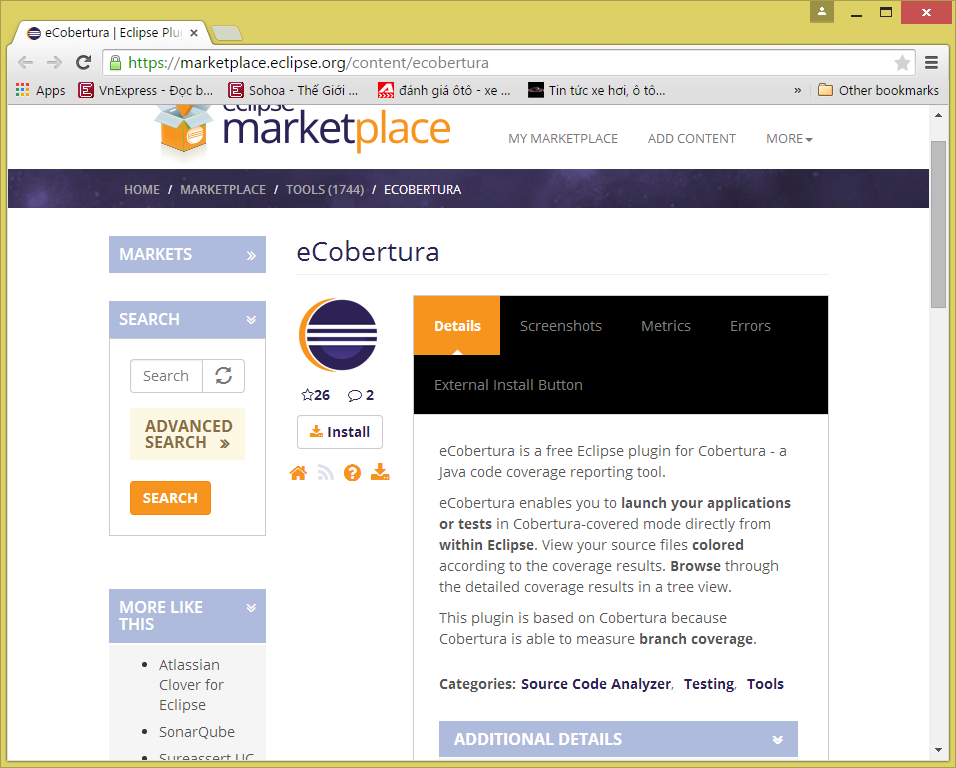
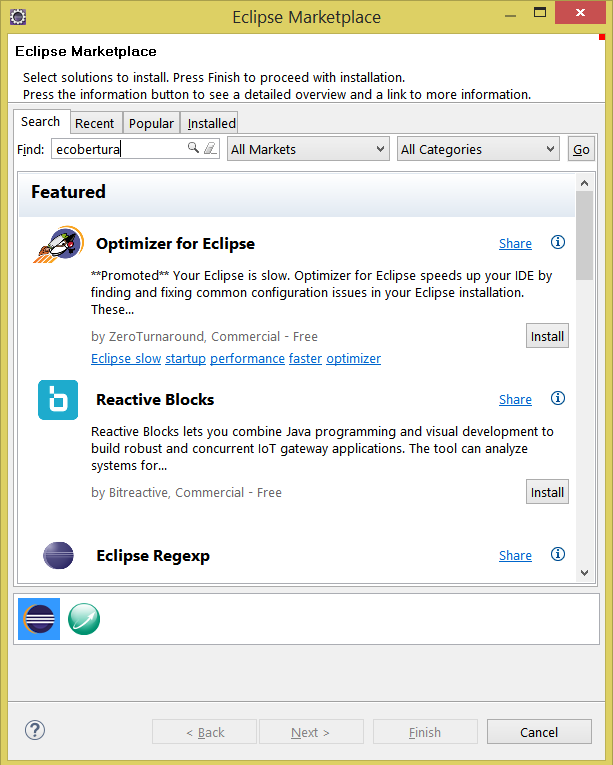
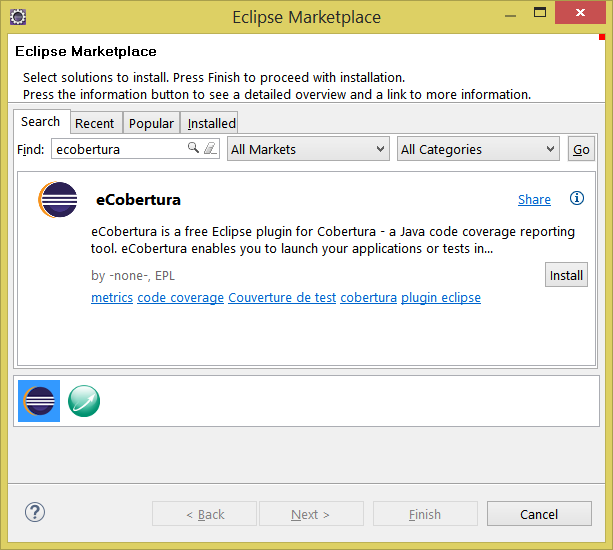
eCobertura Installing



Eclipse Help – Eclipse MarketPlace





Tạo java Project JunitTest

Tạo package cobertura

Tạo class

**package** cobertura;

**public** **class** JUnitDemo {

**public** **int** plus(**int** a, **int** b){

**return** a+b;

}

**public** **int** minus(**int** a, **int** b){

**return** a-b;

}

**public** **int** multiple(**int** a, **int** b){

**return** a\*b;

}

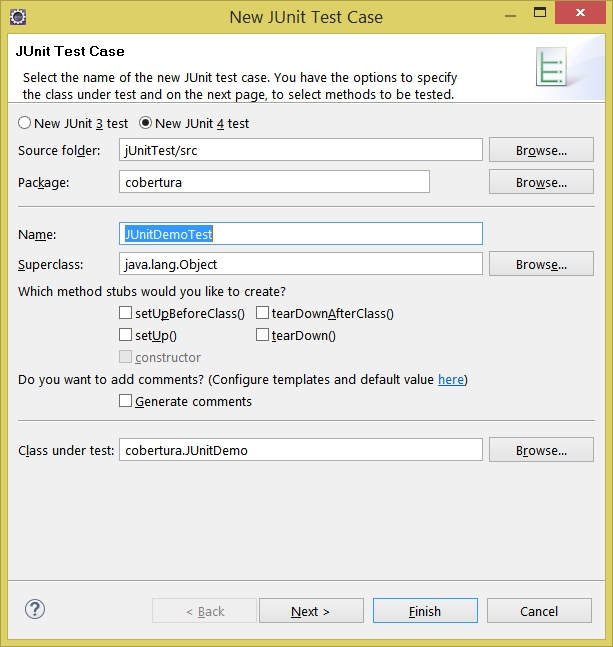
**public** **int** divide(**int** a, **int** b){

**return** a/b;

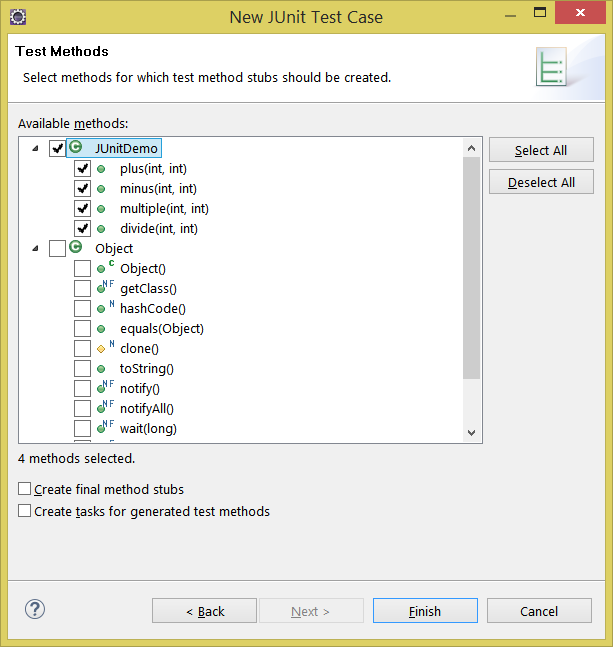
}

}

File – New – Junit Test Case



Next



Run as Junit trước rồi chạy

Coverage as… sau

Saturday, January 21, 2012

**[Cobertura and Eclipse (eCobertura)](http://kennethxu.blogspot.com/2012/01/cobertura-and-eclipse-ecobertura.html)**

[**http://kennethxu.blogspot.com/2012/01/cobertura-and-eclipse-ecobertura.html**](http://kennethxu.blogspot.com/2012/01/cobertura-and-eclipse-ecobertura.html)

**Introduction**

Cobertura is a free Java tool that calculates the percentage of code accessed by tests. It can be used to identify which parts of your Java program are lacking test coverage. It is widely used in the continuous integration process to ensure the unit test was written to exercise the application code. Developer should also run the code coverage on the development environment to make sure there is enough code coverage before commit the changes into center source repository.

**Eclipse Integration**

The most convenient way to use Cobertura is to run it right in the Eclipse IDE using eCobertura (Eclipse Plugin for Cobertura). eCobertura enables you to launch your applications or tests in Cobertura-covered mode directly from within Eclipse. View your source files colored according to the coverage results. Browse through the detailed coverage results in a tree view.

**Install eCobertura**

Use Eclipse menu item “Help->Install New Softwre…” to install the plugin from the location indicated below.

|  |  |
| --- | --- |
| http://lh4.ggpht.com/--UOAm78qBZk/TxY36Se01NI/AAAAAAAABB0/AfM4FvCUNxE/check%25255B3%25255D.gif?imgmax=800 | **Note**For detailed instruction of how to install Eclipse plugins, please refer to [General Installation Instructions section of Install Eclipse and Commonly Used Plugin](http://kennethxu.blogspot.com/2012/01/install-eclipse-and-commonly-used.html#EclipseandCommonPluginInstallation-GeneralInstallationInstructions) page. |

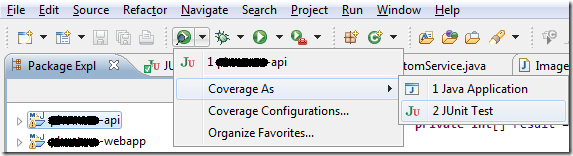
Work with URL: <http://ecobertura.johoop.de/update/>   
Features to be selected to install:

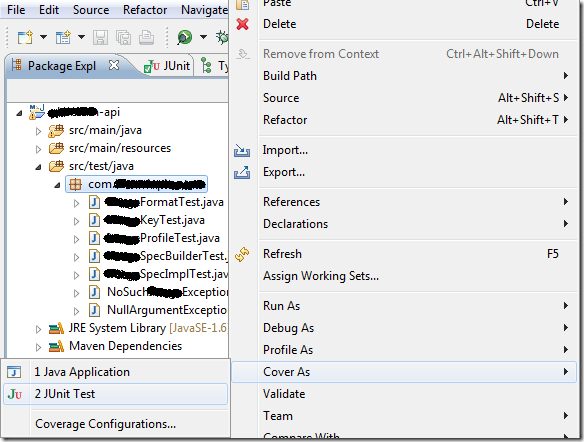
* eCobertura Code Coverage
  + eCobertura

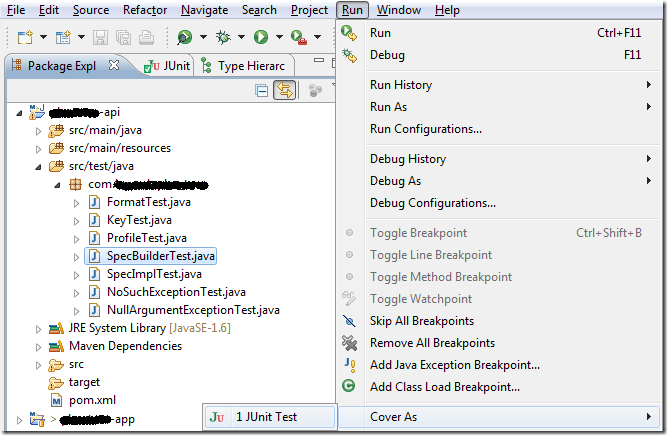
NOTE: If you are trying to install on Kepler, please see the workaround in the end of this post

**Run Coverage**

You can run your applications or tests in Cobertura-covered mode in different ways. We’ll discuss a few of them but our main focus is to show you how to run the JUnit test with eCobertura. Running as application or with a different test framework is similar but will not be discussed here.   
You start by selecting an application, a package, a test class, or a test method in the package explorer, the source editor, or anywhere that JUnit test can run. Then use toolbar button, context menu, or “Run” menu item to run the coverage.

Example 1, use toolbar button’s pull down menu to run all JUnit tests of a project in Cobertura-covered mode. Using toolbar button is the most convenient way launch Cobertura. It maintains a list of recently executed coverage run so that you can access them quickly. You’ll find it is very handy to repeat your last coverage run by directly clicking on the button itself.   


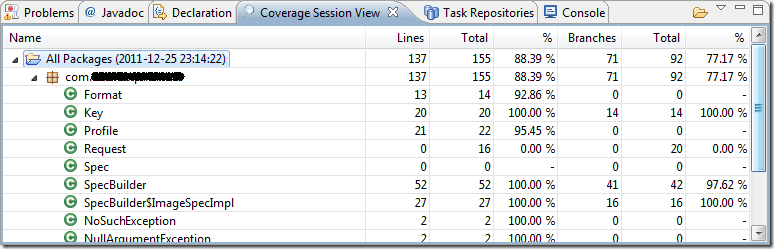
Example 2, use context menu to run all JUnit tests of a selected package in Cobertura-covered mode.   


Example 3, use “Run” menu to run all JUnit tests of a test class in Cobertura-covered mode. Using “Run” menu item maybe convenient for keyboard intensive users. The key sequence is Alt, R, C, then select a number (typically 1 or 2 depends on what you have selected).   


**Check Coverage**

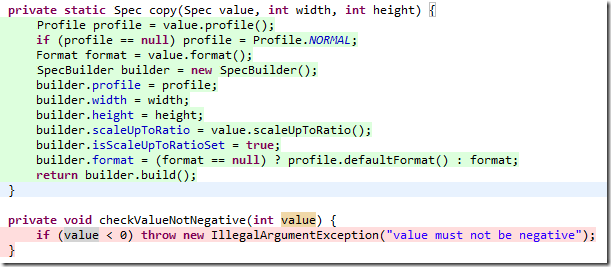
After the coverage run, the detailed coverage result is presented in “Coverage Session View” window and coloring in the source editor.

**Coverage Session View**

The coverage session view let you browse through the detailed coverage results in a tree view. It displays the code coverage at both line and branch level for each class. Results of each class are summarized in subtitle for each package and grant total for all packages. You can double click on the classes in the tree view to jump to their respective source editors.   
   
There are a few known issues with “Coverage Session View” windows, please follow the links to see a solution or workaround.

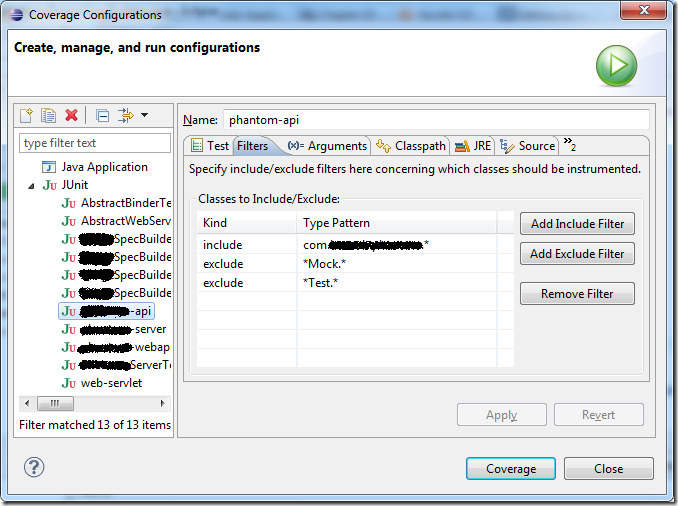
1. The view is hidden by default. You need to [show coverage session view](http://www.blogger.com/blogger.g?blogID=21777272#CoberturaandEclipse(eCobertura)-ShowCoverageSessionView) upon first coverage run.
2. The view isn’t automatically brought to front after every coverage run. Use [shortcut key binding](http://www.blogger.com/blogger.g?blogID=21777272#CoberturaandEclipse(eCobertura)-ShortcutkeybindingtoshowCoverageSessionView) can ease the inconvenience.
3. [Coverage session view stops updating](http://www.blogger.com/blogger.g?blogID=21777272#CoberturaandEclipse(eCobertura)-CoverageSessionViewstopsupdating) itself sometimes. Restart Eclipse when it happens.

**Source code coloring**

eCobertura also colors the source files according to the coverage result. Lines with green background are covered and those with pink background are not. For example,   
   
*Note: eCobertura still cannot visualize branch coverage at this moment so you may need to run the Cobertura along and generate the html report if you must know the missing branch coverage.*

**Configuration**

eCobertura often instruments much more classes than you intended to. Classes do not belong to current project, test classes and mocks add a lot of noise to the coverage result. eCobertura still has no smart way of detecting which class should be include/excluded. Hence, you’ll have to configure your coverage run to get a sensible aggregated percentage. Fortunately, you only need to do it once for reach project.

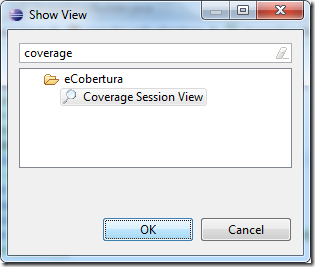
1. Use toolbar button pull down or context menu to open “Coverage Configurations” dialog.    
   
2. Select the project in the left run tree. If you don’t see your project, you can add it manually or simple go back and run the coverage once for the project and come back here.
3. Click on the Filters tab on the right and defined include/exclude filters according to your project. While exclude filters are usually always the same, you'll need one or more filters to include all the packages in your project.
4. Click on Apply button and then close the dialog.

**Known Issues**

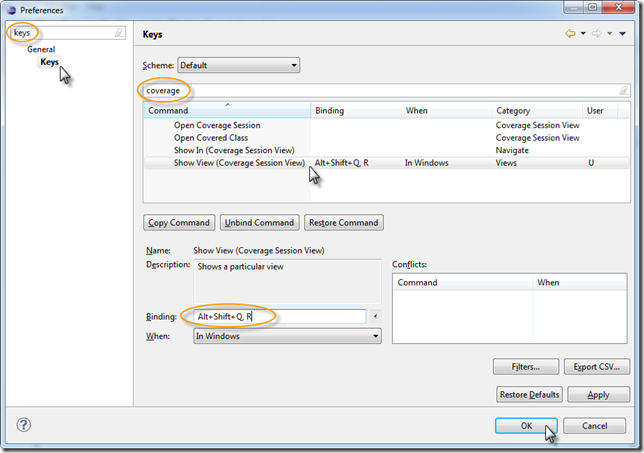
**Show Coverage Session View**

The “Coverage Session View” is hidden by default and you have to show it manually.

|  |  |
| --- | --- |
| http://lh6.ggpht.com/-POXD7lV2Spg/TxY37Bc74KI/AAAAAAAABB4/YUoEfJLvzgk/information%25255B3%25255D.gif?imgmax=800 | **Show View**For detailed instructions on how to show feature window, please refer to [Access Feature Window section of Eclipse and Common Plugin Installation](http://kennethxu.blogspot.com/2012/01/install-eclipse-and-commonly-used.html#EclipseandCommonPluginInstallation-AccessFeatureWindow). |



**Shortcut key binding to show Coverage Session View**

Version 0.9.8 of eCobertura doesn’t automatically bring the “Coverage Session View” window to front after coverage run. You’ll have to click on the Coverage Session View tab every time. Experienced user can ease the inconvenience by defining a shortcut key binding for show view command in preferences.   
[](http://lh4.ggpht.com/-5ASnnrSg6M4/Tzr_z4AeMGI/AAAAAAAABL8/5UxK4fHcqzs/s1600-h/eCobertura-show-view-key%25255B3%25255D.png)

**Coverage Session View stops updating**

It is a known bug in version 0.9.8 of eCobertura that the “Coverage Session View” may sometimes stop updating itself after coverage run. You'll notice the issue when the execution time following the “All Packages” doesn’t update any more; and the tree remains opened from run to run. When this happens, the only known fix is to restart Eclipse.

Update (9/2/2012): I have now switched to [EclEmma](http://www.eclemma.org/) *(not complete see 2nd update below)*. Don’t be fooled by the name. Emma is just the history, it no longer uses Emma. Instead, it uses JaCoCo coverage engine that can provide save level of granularity of coverage information but integrates with Eclipse much better than eCobertura. The only thing I have missed is the ability to see both line and branch coverage in the eCobertura’s coverage session view. EclEmma only allows you to see one at a time.