

Experiment No: 7

Calculate the coverage analysis of the programs given in exp-5 using Eclemma Free open-source Tool.

Eclemma is a free Java code coverage tool for Eclipse, available under the Eclipse Public License. It brings code coverage analysis directly into the Eclipse workbench:

- **Fast develop/test cycle:** Launches from within the workbench like JUnit test runs can directly be analyzed for code coverage.
- **Rich coverage analysis:** Coverage results are immediately summarized and highlighted in the Java source code editors.
- **Non-invasive:** Eclemma does not require modifying your projects or performing any other setup.

Since version 2.0 Eclemma is based on the JaCoCo code coverage library. The Eclipse integration has its focus on supporting the individual developer in an highly interactive way. For automated builds please refer to JaCoCo documentation for integrations with other tools. Originally Eclemma was inspired by and technically based on the great EMMA library developed by Vlad Roubtsov.

The update site for Eclemma is <http://update.eclemma.org/>. Eclemma is also available via the Eclipse Marketplace Client, simply search for "Eclemma".

Features

Launching

Eclemma adds a so called launch mode to the Eclipse workbench. It is called Coverage mode and works exactly like the existing Run and Debug modes. The Coverage launch mode can be activated from the Run menu or the workbench's toolbar:

Simply launch your applications or unit tests in the Coverage mode to collect coverage information. Currently the following launch types are supported:

- Local Java application
- Eclipse/RCP application
- JUnit plug-in test
- JUnit RAP test
- SWTBot test
- Scala application

Analysis

On request or after your target application has terminated code coverage information is automatically available in the Eclipse workbench:

- **Coverage overview:** The *Coverage* view lists coverage summaries for your Java projects, allowing drill-down to method level.
- **Source highlighting:** The result of a coverage session is also directly visible in the Java source editors. A customizable color code highlights fully, partly and not covered lines. This works for your own source code as well as for source attached to instrumented external libraries.

Additional features support analysis for your test coverage:

- **Different counters:** Select whether instructions, branches, lines, methods, types or cyclomatic complexity should be summarized.
- **Multiple coverage sessions:** Switching between coverage data from multiple sessions is possible.
- **Merge Sessions:** If multiple different test runs should be considered for analysis coverage sessions can easily be merged.

Import/Export

While Eclemma is primarily designed for test runs and analysis within the Eclipse workbench, it provides some import/export features.

- **Execution data import:** A wizard allows to import JaCoCo *.exec execution data files from external launches.
- **Coverage report export:** Coverage data can be exported in HTML, XML or CSV format or as JaCoCo execution data files (*.exec).

The screenshot displays the Eclipse IDE interface during a test run. The top editor shows the source code of `CursorableLinkedList.java`. The left sidebar shows the test run progress, indicating it finished after 34,293 seconds with 1,300,913,009 runs, 0 errors, and 0 failures. The bottom-right pane shows the Coverage report for `TestAllPackages (31.10.2006 15:04:14)`.

Element	Coverage	CoveredLines	TotalLines
java - commons-collections	79,5 %	10607	13328
org.apache.commons.collections	74,1 %	3842	5183
ArrayStack.java	86,5 %	32	37
BagUtils.java	86,7 %	13	15
BeanMap.java	72,4 %	155	214
BinaryHeap.java	87,6 %	127	145
BoundedFifoBuffer.java	93,2 %	62	66
BufferOverflowException.java	55,6 %	5	9
BufferUnderflowException.java	88,9 %	8	9
BufferUtils.java	30,8 %	4	13
ClosestUtils.java	93,9 %	31	33
CollectionUtils.java	92,4 %	293	317
ComparatorUtils.java	6,6 %	3	35
CursorableLinkedList.java	85,4 %	444	520