Derby Code Coverage Using EMMA

Date: March 07, 2005

Author: Ramandeep Kaur
ramank@yngvi.org

Table of Contents

1 Introduction

Several open source code coverage tools were evaluated as candidates for Derby code coverage. These included EMMA, Clover, and Jcoverage. After several iterations of testing, it was found that EMMA would provide most value to Derby as a code coverage tool. This document is intended to provide information to the Derby community re the use of EMMA as a code coverage tool.

The document initially provides a brief introduction to Code Coverage and EMMA, and then provides the information on installing and setting up EMMA to run with Derby**.**

1.1 Notes

- This document contains references to urls. The urls may change anytime. If there is any problem with any url, post it to the Derby mailing list (derby-dev@db.apache.org).

- All examples or commands are given in windows conventions. For unix, modify the examples or commands as per unix conventions.

2 Code Coverage

Code coverage is the process of finding out how much of the code base is being tested by executing the test cases. This helps to identify untested portions of the code and to improve the quality of test cases. Making code coverage part of the software development process improves software quality.

Code coverage tools can be used at any time during the software life cycle. In development, code coverage can be used for unit testing to find out how much of the code is covered by unit test. In functional testing, code coverage can be used to determine test suites effectiveness.

Code coverage can be measured in various ways:

- **Line coverage** measures if each individual line/statement is being executed during test execution.
- **Basic Block coverage** measures if each sequence of non-branching statements is being executed during test execution.
- **Method coverage** measures if each method within a class is being invoked during test execution.
- **Class coverage** measures if each class is being invoked during test execution.

3 EMMA Overview

EMMA is an open-source tool kit for measuring and reporting code coverage for java applications. EMMA measures the code coverage from various ways such as class, method, line, and basic block coverage at package, source file, class, and method levels. EMMA's fundamental units of coverage are basic blocks and all other types of coverages are derived from basic block coverage.

EMMA provides a command line interface and ANT tasks. The examples in this document will primarily use EMMA command line interface.

4 Setting up EMMA and Derby

4.1 System Requirements

The system requirements for running EMMA are as follows:

JVM: EMMA works with any JVM version. EMMA tools and runtime performance is better if used with JVM 1.3 and later versions.

ANT: EMMA ANT tasks works with Apache ANT 1.4.1 and later versions. ANT is only required for EMMA ant tasks.

Operating System: EMMA is a java tool and it works on any operating system that supports JVM.

4.2 Installing EMMA

Install EMMA as follows:

1. Go to URL http://emma.sourceforge.net/downloads.html

2. Download "emma-major.minor.build-lib.zip**"** from section "release distribution**".**

3. Unzip the downloaded files to any directory. This directory will be referred to as the ${emma.dir} directory in the rest of this document.

4. In directory ${emma.dir}, there will be 2 jar files - emma.jar, emma_ant.jar. To run EMMA command line interface, you only need to add emma.jar to the JVM classpath. You can add emma.jar to the JVM classpath by either one of the methods as follows:

- Using JVM option -cp while calling java command.

  Example: *java -cp ${emma.dir}\emma.jar emma* ***emma_command command_options***

- Setting up CLASSPATH to emma.jar from command line before calling emma command line tools.

  Example: *set CLASSPATH=${emma.dir}\emma.jar;%CLASSPATH%*

- Adding emma.jar as an installed JRE/JVM extension. To add emma.jar as an installed JVM/JRE extension, simply add emma.jar to subdirectory lib\ext of the directory where you have installed JVM.

  Example: *${java_home}\jre\lib\ext\emma.jar*

where:

${emma.dir} is the directory where you have installed EMMA.
${java.home} is the directory where you have installed JVM.

4.3 Setting up Derby Test Environment

To run EMMA to gather code coverage for Derby code, Derby test cases/suites need to be executed for which you will need to set up Derby test environment by following instructions in readme file:
https://svn.apache.org/viewcvs.cgi/*checkout*/incubator/derby/code/trunk/java/testing/README.htm

5 Running EMMA with Derby

To run EMMA code coverage during Derby testing, do the following:

1. Check if emma.jar is added to your JVM classpath as described in step 4 of section 4.2 of this document.

2. Locate the directory where you have all the derby jar files. This directory will be referred to as the ${derby.jar.dir} directory in the rest of this document.

3. The first step in EMMA code coverage is instrumentation of Derby jar files. Instrumentation is adding reporting code into an executable. To instrument Derby jar files (derby.jar, derbynet.jar, derbytools.jar, derbyclient.jar), run EMMA command as follows:

   > java emma instr -m overwrite -ip derby.jar,derbynet.jar,derbytools.jar,derbyclient.jar
   OR
   > java -cp ${emma.dir}\emma.jar emma instr -m overwrite -ip
   derby.jar,derbynet.jar,derbytools.jar,derbyclient.jar

   The output of the above command will look similar to the following:

```
EMMA: [EMMA v2.0, build 4217 (2004/07/17 12:19:29)]
EMMA: instrumentation path:
EMMA: {
EMMA:   C:\derbyjars\derby.jar
EMMA:   C:\derbyjars\derbytools.jar
EMMA:   C:\derbyjars\derbynet.jar
EMMA:   C:\derbyjars\derbyclient.jar
EMMA: }
EMMA: instrumentation output mode: overwrite
EMMA: metadata output file: C:\derbyjars\coverage.em
EMMA: metadata output merge mode: true
EMMA: processing archive path entry [C:\derbyjars\derby.jar] ...
EMMA: processing archive path entry [C:\derbyjars\derbytools.jar] ...
EMMA: processing archive path entry [C:\derbyjars\derbynet.jar] ...
EMMA: processing archive path entry [C:\derbyjars\derbyclient.jar] ...
EMMA: instrumentation path processed in 5390 ms
EMMA: [1178 class(es) instrumented, 447 resource(s) copied]
EMMA: metadata contains 1159 entries
EMMA: metadata merged into [C:\derbyjars\coverage.em] {in 516 ms}
```

The above command will instrument all the class files within jar archives and will overwrite original jar files with their instrumented versions. Note that after running the above command, the size of all jar files will increase by more than 100%. The above command will also create metadata file with default name "coverage.es" in the current directory.

4. Second step in EMMA code coverage is to execute Derby test cases/test suites. Execute Derby test cases/test suites as follows:

   - Set up classpath as described in testing readme file (https://svn.apache.org/viewcvs.cgi/*checkout*/incubator/derby/code/trunk/java/testing/README.htm) to include all derby jar files and other required jar files in JVM classpath. Make sure emma.jar is still in your classpath as it is needed during execution of each class that is been instrumented by EMMA.

   - Create a directory with any name (e.g.: testing) and cd to it. The purpose of creating this directory is to keep test results for multiple suites in one single directory.
     > cd testing

   - Run either of the following command:

     >java -Djvmflags="-Demma.verbosity.level=silent" -Dverbose=true
     org.apache.derbyTesting.functionTests.harness.RunSuite <suitename>

     OR

     >java -Djvmflags="-Demma.verbosity.level=silent" -Dverbose=true
     org.apache.derbyTesting.functionTests.harness.RunTest <testcasename>

Examples:

>java -Djvmflags="-Demma.verbosity.level=silent" -Dverbose=true
org.apache.derbyTesting.functionTests.harness.RunSuite derbyall

OR

>java -Djvmflags="-Demma.verbosity.level=silent" -Dverbose=true
org.apache.derbyTesting.functionTests.harness.RunTest lang/supersimple.sql

During the execution of test cases/test suites, by default a coverage data file with name "coverage.ec" will get created in directory where you ran the above command. The coverage data file contains runtime coverage profile for executables that were instrumented. The name of the coverage data file and directory in which it gets created can be changed by setting emma property *coverage.out.file*. Check out EMMA reference guide for more information about this property.

5. Third step in EMMA code coverage is to create code coverage report. Code coverage report is created from coverage metadata file created during instrumentation of derby jar files, from coverage data files created during execution of one or more of test cases/test suites, and from derby source files. Derby source code is only needed if you want code coverage reports to be able to link to Derby source code to highlight tested and untested source code. Create coverage report as follows:

   - Run the following command:

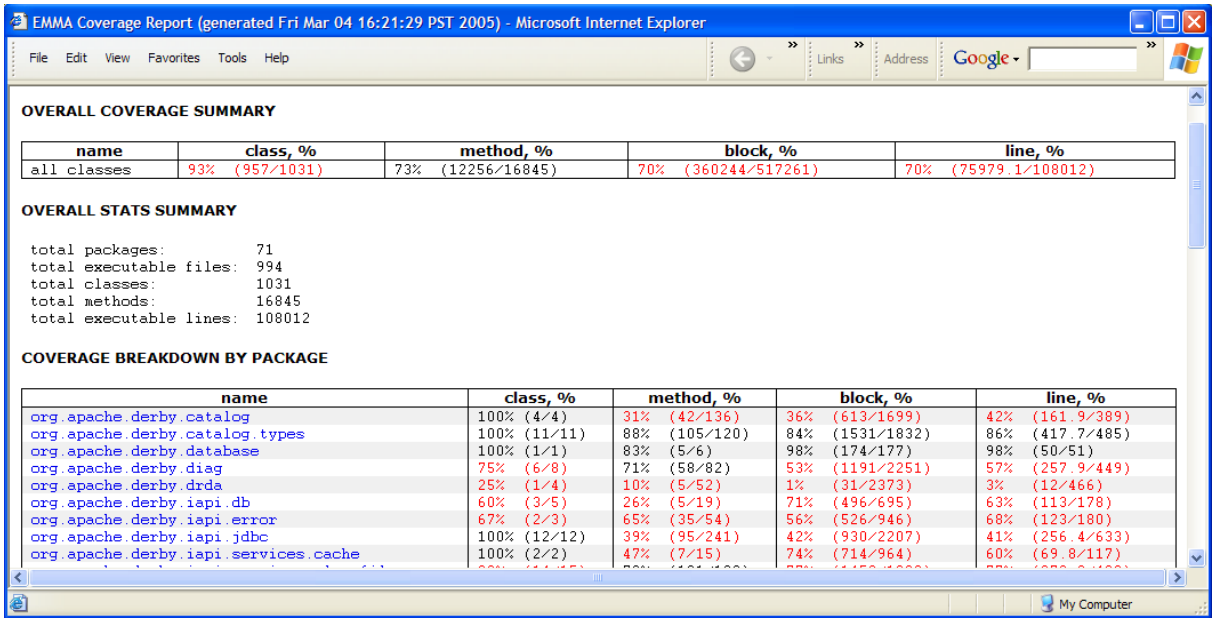
   >java emma report -r html -in *${path}*/coverage.es,*${path}*/testing/coverage.ec -sourcepath *${sourcecodedir}* -Dreport.sort=+name,+class,+method,+block,+line -Dreport.out.file=*${coverage_report_dir}*

   Where
   ${path} is the location of coverage files
   ${sourcecodedir} is the location of source code
   ${coverage_report_dir} is the location where report files will be created.

   Notes:
   - If you ran multiple test suites or cases from same directory or specified the same name and location for coverage data file even though you ran multiple test suites or cases from different directories, you will have one coverage data file.

   - If you ran multiple test suites or cases from different directories and you did not specify same name and location for coverage data file, by default, you will have coverage data file in each directory in which you ran test suites or cases. In this case, you can specify multiple coverage data files separated by comma while calling emma report command.

The above command will create a report file with default name "index.html" in directory *${coverage_report_dir}*. The report file "index.html" will be the starting point to check code coverage results. The main report (index.html) shows the overall and package level code coverage. The sample report is as follows:



EMMA Coverage Report (generated Fri Mar 04 16:21:29 PST 2005) - Microsoft Internet Explorer

File Edit View Favorites Tools Help

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	93% (957/1031)	73% (12256/16845)	70% (360244/517261)	70% (75979.1/108012)

OVERALL STATS SUMMARY

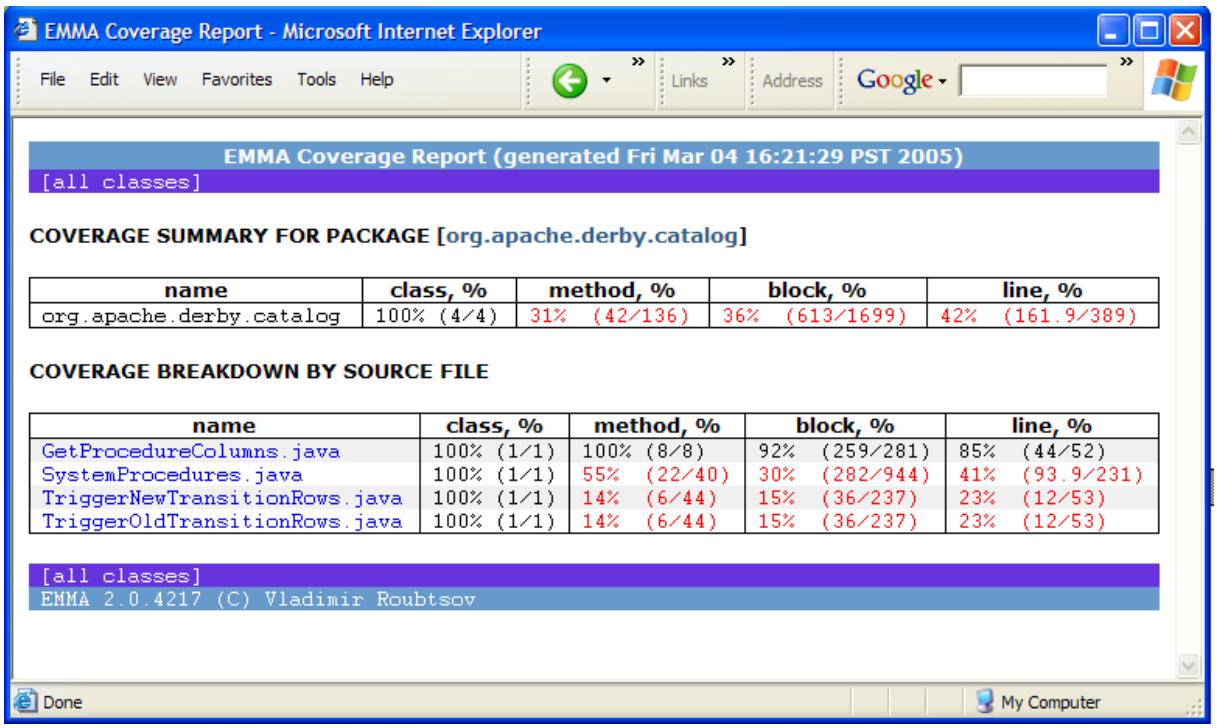
total packages:	71
total executable files:	994
total classes:	1031
total methods:	16845
total executable lines:	108012

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
org.apache.derby.catalog	100% (4/4)	31% (42/136)	36% (613/1699)	42% (161.9/389)
org.apache.derby.catalog.types	100% (11/11)	88% (105/120)	84% (1531/1832)	86% (417.7/485)
org.apache.derby.database	100% (1/1)	83% (5/6)	98% (174/177)	98% (50/51)
org.apache.derby.diag	75% (6/8)	71% (58/82)	53% (1191/2251)	57% (257.9/449)
org.apache.derby.drda	25% (1/4)	10% (5/52)	1% (31/2373)	3% (12/466)
org.apache.derby.iapi.db	60% (3/5)	26% (5/19)	71% (496/695)	63% (113/178)
org.apache.derby.iapi.error	67% (2/3)	65% (35/54)	56% (526/946)	68% (123/180)
org.apache.derby.iapi.jdbc	100% (12/12)	39% (95/241)	42% (930/2207)	41% (256.4/633)
org.apache.derby.iapi.services.cache	100% (2/2)	47% (7/15)	74% (714/964)	60% (69.8/117)

My Computer

Clicking on package name in report shown above provides code coverage summary for all the classes in that package. The sample report is as follows:



EMMA Coverage Report - Microsoft Internet Explorer

File Edit View Favorites Tools Help

EMMA Coverage Report (generated Fri Mar 04 16:21:29 PST 2005)

[all classes]

COVERAGE SUMMARY FOR PACKAGE [org.apache.derby.catalog]

name	class, %	method, %	block, %	line, %
org.apache.derby.catalog	100% (4/4)	31% (42/136)	36% (613/1699)	42% (161.9/389)

COVERAGE BREAKDOWN BY SOURCE FILE

name	class, %	method, %	block, %	line, %
GetProcedureColumns.java	100% (1/1)	100% (8/8)	92% (259/281)	85% (44/52)
SystemProcedures.java	100% (1/1)	55% (22/40)	30% (282/944)	41% (93.9/231)
TriggerNewTransitionRows.java	100% (1/1)	14% (6/44)	15% (36/237)	23% (12/53)
TriggerOldTransitionRows.java	100% (1/1)	14% (6/44)	15% (36/237)	23% (12/53)

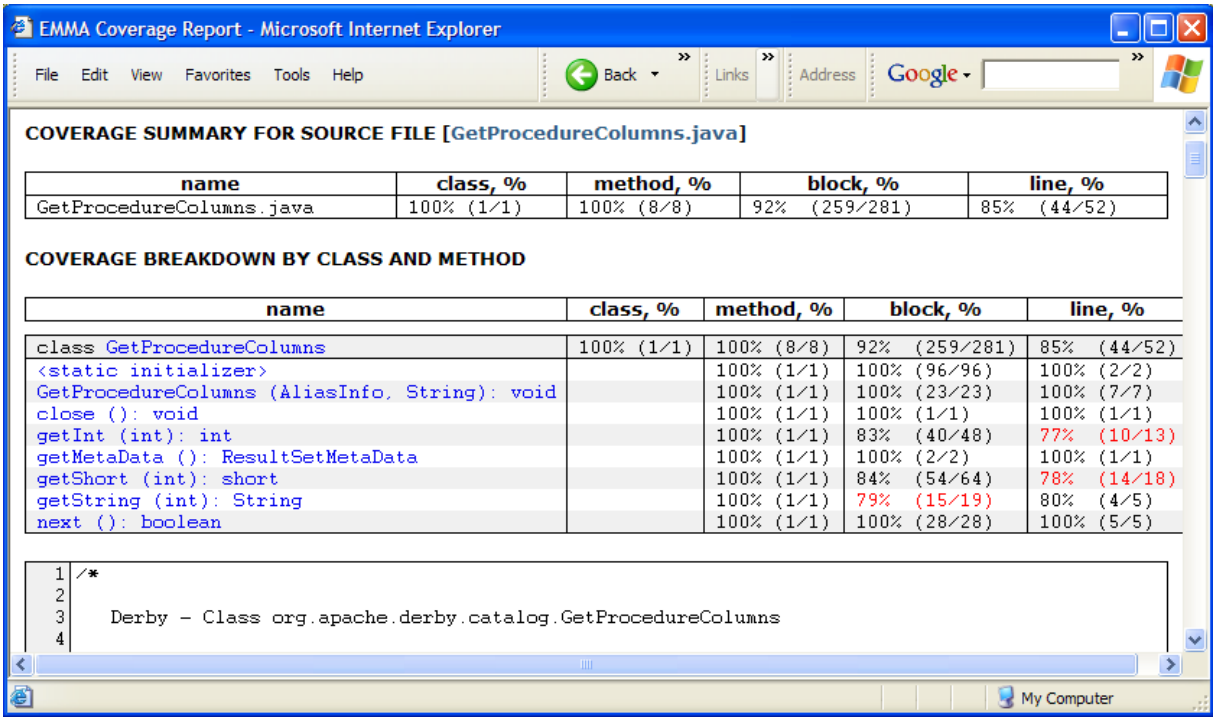
[all classes]

EMMA 2.0.4217 (C) Vladimir Roubtsov

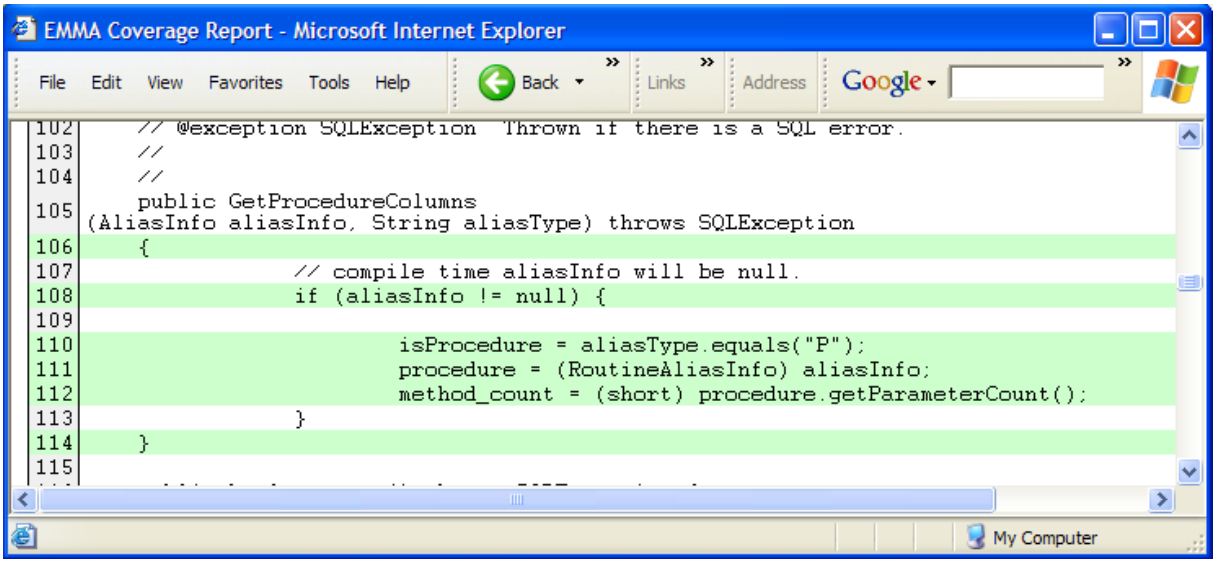
Done

My Computer

Clicking on class name in report shown above provides code coverage summary for all the methods in that class. The sample report is as follows:



Another sample report for source code highlighting is as follows:



6 Tips/Hints

6.1 emma.verbosity.level

As Derby test cases are based on the comparison of expected output with actual output from a test case, it is very important to set the jvmflag emma.verbosity.level to silent while calling derby tests with EMMA. If the emma.verbosity.level is not set to silent, EMMA prints out some EMMA related information while running test cases which makes the actual output of a test case different from expected output. Difference in expected and actual output results in test case marked as fail, when in reality, the test case may have passed.

6.2 EMMA coverage for selected class files

At various times, developers may need to get the code coverge data only for few classes. This could be needed while doing unit testing. Moreover, when developers are doing unit testing, they may not be interested in looking at the reports which gives coverage for all the packages in an application. As developers may be running only few test cases, most of the packages will show 0% coverage as they are not covered by the tests that they have executed. To get code coverage for only selected files, developers should instrument only those class files that they expect to be invoked while running the test cases. Developers can instrument a set of class files by using filters. The example is as follows:

If developer wants to get code coverage for class files in org.apache.derby.tools package only, developer should give the following command:

>java emma instr -m overwrite -ip derbytools.jar -ix +org.apache.derby.tools.*

The above command will only instrument class files in package org.apache.derby.tools and will update the derbytools.jar with instrumented versions of class files in package org.apache.derby.tools. If you do “jar -tvf derbytools.jar”, you will notice that the size of class files in package org.apache.derby.tools have increased in size.

Developers can now run test cases and run report as explained in step 4 and 5 of section 5 of this document. During execution of test cases, EMMA will be able to collect code coverage information for only those files that were instrumented.

6.3 EMMA commands documentation

EMMA commands (instr, report, merge, emmarun) have additional attributes and parameters which can be used as per need of individual developers. Check out EMMA commands documentation at http://emma.sourceforge.net/reference_single/reference.html.

6.4 EMMA Frequently Asked Questions

EMMA web page has a very good listing of frequently asked questions. Check out EMMA FAQ at http://emma.sourceforge.net/faq.html

6.5 EMMA Forum

EMMA has forum available for users to post any questions/inquires. The response to the inquiries is very quick. Check out EMMA forum at
http://sourceforge.net/forum/forum.php?forum_id=373865

7 Summary

This document has given a brief overview of EMMA. Instructions are given to run EMMA for Derby code coverage. As EMMA command line interface provides a number of input attributes and parameters, developers may use them as per their need. Developers are encouraged to read EMMA user and reference guide to get more information on additional attributes and parameters to EMMA commands.

8 References

[1] Vlad Roubtsov, EMMA documentation
http://emma.sourceforge.net/index.html