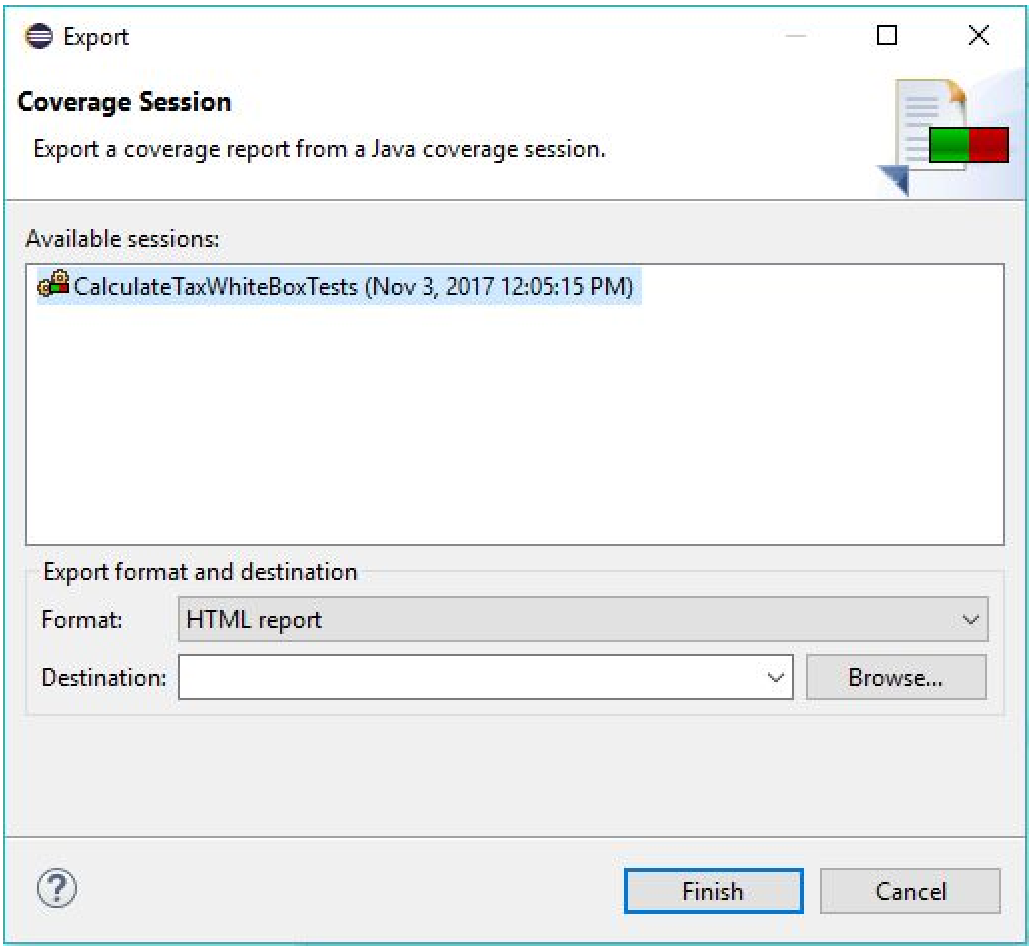
Q: How to export result into different type?

For user to export test result into different type, we will have to first run the test as shown in the example, then right click on your mouse on the coverage result panel like below:



Element	Coverage	Covered Instructio...
422_CodeCoverage_WhiteBox	100.0 %	153

There will be different options to choose from, but what we want is to click on “Export Session…”, and a popup window will show up as below:



Coverage Session

Export a coverage report from a Java coverage session.

Available sessions:

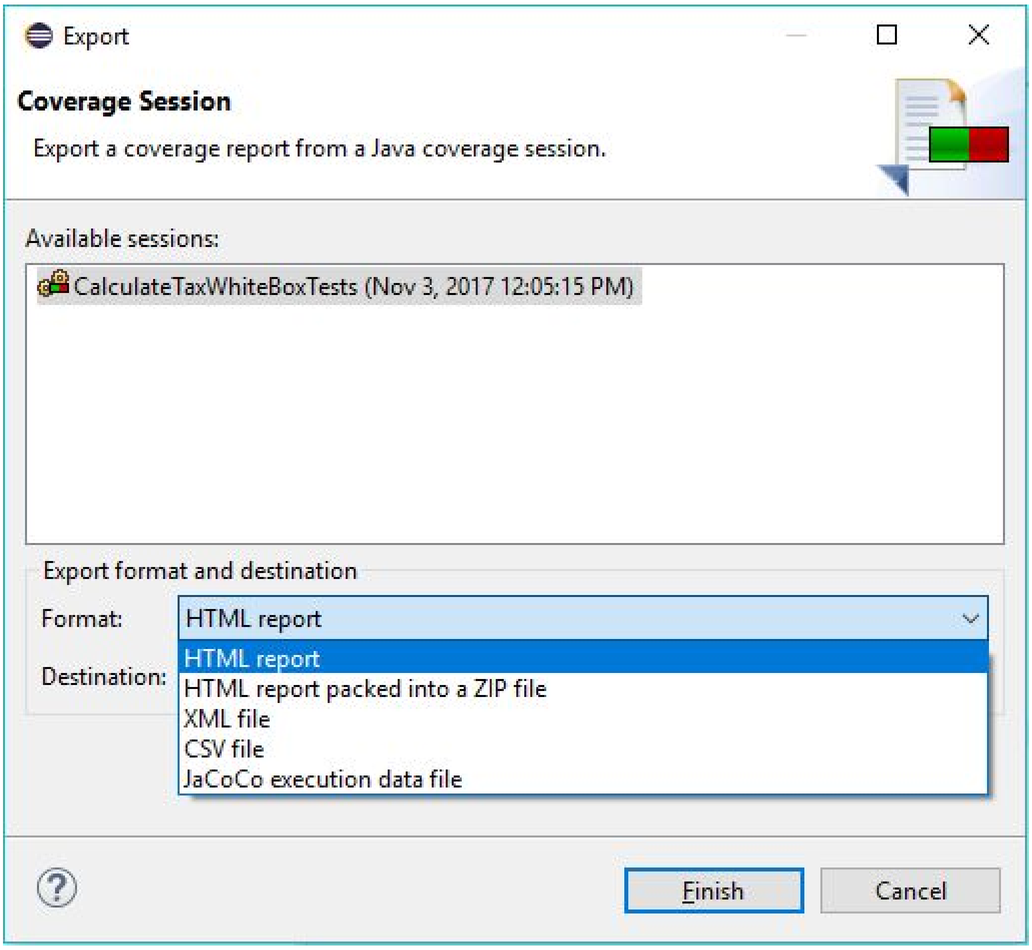
CalculateTaxWhiteBoxTests (Nov 3, 2017 12:05:15 PM)

Export format and destination

Format: HTML report

Destination:

The “Format” drop down menu have different type of report you can choose from.



Export

Coverage Session

Export a coverage report from a Java coverage session.

Available sessions:

CalculateTaxWhiteBoxTests (Nov 3, 2017 12:05:15 PM)

Export format and destination

Format: HTML report

Destination:

- HTML report
- HTML report packed into a ZIP file
- XML file
- CSV file
- JaCoCo execution data file

Pick the type of report you are looking for, then choose the “Destination” you want the file be save at, and then press the “Finish” button, and the file you request for will appear. Now you will have the type of file you requested and waiting to be review at the Destination location you choose from.

Be Aware

- Powermock and EclEmma do not goes well together, and whenever they comes together, the return coverage always show 0%.
This is a known issue according to internet, and here is some of the discussion on github and stackoverflow about this issue:
https://github.com/powermock/powermock/issues/422
https://github.com/jacoco/eclemma/issues/15
https://stackoverflow.com/questions/23363212/powermock-eclemma-coverage-issue

- Understanding every plugin have their own limitation is important. In this case some suggests using eCobertura. It is basically same as EclEmma, but it will not have the same problem with the coverage display. However, this plugin also has their own limitation, where once the test is run, the color marker will never goes away.
Here is the discussion on bitbucket:
https://bitbucket.org/jmhofer/ecobertura/issues/6/cant-find-a-way-removing-ecobertura
One of the solution:
https://github.com/jmhofer/eCobertura/issues/8#issuecomment-431087
Official eCobertura websit:
http://ecobertura.johoop.de/

Tips

- Normally if statement can only cover 1 branch, for complete with all branch you will need extra else if statement and else statement to make the yellow lines turn into green.

- Fix and try to hit all cases. Fix it all one case by one case. Be patient and at the end you will be able to get 100% coverage.