
基于 Emma 的独立测试用例代码覆盖率统计

作者：洗志超，新炬网络技术专家

简介

EMMA 是一个开源、面向 Java 程序测试覆盖率收集和报告工具。它通过对编译后的 Java 字节码文件进行插装，在测试执行过程中收集覆盖率信息，并通过支持多种报表格式对覆盖率结果进行展示。

优点

EMMA 收集的数据包括类覆盖率、方法覆盖率、块覆盖率和行覆盖率，这些数据以包为单位进行组织。

大多数功能测试中，测试人员一般不能直接得到被测源代码，源代码也不是测试人员关心的重点。在具体的测试过程中，功能测试人员一般以一个有意义的功能模块作为测试关心的重点，而能够反映一定功能含义的类和方法的覆盖率在功能测试中更有价值。因此，在功能测试中，类覆盖率和方法覆盖率是测试人员关心的重点，行和块覆盖率则作为测试的参考。

测试覆盖率报告中包含了两个方面的内容，测试覆盖的部分和未被测试覆盖的部分。尽管百分之百的测试覆盖率不能代表被测对象完全没有问题，但是测试覆盖的部分以及覆盖比率可以增加测试者对测试工作的信心，指导测试执行以及测试的方向。另一方面，当测试用例执行出现异常时，针对每个测试用例的测试报告还可以提供可疑代码的范围，为代码纠错提供帮助。

测试覆盖率报告中未覆盖的部分也同样有价值：

- ✓ 表明测试可能不完整，有些功能、代码没有被测试覆盖到。
- ✓ 为测试用例的设计提供指导建议。在覆盖率报告的指导下，测试人员有目的地与开发人员进行讨论，确定未覆盖部分是测试的空白还是不需要测试的部分。
- ✓ 帮助开发人员发现无用代码，为修改，完善代码提供依据。

在使用 EMMA 获得测试覆盖率过程中，类、方法等覆盖的百分比报告，可以方便测试人员更好的评估测试。测试人员通过对照覆盖率报告与测试用例设计文档，需求文档可以迅速找到测试的不足。通过与开发人员进行讨论，可以更好的评估测试力度，并指导进一步的测试。因此在功能测试中引入覆盖率信息，能够完善测试结果报告，确保测试质量和力度，保证测试按质、按量地完成。

环境准备

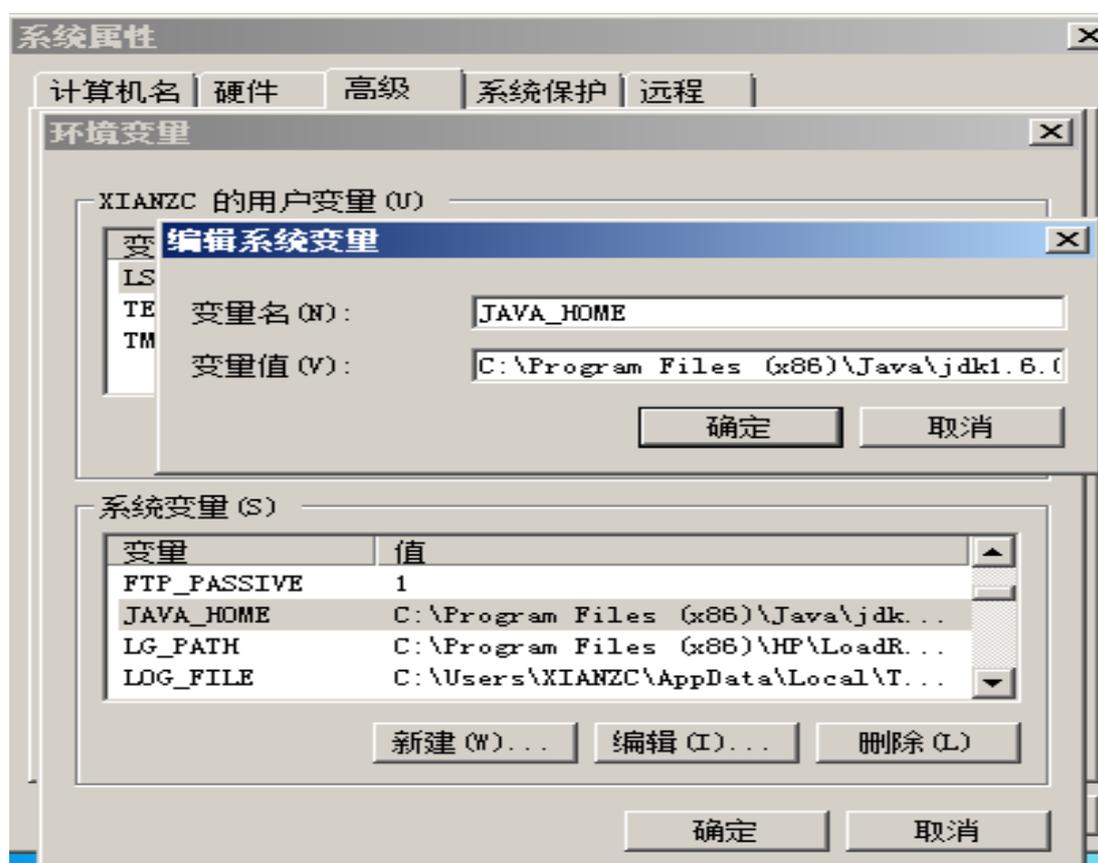
操作系统

EMMA 覆盖率测试工具是由纯 Java 开发，可运行在任何支持 JRE1.6+版本的系统内，包括 Windows、Linux。

JDK 环境

Emma 覆盖率测试工具运行时需使用 JRE，配置环境变量：

- ✓ JAVA_HOME: C:\Program Files (x86)\Java\jdk1.6.0_38
- ✓ Path: C:\Program Files (x86)\Java\jdk1.6.0_38\bin;
- ✓ CLASSPATH: .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;



配置好后，命令窗口中分别输入 `java -version` 和 `javac`，出现如下图的信息，即为配置成功。

```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\XIANZC>java -version
java version "1.6.0_38-ea"
Java(TM) SE Runtime Environment (build 1.6.0_38-ea-b04)
Java HotSpot(TM) Client VM (build 20.13-b02, mixed mode, sharing)

C:\Users\XIANZC>javac
用法: javac <选项> <源文件>
其中, 可能的选项包括:
-g 生成所有调试信息
-g:none 不生成任何调试信息
-g:<lines,vars,source> 只生成某些调试信息
-nowarn 不生成任何警告
-verbose 输出有关编译器正在执行的操作的消息
-deprecation 输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径> 指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 覆盖引导类文件的位置
-extdirs <目录> 覆盖安装的扩展目录的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-proc:<none,only> 控制是否执行注释处理和/或编译。
-processor <class1>[,<class2>,<class3>... ] 要运行的注释处理程序的名称; 绕过默认

```

示例 WEB 应用部署

使用 Tomcat 部署开源 JSPXCMS 的演示站点, 包含了新闻、图集、视频、下载、在线留言、人员招聘、BBS 等模块。jspxcms 是一个开源的、基于 Java 的内容管理系统 (CMS), 依托 Java 的开源精神和强大的企业级功能, 致力于实现 Java CMS 网站制作、维护。

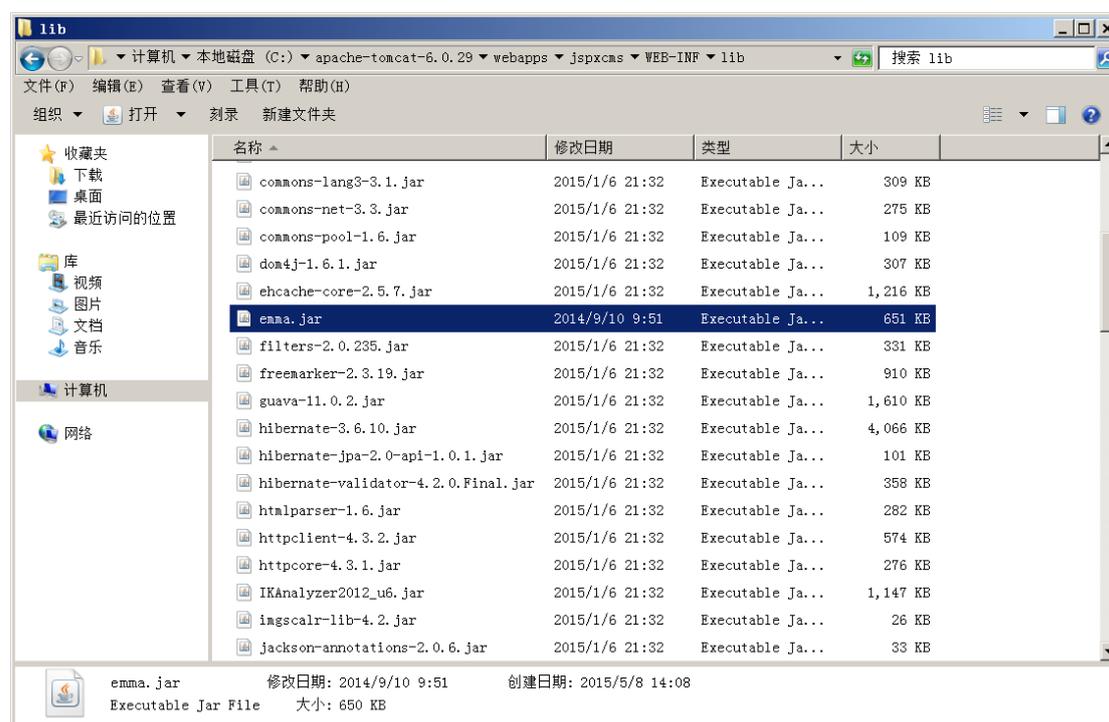


EMMA 工具配置

在执行插装操作之前，首先应该扩展 Java 虚拟机，即将 emma.jar 放到被测组件运行使用的 JRE 目录下面作为 JRE 的扩展，以便 EMMA 能够被调用。emma.jar 包含了 EMMA 核心功能模块的实现和 EMMA 运行时所需的类文件，这些文件是使用 EMMA 所必需的。

- ✓ 把 emma.jar 放到 C:\Program Files (x86)\Java\jre6\lib\ext 目录下；
- ✓ 把 emma.jar 放到示例工程项目 lib 目录下；

Emma.jar 放置路径为：C:\apache-tomcat-6.0.29\webapps\jspxcms\WEB-INF\lib



EMMA 在测试执行中的应用

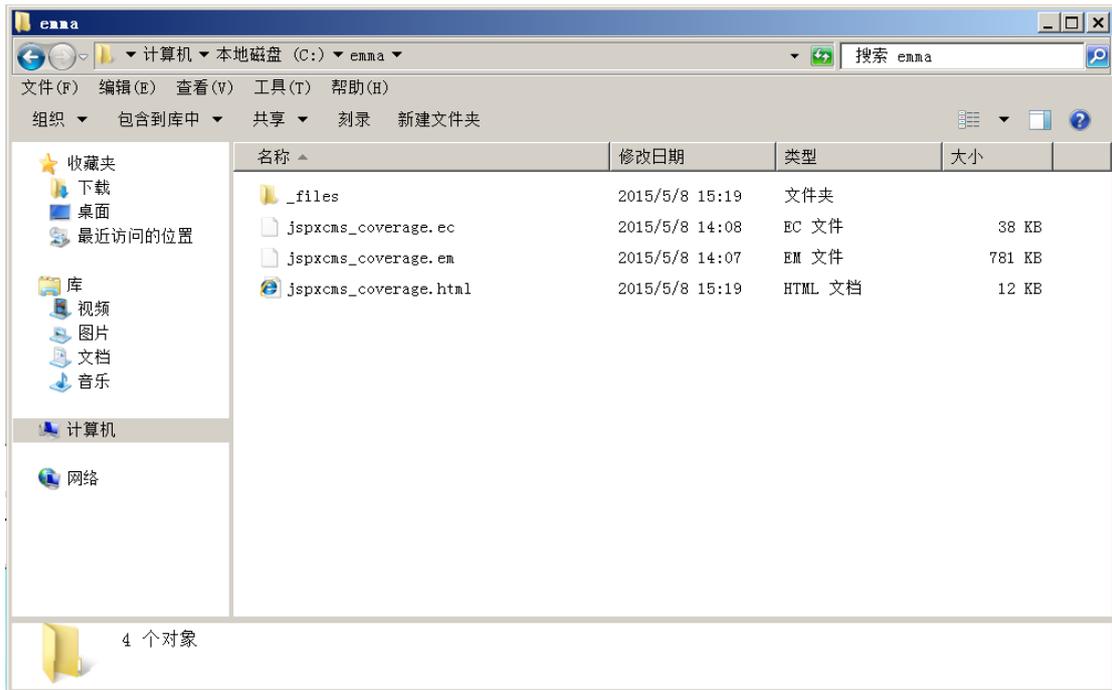
EMMA 使用基本步骤是：插桩（instr），运行，收集（ctl），报告（report）

EMMA 生成文件目录：

插桩生成的二进制文件：jspxcms_coverage.em

收集覆盖率信息生成的二进制文件：jspxcms_coverage.ec

生成覆盖率报告的文件：jspxcms_coverage.html

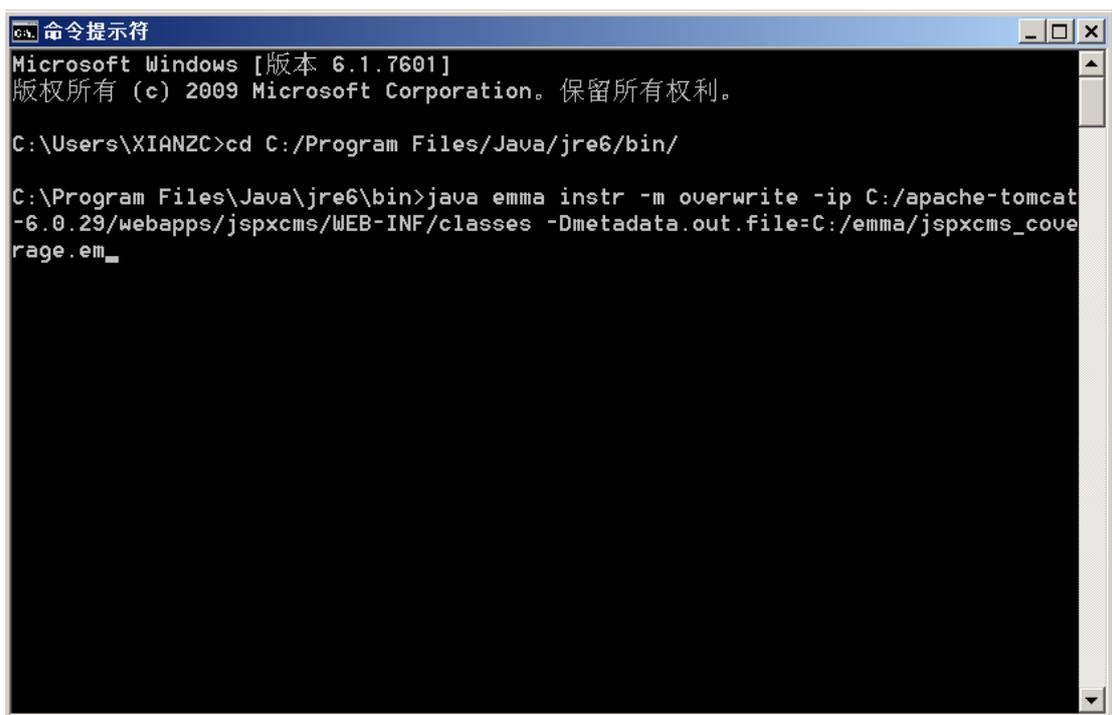


插桩

EMMA 中提供了 “instr” 命令完成插装操作。插装操作可以面向 JAR 包、WAR 包、类文件和目录。

插桩前，启动应用服务。

在命令窗口中 cd C:/Program Files/Java/jre6/bin/（或直接配置 emma.jar 的 classpath 环境变量），插桩命令：



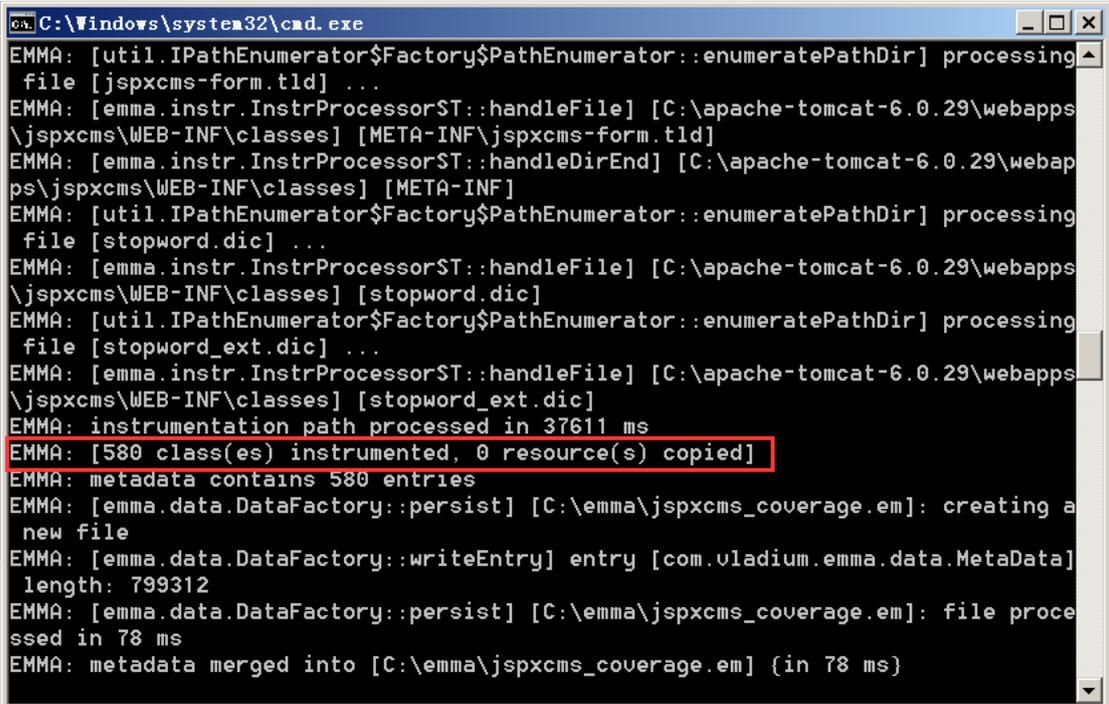
参数 “m” 代表插装后文件输出的模式。有三个值可供选择：“copy”，“overwrite” 和 “fullcopy”。其中，“copy” 和 “fullcopy” 这两种模式将会改变插装文件所在的目录，并需要手动为其生成所需的包，使用起来比较复杂。“overwrite” 模式直接使用插装后的文件覆盖插装前文件，使用方便。但是由于同一时间生成的文件只能插装一次，在 “overwrite” 模式下，插装前的文件已经丢失，无法重复插装操作，因此在插装之前先将需要插装的文件和包进行备份。

参数 “ip” 和 “cp” 用来提供插装路径，其中 “cp” 用来指明一个文件夹，“ip” 指定单独的文件或者 JAR 包。

参数 “Dmetadata.out.file” 用来指定插装得到的元数据文件保存的路径。

命令中通过 “ix” 参数指定文件的包含和排除关系，其中在 “+” 符号后的文件为包含进的文件，“-” 后面的内容为排除在外的文件。支持多个+和-，用逗号分隔。

插桩成果结果显示：(580 个类文件插桩成功)



```
C:\Windows\system32\cmd.exe
EMMA: [util.IPathEnumerator$Factory$PathEnumerator::enumeratePathDir] processing
file [jspxcms-form.tld] ...
EMMA: [emma.instr.InstrProcessorST::handleFile] [C:\apache-tomcat-6.0.29\webapps
\jspxcms\WEB-INF\classes] [META-INF\jspxcms-form.tld]
EMMA: [emma.instr.InstrProcessorST::handleDirEnd] [C:\apache-tomcat-6.0.29\webap
ps\jspxcms\WEB-INF\classes] [META-INF]
EMMA: [util.IPathEnumerator$Factory$PathEnumerator::enumeratePathDir] processing
file [stopword.dic] ...
EMMA: [emma.instr.InstrProcessorST::handleFile] [C:\apache-tomcat-6.0.29\webapps
\jspxcms\WEB-INF\classes] [stopword.dic]
EMMA: [util.IPathEnumerator$Factory$PathEnumerator::enumeratePathDir] processing
file [stopword_ext.dic] ...
EMMA: [emma.instr.InstrProcessorST::handleFile] [C:\apache-tomcat-6.0.29\webapps
\jspxcms\WEB-INF\classes] [stopword_ext.dic]
EMMA: instrumentation path processed in 37611 ms
EMMA: [580 class(es) instrumented, 0 resource(s) copied]
EMMA: metadata contains 580 entries
EMMA: [emma.data.DataFactory::persist] [C:\emma\jspxcms_coverage.em]: creating a
new file
EMMA: [emma.data.DataFactory::writeEntry] entry [com.vladium.emma.data.MetaData]
length: 799312
EMMA: [emma.data.DataFactory::persist] [C:\emma\jspxcms_coverage.em]: file proce
ssed in 78 ms
EMMA: metadata merged into [C:\emma\jspxcms_coverage.em] {in 78 ms}
```

完成插装操作以后，在指定的路径下会产生一些名为 “*coverage.em” 的文件，这些文件保存了插装的元信息，这些信息主要是记录插装过程中的插装点在被测代码中的位置。如果在插装过程中，指定这些文件到同一文件的话，EMMA 默认将元数据进行合并。如果未指定路径，或者希望得到独立的元文件，这些文件将分别产生在默认或指定的目录下。可以通过使用 “merge” 命令手动将这些元文件进行合并，保证生成的覆盖率报表的全面性。

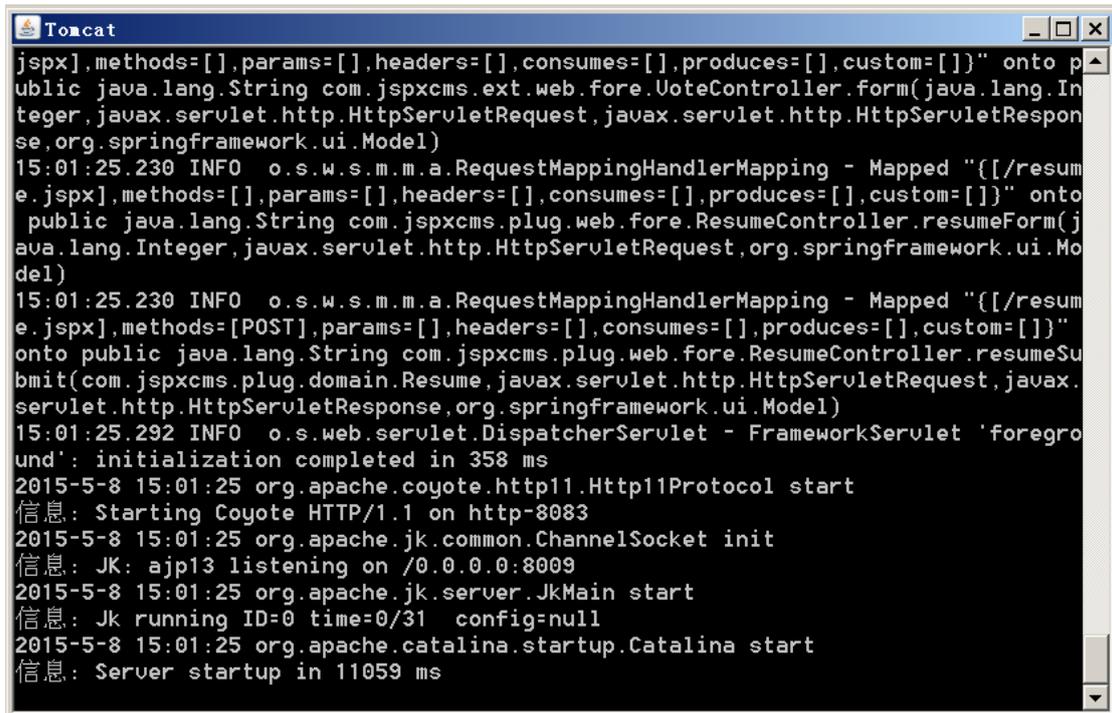
命令: java emma merge -input path1>/coverage1.em,<path2>/coverage2.em -out <path>/coverage.em

在 “input” 后面的参数为待合并的文件名，在 “out” 后面的参数为合并以后的结果文件。

运行

确保 emma.jar 已放在 tomcat 部署 JSPXCMS 的 lib 目录下，一般情况下，直接运行应用程序即可。EMMA 会启动一个监听端口，用来后面收集信息（ctl）。这个端口是固定的，47653。

启动 TOMCAT:



```
Tomcat
jsp],methods=[],params=[],headers=[],consumes=[],produces=[],custom=[]]" onto p
public java.lang.String com.jspxcms.ext.web.fore.VoteController.form(java.lang.In
Integer, javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse, org.springframework.ui.Model)
15:01:25.230 INFO  o.s.w.s.m.m.a.RequestMappingHandlerMapping - Mapped "{[/resum
e.jsp],methods=[],params=[],headers=[],consumes=[],produces=[],custom=[]}" onto
public java.lang.String com.jspxcms.plug.web.fore.ResumeController.resumeForm(j
ava.lang.Integer, javax.servlet.http.HttpServletRequest, org.springframework.ui.Mo
del)
15:01:25.230 INFO  o.s.w.s.m.m.a.RequestMappingHandlerMapping - Mapped "{[/resum
e.jsp],methods=[POST],params=[],headers=[],consumes=[],produces=[],custom=[]}"
onto public java.lang.String com.jspxcms.plug.web.fore.ResumeController.resumeSu
bmit(com.jspxcms.plug.domain.Resume, javax.servlet.http.HttpServletRequest, javax.
servlet.http.HttpServletResponse, org.springframework.ui.Model)
15:01:25.292 INFO  o.s.web.servlet.DispatcherServlet - FrameworkServlet 'foregro
und': initialization completed in 358 ms
2015-5-8 15:01:25 org.apache.coyote.http11.Http11Protocol start
信息: Starting Coyote HTTP/1.1 on http-8083
2015-5-8 15:01:25 org.apache.jk.common.ChannelSocket init
信息: JK: ajp13 listening on /0.0.0.0:8009
2015-5-8 15:01:25 org.apache.jk.server.JkMain start
信息: Jk running ID=0 time=0/31 config=null
2015-5-8 15:01:25 org.apache.catalina.startup.Catalina start
信息: Server startup in 11059 ms
```

打开 JSPXCMS 网站，执行一系列功能测试操作。

收集

执行收集命令前，应用服务需保持启用状态。

本地覆盖率数据收集命令:

```
命令提示符
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\XIANZC>cd C:/Program Files/Java/jre6/bin/

C:\Program Files\Java\jre6\bin>java -cp emma.jar emma ctl -connect localhost:47653 -command coverage.get,C:/emma/jspxcms_coverage.ec
```

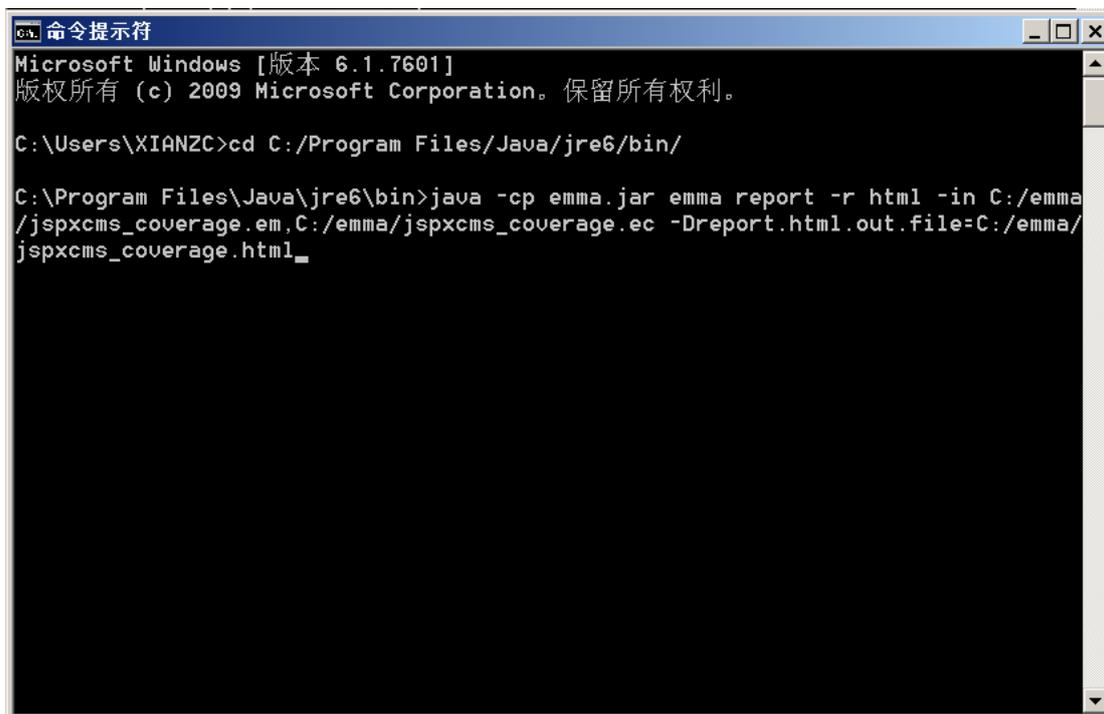
收集成功显示:

```
C:\Windows\system32\cmd.exe
EMMA: }
EMMA: connecting to [localhost:47653] ...
EMMA: executing [coverage.get] with arguments:
EMMA: {
EMMA:   C:/emma/jspxcms_coverage.ec
EMMA:   true
EMMA:   true
EMMA: }
EMMA: [emma.rt.RTControllerClientProxy::execute] request [coverage.get (C:/emma/jspxcms_coverage.ec,true,true)] sent
EMMA: [emma.rt.RTControllerClientProxy::execute] reading response ...
EMMA: [emma.rt.RTControllerClientProxy::execute] response received [com.vladium.emma.rt.rpc.Response@22998b08]
EMMA: [emma.ctl.CtlProcessor::_run] coverage.get: RPC call completed in 31 ms
EMMA: coverage.get: coverage data contains 237 entries
EMMA: [emma.data.DataFactory::persist] [C:\emma\jspxcms_coverage.ec]: creating a new file
EMMA: [emma.data.DataFactory::writeEntry] entry [com.vladium.emma.data.CoverageData] length: 38133
EMMA: [emma.data.DataFactory::persist] [C:\emma\jspxcms_coverage.ec]: file processed in 47 ms
EMMA: coverage.get: local copy of coverage data merged into [C:\emma\jspxcms_coverage.ec] (in 47 ms)
EMMA: coverage.get: command completed in 78 ms
EMMA: control command sequence complete
```

收集到的信息被保存在 “coverage.ec” 中。

报告

生成代码覆盖率报告命令:

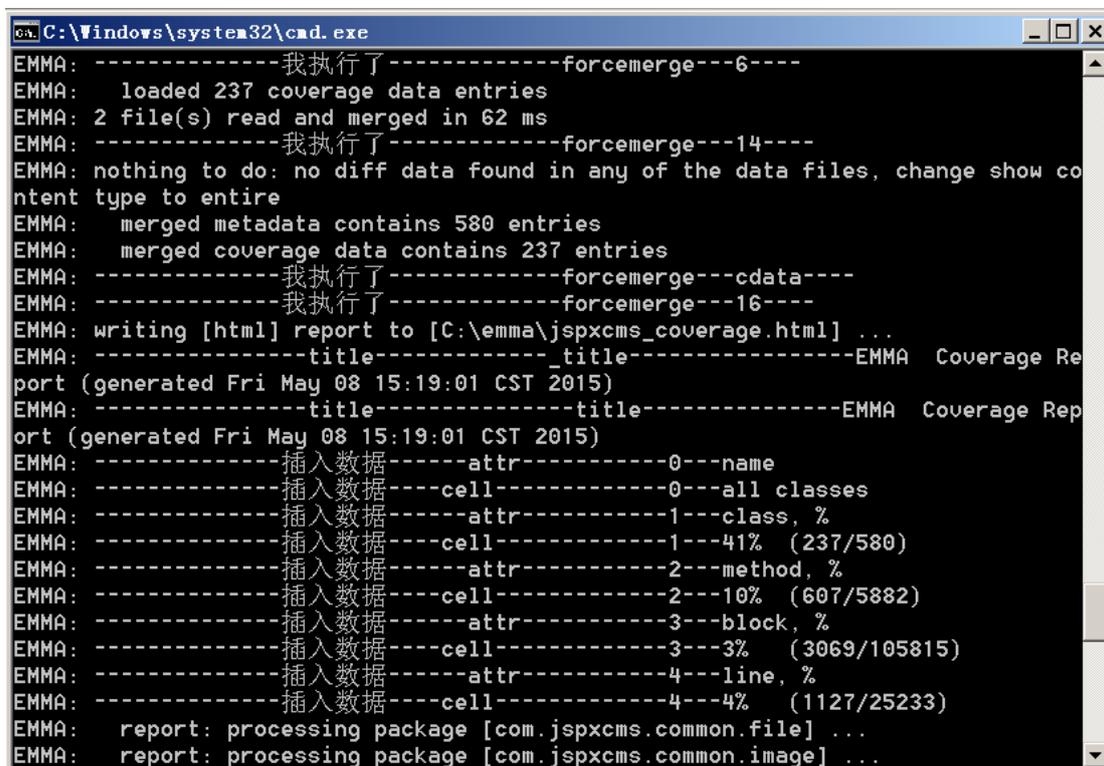


```
命令提示符
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\XIANZC>cd C:/Program Files/Java/jre6/bin/

C:\Program Files\Java\jre6\bin>java -cp emma.jar emma report -r html -in C:/emma/
jspxcms_coverage.em,C:/emma/jspxcms_coverage.ec -Dreport.html.out.file=C:/emma/
jspxcms_coverage.html_
```

执行成功会显示:



```
C:\Windows\system32\cmd.exe
EMMA: -----我执行了-----forcemerge---6----
EMMA:   loaded 237 coverage data entries
EMMA: 2 file(s) read and merged in 62 ms
EMMA: -----我执行了-----forcemerge---14----
EMMA: nothing to do: no diff data found in any of the data files, change show co
ntent type to entire
EMMA:   merged metadata contains 580 entries
EMMA:   merged coverage data contains 237 entries
EMMA: -----我执行了-----forcemerge---cdata---
EMMA: -----我执行了-----forcemerge---16----
EMMA: writing [html] report to [C:\emma\jspxcms_coverage.html] ...
EMMA: -----title-----_title-----EMMA Coverage Re
port (generated Fri May 08 15:19:01 CST 2015)
EMMA: -----title-----title-----EMMA Coverage Rep
ort (generated Fri May 08 15:19:01 CST 2015)
EMMA: -----插入数据-----attr-----0---name
EMMA: -----插入数据-----cell-----0---all classes
EMMA: -----插入数据-----attr-----1---class, %
EMMA: -----插入数据-----cell-----1---41% (237/580)
EMMA: -----插入数据-----attr-----2---method, %
EMMA: -----插入数据-----cell-----2---10% (607/5882)
EMMA: -----插入数据-----attr-----3---block, %
EMMA: -----插入数据-----cell-----3---3% (3069/105815)
EMMA: -----插入数据-----attr-----4---line, %
EMMA: -----插入数据-----cell-----4---4% (1127/25233)
EMMA:   report: processing package [com.jspxcms.common.file] ...
EMMA:   report: processing package [com.jspxcms.common.image] ...
```


下图是包 com.jspxcms.common.fulltext 类的执行情况：

EMMA Coverage: Report Fri May 08 15:19:01 CST 2015				
[all classes]				
COVERAGE SUMMARY FOR PACKAGE [com.jspxcms.common.fulltext]				
name	class, %	method, %	block, %	line, %
com.jspxcms.common.fulltext	57% (4/7)	36% (24/66)	10% (128/1280)	13% (47.3/357)
COVERAGE BREAKDOWN BY SOURCE FILE				
name	class, %	method, %	block, %	line, %
LuceneException.java	0% (0/1)	0% (0/2)	0% (0/9)	0% (0/4)
NFTLuceneIndexTemplate.java	0% (0/1)	0% (0/17)	0% (0/481)	0% (0/123)
SearchManagerFactoryBean.java	0% (0/3)	0% (0/7)	0% (0/78)	0% (0/24)
DefaultLuceneIndexTemplate.java	100% (1/1)	28% (5/18)	4% (25/563)	8% (11.3/151)
FSDirectoryFactoryBean.java	100% (1/1)	86% (6/7)	67% (34/51)	63% (12/19)
IndexWriterFactoryBean.java	100% (1/1)	88% (7/8)	63% (37/54)	65% (13/20)
SearcherManagerFactoryBean.java	100% (1/1)	88% (6/7)	73% (32/44)	69% (11/16)
[all classes]				
EMMA 3.2.1 (unsupported private build) (C) Vladimir Roubtsov				

下图是包 com.jspxcms.common.fulltext 类 DefaultLuceneIndexTemplate.java 的方法执行情况：

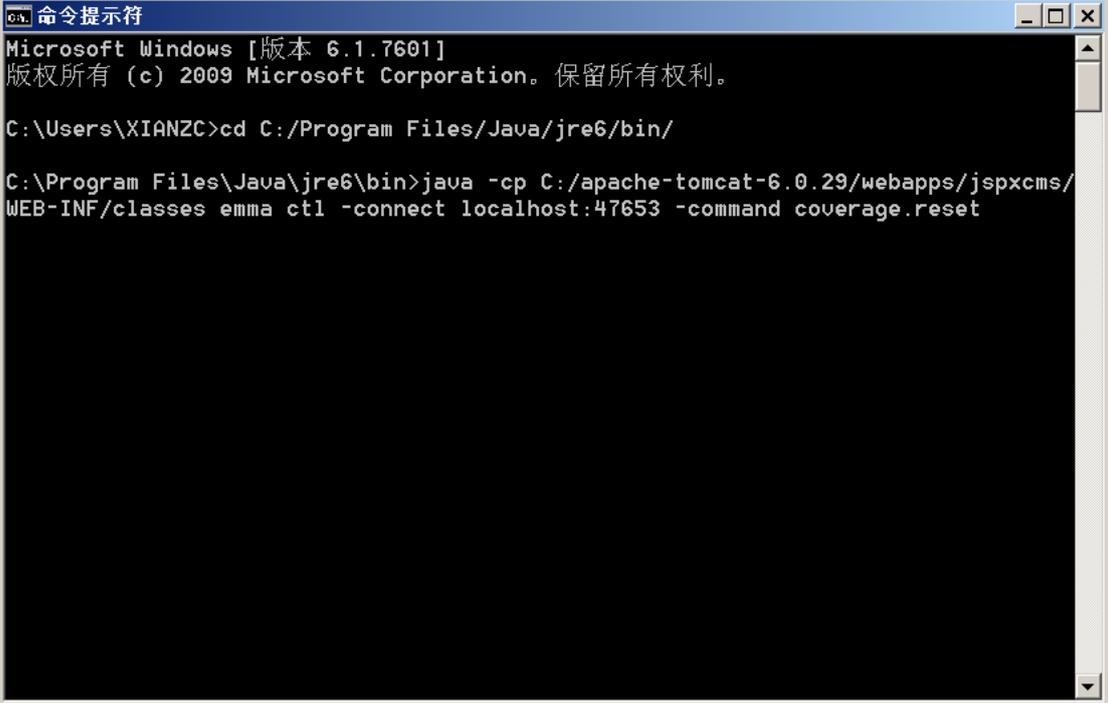
EMMA Coverage: Report Fri May 08 15:19:01 CST 2015				
[all classes][com.jspxcms.common.fulltext]				
COVERAGE SUMMARY FOR SOURCE FILE [DefaultLuceneIndexTemplate.java]				
name	class, %	method, %	block, %	line, %
DefaultLuceneIndexTemplate.java	100% (1/1)	28% (5/18)	4% (25/563)	8% (11.3/151)
COVERAGE BREAKDOWN BY CLASS AND METHOD				
name	class, %	method, %	block, %	line, %
class DefaultLuceneIndexTemplate	100% (1/1)	28% (5/18)	4% (25/563)	8% (11.3/151)
DefaultLuceneIndexTemplate (IndexWriter, SearcherManager): void		0% (0/1)	0% (0/12)	0% (0/5)
addDocument (Document): void		0% (0/1)	0% (0/19)	0% (0/6)
addDocuments (Collection): void		0% (0/1)	0% (0/19)	0% (0/6)
deleteAll (): void		0% (0/1)	0% (0/18)	0% (0/6)
deleteDocuments (Query []): void		0% (0/1)	0% (0/19)	0% (0/6)
deleteDocuments (Query): void		0% (0/1)	0% (0/19)	0% (0/6)
deleteDocuments (Term []): void		0% (0/1)	0% (0/19)	0% (0/6)
deleteDocuments (Term): void		0% (0/1)	0% (0/19)	0% (0/6)
list (Predicate, EntityPath, Limitable): List		0% (0/1)	0% (0/85)	0% (0/20)
list (Query, String, Limitable, Sort): List		0% (0/1)	0% (0/93)	0% (0/19)
page (Predicate, EntityPath, Pageable): Page		0% (0/1)	0% (0/81)	0% (0/22)
page (Query, String, Pageable, Sort): Page		0% (0/1)	0% (0/105)	0% (0/24)
updateDocument (Term, Document): void		0% (0/1)	0% (0/20)	0% (0/6)
afterPropertiesSet (): void	100% (1/1)		41% (7/17)	60% (3/5)
DefaultLuceneIndexTemplate (): void	100% (1/1)	100% (6/6)	100% (6/6)	100% (3/3)
setAutoCommit (boolean): void	100% (1/1)	100% (4/4)	100% (4/4)	100% (2/2)
setIndexWriter (IndexWriter): void	100% (1/1)	100% (4/4)	100% (4/4)	100% (2/2)
setSearcherManager (SearcherManager): void	100% (1/1)	100% (4/4)	100% (4/4)	100% (2/2)
[source file 'com.jspxcms/common/fulltext/DefaultLuceneIndexTemplate.java' not found in sourcepath]				
[all classes][com.jspxcms.common.fulltext]				
EMMA 3.2.1 (unsupported private build) (C) Vladimir Roubtsov				

独立测试用例覆盖率

为得到独立的测试报告，需要在每次执行测试用例前，将内存中的执行信息清除。目前有两种方法支持清除记录，测试人员可在测试过程中，根据需要选择合适的方法。

- ✓ 每次运行完一个测试用例，重启 JVM。这种方法能够完整的清除内存中记录的执行信息，但是每次重启 JVM 给测试带来很多麻烦。
- ✓ 使用“coverage.reset”命令，该命令可以在不重启 JVM 的情况下，清除内存中记录的方法、块、行的执行信息，但是无法清除类覆盖信息。如果用户关注的重点在方法的覆盖信息上，可以选择这种方法。

清除内存中覆盖率信息命令:

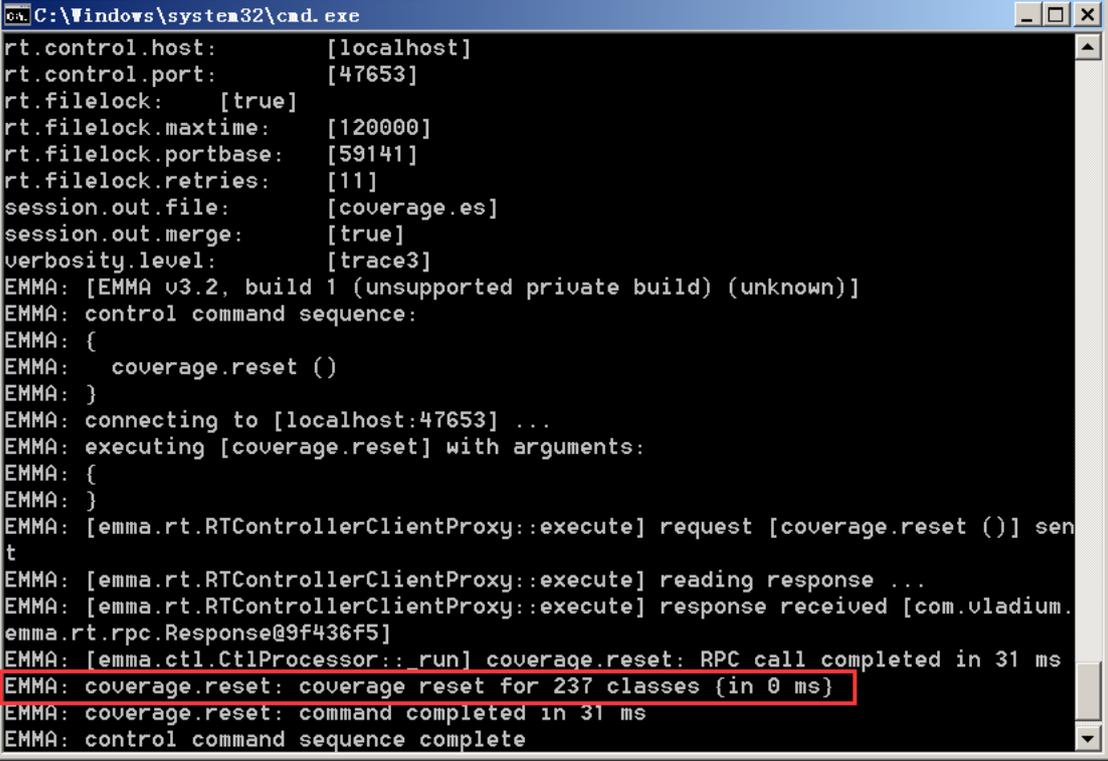


```
命令提示符
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\XIANZC>cd C:/Program Files/Java/jre6/bin/

C:\Program Files\Java\jre6\bin>java -cp C:/apache-tomcat-6.0.29/webapps/jspxcms/WEB-INF/classes emma ctl -connect localhost:47653 -command coverage.reset
```

执行成功会显示:



```
C:\Windows\system32\cmd.exe
rt.control.host: [localhost]
rt.control.port: [47653]
rt.filelock: [true]
rt.filelock.maxtime: [120000]
rt.filelock.portbase: [59141]
rt.filelock.retries: [11]
session.out.file: [coverage.es]
session.out.merge: [true]
verbosity.level: [trace3]
EMMA: [EMMA v3.2, build 1 (unsupported private build) (unknown)]
EMMA: control command sequence:
EMMA: {
EMMA:   coverage.reset ()
EMMA: }
EMMA: connecting to [localhost:47653] ...
EMMA: executing [coverage.reset] with arguments:
EMMA: {
EMMA: }
EMMA: [emma.rt.RTControllerClientProxy::execute] request [coverage.reset ()] sent
EMMA: [emma.rt.RTControllerClientProxy::execute] reading response ...
EMMA: [emma.rt.RTControllerClientProxy::execute] response received [com.vladium.emma.rt.rpc.Response@9f436f5]
EMMA: [emma.ctl.CtlProcessor::run] coverage.reset: RPC call completed in 31 ms
EMMA: coverage.reset: coverage reset for 237 classes {in 0 ms}
EMMA: coverage.reset: command completed in 31 ms
EMMA: control command sequence complete
```

下图可见，执行 coverage.reset 后，内存中记录的方法、块、行的执行信息被清除，但是无法清除类覆盖信息。

EMMA Coverage Report Fri May 08 16:21:15 CST 2015

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	41% (237/582)	0% (0/5882)	0% (0/105815)	0% (0/25233)

OVERALL STATS SUMMARY

total packages: 43
total executable files: 480
total classes: 580
total methods: 5882

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
com.jspxcms.common.file	0% (0/50)	0% (0/236)	0% (0/4666)	0% (0/969)
com.jspxcms.common.firemarker	48% (10/21)	0% (0/157)	0% (0/2227)	0% (0/750)
com.jspxcms.common.fulltext	57% (4/7)	0% (0/66)	0% (0/1280)	0% (0/357)
com.jspxcms.common.image	0% (0/5)	0% (0/97)	0% (0/2868)	0% (0/650)
com.jspxcms.common.ip	0% (0/4)	0% (0/40)	0% (0/1449)	0% (0/398)
com.jspxcms.common.ora	21% (3/14)	0% (0/93)	0% (0/2347)	0% (0/472)
com.jspxcms.common.security	31% (4/13)	0% (0/56)	0% (0/889)	0% (0/237)
com.jspxcms.common.upload	0% (0/3)	0% (0/42)	0% (0/402)	0% (0/127)
com.jspxcms.common.util	12% (1/8)	0% (0/77)	0% (0/1759)	0% (0/428)
com.jspxcms.common.web	31% (5/16)	0% (0/120)	0% (0/2549)	0% (0/574)
com.jspxcms.common.web.tags	0% (0/16)	0% (0/199)	0% (0/1989)	0% (0/597)
com.jspxcms.core.domain	87% (54/62)	0% (0/1701)	0% (0/14753)	0% (0/4280)
com.jspxcms.core.domainsdl	0% (0/55)	0% (0/318)	0% (0/7087)	0% (0/1179)
com.jspxcms.core.fulltext	50% (4/8)	0% (0/52)	0% (0/1790)	0% (0/354)
com.jspxcms.core.helper	0% (0/5)	0% (0/42)	0% (0/513)	0% (0/136)
com.jspxcms.core.html	12% (1/8)	0% (0/45)	0% (0/1317)	0% (0/331)
com.jspxcms.core.listbase	33% (1/3)	0% (0/21)	0% (0/27)	0% (0/21)
com.jspxcms.core.repository.impl	100% (15/18)	0% (0/67)	0% (0/2989)	0% (0/588)
com.jspxcms.core.security	80% (4/5)	0% (0/41)	0% (0/689)	0% (0/183)
com.jspxcms.core.service.impl	74% (52/70)	0% (0/822)	0% (0/17019)	0% (0/3961)
com.jspxcms.core.setup	0% (0/1)	0% (0/13)	0% (0/707)	0% (0/188)
com.jspxcms.core.support	42% (5/12)	0% (0/143)	0% (0/2951)	0% (0/821)
com.jspxcms.core.web.back	0% (0/37)	0% (0/397)	0% (0/10982)	0% (0/2933)
com.jspxcms.core.web.back.f7	0% (0/6)	0% (0/21)	0% (0/1021)	0% (0/232)
com.jspxcms.core.web.directive	96% (25/26)	0% (0/58)	0% (0/2083)	0% (0/479)
com.jspxcms.core.web.fore	0% (0/13)	0% (0/94)	0% (0/3164)	0% (0/809)
com.jspxcms.core.web.method	100% (2/2)	0% (0/4)	0% (0/96)	0% (0/25)
com.jspxcms.est.collect	0% (0/2)	0% (0/10)	0% (0/243)	0% (0/62)
com.jspxcms.est.domain	92% (11/12)	0% (0/359)	0% (0/2733)	0% (0/857)
com.jspxcms.est.domainsdl	0% (0/8)	0% (0/48)	0% (0/1028)	0% (0/181)
com.jspxcms.est.quartz	0% (0/1)	0% (0/3)	0% (0/51)	0% (0/15)
com.jspxcms.est.repository.impl	100% (11/11)	0% (0/27)	0% (0/684)	0% (0/134)
com.jspxcms.est.service.impl	50% (10/17)	0% (0/17)	0% (0/3067)	0% (0/207)
com.jspxcms.est.web.back	0% (0/11)	0% (0/98)	0% (0/2714)	0% (0/610)
com.jspxcms.est.web.directive	100% (8/8)	0% (0/16)	0% (0/585)	0% (0/123)
com.jspxcms.est.web.fore	0% (0/1)	0% (0/1)	0% (0/1)	0% (0/1)

当完成所有单独执行测试用例后，测试覆盖信息可以被合并在一起，得到整个测试的覆盖报告。覆盖率结果文件通过 “merge” 命令合并 “*.ec” 文件实现的。

由于 EMMA 中测试覆盖率是通过与 “*.em” 文件关联获得代码信息的，因此当代码发生变化时，已经运行过的测试不必完全重复，只需将得到的 “*.ec” 文件合并（新得到的 “*.ec” 文件放在后面），然后关联最新的 “*.em” 文件即可得到代码变化后的覆盖率信息，这方便了 EMMA 支持版本变化的测试。在生成新的测试报告的时候，需要注意 “*.ec” 的时间一定要晚于 “*.em” 文件。

- ✓ 自动合并。如果生成的.ec 文件名字相同，自动合并。
- ✓ 命令行合并：`java -cp emma merge -input jspxcms1_coverage.ec, jspxcms2_coverage.ec -outfile jspxcms_coverage.ec`

举例说明：（收集独立测试用例“搜索功能”的代码覆盖率数据，并定位其具体实现类和方法）

独立测试用例“搜索功能”执行前，如下图覆盖率为 0%，被执行的类和方法同样为 0：

EMMA Coverage Report Fri May 08 10:34:06 CST 2015

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	0% (0/566)	0% (0/5789)	0% (0/103468)	0% (0/24761)

OVERALL STATS SUMMARY

total packages: 42
total executable files: 478
total classes: 566
total methods: 5789

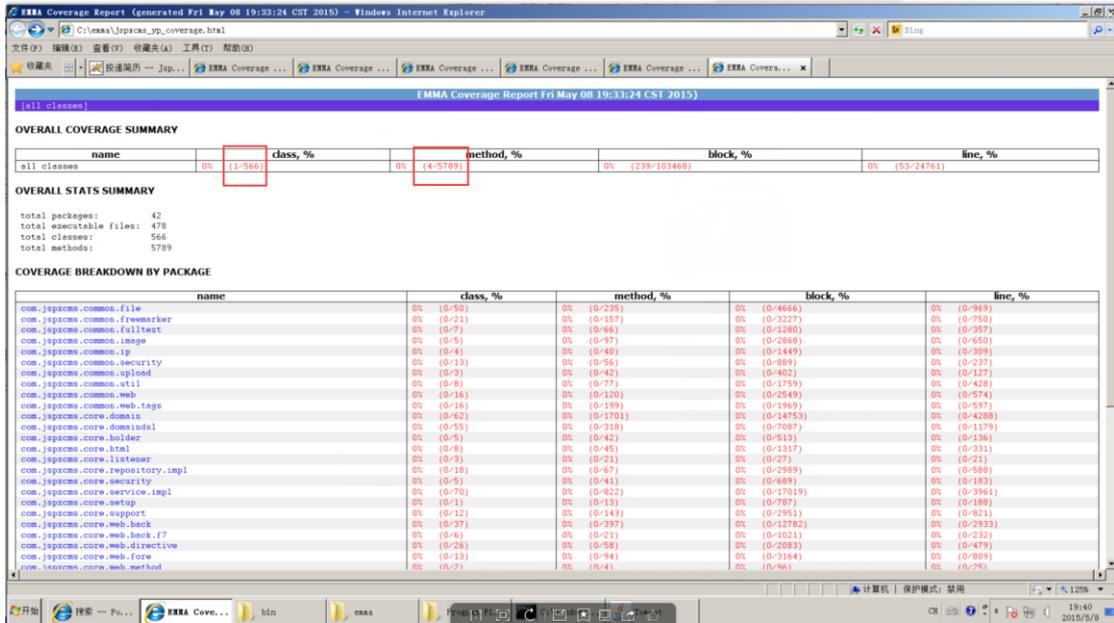
COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
com.jspxcms.common.file	0% (0/50)	0% (0/235)	0% (0/8666)	0% (0/969)
com.jspxcms.common.freemarker	0% (0/21)	0% (0/157)	0% (0/2227)	0% (0/750)
com.jspxcms.common.fulltext	0% (0/7)	0% (0/66)	0% (0/1280)	0% (0/357)
com.jspxcms.common.image	0% (0/5)	0% (0/97)	0% (0/2868)	0% (0/650)
com.jspxcms.common.ip	0% (0/4)	0% (0/40)	0% (0/1448)	0% (0/309)
com.jspxcms.common.security	0% (0/13)	0% (0/56)	0% (0/889)	0% (0/237)
com.jspxcms.common.upload	0% (0/3)	0% (0/42)	0% (0/402)	0% (0/127)
com.jspxcms.common.util	0% (0/8)	0% (0/77)	0% (0/1759)	0% (0/418)
com.jspxcms.common.web	0% (0/16)	0% (0/120)	0% (0/2549)	0% (0/574)
com.jspxcms.common.web.tags	0% (0/16)	0% (0/199)	0% (0/1969)	0% (0/597)
com.jspxcms.core.domain	0% (0/62)	0% (0/1701)	0% (0/14753)	0% (0/4388)
com.jspxcms.core.domainid	0% (0/55)	0% (0/318)	0% (0/7087)	0% (0/1179)
com.jspxcms.core.fulltext	0% (0/8)	0% (0/62)	0% (0/1798)	0% (0/394)
com.jspxcms.core.holder	0% (0/5)	0% (0/42)	0% (0/513)	0% (0/136)
com.jspxcms.core.html	0% (0/8)	0% (0/45)	0% (0/1317)	0% (0/331)
com.jspxcms.core.listener	0% (0/3)	0% (0/21)	0% (0/27)	0% (0/21)
com.jspxcms.core.repository.impl	0% (0/28)	0% (0/67)	0% (0/288)	0% (0/588)
com.jspxcms.core.security	0% (0/5)	0% (0/41)	0% (0/689)	0% (0/183)
com.jspxcms.core.service.impl	0% (0/70)	0% (0/822)	0% (0/17019)	0% (0/3961)
com.jspxcms.core.setup	0% (0/2)	0% (0/13)	0% (0/707)	0% (0/188)
com.jspxcms.core.support	0% (0/12)	0% (0/143)	0% (0/2951)	0% (0/821)
com.jspxcms.core.web.back	0% (0/37)	0% (0/397)	0% (0/12782)	0% (0/2933)
com.jspxcms.core.web.back.ft	0% (0/6)	0% (0/21)	0% (0/1021)	0% (0/232)
com.jspxcms.core.web.directive	0% (0/26)	0% (0/58)	0% (0/2083)	0% (0/479)
com.jspxcms.core.web.force	0% (0/13)	0% (0/94)	0% (0/3164)	0% (0/809)
com.jspxcms.core.web.method	0% (0/8)	0% (0/4)	0% (0/46)	0% (0/25)
com.jspxcms.est.collect	0% (0/2)	0% (0/10)	0% (0/243)	0% (0/62)
com.jspxcms.est.domain	0% (0/12)	0% (0/359)	0% (0/2733)	0% (0/857)
com.jspxcms.est.domainid	0% (0/8)	0% (0/48)	0% (0/1026)	0% (0/241)
com.jspxcms.est.quartz	0% (0/1)	0% (0/3)	0% (0/51)	0% (0/15)
com.jspxcms.est.repository.impl	0% (0/11)	0% (0/27)	0% (0/684)	0% (0/134)
com.jspxcms.est.service.impl	0% (0/17)	0% (0/177)	0% (0/3067)	0% (0/707)
com.jspxcms.est.web.back	0% (0/11)	0% (0/98)	0% (0/2714)	0% (0/610)
com.jspxcms.est.web.directive	0% (0/8)	0% (0/16)	0% (0/585)	0% (0/123)
com.jspxcms.est.web.force	0% (0/7)	0% (0/24)	0% (0/871)	0% (0/212)
com.jspxcms.plugin.backup	0% (0/0)	0% (0/0)	0% (0/600)	0% (0/138)

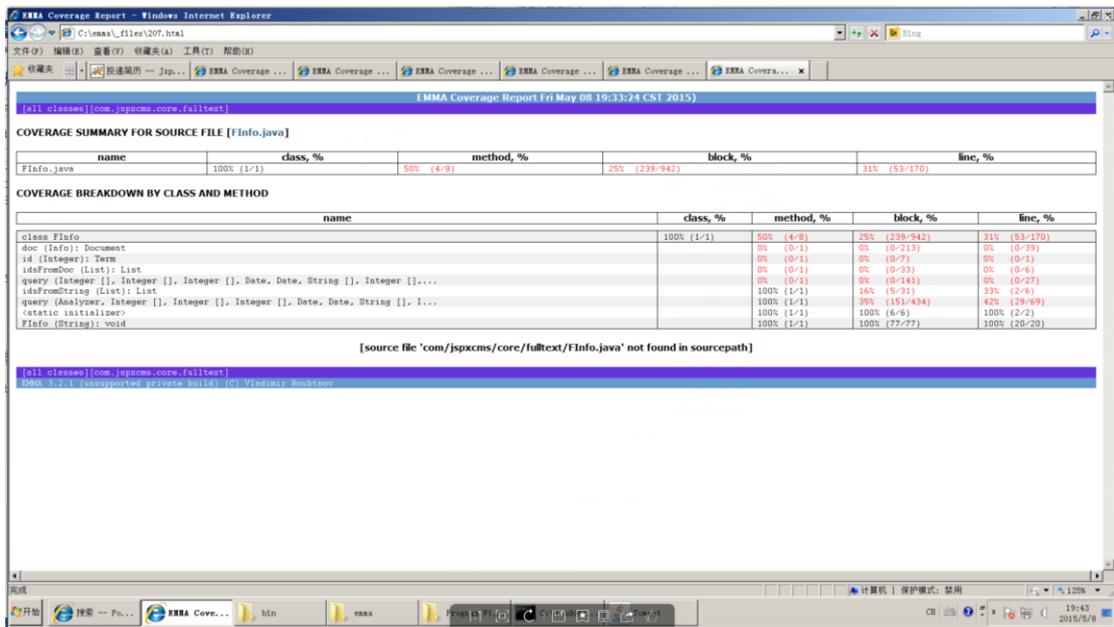
在网站页面，执行搜索操作：



收集覆盖率信息，生成报告：



由上图可见，执行搜索测试用例后，1 个类和 4 个方法被调用。进入查看明细：



从上图报告明细可见，搜索测试用例关联的类为 FInfo.java。即是可通过测试用例反推定位找到具体实现的类和方法。

存在问题：

一般情况下，系统首页加载，系统登陆，页面跳转等情况可能会执行相关类和方法，故测试人员进入要测功能模块页面，未测试时，已经产生代码覆盖率数据。使用 coverage.reset 命令可清除方法、块、行执行信息，但无法清除类的执行信息。如果通过重启 JVM 实现清除类、方法、块、行执行信息，除了操作上较麻烦外，还因为对于一般大型项目 web 系统，在 tomcat、websphere 或 weblogic 启动时，应用程序的一些后台服务，配置、数据字典等相关类和方法已经执行，故重启 JVM 的方法同样可能存在独立测试用例未执行已经存在代码覆盖率数据。所以独立用例关联到的类可能会包含其他类。故如果需要收集独立测试用例覆

盖率信息，在执行独立测试用例之前，确保类、方法的覆盖率为 0%。执行独立测试用例后，生成的覆盖率数据为干净的独立测试用例覆盖率数据。

结束语

在功能测试过程中，为每个单独的测试用例生成独立的覆盖率报告能够给测试过程带来很大的帮助：

- ✓ 当测试用例失败或者抛出异常时，可以通过覆盖率报告找到该测试用例对应的代码，这样就可以为测试人员提供可能出错代码的范围。这一报告不仅可以帮助测试人员在提交问题时更加详细的描述错误，提供更详细的信息，还可以为开发人员跟踪问题提供线索，缩短解决问题的周期。
- ✓ 测试人员可以从独立的测试报告中获得代码和功能模块的对应关系，更好的理解测试用例的作用。
- ✓ 独立的测试报告可以帮助测试人员改进测试用例的设计，删除重复的测试用例，将覆盖点较多的测试用例进行拆分。