

PDFUnit

Automated PDF Tests

Purchase orders, invoices and other PDF documents are the output of individual programs.

They might have errors. They can be tested.

more ...

Content

- General requirements for PDF testing tools
- Features of PDFUnit-Java
- PDFUnit-Monitor
- Utilities
- Code Examples, PDFUnit-Java
- Code Examples, PDFUnit-XML
- Code Examples, PDFUnit-Perl
- Code Examples, PDFUnit-NET
- History of PDFUnit
- Inside PDFUnit-Java
- Contact

General Requirements for every Testing Tool

- A testing tool has to be easy to use
- A testing tool has to cover the functionality requested by the market
- A testing tool has to integrate with common languages
- A testing tool has to be tested itself

PDFUnit fulfills all these requirements

Java - available
XML - available
Perl - available
.NET - available 03/2016

Features of PDFUnit (I)

- Checking for **text** on specific pages, also inside a given **clipping area**
- Verifying **rotated text**, also in a clipping area
- Validating **bookmarks**, **destinations**
- Validating **JavaScript**
- Validating a lot of **actions**
- Verifying keywords, title and other **metadata**, also **custom** properties
- Verifying the names of **layers**
- Verifying the **fonts**, **font subsets** and **font types**
- Checking a lot of details of **acrofields**
- Verifying **text overflow in acrofields**

No other test tool does this

Features of PDFUnit (II)

- Comparing **images** inside PDF against image files
- Counting **visible images** on selected pages
- Verifying **fast-web-view**
- Verifying any page **format**
- Comparing a test PDF with a **master** PDF
- Comparing a **rendered PDF** page against a **rendered master**
- Comparing a **rendered PDF** page against an **image file**
- Checking **XFA**-data
- Checking **XMP**-data

A diff image is created
in case of an error

Features of PDFUnit (III)

- Working with [Unicode](#)
- Verifying [signatures](#) inside a PDF
- Comparing [embedded files](#) against external files
- Using [XPath](#) in several test scopes
- Different [whitespace processing](#) for comparison is available
- Individual [page selection](#) is possible



Many signature data can be checked

PDFUnit-Monitor

Non-developers can use the PDFUnit-Monitor for PDF tests

The screenshot shows the PDFUnit Monitor application window. It features a 'Watched Folders' tree on the left, a 'Filter' section at the top right, a 'Validation Results' table in the center, and a 'Details' section at the bottom. Callouts point to specific elements: 'Message filter' points to the filter dropdown, 'Overview files' points to the folder tree, 'Overview messages' points to the results table, and 'Message details' points to the details section.

Message filter

Overview files

Overview messages

Message details

Validation Results:

Time	Error Level	PDF Document	Constraint
2015-10-24 10:19:21	ERROR	din5008_letter-formA_1-page_text-in-left-mar...	CheckDIN5008_FormA_individual.xls
2015-10-24 10:19:21	ERROR	din5008_letter-formA_1-page_text-in-left-mar...	CheckDIN5008_FormA_general.xls
2015-10-24 10:19:21	ERROR	din5008_letter-formA_1-page_content-with-er...	CheckDIN5008_FormA_individual.xls
2015-10-24 10:19:20	ERROR	din5008_letter-formA_1-page_content-with-er...	CheckDIN5008_FormA_general.xls

Details:

Error Level: ERROR

Message: Die Seitenangabe hat nicht das richtige Format.

Page 1 of 'C:\daten\p...in5008_letter-formA_1-page_text-in-left-margin.pdf' does not match expression '.*Seite \d von \d.*'.

PDF File: din5008_letter-formA_1-page_text-in-left-margin.pdf

Folder: C:\daten\pdfunit\workspace_pdfunit_all\pdfmonitor\src\test\resources\folder-for-pdfmonitor\

Constraint: CheckDIN5008_FormA_general.xls

Validation Time: 2015-10-24 10:19:21

Buttons at the bottom: Interrupt monitoring, Validate all, Reset filter, Clear data, Import, Export, Exit.

Utilities

A couple of utilities are provided to extract data/images from PDF:

- Extract acrofield infos to XML
- Extract bookmarks to XML
- Extract embedded files
- Extract font infos to XML
- Extract images to PNG
- Extract JavaScript
- Extract named destinations to XML
- Extract signature infos to XML
- Extract XFA-data to XML
- Extract XMP-data to XML
- Render PDF pages into PNG
- Render sections of a PDF page into PNG
- Convert any Unicode string into Unicode-Hex-Values (\uXXXX)

Examples Java (I)

```
AssertThat.document(filename)
    .hasText(ON_FIRST_PAGE)
    .startingWith("PDFUnit")
    .containing("Content on first page.")
    .endingWith("Page # 1 of 4")
;
```

Different types to compare text

```
int upperLeftX = 222; // in millimeter
int upperLeftY = 720;
int width      = 111;
int height     = 30;
ClippingArea inClippingArea = new ClippingArea(upperLeftX, upperLeftY, width, height);

AssertThat.document(filename)
    .hasText(ON_FIRST_PAGE, inClippingArea)
    .startingWith("Content")
;
```

Text in specified regions

```
AssertThat.document(filename)
    .hasField("Textfield-1_left-aligned").withWidth(200, POINTS).withHeight(25, POINTS)
    .hasField("Textfield-2_right-aligned").withWidth(200, POINTS).withHeight(60, POINTS)
    .hasField("Textfield-3_centered").withWidth(150, POINTS).withHeight(60, POINTS)
;
```

Form field validation

Examples Java (II)

```
AssertThat.document(filenameTest)
    .and(filenameMaster)
    .haveSameText(ON_FIRST_PAGE, inClippingArea, WhitespaceProcessing.NORMALIZE)
;
```

Comparing text in a region

```
AssertThat.document(filenameTest)
    .and(filenameMaster)
    .haveSameTitle()
    .haveSameSubject()
    .haveSameKeywords()
    .haveSameAuthor()
;
```

Comparing meta data

```
AssertThat.document(filename)
    .hasSignature("Signature2")
    .withCertificate()
    .havingSubjectField("CN").withValue("John Doe")
    .havingSubjectField("C").withValue("US")
;
```

Test for signatures

```
defaultNS = new DefaultNamespace("http://purl.org/dc/elements/1.1/");
expression = new XPathExpression("//dc:format = 'application/pdf'", defaultNS);
AssertThat.document(filename)
    .hasXMPData()
    .matchingXPath(expression)
;
```

Using XPath to check XMP data

Examples XML

```
<testcase name="hasText_OnFirstPage_InClippingArea">
  <assertThat testDocument="&pdfdir;/content/documentForTextClipping.pdf">
    <hasText on="FIRST_PAGE" >
      <inClippingArea upperLeftX="17.6" upperLeftY="45.8" width="60.0" height="8.8" >
        <startingWith>Content</startingWith>
        <containing>on first</containing>
        <endingWith>page.</endingWith>
      </inClippingArea>
    </hasText>
  </assertThat>
</testcase>
```

Text in a specified region

```
<testcase name="hasSignaturesMatchingXPath_CoveringWholeDocument_nodeExists">
  <assertThat testDocument="&pdfdir;/signed/sampleSignedPDFDocument.pdf">
    <hasSignatures matchingXPath="//signature[@name='Signature2'][@coversWholeDocument='true']" />
  </assertThat>
</testcase>
```

Test for signatures using XPath

- Compatible with the Java-API
- Same reporting as in PDFUnit-Java

Examples Perl

```
6 #
7 # Test hasAuthor
8 #
9 ok(
10 com::pdfunit::AssertThat
11     ->document("documentInfos/documentInfos_allInfos.pdf")
12     ->hasAuthor()
13     ->matchingComplete("PDFUnit.com")
14 , "Document does not have the expected author");
15
16 #
17 # Test hasFormat
18 #
19 ok(
20 com::pdfunit::AssertThat
21     ->document("documentInfos/documentInfos_allInfos.pdf")
22     ->hasFormat($com::pdfunit::Constants::A4_PORTRAIT)
23 , "Document does not have the expected format A4 portrait")
24 ;
25
```

Using Inline::Java and TEST::More.

CPAN-Modul PDF::PDFUnit available 12/2015

Examples C#

```
[TestMethod]
public void HasAuthor()
{
    String filename = path + "documentInfos/documentInfos_allInfos.pdf";

    AssertThat.document(filename)
        .hasAuthor()
        .matchingComplete("PDFUnit.com")
    ;
}
```

The DLL is created from JAR files

So:

- 100% compatibel with Java
- Method names are in lower case

```
[TestMethod]
[ExpectedException(typeof(PDFUnitError))]
public void HasAuthor_NoAuthorInPDF()
{
    String filename = path + "documentInfos/documentInfos_noAuthor.pdf";

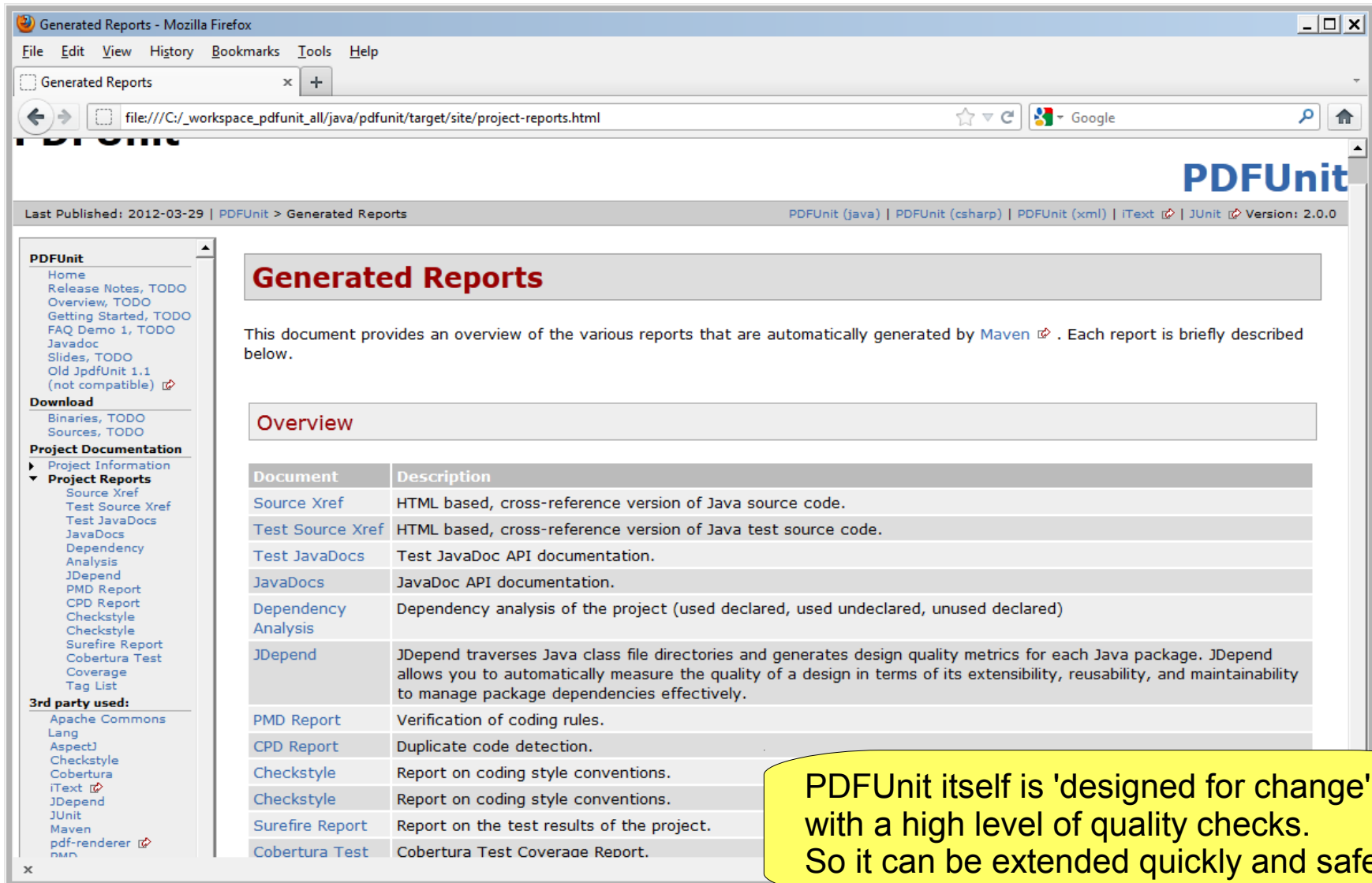
    AssertThat.document(filename)
        .hasAuthor()
    ;
}
```

Available in 03/2016

History of PDFUnit-Java

- Before 10/2010 - Training with Selenium, Canoo-Webtest
- 10/2010 - Sending a bugfix to jpdfunit 1.1
(<http://jpdfunit.sourceforge.net/>)
Joining the project as a developer.
- 01/2011 - PDFBox separated by an interface.
New implementation with iText.
- 08/2011 - PDFBox dropped off. Total Redesign
- 12/2011 - Proof of concept for an XML-API
- 03/2012 - Proof of concept for a C#-implementation
- 07/2012 - Implementation of the XML-API
- 12/2012 - Finishing the Java-API with a lot of new functions
- 06/2014 - New release with more functions and less bugs
- 10/2015 - PDFMonitor added, PDFUnit-Perl released

Inside PDFUnit-Java – Analyzing continuously



The screenshot shows a Mozilla Firefox browser window displaying the PDFUnit website. The address bar shows the URL: file:///C:/_workspace_pdfunit_all/java/pdfunit/target/site/project-reports.html. The page title is "Generated Reports - Mozilla Firefox". The page content includes a sidebar with navigation links, a main heading "Generated Reports", a paragraph describing the reports, and a table titled "Overview" listing various reports and their descriptions.

Generated Reports

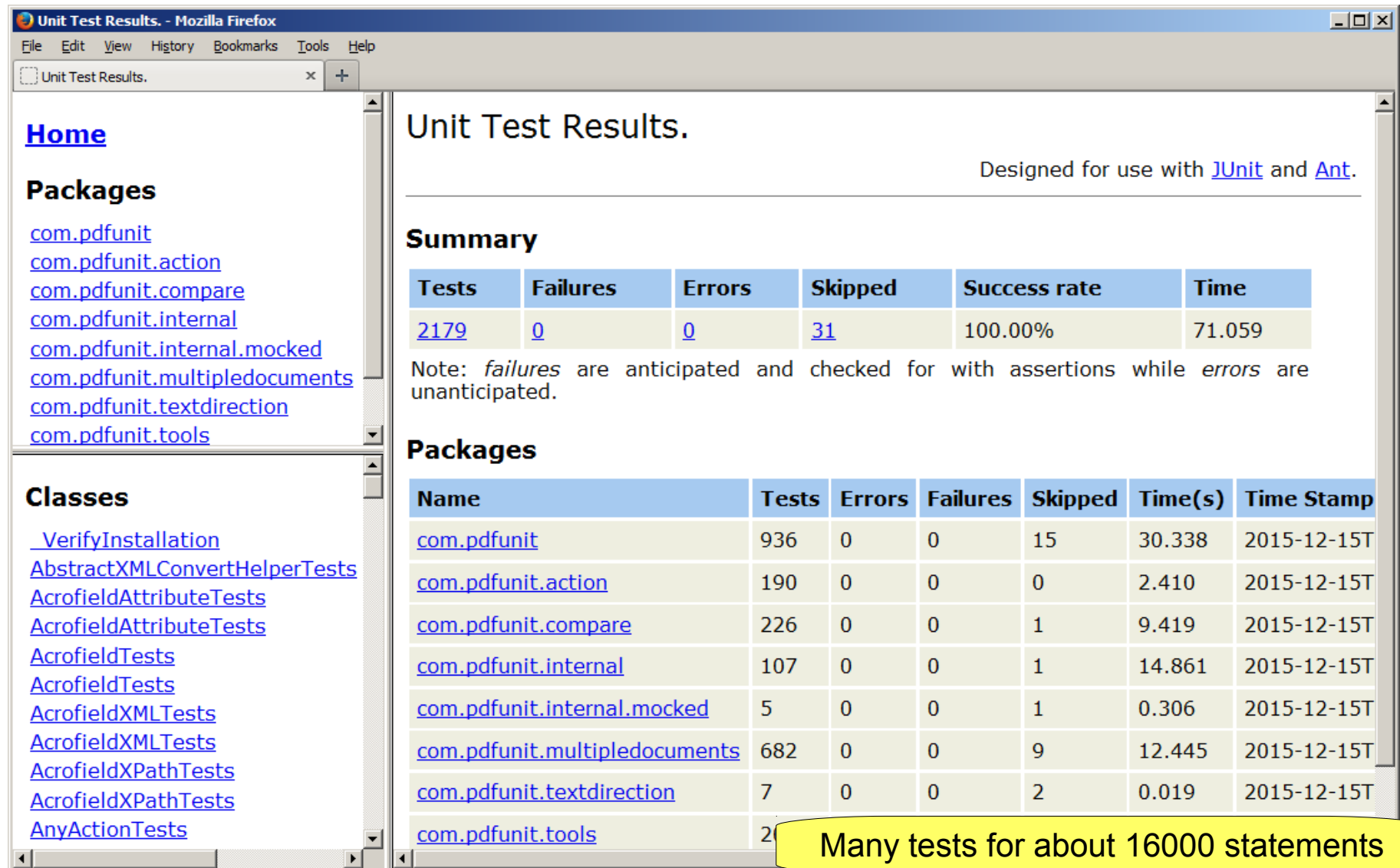
This document provides an overview of the various reports that are automatically generated by [Maven](#). Each report is briefly described below.

Overview

Document	Description
Source Xref	HTML based, cross-reference version of Java source code.
Test Source Xref	HTML based, cross-reference version of Java test source code.
Test JavaDocs	Test JavaDoc API documentation.
JavaDocs	JavaDoc API documentation.
Dependency Analysis	Dependency analysis of the project (used declared, used undeclared, unused declared)
JDepend	JDepend traverses Java class file directories and generates design quality metrics for each Java package. JDepend allows you to automatically measure the quality of a design in terms of its extensibility, reusability, and maintainability to manage package dependencies effectively.
PMD Report	Verification of coding rules.
CPD Report	Duplicate code detection.
Checkstyle	Report on coding style conventions.
Checkstyle	Report on coding style conventions.
Surefire Report	Report on the test results of the project.
Cobertura Test	Cobertura Test Coverage Report.

PDFUnit itself is 'designed for change' with a high level of quality checks. So it can be extended quickly and safely.

Inside PDFUnit-Java – Many Tests



Unit Test Results. - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Unit Test Results.

Home

Packages

- [com.pdfunit](#)
- [com.pdfunit.action](#)
- [com.pdfunit.compare](#)
- [com.pdfunit.internal](#)
- [com.pdfunit.internal.mocked](#)
- [com.pdfunit.multipledocuments](#)
- [com.pdfunit.textdirection](#)
- [com.pdfunit.tools](#)

Classes

- [VerifyInstallation](#)
- [AbstractXMLConvertHelperTests](#)
- [AcrofieldAttributeTests](#)
- [AcrofieldAttributeTests](#)
- [AcrofieldTests](#)
- [AcrofieldTests](#)
- [AcrofieldXMLTests](#)
- [AcrofieldXMLTests](#)
- [AcrofieldXPathTests](#)
- [AcrofieldXPathTests](#)
- [AnyActionTests](#)

Unit Test Results.

Designed for use with [JUnit](#) and [Ant](#).

Summary

Tests	Failures	Errors	Skipped	Success rate	Time
2179	0	0	31	100.00%	71.059

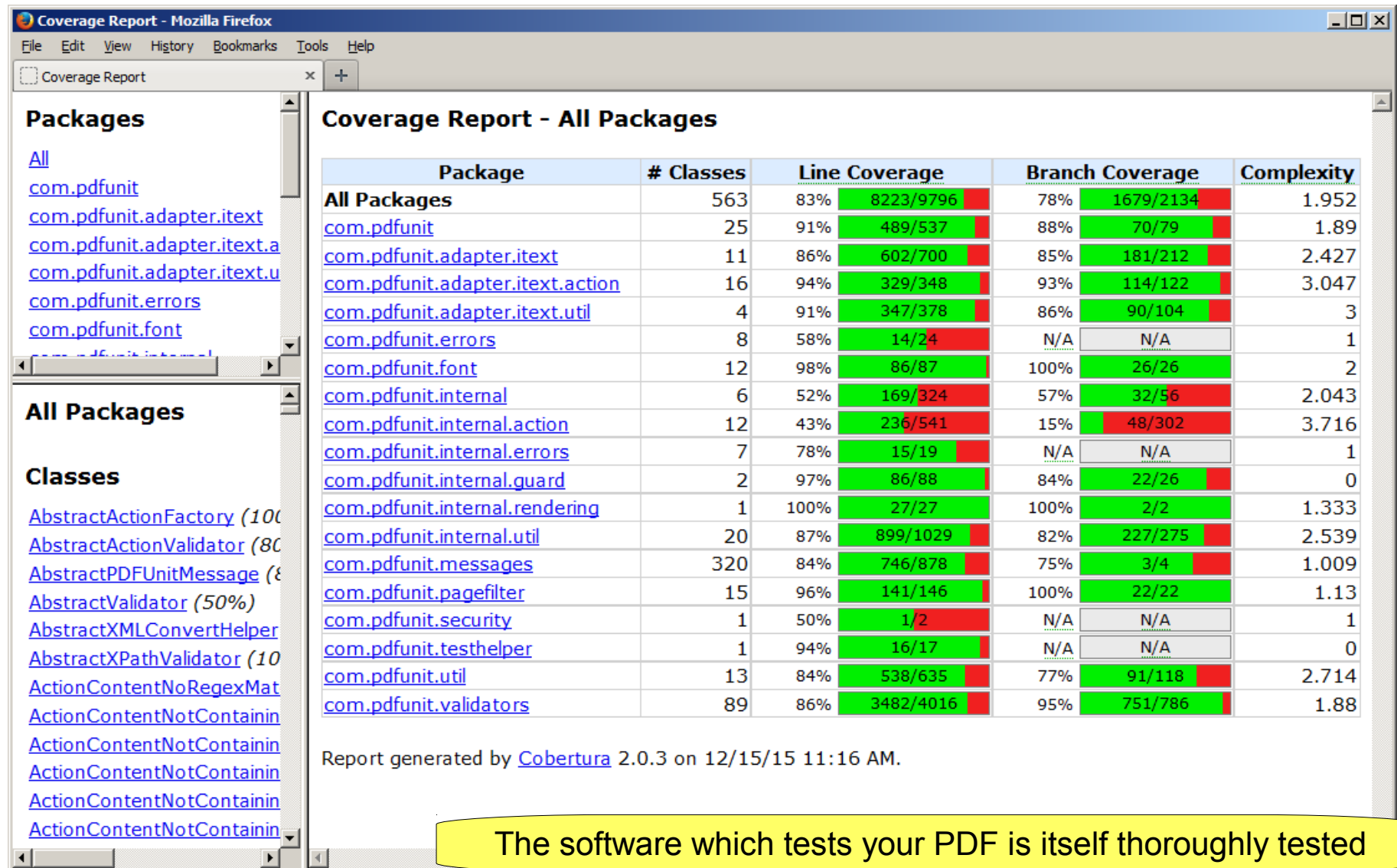
Note: *failures* are anticipated and checked for with assertions while *errors* are unanticipated.

Packages

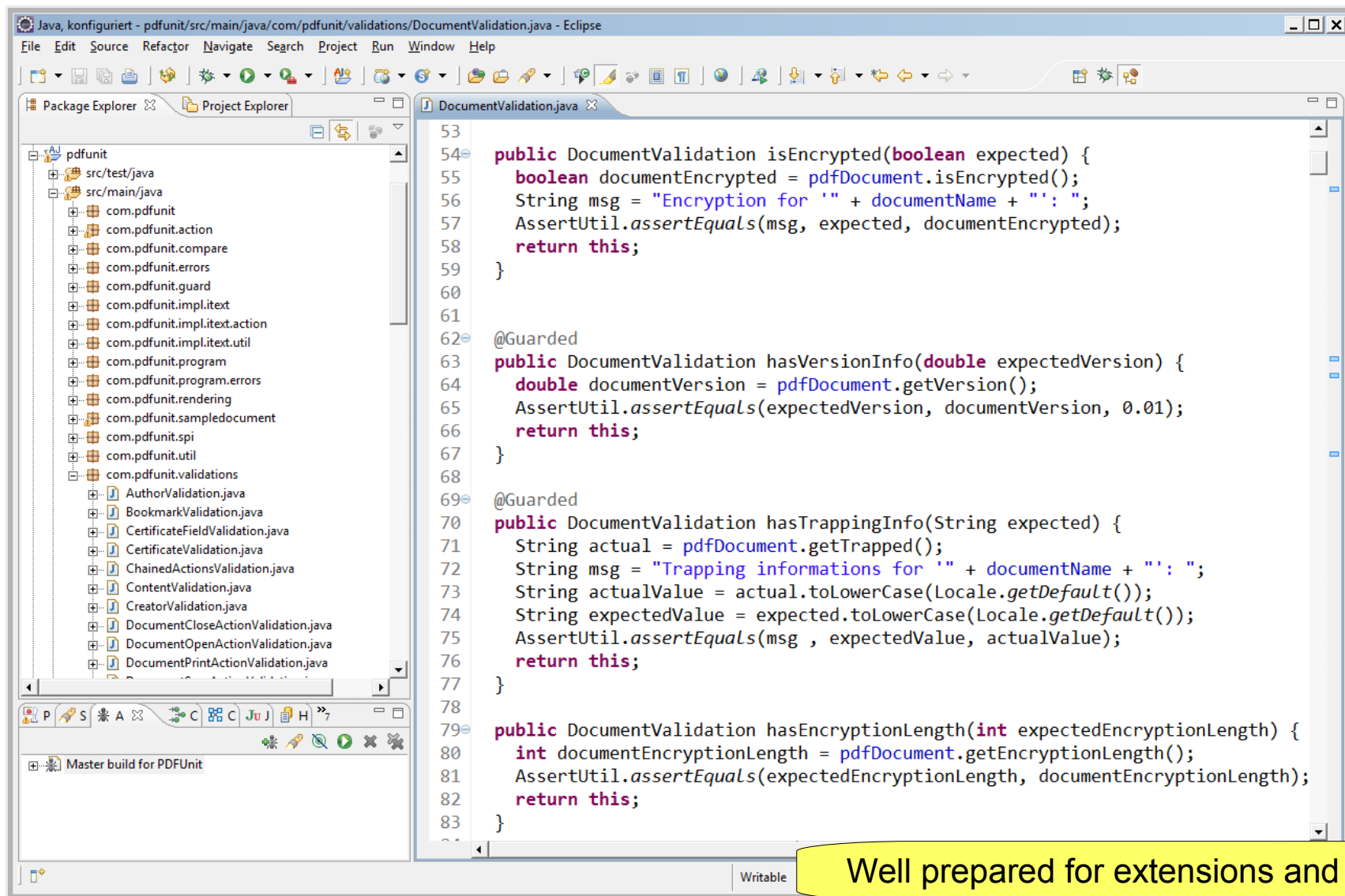
Name	Tests	Errors	Failures	Skipped	Time(s)	Time Stamp
com.pdfunit	936	0	0	15	30.338	2015-12-15T
com.pdfunit.action	190	0	0	0	2.410	2015-12-15T
com.pdfunit.compare	226	0	0	1	9.419	2015-12-15T
com.pdfunit.internal	107	0	0	1	14.861	2015-12-15T
com.pdfunit.internal.mocked	5	0	0	1	0.306	2015-12-15T
com.pdfunit.multipledocuments	682	0	0	9	12.445	2015-12-15T
com.pdfunit.textdirection	7	0	0	2	0.019	2015-12-15T
com.pdfunit.tools	2					

Many tests for about 16000 statements

Inside PDFUnit-Java – Good Test Coverage



Inside PDFUnit-Java – Clean Code



Well prepared for extensions and fixes

Contact

- Contact us for further information:

PDFUnit.com
Carsten Siedentop
Leostr. 41
51145 Cologne
Germany

Phone: +49 (0)178 7997141
Mail: info@pdfunit.com

- Website www.pdfunit.com

Thank you.