

5 minutes for Maven

Arnaud Nauwynck

arnaud.nauwynck@gmail.com

First Step Using Maven

- Download + Unzip from maven.apache.org
- setup Environment variables
 - JAVA_HOME
 - PATH
- <http://maven.apache.org/guides/getting-started>
- Create the first “pom.xml” project file
 - mvn archetype:generate
 - Enter packaging, groupId, artifactId, version

Project Archetype (Minimalist Hello World)

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>an</groupId>          ← GroupId = domain name / directory / package name
  <artifactId>myTestSpring</artifactId> ← Artifact = name of result built file
  <packaging>jar</packaging>        ← Packaging = type of project: pom, jar, war, ear, ...
  <version>1.0-SNAPSHOT</version>    ← Version
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency> <groupId>junit</groupId> <artifactId>junit</artifactId> <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Step 2 : Customize Project

- change general purpose info
- customize project layout ? ... default is ok :
 - src/main/java
 - src/test/java
 - ... target/classes
- add dependencies
- add maven plugins

Commands to Know

- build all and install in local repository:
 - mvn install
- (compile only: mvn compile, test only: mvn test)
- generate eclipse project:
 - mvn eclipse:eclipse
- Deploy installed artifact:
 - mvn deploy
- build web site with all html reports:
 - mvn site

Adding Dependency

- Ex: adding dependency... Logback
- Search in Nexus...
<http://repository.sonatype.org>
- Copy & Paste <dependency> in pom.xml

```
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-core</artifactId>
  <version>0.9.15</version>
</dependency>
```

Nexus : Maven Repository

Sonatype Nexus™ Professional Edition Log In
Version 1.4.0 Professional

Sonatype™ Servers Nexus

Artifact Search Advanced Search

Views/Repositories Repositories

Help

Welcome Search

Keyword Search ▾ logback

Source Index Group Artifact Version Packaging Classifier

Central Proxy (Cac ch.qos.logback) log4j-bridge
Central Proxy (Cac ch.qos.logback) log4j-bridge 0.9.1
Central Proxy (Cac ch.qos.logback) logback-classic 0.9.17 jar
Central Proxy (Cac ch.qos.logback) logback-classic 0.9.15 jar
Central Proxy (Cac ch.qos.logback) logback-classic 0.9.15 iar sources
Central Proxy (Cac ch.qos.logback) logback-classic 0.9

Displaying 13 records Clear Results

Artifact Information

Group: ch.qos.logback XML: <dependency>

Artifact: logback-classic </groupId>ch.qos.logback</groupId>

Version: 0.9.17 </artifactId>logback-classic</artifactId>

Download: pom, artifact </version>0.9.17</version>

</dependency>

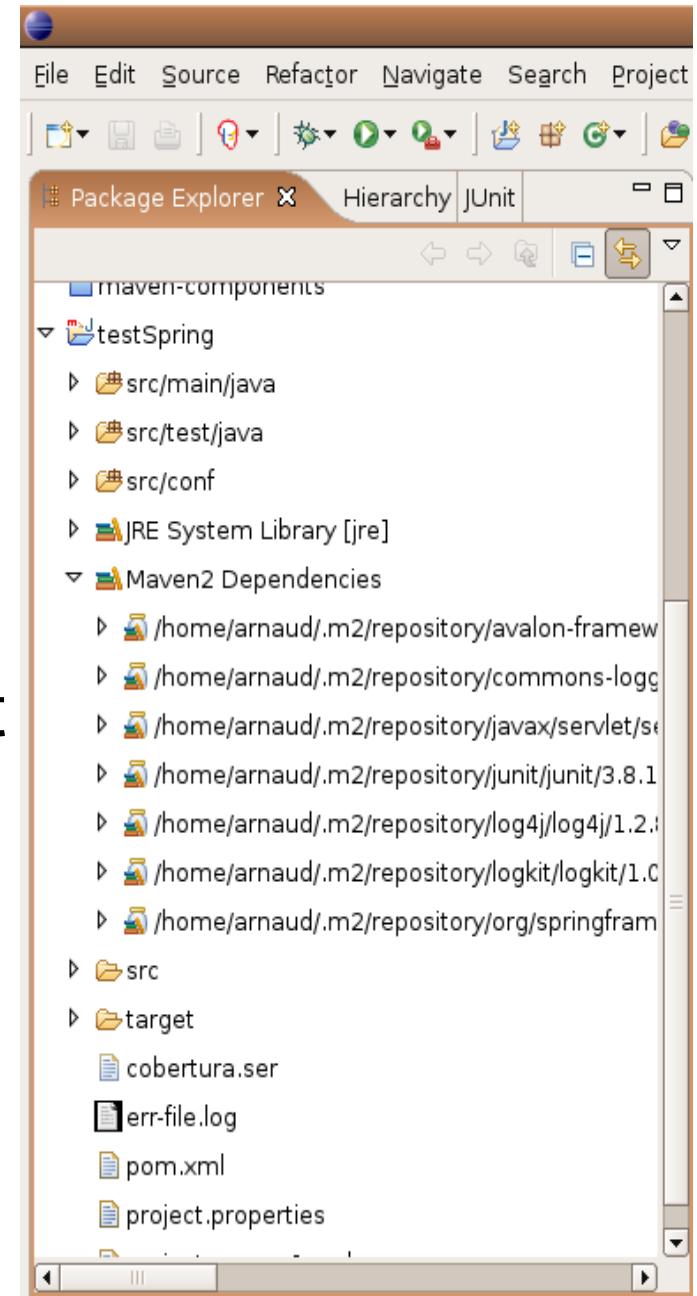
Search library by group/artifact

select results with version

Copy & Paste in pom.xml <dependencies>

Downloading Dependency

- mvn install
 - => auto download dependency in local repository!
- re-generate eclipse project:
 - mvn eclipse:eclipse
 - in eclipse, do “refresh” on project
 - check M2_REPO var is set



Maven Site, Reports

- mvn site:site
 - by default... very simple site, with project info only

testjsp Maven Webapp

Last Published: 2009-12-09 [testjsp Maven Webapp](#)

Project Documentation

- ▼ Project Information
 - [About](#)
 - [Continuous Integration](#)
 - [Dependencies](#)
 - [Issue Tracking](#)
 - [Mailing Lists](#)
 - [Plugin Management](#)
 - [Project License](#)
 - [Project Summary](#)
 - [Project Team](#)
 - [Source Repository](#)

Built by: 

About testjsp Maven Webapp

There is currently no description associated with this project.

- You can configure <reports> in pom.xml
 - Junit, TestCoverage, FindBugs, CheckStyle!!
- or use Sonar... (prepackaged + dashboard)

List of Reports....

- Hundreds of maven plugins, and reports...

The screenshot shows a Mozilla Firefox browser window with the title "MyTestSpring - Generated Reports - Mozilla Firefox". The address bar displays the URL "file:///mnt/hda2/home/arnaud/perso/devPerso/testSpring/target/site/project-reports.html". The page content is titled "MyTestSpring" and indicates it was last published on "10/05/2006". A sidebar on the left is titled "Project Documentation" and includes a "Project Reports" section with links to Checkstyle, Cobertura Test Coverage, CPD Report, FindBugs Report, JavaDocs, JDepend, Maven Surefire Report, PMD Report, Source Xref, and Tag List. A "built by: maven" logo is also present. The main content area is titled "Generated Reports" and contains a paragraph stating: "This document provides an overview of the various reports that are automatically generated by Maven. Each report is briefly described below." Below this is a section titled "Overview" which lists the following reports with their descriptions:

Document	Description
Checkstyle	Report on coding style conventions.
Cobertura Test Coverage	Cobertura Test Coverage Report.
CPD Report	Duplicate code detection.
FindBugs Report	Generates a source code report with the FindBugs Library.
JavaDocs	JavaDoc API documentation.
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.
Maven Surefire Report	Report on the test results of the project.
PMD Report	Verification of coding rules.
Source Xref	HTML based, cross-reference version of Java source code.
Tag List	Report on various tags found in the code.

Reports, Sonar

Sonar - Iceweasel

File Edit View History Bookmarks Tools Help

http://localhost:9000/project/index/fr.an:HelloWorld

ApacheTc tomcat cc Sonar

Sonar Home Search projects HelloWorld Configuration Log out

Version 1.0-SNAPSHOT on Thu, 10 Sep 2009 07:47 , using profile Sonar way.

Dashboard

- Violations drilldown
- Time machine
- Clouds
- Hotspots
- Settings

sonar

Lines of code **Comments** **Duplications**

8 **11.1%** **0.0%**

13 lines 1 lines 0 lines
1 packages 50.0% docu. API 0 blocks
1 classes 0 files
1 methods 0 statements
0 accessors

Complexity

1.0 / method **1.0** / class
1 cmplx 1 statements

Methods **Classes**

Rules compliance **Violations**

25.0% **2**

Rel. Usa. Eff.
Por. Mai.

- Blocker 0
- Critical 0
- Major 2
- Minor 0
- Info 0

Code coverage **Test success**

0.0% **100.0%**

1 tests 0 failures
235 ms 0 errors

[Add a measure](#)

Events All

2009-09-10 Version 1.0-SNAPSHOT [Edit](#) [Delete](#)

[Add an event](#)

Key : fr.an:HelloWorld
Language : java

[Home](#)

Components

fr.an

Size Color 0% 100%

Lines of code Rules compliance

[Set as default](#)

Powered by SonarSource - Open Source [LGPL](#) - v.1.10.1 - [Plugins](#) - [Documentation](#) - [Ask a question](#) - [Bug/feature request](#)

Done

Junit tests

- By default, a maven project ALWAYS run all Junit tests !!
- The plugin is called “surefire”
- To run test only:
 - mvn test
- default config : put all Junit tests in
 - src\test\java
- (default in “pom.xml”)

```
<build>
  <sourceDirectory>src/main/java</sourceDirectory>
  <testSourceDirectory>src/test/java</testSourceDirectory>
  ...
</build>
```

Junit results

- maven logs:

```
[INFO] [surefire:test]
[INFO] Surefire report directory: .../target/surefire-reports
```

```
-----  
TESTS  
-----
```

```
.....
```

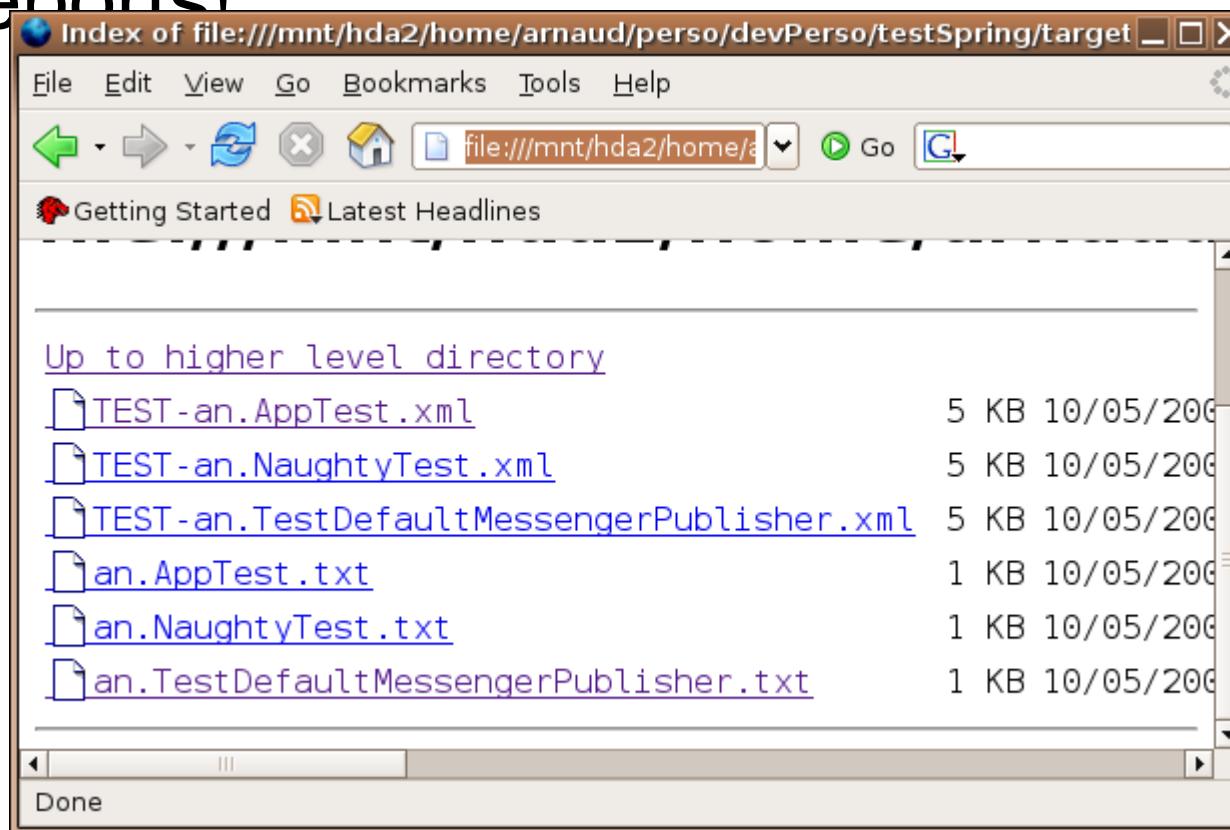
Results :

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

```
[INFO] -----  
[INFO] BUILD SUCCESSFUL  
[INFO] -----  
[INFO] Total time: 13 seconds  
[INFO] Finished at: Thu Oct 05 22:19:25 CEST 2006  
[INFO] Final Memory: 3M/7M  
[INFO] -----
```

Junit Results

- results : file:///...../target/surefire-reports
- By default: xml and text files
not a friendly Html report... must be configured in reports!



Junit Html Reports

- edit the pom.xml :

```
<build>
  ...
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-report-plugin</artifactId>
    </plugin>
  </plugins>
  ...
</build>
```

- consult report in Html:

file:///..../target/site/surefire-reports.html

- (Note: need to configure Xref reports, or disable linkXRef)

Junit Html Reports (bis)

The screenshot shows a Mozilla Firefox browser window displaying a JUnit HTML report. The URL in the address bar is `rso/testSpring/target/site/surefire-report.html`. The page has a navigation bar with links to "Getting Started" and "Latest Headlines". A timestamp at the top right indicates the report was last published on Thursday, October 5, 2006, at 22:49:06 CEST.

Project Documentation sidebar:

- Maven Surefire Report
- built by: maven

Summary

[Summary][Package List][Test Cases]

Tests	Errors	Failures	Success Rate	Time
3	0	0	100%	5.355

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

Package List

[Summary][Package List][Test Cases]

Package	Tests	Errors	Failures	Success Rate	Time
an	3	0	0	100%	5.355

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

an

Class	Tests	Errors	Failures	Success Rate	Time
NaughtyTest	1	0	0	100%	0.129
AppTest	1	0	0	100%	0.087
TestDefaultMessengerPublisher	1	0	0	100%	5.139

Test Cases

[Summary][Package List][Test Cases]

Done

TestCoverage (Cobertura) Plugin

- edit the pom.xml file:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>Cobertura-maven-plugin</artifactId>
      <executions>
        <execution> <goals> <goal>clean</goal> </goals> </execution>
      </executions>
    </plugin>
  </plugins>
</build>

<reporting>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>cobertura-maven- plugin</artifactId>
    </plugin>
  </plugins>
</reporting>
```

Running Cobertura

- for report in html...
 - mvn site

==>

```
[INFO] [site:site]
```

```
.....
```

```
[INFO] Generate "Cobertura Test Coverage" report.
```

```
[INFO] Cobertura 1.7 - GNU GPL License (NO WARRANTY) - See COPYRIGHT file
```

```
Cobertura: Loaded information on 8 classes.
```

```
[cobertura] INFO [main] net.sourceforge.cobertura.reporting.ComplexityCalculator - Cannot find source file  
during CCN computation, source=[an/TestDefaultMessengerPublisher.java]
```

```
[cobertura] INFO [main] net.sourceforge.cobertura.reporting.html.HTMLReport - Could not generate HTML  
file for source file an/TestDefaultMessengerPublisher.java: Cannot find source file,  
name=[an/TestDefaultMessengerPublisher.java]
```

```
Report time: 996ms
```

```
[INFO] Cobertura Report generation was successful.
```

```
.....
```

```
[INFO] Generate "Continuous Integration" report.
```

Cobertura TestCoverage Reports

Coverage Report - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Getting Started Latest Headlines

Packages

All
an
an.ext

Coverage Report - All Packages

Package	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	8	49%	100%	1
an	5	21%	N/A	1
an.ext	3	75%	100%	1

Report generated by Cobertura 1.7 on 10/5/06 10:01 PM.

All Packages

Classes

[DefaultMessengerEventPubl](#)
[Main \(0%\)](#)
[MessengerEvent \(100%\)](#)
[MessengerEventListener \(N\)](#)
[MessengerEventLogger \(100%\)](#)
[MessengerEventLogger2 \(100%\)](#)
[MessengerEventPublisher \(100%\)](#)
[TestDefaultMessengerPublis](#)

Done

FindBugs Report

MyTestSpring - FindBugs Bug Detector Report - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Getting Started Latest Headlines

MyTestSpring

Last Published: 10/05/2006 MyTestSpring

Project Documentation

- ▶ Project Information
- ▼ Project Reports
 - Checkstyle
 - Cobertura Test
 - Coverage
 - CPD Report
- FindBugs Report**
- JavaDocs
- JDepend
- Maven Surefire
- Report
- PMD Report
- Source Xref
- Tag List

built by:  maven

FindBugs Bug Detector Report

The following document contains the results of [FindBugs Report](#)

FindBugs Version is 1.0.0

Threshold is *Low*

Effort is *Default*

Files

© 2006

Done

Continuous Integration

- On Commit in SCM => Auto Compile in Maven
- Use Continuous Integration Server : Hudson

Hudson

search ?

[DISABLE AUTO REFRESH](#)

[New Job](#) [add description](#)

[Manage Hudson](#)

[People](#)

[Build History](#)

[Project Relationship](#)

[Check File Fingerprint](#)

Tous +

S	W	Job ↓	Last Success	Last Failure	Last Duration
		helloworld	7 hr 3 min (#8)	7 hr 53 min (#2)	1 min 52 sec

Icon: [S](#) [M](#) [L](#)

[Legend](#) for all for failures for just latest builds

Build Queue
No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

Page generated: Sep 29, 2009 5:34:09 AM [Hudson ver. 1.323](#)

SCM Commit => Maven Build

The screenshot illustrates the process of triggering a Maven build via an SCM commit. It shows two main views: a terminal window at the top and a Hudson interface below.

Terminal Window:

```
29 sept. 2009 05:45:27 hudson.triggers.SCMTrigger$Runner run
INFO: SCM changes detected in helloworld. Triggering #11
```

Hudson Interface:

- Build Queue:** Shows a single build item: "helloworld". A red arrow points from this item to the "Build Executor Status" table.
- Build Executor Status:** Shows the status of two executors:

#	Status
1	Idle
2	Idle

A tooltip message "In the quiet period. Expires in 25 sec" is displayed over the first executor.
- Build Queue (Right Panel):** Shows the build queue after the build has started. It displays the message "No builds in the queue." and the "Build Executor Status" table, which now includes the active build:

#	Status
1	Building helloworld #10
2	Idle

The build progress bar for executor 1 is partially filled.

Hudson Project Dashboard

Hudson

[Hudson](#) > [helloworld](#)

[Back to Dashboard](#) [Status](#) [Changes](#) [Workspace](#) [Build Now](#) [Delete Project](#) [Configure](#) [Modules](#) [Subversion Polling Log](#)

[Build History](#) ([trend](#))

- #8 Sep 28, 2009 10:30:51 PM
- #7 Sep 28, 2009 10:07:00 PM
- #6 Sep 28, 2009 9:53:20 PM
- #5 Sep 28, 2009 9:51:21 PM
- #4 Sep 28, 2009 9:50:22 PM
- #3 Sep 28, 2009 9:42:55 PM
- #2 Sep 28, 2009 9:40:37 PM
- #1 Sep 28, 2009 9:40:26 PM

Project helloworld

[add description](#)

Test Result Trend

count

0 1

0 2 4 6 8 10 12 14 16 18 20

(just show failures) [enlarge](#)

[Workspace](#) [Recent Changes](#) [Latest Test Result \(no failures\)](#)

search [?](#)

[DISABLE AUTO REFRESH](#)

Conclusion

- only a very very short introduction to maven
- Maven is much much more
- Modern Java Project Tools =
SCM + Eclipse + Maven + Hudson + Nexus +...