**Context:**

Most of the teams are measuring code coverage for unit tests with the plugins available in integration with Jenkins as part of CI.

Solution provided below is to measure code coverage for Functional tests for Services.

This is a proposal using EMMA and ANT script for achieving reliable quality through identifying untested areas of application for quick benefits.

**Who all can Use:**

Services Teams with Automation frameworks, Manual functional Tests can use to measure code coverage for respective components (jar's, war's and ear's).

**Gist:**

Code Coverage can be generated for required builds by following execution process:

            i. Integrate Instrumentation with Build

            ii. Deploy instrumented application

            iii. Collect coverage data during/post functional testing

            iv. Final Report Generation

Apart from common coverage tool features, POC is capable of:

* Collecting coverage from a JVM running instrumented classes via TCP socket
* Analyze coverage for N number of binaries
* Link source code to reports
* Extendable to merge coverage data for consolidated reports
* Extendable for CI

**Pre-Requisites**

•          Ant installed: <http://ant.apache.org/>

•          Emma installed: <http://emma.sourceforge.net/>

**Installation**:

•          Create Emma directory and copy the contents (lib, build.xml).

Ex:  Emma/build.xml

•          Under Emma/ dir, execute “ant -f build.xml”

•          Restart Jboss

•          Run Tests

•          Under Emma/ directory, execute “ant –f build.xml report”

•          Copy coverage.html and \_files to view the report.

**Explanation of Script:**

Step 1: Set class path to include Emma and ANT binaries

Step 2: Settings to use ANT tasks

Step 3: Initialize directories

Step 4: Instrument the .jar's/.classes/.war's/.ear's

Step 5: Collect the data from JVM Port runtime

Step 6: Generate Coverage Report

**Other Details:**

**Command line support for CI Integration:**

There are 3 parameters emma.dir, deploy.dir, jboss.bin.dir

that can be passed as command line arguments. Comment this parameters in build.xml in case passing as command line arguments.

**For Instrumenting:**

ant –f build.xml -Demma.dir=<Emma directory> -Ddeploy.dir=<AppServer deploy path>

-Djboss.bin.dir=<AppServer bin directory>

**For report Generation:**

ant -f build.xml -Demma.dir=. report

**Script snippets for instrumenting binaries and coverage collection:**

<emma verbosity="verbose" enabled="${emma.enabled}">

<instr mode="overwrite"

merge="true"

destdir="${archive.dir}"

metadatafile="${coveragemetadata}/${archive.file}-metadata.em">

<!-- Explicitly include/exclude if anything specific based on the requirement -->

<instrpath>

<fileset dir="${archive.dir}" includes="\*\*/\*.jar"/>

</instrpath>

</instr>

</emma>

<emma>

<ctl connect="${jvm.host.port}">

<command name="coverage.get" args="${coverage.dir}/${coverage.file},true"/>

<command name="coverage.reset"/>

</ctl>

</emma>

**Tool Evaluation:**

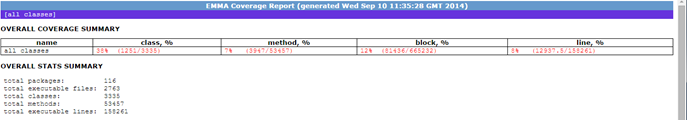
|  |  |
| --- | --- |
| **Parameters** | **Comments** |
| **Coverage Levels** |  |
| Package |  |
| Class |  |
| Method |  |
| Block |  |
| Line |  |
| File |  |
| **Report Clarity** |  |
| Hit Count | No. of times the statement/code block is hit |
| Source Linking | Ability to link coverage Report and Source Code |
| **Exclusion Management** |  |
| Source File Exclusion |  |
| Exclusion Patterns Support | Ability to exclude certain areas of code from reporting |
| **Advanced Reporting** |  |
| HTML Reports |  |
| Incremental Reporting |  |
| Base lining & Versioning | Only for newly added code |
| **Platform Support** |  |
| Command Line |  |
| **Licensing** |  |
| Well Documented |  |
| Open Source |  |
| Community based Support |  |
| **Technical Aspects** |  |
| Source Level Instrumentation |  |
| Merging |  |
| Performance |  |

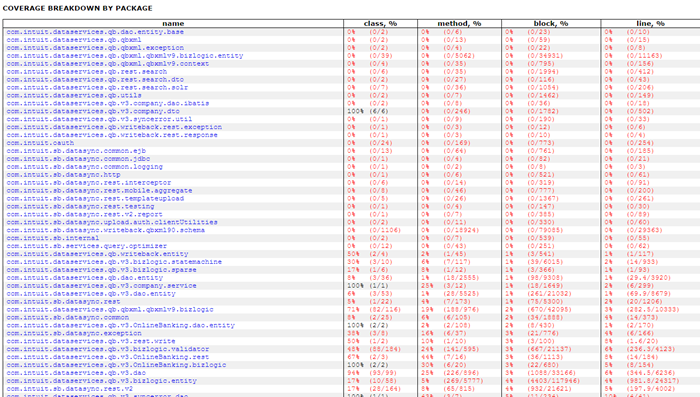
**Tool References:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Technology** | **Tools** |
| 1 | Java | Emma, Cobertura |
| 2 | .NET | NCover, PartCover |
| 3 | C/C++ | BullsEye, CoverageMeter |

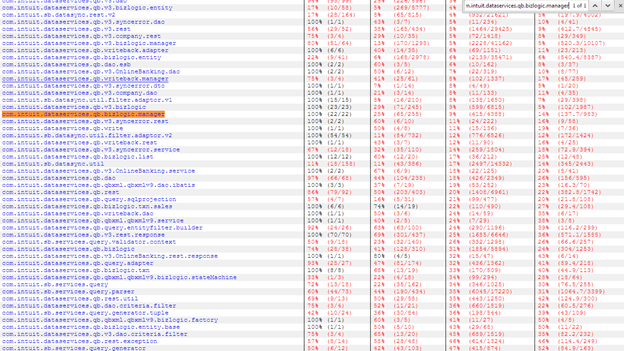
**Sample Reports:**

Overall Coverage

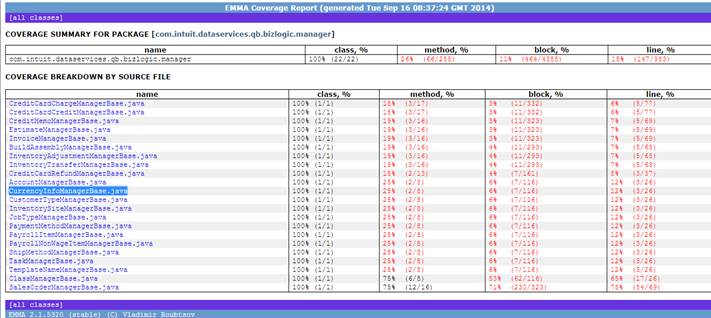




Package Level



Class Level



Method Level

