chem-bla-ics

Code coverage: making sure your code is tested

Egon Willighagen

Published November 28, 2006

Citation
Willighagen, E. (2006). Code coverage: making sure your code is tested. In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/gtt31-tee03

Keywords
Opensource, Cdk, Junit

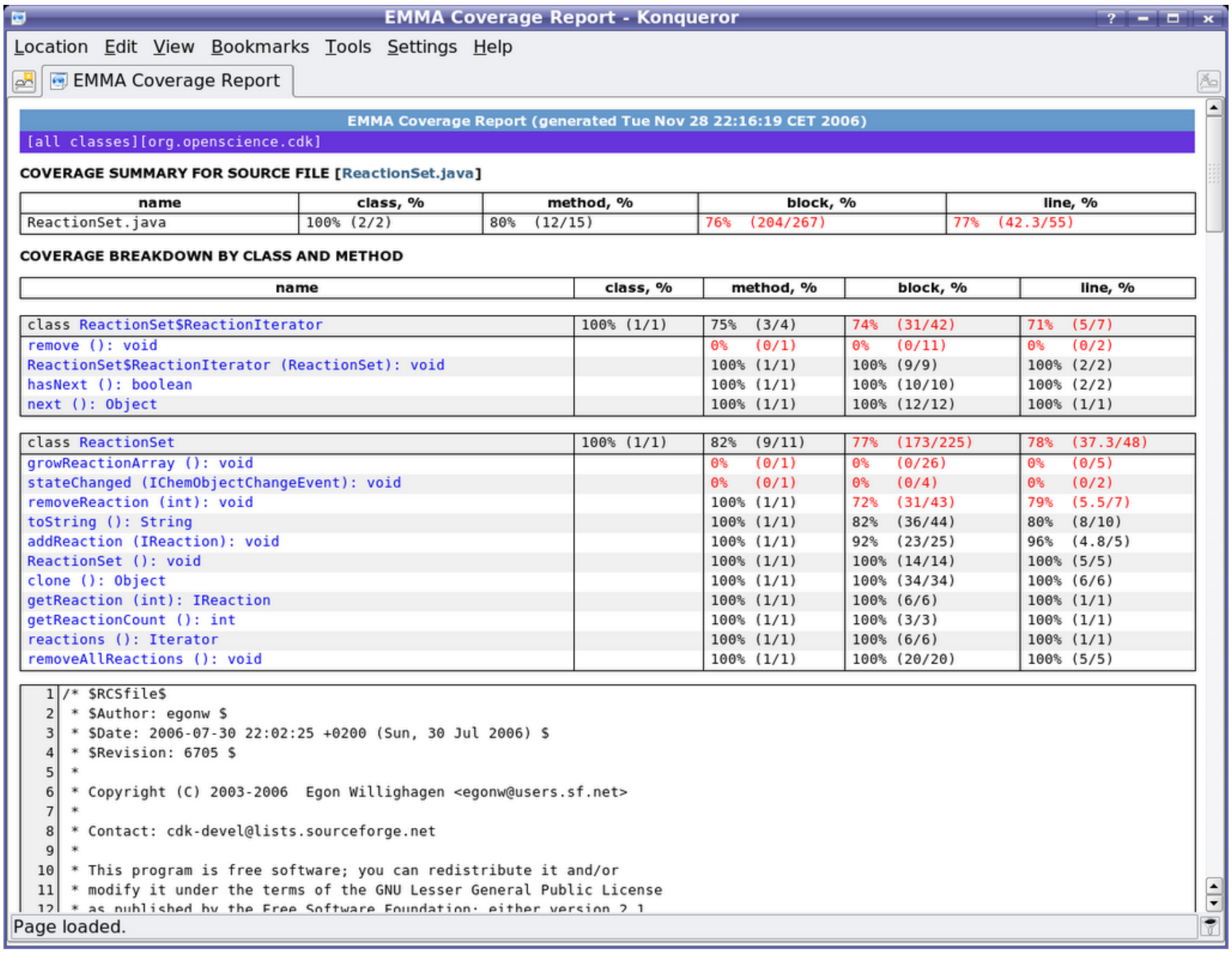
Abstract
Recently I discussed JUnit testing from within Eclipse , and blogged at several occasions about it in other situations. I cannot stress enough how useful unit testing is: it adds this extra set of eyeballs to make bugs shallow. And it does that, indeed.

Copyright

Recently I discussed JUnit testing from within Eclipse , and blogged at several occasions about it in other situations. I cannot stress enough how useful unit testing is: it adds this extra set of eyeballs to make bugs shallow. And it does that, indeed.

Ensuring that you actually test all the code you write, however, is not easy. A couple of years back I read an article about Hansel, which does code coverage checking, but never got it nicely working for the CDK project. Never looked at that lately, so no idea how the current release would work out. Hansel is an extension of JUnit, and requires hard coding class names, which conflicts with CDK's module setup.

Thomas Kuhn pointed me last week to Emma, which seems a nice tool. It does not require hacking our source, and generates cool HTML:


EMMA Coverage Report - Konqueror
Location Edit View Bookmarks Tools Settings Help
EMMA Coverage Report
Page loaded.

EMMA Coverage Report (generated Tue Nov 28 22:16:19 CET 2006)

[all classes][org.openscience.cdk]

COVERAGE SUMMARY FOR SOURCE FILE [ReactionSet.java]

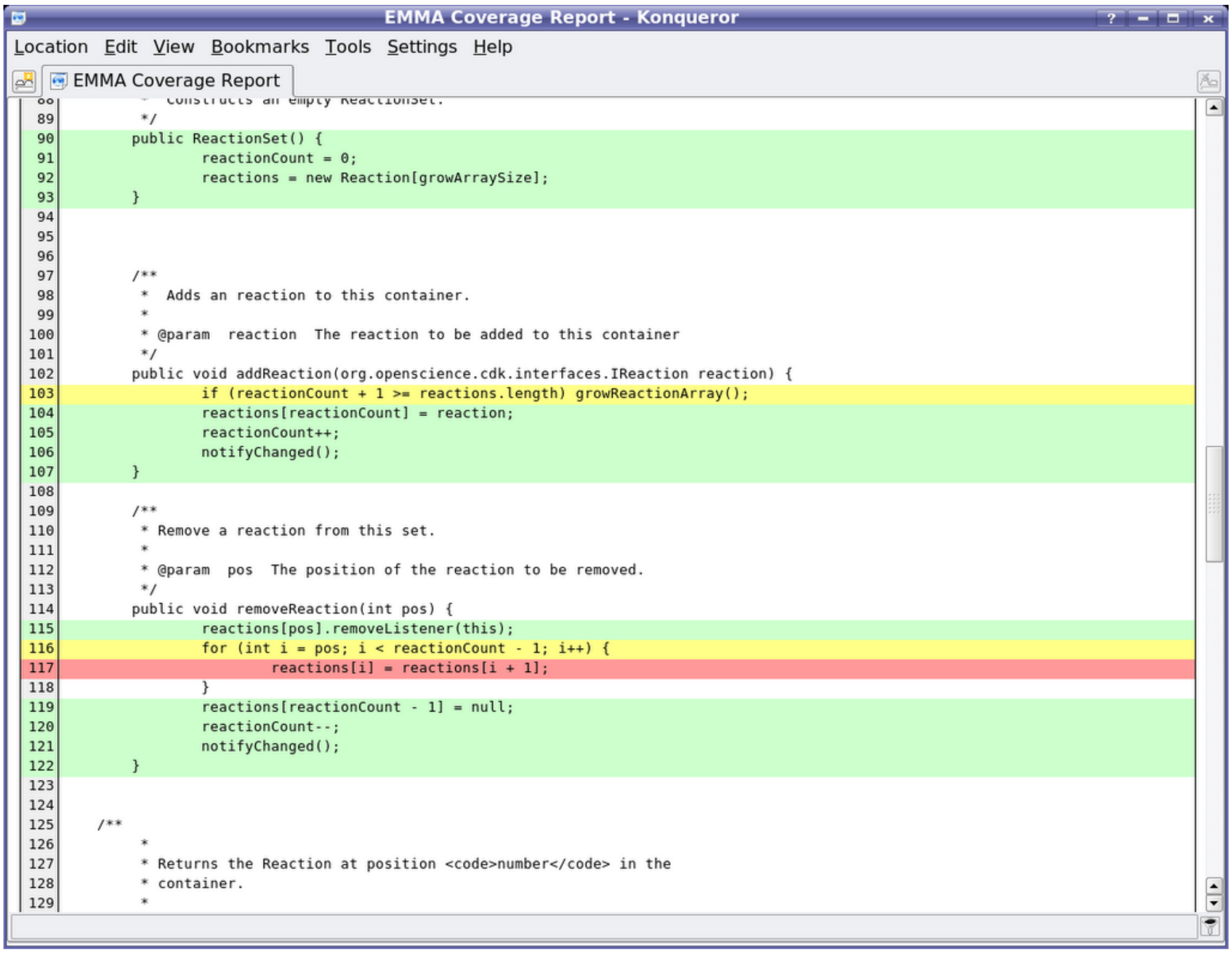
name	class, %	method, %	block, %	line, %
ReactionSet.java	100% (2/2)	80% (12/15)	76% (204/267)	77% (42.3/55)

COVERAGE BREAKDOWN BY CLASS AND METHOD

name	class, %	method, %	block, %	line, %
class ReactionSet$ReactionIterator	100% (1/1)	75% (3/4)	74% (31/42)	71% (5/7)
remove (): void		0% (0/1)	0% (0/11)	0% (0/2)
ReactionSet$ReactionIterator (ReactionSet): void		100% (1/1)	100% (9/9)	100% (2/2)
hasNext (): boolean		100% (1/1)	100% (10/10)	100% (2/2)
next (): Object		100% (1/1)	100% (12/12)	100% (1/1)
class ReactionSet	100% (1/1)	82% (9/11)	77% (173/225)	78% (37.3/48)
growReactionArray (): void		0% (0/1)	0% (0/26)	0% (0/5)
stateChanged (IChemObjectChangeEvent): void		0% (0/1)	0% (0/4)	0% (0/2)
removeReaction (int): void		100% (1/1)	72% (31/43)	79% (5.5/7)
toString (): String		100% (1/1)	82% (36/44)	80% (8/10)
addReaction (IReaction): void		100% (1/1)	92% (23/25)	96% (4.8/5)
ReactionSet (): void		100% (1/1)	100% (14/14)	100% (5/5)
clone (): Object		100% (1/1)	100% (34/34)	100% (6/6)
getReaction (int): IReaction		100% (1/1)	100% (6/6)	100% (1/1)
getReactionCount (): int		100% (1/1)	100% (3/3)	100% (1/1)
reactions (): Iterator		100% (1/1)	100% (6/6)	100% (1/1)
removeAllReactions (): void		100% (1/1)	100% (20/20)	100% (5/5)

```
/* $RCSfile$
 * $Author: egonw $
 * $Date: 2006-07-30 22:02:25 +0200 (Sun, 30 Jul 2006) $
 * $Revision: 6705 $
 *
 * Copyright (C) 2003-2006  Egon Willighagen <egonw@users.sf.net>
 *
 * Contact: cdk-devel@lists.sourceforge.net
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public License
 * as published by the Free Software Foundation; either version 2.1
```

And even highlights the source code:


EMMA Coverage Report - Konqueror
Location Edit View Bookmarks Tools Settings Help
EMMA Coverage Report

```
         * Constructs an empty ReactionSet.
         */
        public ReactionSet() {
                reactionCount = 0;
                reactions = new Reaction[growArraySize];
        }



        /**
         *  Adds an reaction to this container.
         *
         * @param  reaction  The reaction to be added to this container
         */
        public void addReaction(org.openscience.cdk.interfaces.IReaction reaction) {
                if (reactionCount + 1 >= reactions.length) growReactionArray();
                reactions[reactionCount] = reaction;
                reactionCount++;
                notifyChanged();
        }

        /**
         * Remove a reaction from this set.
         *
         * @param  pos  The position of the reaction to be removed.
         */
        public void removeReaction(int pos) {
                reactions[pos].removeListener(this);
                for (int i = pos; i < reactionCount - 1; i++) {
                        reactions[i] = reactions[i + 1];
                }
                reactions[reactionCount - 1] = null;
                reactionCount--;
                notifyChanged();
        }


    /**
         *
         * Returns the Reaction at position <code>number</code> in the
         * container.
         *
```

BTW, I seem to be in good company: Classpath is using it too.

Below is the command I issued to generate the HTML output. Rajarshi, maybe this can be integrated into Nightly? Note that it only runs the tests for the data module:

```
ant dist-large dist-test-large
java -cp ~/tmp/emma-2.0.5312/lib/emma.jar emmarun \
  -cp develjar/junit.jar:dist/jar/cdk-svn-20061128.jar:dist/jar/cdk-test-svn-20061128.ja
  -r html -sp src junit.textui.TestRunner org.openscience.cdk.test.MdataTest
```