Quick answers to common problems

Apache Maven 3 Cookbook

Over 50 recipes towards optimal Java software engineering with Maven 3

Srirangan

[PACKT] open source* PUBLISHING
community experience distilled

Apache Maven 3 Cookbook

Over 50 recipes towards optimal Java software engineering with Maven 3

Srirangan

[PACKT] open source
community experience distilled
PUBLISHING

BIRMINGHAM - MUMBAI

Apache Maven 3 Cookbook

First published: August 2011

Production Reference: 1180811

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-849512-44-2

www.packtpub.com

Cover Image by Parag Kadam (paragvkadam@gmail.com)

Credits

Author
Srirangan

Reviewer
Carsten Ziegeler
Emmanuel Venisse

Acquisition Editor
Sarah Cullington

Development Editor
Chris Rodrigues

Technical Editor
Priyanka S

Copy Editor
Leonard D'Silva

Project Coordinator
Michelle Quadros

Proofreader
Lisa Brady

Indexer
Hemangini Bari

Graphics
Nilesh Mohite

Production Coordinator
Aparna Bhagat

Cover Work
Aparna Bhagat

About the Author

Srirangan is a passionate programmer with nine years of experience in freelance, open source, and Enterprise. He has executed projects in a broad range of technologies including Python, PHP, Scala, Java, Adobe Flex, HTML5, Javascript, and so on.

He is the creator of Review19 (`http://www.review19.com`); an innovative, real-time Agile team collaboration and project management tool. He is also involved with India Defence (`http://www.indiadefence.in`) which is India's largest web property dedicated to the defense sector.

Sri is a senior consultant in Inphina Technologies (`http://www.inphina.com`), a rapidly expanding, high-end technology startup in New Delhi focusing on cloud computing (Google App Engine, Hadoop) and emerging technologies.

He is an enthusiastic open source contributor and his open source projects are available on GitHub and BitBucket:

`https://github.com/Srirangan`

`https://bitbucket.org/srirangan`

To know more you can also visit the following links:

Blog - `http://srirangan.net`

Twitter - `http://twitter.com/srirangan`

LinkedIn - `http://www.linkedin.com/in/srirangan`

About the Reviewer

Carsten Ziegeler is a senior developer and software architect for JEE and portal applications at Adobe Systems. He is a member of the Apache Software Foundation and has been participating for more than twenty years in several open source projects. Carsten is a member of several Apache communities and project management committees like Felix, Sling, and Portals. In addition, Carsten is frequently writing articles, reviewing books, and presenting at various conferences.

Emmanuel Venisse has been developing, architecturing, and integrating J2EE applications for thirteen years for banks, government, holiday company projects, and so on. For the last six years, he has worked as a freelancer. For the last eight years, he's been working, in his spare time, on Apache Maven, Continuum and Archiva projects as a core developer and he's also the Continuum project leader. He has contributed to the majority of books written about Apache Maven.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

Apache Maven is more than just build automation. When positioned at the very heart of your development strategy, Apache Maven can become a force multiplier not just for individual developers but for Agile teams and managers. This book covers implementation of Apache Maven with popular enterprise technologies/frameworks and introduces Agile collaboration techniques and software engineering best practices integrated with Apache Maven.

The *Apache 3 Maven Cookbook* is a real-world collection of step-by-step solutions for individual programmers, teams, and managers to explore and implement Apache Maven and the engineering benefits it brings into their development processes.

This book helps with the basics of Apache Maven and with using it to implement software engineering best practices and Agile team collaboration techniques. It covers a broad range of emergent and enterprise technologies in the context of Apache Maven, and concludes with recipes on extending Apache Maven with custom plugins.

We look at specific technology implementations through Apache Maven, including Java Web Applications, Enterprise Java Frameworks, cloud computing, mobile / device development, and more. We also look at Maven integration with popular IDEs, including Eclipse, NetBeans, and IntelliJ IDEA.

The book is rounded off by exploring how to extend the Apache Maven platform by building custom plugins, integrating them with existing projects, and executing them through explicit command-line calls or with Maven build phases.

What this book covers

Chapter 1, *Basics of Apache Maven*, assists you while you take your first steps with Apache Maven. It will cover setting up Apache Maven on your operating system, verifying the setup, and getting started with your first Apache Maven project.

Chapter 2, *Software Engineering Techniques*, introduces us to implementing software engineering best practices and techniques (such as Test Driven Development, build automation, dependency management, and so on) with Apache Maven.

Chapter 3, *Agile Team Collaboration*, helps you implement collaboration mechanisms with Apache Maven that aid team and distributed development.

Chapter 4, *Reporting and Documentation*, helps you generate reports and documentation, code quality, test coverage reports, and other related metrics and bundles them in a Maven site.

Chapter 5, *Java Development with Maven*, helps you take on web application and enterprise application development challenges with Java while leveraging popular frameworks including Spring and Hibernate.

Chapter 6, *Google Development with Maven*, gets you up and running with Android, Google App Engine, Google Web Toolkit development, and testing with Apache Maven.

Chapter 7, *Scala, Groovy, and Flex*, discusses these popular upcoming technologies and frameworks and gets you started on these with Apache Maven.

Chapter 8, *IDE Integration*, looks at working with Apache Maven directly from your IDE without having to switch back to the terminal.

Chapter 9, *Extending Apache Maven*, shows you how to extend Apache Maven features by development in Java, Apache Ant, and JRuby.

What you need for this book

Apache Maven is based on the Java platform, thus the main requirement is the Java SDK. Recipes in the first chapter guide you on how to set up the Java SDK on your machine. Other than this, Apache Maven automatically downloads all dependencies during execution. Make sure you have fairly good internet access available while working with Apache Maven.

For coding popular IDEs such as IntelliJ, IDEA, Eclipse, and NetBeans can be used. I prefer and recommend IntelliJ IDEA, but even a text editor such as (Scite, TextMate, or Notepad++) should be good enough.

Who this book is for

This book is for Java developers, teams, and managers who want to implement Apache Maven in their development process, leveraging the software engineering best practices and Agile team collaboration techniques it brings along. The book is also specifically for the developer who wishes to get started in Apache Maven and use it with a range of emergent and enterprise technologies including Enterprise Java, Frameworks, Google App Engine, Android, and Scala.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "The `archetype: generate` command would have generated a sample Apache Maven project for us".

A block of code is set as follows:

```
<mirrors>
  <mirror>
    <id>TestRepository</id>
    <name>My test repository</name>
    <url>http://localhost:8080/nexus-webapp-
      1.8.0/content/repositories/TestRepository/</url>
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
<groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <configuration>
        <skip>true</skip>
      </configuration>
```

Any command-line input or output is written as follows:

```
$ mvn install -Dmaven.test.skip=true
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Enter the administrative **Username** and **Password**."

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on `www.packtpub.com` or e-mail `suggest@packtpub.com`.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.PacktPub.com`. If you purchased this book elsewhere, you can visit `http://www.PacktPub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Basics of Apache Maven

In this chapter, we will cover:

- Setting up Apache Maven on Windows
- Setting up Apache Maven on Linux
- Setting up Apache Maven on Mac
- Verifying the Apache Maven installation
- Creating a new project
- Compiling and testing a project
- Understanding the Project Object Model
- Understanding the build lifecycle
- Understanding build profiles

Apache Maven originated as an attempt to simplify the build process for the now defunct Apache Jakarta Alexandria project. Its formative years were then spent in the Apache Turbine project where it eventually came to replace a brittle and fragile build system based on Apache ANT.

Given Maven's tremendous potency and effectiveness in solving a majority of our day-to-day challenges, it has become hugely popular and is now widely used not only by developers but by other roles in a team including scrum masters, product owners, and project managers.

In recent years, Maven has clearly emerged as an important force-multiplier for Agile teams and organizations.

On its official website, `http://maven.apache.org`, Apache Maven's objectives are listed as:

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Providing guidelines for best practices in development
- Allowing transparent migration to new features

Whatever the reasons that made you choose Maven (be it build automation, dependency management, standardization, testability, lifecycle management, continuous integration, or any other industry best practice), the recipes in this book will get you up and running in the shortest time possible.

In the upcoming recipes, we will set up Maven on various platforms and host environments followed by selectively exploring the core concepts of the **Project Object Model** and the **Maven build lifecycle**.

Setting up Apache Maven on Windows

We will look at installing and setting up Apache Maven on the Windows operating system. Maven is a command-line tool and needs to be integrated with the Windows environment variables. The process is quite simple and Java dependent.

There is a chance Apache Maven may have been pre-installed on your machine. Verify that Maven isn't already installed before proceeding.

See the recipe *Verifying the Maven installation* in this chapter

Getting ready

As mentioned, a prerequisite for working with Maven is the availability of the Java Development Kit. Make sure that the JDK is available before proceeding. This can be verified by running the following command line:

```
Java -version
```

It will give the following output:

```
java version "1.6.0_21"
Java(TM) SE Runtime Environment (build 1.6.0_21-b06)
Java HotSpot(TM) Client VM (build 17.0-b16, mixed mode, sharing)
```

If you do not have JDK installed, you can download it at `http://www.oracle.com/technetwork/java/javase/downloads/index.html`.

Got JDK? Next you need to get your hands on Maven. You can download it from:

```
http://maven.apache.org/download.html
```

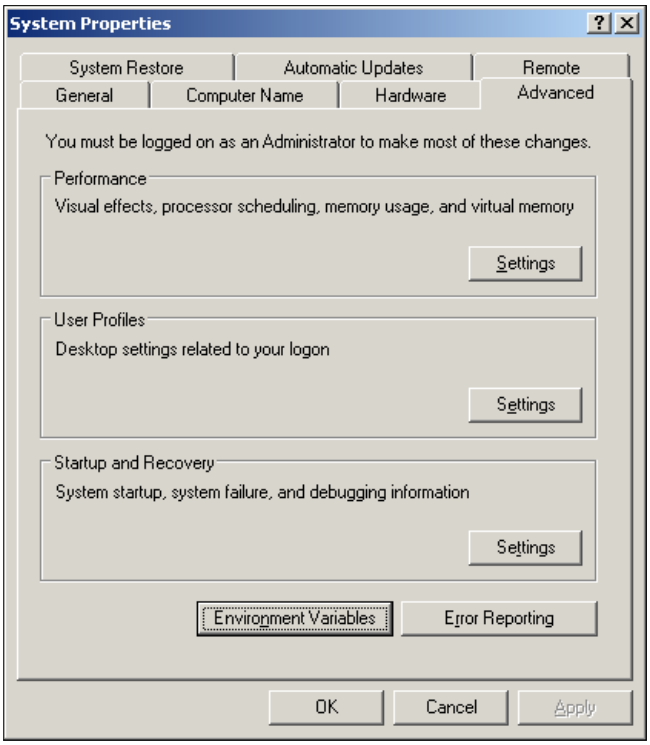
After downloading Maven, extract the archive into any folder. For Windows, it is advised that the path doesn't contain any white-space characters. I extracted Maven in the `D` drive.

```
D:\apache-maven-3.0.2\
```

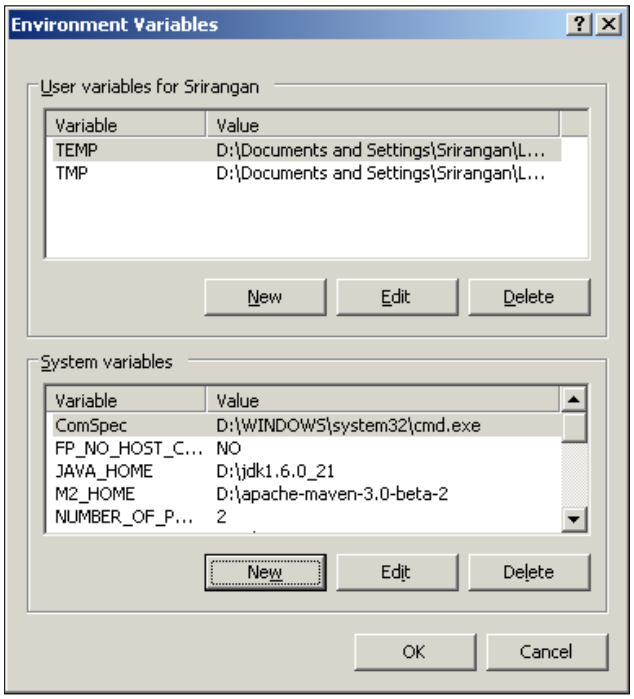
How to do it...

To start using Maven, we need to configure Windows environment variables. The `M2_HOME` variable needs to be set and the `PATH` variable needs to be modified to include the Maven binaries folder.

You can set the environment variables by accessing the **System** settings in the **Control Panel.**



Select **Environment Variables** and then the **New** button to create a new environment variable.



Create a new environment variable for `M2_HOME` pointing to the Maven base directory. For me, the value of `M2_HOME` will be `D:\apache-maven-3.0.2\`.

The `PATH` environment variable will already exist. Select it and click **Edit** to modify. It must be modified by appending the following text at the end:

```
;%M2_HOME%\bin
```

Apache Maven is now ready and available for use in the command line. It is also available for integration with IDEs and other development tools, but more on that in upcoming chapters.

See also

- *Setting up Apache Maven on Linux* in this chapter
- *Setting up Apache Maven on Mac* in this chapter
- *Verifying your Apache Maven installation* in this chapter
- *Working with Eclipse and Maven* in *Chapter 8*, *IDE Integration*
- *Working with NetBeans and Maven* in *Chapter 8*, *IDE Integration*
- *Working with IntelliJ and Maven* in *Chapter 8*, *IDE Integration*

Setting up Apache Maven on Linux

The Linux distribution used in this book is Ubuntu 10.04. If you use any other Linux distribution, the steps should nevertheless remain similar. Apache Maven is a command-line tool; it just needs to be extracted and configured with the operating system's environment.

There is a chance Apache Maven may have been pre-installed on your machine. Verify that Maven isn't already installed before proceeding.

See the recipe *Verifying the Maven installation* in this chapter

Getting ready

A prerequisite for working with Maven is the availability of the Java Development Kit. Make sure the JDK is available before proceeding. This can be verified by running the command line:

```
java -version
```

It should give the following output:

```
java version "1.6.0_18"
OpenJDK Runtime Environment (IcedTea6 1.8.1) (6b18-1.8.1-0ubuntu1)
OpenJDK Client VM (build 16.0-b13, mixed mode, sharing)
```

If you do not have JDK installed, you can download it at: `http://www.oracle.com/technetwork/java/javase/downloads/index.html`.

Got JDK? Next you need to get your hands on Maven. You can download it from: `http://maven.apache.org/download.html`.

After downloading Maven, extract the archive into a folder. For example, I extracted Maven into my home folder, `/home/srirangan/apache-maven-3.0.2/`. This will be an installation for a single user.

How to do it...

The next step is to add commands for exporting the `PATH` and `M2_HOME` environment variables in the `.bashrc` file. This file can be found in the user's home folder, which for me is `/home/srirangan/.bashrc`.

```
export M2_HOME=/home/srirangan/apache-maven-3.0.2
export PATH=${PATH}:${M2_HOME}/bin
```

Apache Maven is now ready and available for use in the command line. It is also available for integration with IDEs and other development tools, but more on that in upcoming chapters.

See also

- *Setting up Apache Maven on Windows* in this chapter
- *Setting up Apache Maven on Mac* in this chapter
- *Verifying your Apache Maven installation* in this chapter
- *Working with Eclipse and Maven* in *Chapter 8*, *IDE Integration*
- *Working with NetBeans and Maven* in *Chapter 8*, *IDE Integration*
- *Working with IntelliJ and Maven* in *Chapter 8*, *IDE Integration*

Setting up Apache Maven on Mac

Installing Maven on the Mac OS X isn't very different from the installation and setup on Linux. This really shouldn't be a surprise because OS X is built on top of BSD Linux in the first place. Apache Maven is a command-line tool; it just needs to be extracted and configured with the operating system's environment.

There is a chance Apache Maven may have been pre-installed on your machine. Verify that Maven isn't already installed before proceeding.

See the recipe *Verifying the Maven installation* in this chapter

Getting ready

A pre-requisite for working with Maven is the availability of the Java Development Kit. Make sure the JDK is available before proceeding. This can be verified by running the following command line:

```
java -version
```

It should give the following output:

```
java version "1.5.0_19"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_19-b02-306)
Java HotSpot(TM) Client VM (build 1.5.0_19-138, mixed mode, sharing)
```