

杨世文 孙会军 编著

Maven 应用实战

Java项目构建
热门工具

+

QQ
在线答疑

+

配套
源代码

清华大学出版社



杨世文 孙会军 编著

Maven

应用实战

常州大学图书馆
藏书章

清华大学出版社
北京

内 容 简 介

Maven 是 Java 项目构建工具,由资深 Java 讲师结合多年的教学经验编写,是为数不多的帮助程序员从零开始认识 Maven,使用 Maven,再到熟练掌握 Maven 的辅导书。全书总体分成三个层次:Maven 的安装使用、Maven 的核心概念和运行原理以及 Maven 的高级应用。其中通过穿插案例,介绍了 Maven 的安装与 Eclipse 的集成配置,搭建 Archiva 服务器的方法,Maven 的架构、运行生命周期、仓库、依赖和插件,基于 Maven 生成项目站点、生成项目报告文档和软件测试等内容。全书以实践为宗旨,一切源于实践,又回归于实践。

本书适合 Java 程序员和项目经理阅读,也可作为相关领域的培训教材和业余爱好者的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Maven 应用实战/杨世文,孙会军编著. —北京:清华大学出版社,2018

ISBN 978-7-302-48582-7

I. ①M… II. ①杨… ②孙… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2017)第 249864 号

责任编辑:张龙卿

封面设计:常雪影

责任校对:袁 芳

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, cs-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>; 010-62770175-4278

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:15

字 数:344 千字

版 次:2018 年 1 月第 1 版

印 次:2018 年 1 月第 1 次印刷

印 数:1~2500

定 价:49.00 元

产品编号:073762-01

前言

写作背景

Maven 是一款由 Apache 软件基金会开发的，用来管理项目的构建，生成报告和文档的 Java 项目管理工具。

关于 Maven 的资料和书籍很多。有的非常简洁，就一本小册子；有的非常详细，厚厚的一本，把各个细节都阐述得面面俱到。综观所有的文献资料，以项目实践为需求导向，能剔除不用过多理解的概念，指导读者快速在项目中上手使用 Maven 的书籍还是凤毛麟角。

要知道，很多程序员在开发项目时，经常会被许多零碎的资料困扰很久，从而感到手足无措、焦头烂额。他们需要的是一看就明白，就能使用到项目中去的第一手资料。

为了帮助程序员解决这些问题，引导读者学有所用，我们编写了此书。本书的目的就是利用简洁实用的语言，以实际项目为案例，按项目自身发展为线索，介绍 Maven 在项目中每个环节的使用方法，使读者由浅入深地学习使用 Maven。

本书内容

第 1~3 章，介绍 Maven 的作用及其安装配置，并用命令行构建 Maven 项目，体验 Maven 的基本操作。

第 4 章，介绍在 Eclipse 上安装配置 M2Eclipse 插件，并且构建简单的 Maven 项目，体验在 Eclipse 上构建 Maven 项目的过程。

第 5~8 章，详细介绍基于 Eclipse 的 M2Eclipse 插件开发 Web 应用和流行框架，开发企业级 Web 应用。

第 9 章，详细介绍 Maven 构建的生命周期与核心概念。

第 10~13 章，介绍 Maven 在项目中比较常见的使用方法。

本书特色

本书从零基础开始讲解 Maven，然后由浅入深，循序渐进地通过实例指导读者慢慢熟练掌握。

本书的内容是先实践，再理论，最后又归于实践。如果公司的项目马上要启动了，并且对 Maven 的要求不是太高，那么至少可以先使用起来，再慢慢深入了解。

当然，对于那些已经对 Maven 有了基本的了解并打算再深入研究并灵活运用到项目中去去的程序员和项目管理员，也可以借鉴本书后面部分的内容。

读者对象

本书适合以下读者阅读。

从事 Java 编程行业的开发人员和项目管理员。

大中专院校的老师 and 学生。

相关培训机构的老师和学员。

Java 编程爱好者。

读者服务

为了方便读者解决学习过程中遇到的问题，本书提供书中各章配套的开发源代码及相关资源，欢迎通过编著者索取或出版社官方网站下载。

本书由杨世文与孙会军合作编写，杨世文负责统稿，孙会军负责整理资料并调试代码。由于水平有限，书中难免存在不妥之处，敬请广大读者批评、指正，编著者 QQ: 775488842。

编著者

2017年12月

内容目录

目 录

第 1 章 课前准备	1
1.1 项目经理的工作	1
1.2 Maven 的作用	1
第 2 章 开始学习 Maven	2
2.1 Maven 简介	2
2.2 安装 Maven 前的准备	2
2.2.1 下载合适的 JDK 安装软件安装	2
2.2.2 配置 JDK 环境变量	3
2.2.3 测试 JDK 是否安装成功	5
2.3 Maven 的安装与配置	5
第 3 章 使用 Maven 开发第一个案例	8
3.1 创建 Maven 项目	8
3.1.1 使用命令向导一步步创建项目	9
3.1.2 在命令中输入所有必要信息直接创建项目	9
3.2 添加样例代码	10
3.2.1 HelloWorld.java	11
3.2.2 TestHelloWorld.java	11
3.3 编写 Maven 骨架文件	12
3.4 编译和测试	13
3.5 生成站点和报告文档	15
3.5.1 生成站点信息	15
3.5.2 生成 API Doc 文档	16
第 4 章 基于 Eclipse 安装 Maven 插件	18
4.1 搭建 Eclipse+M2Eclipse 的必要性	18
4.2 安装配置 M2Eclipse 插件	18
4.3 创建 Maven 项目	23
4.4 构建 Maven 项目	25

4.5	基于 M2Eclipse 完成所有工作	26
4.5.1	运行测试	27
4.5.2	生成 javadoc API 帮助文档	29
4.5.3	生成站点	30
4.5.4	测试报告	30
第 5 章	基于 Maven 开发 Web 应用	32
5.1	开发 Web 应用的思路	32
5.2	实现 Web 版 HelloWorld	32
5.2.1	安装配置 Web 应用的 Archetype Catalog	32
5.2.2	基于 Archetype 向导创建 Web 工程	34
5.2.3	编写样例代码	35
5.2.4	构建 Web 项目	35
5.2.5	测试	39
5.3	基于 Maven 开发用户模块的 CRUD	39
5.3.1	创建 Web 工程和初始化数据库	39
5.3.2	添加相关依赖	40
5.3.3	添加注册代码	41
5.3.4	构建项目	42
5.3.5	测试	43
第 6 章	开发企业级 Web 应用	44
6.1	企业 Web 应用简介	44
6.2	搭建 Maven 私服	44
6.2.1	下载 Archiva	44
6.2.2	启动服务器	45
6.2.3	初始化和配置 Archiva 服务器	46
6.2.4	在开发人员端配置对私服的使用	48
6.3	实现 Struts2+Spring+Hibernate 框架应用	50
6.3.1	创建公共 POM 模块	51
6.3.2	实现 Hibernate DAO 模块	57
6.3.3	实现 Service 模块	64
6.3.4	实现 Struts2 Web 模块	68
6.3.5	整合成 SSH	73
6.4	实现 SpringMVC+Spring+MyBatis 框架应用	78
6.4.1	创建公共 POM	78
6.4.2	实现 MyBatis DAO 模块	82
6.4.3	实现 Spring 的 Service 层模块	85

6.4.4	实现 SpringMVC Web 模块	86
6.4.5	整合成 SSM	93
第 7 章	生成项目站点	95
7.1	生成基本站点	95
7.1.1	简单站点	95
7.1.2	完善站点信息	97
7.2	添加插件丰富站点信息	99
7.2.1	JavaDoc 插件	99
7.2.2	源代码插件	101
7.2.3	测试报告插件	102
7.2.4	源代码分析插件	104
7.3	个性化站点	105
7.3.1	修饰外观	105
7.3.2	自定义页面	110
7.3.3	国际化	115
7.4	部署站点	116
7.4.1	在 Tomcat 7 上的 DAV 服务	117
7.4.2	设置 Tomcat 7 的用户名和密码	118
7.4.3	配置 Maven 的 DAV 自动部署	118
第 8 章	版本管理	121
8.1	简介	121
8.2	专业术语	121
8.2.1	快照版本	121
8.2.2	发布版本	121
8.2.3	版本管理关系	121
8.2.4	版本号约定	122
8.2.5	主干、分支、标签	122
8.3	自动版本发布	123
8.3.1	准备环境	124
8.3.2	创建仓库	125
8.3.3	创建样例项目	125
8.4	GPG 签名验证	130
8.4.1	第 1 阶段：手动操作 GPG	131
8.4.2	第 2 阶段：基于 Maven 插件使用 GPG	135

第 9 章 Maven 核心概念	138
9.1 简介	138
9.2 生命周期	138
9.2.1 生命周期简介	138
9.2.2 深入生命周期	139
9.2.3 调用生命周期阶段	140
9.3 插件	144
9.3.1 插件的作用和目标	144
9.3.2 插件同生命周期阶段的绑定	144
9.3.3 插件参数配置	147
9.3.4 获取插件信息	149
9.3.5 调用插件	153
9.3.6 解析插件	154
9.4 坐标	156
9.4.1 groupId	157
9.4.2 artifactId	157
9.4.3 version	157
9.4.4 packaging	157
9.4.5 classifier	158
9.5 仓库	158
9.5.1 Maven 仓库的定义	158
9.5.2 仓库的管理方式	159
9.5.3 仓库的种类	160
9.5.4 配置远程仓库	163
9.5.5 快照版本	166
9.5.6 从仓库中解析依赖的机制	167
9.5.7 镜像	167
9.5.8 仓库搜索服务	169
9.6 依赖	169
9.6.1 依赖是什么	169
9.6.2 依赖的配置	170
9.6.3 依赖的范围	170
9.6.4 传递性依赖	172
9.6.5 依赖的调解	173
9.6.6 排除依赖	175
9.6.7 归类依赖	175
9.6.8 优化依赖	177

9.7 继承和聚合	177
第 10 章 Maven 测试	186
10.1 测试简介	186
10.2 测试框架	186
10.2.1 JUnit 单元测试框架	186
10.2.2 TestNG 测试框架	187
10.3 Maven 测试插件	189
10.3.1 Surefire 插件简介	189
10.3.2 跳过测试	190
10.3.3 个性化指定运行测试	191
10.3.4 包含和排除测试	192
10.4 测试报告	192
10.4.1 基本测试报告	193
10.4.2 测试覆盖率报告	194
10.5 重用测试代码	194
第 11 章 灵活构建 Maven 项目	196
11.1 Maven 属性	196
11.1.1 内置属性	197
11.1.2 POM 属性	197
11.1.3 自定义属性	197
11.1.4 Settings 属性	197
11.1.5 Java 系统属性	198
11.1.6 环境变量属性	198
11.2 需要灵活处理的构建环境	198
11.3 资源过滤	198
11.4 Maven 的 profile	200
11.4.1 针对不同环境的 profile 的配置	200
11.4.2 激活 profile 配置	201
11.4.3 profile 的种类	204
11.5 Web 资源过滤	205
11.6 在 profile 中激活集成测试	206
第 12 章 自定义 Maven 插件	208
12.1 自定义 Maven 插件简介	208
12.2 自定义 Maven 插件案例	209
12.3 自定义 Maven 插件的详细说明	215

12.3.1	Mojo 标记	216
12.3.2	Mojo 参数	217
12.4	自定义 Maven 插件中的错误处理和日志	221
第 13 章	Archetype 扩展	222
13.1	Archetype 使用概述	222
13.1.1	maven-archetype-quickstart	223
13.1.2	maven-archetype-webapp	223
13.1.3	AppFuse Archetype	223
13.2	自定义 Archetype	224
13.3	Archetype 数据库	224
13.3.1	Archetype 数据库简介	224
13.3.2	使用本地 Archetype 数据库	225
13.4	在 M2Eclipse 中配置 Archetype Catalogs	226
参考文献		227
附录		228

课前准备

关于 Maven 的资料和书籍很多。有的比较简洁,点到为止;有的是厚厚的一本书,把方方面面都阐述得很详细;还有些是针对某个方面进行说明讲解。

本书的目的和追求:用通俗易懂的语言,以实际项目为案例,按项目自身发展线索,介绍 Maven 在项目中每个环节的使用方法,使读者由浅入深地了解 Maven、运用 Maven,最后实践与理论结合,掌握 Maven。

1.1 项目经理的工作

一支团队在接到公司下达的项目后,项目经理或项目架构师往往要在团队正式介入以前,做好充分的前期准备。具体工作如下。

- (1) 确定系统架构。
- (2) 收集框架相关的 jar 包。
- (3) 搭建 SSM 框架。
- (4) 编写测试代码。
- (5) 寻找框架依赖的 jar 包。
- (6) 剔除冲突 jar 包。
- (7) 制定需求设计文档规范。
- (8) 测试文档和代码规范。
- (9) 相关报告文档规范。

这里面的内容很多、很杂,必须进行充分细致的准备。一支团队在每承接一个项目时,都要重复做类似的事情。这种工作劳动强度大,而且缺少技术含量。

1.2 Maven 的作用

上述问题用 Maven 可以直接解决。

- (1) Maven 统一集中管理好所有的依赖包,不需要程序员再去寻找。
- (2) 对应第三方组件用到的共同 jar, Maven 自动解决重复和冲突问题。
- (3) Maven 作为一个开放的架构,提供了公共接口,方便同第三方插件集成。程序员可以将自己需要的插件,动态地集成到 Maven,从而扩展新的管理功能。
- (4) Maven 可以统一每个项目的构建过程,实现不同项目的兼容性管理。

第2章

开始学习 Maven

2.1 Maven 简介

Maven 是 Apache 开源组织奉献的一个开源项目。Maven 的本质是一个项目管理工具,将项目开发和管理过程抽象成一个项目对象模型(POM)。开发人员只需做一些简单的配置,就可以批量完成项目的构建、报告和文档的生成工作。

当然,Maven 除了是一个优秀的项目构建方面的管理工具外,还有项目管理相关的其他特殊优势。比如,项目相关的第三方依赖包,这是每个 Java 程序员不可回避的问题。

每个老程序员都在自己的计算机里有个专门目录,分类保存过去项目开发过程中使用的第三方 jar 包,需要的时候从里面筛选。新程序员就麻烦了,测试项目的时候,经常会遇到 Class No Found Exception,导致一整天在搜索和重启中度过。

Maven 可以统一管理所有的依赖 jar,甚至是不同的版本。程序员也可以动态地将自己写好的模块打包成 jar 包让它管理。需要的时候,可以直接通过简单的描述文件告诉 Maven,它会自动帮助程序员找出来,集成到项目中。

本书的所有案例都是基于 Maven 3.3.9 版本测试的。

2.2 安装 Maven 前的准备

因为 Maven 本身就是基于 Java 写的,所以在安装配置 Maven 之前,有必要将 Java 的运行环境安装配置好。接下来介绍 Java 的安装和配置过程,有相关基础的读者可以跳过本节。但要注意,Maven 3.3.9 版本要求的 JDK 一定是 JDK 1.7 或以后的版本。

JDK 的安装和配置主要分如下步骤。

2.2.1 下载合适的 JDK 安装软件安装

Oracle 软件的下载链接为 <https://www.oracle.com/downloads/index.html>。

程序员可以在里面找到需要的 JDK 安装包。本书是在 Windows 操作系统中完成的所有案例,所以请下载 Windows 版本。下载时注意匹配跟自己的系统统一的安装包(32 位或 64 位)。现在的计算机基本上是 64 位,所以建议 Java 相关软件统一安装 64 位的。

安装好后的目录结构如图 2-1 所示。

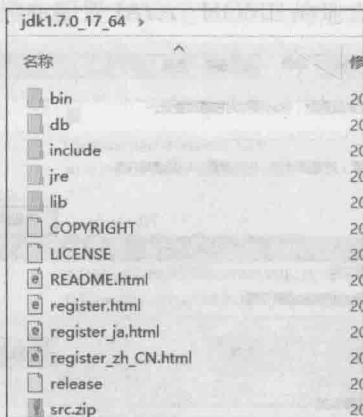


图 2-1 JDK 安装目录

jdk1.7.0_17_64 是安装 JDK 后的总目录,bin、db、include 等是它里面的子目录和文件。

2.2.2 配置 JDK 环境变量

JDK 的环境变量只要配置 JAVA_HOME 和 Path。右击“我的电脑”,选择“属性”命令,出现如图 2-2 所示窗口。



图 2-2 Windows 系统信息

单击左边的“高级系统设置”按钮,出现如图 2-3 所示对话框。

单击“高级”标签中的“环境变量”按钮,显示如图 2-4 所示对话框。

(1) 配置 JAVA_HOME 目录。查看“系统变量”中有没有 JAVA_HOME(之前有配置过,就会有 JAVA_HOME 这项显示)。如果没有,单击“新建”按钮,在弹出的窗口中输入 JAVA_HOME 和 Java 的安装目录。

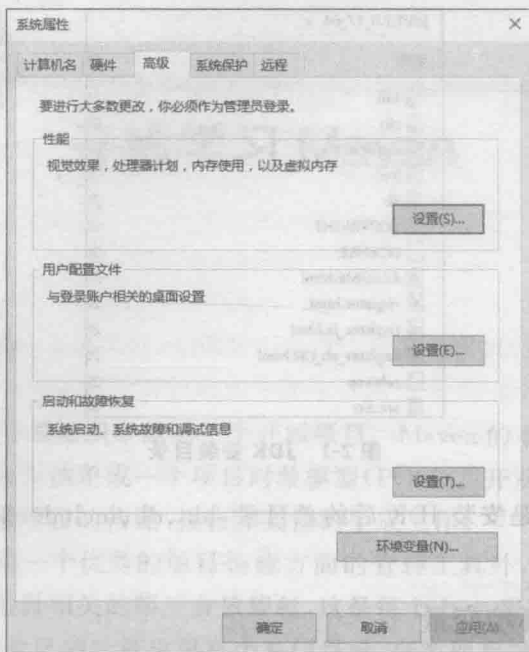


图 2-3 系统属性



图 2-4 系统环境变量

环境 (2) 配置 Path 目录。同样在配置 JAVA_HOME 的地方查找 Path 变量,如图 2-5 所示。

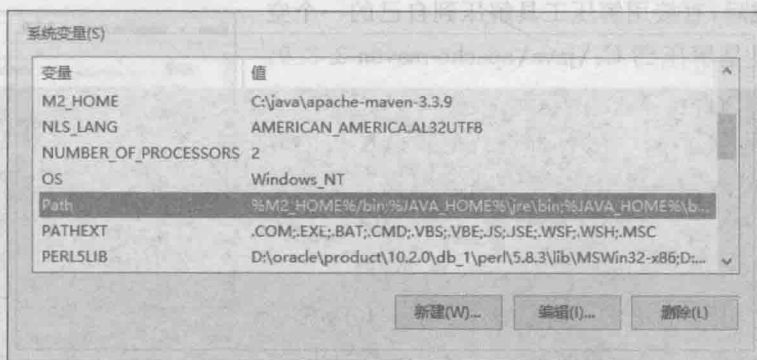


图 2-5 系统变量

单击“编辑”按钮,在以前的 Path 值的前面新添加 JDK 的 bin 目录。这里的 JDK 的 bin 目录是 C:\java\jdk1.7.0_17_64\bin,就是前面的 JAVA_HOME 目录后面添加\bin。添加完成后,同后面的内容要用分号分开,而且是英文输入法的分号。结果如下:

C:\java\jdk1.7.0_17_64\bin;以前的内容

2.2.3 测试 JDK 是否安装成功

打开一个 CMD 窗口,分别输入 javac -version 命令和 java -version 命令,出现如图 2-6 所示的 Java 版本信息显示,表示 JDK 安装成功。

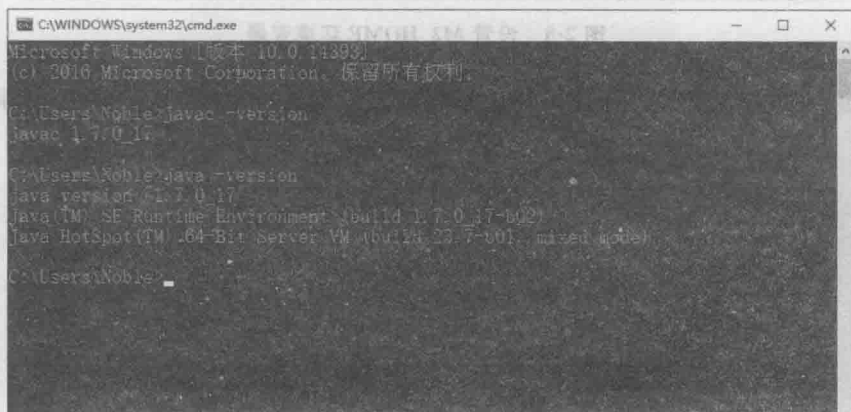


图 2-6 DOS 命令窗口

2.3 Maven 的安装与配置

JDK 环境安装好了,接下来正式安装配置 Maven。先要下载 Maven(<http://maven.apache.org/download.cgi>)。

它是以压缩包形式提供的,下载 Binary 形式的压缩包就行。

下载完成后,直接用解压工具解压到自己的一个空目录下。这里是解压到 C:\java\apache-maven-3.3.9,最后的目录结构如图 2-7 所示。

安装好后,接下来对 Maven 进行配置。配置内容同 JDK 的配置内容一样,也是两部分:一部分是在系统环境变量中配置一个 M2_HOME;另一部分也是将 Maven 里面的 bin 目录添加到 Path 环境变量。

(1) 配置 M2_HOME。同配置 JAVA_HOME 一样,在系统环境变量中添加 M2_HOME,参考 JDK 的 JAVA_HOME 配置,这里就不再赘述了,效果如图 2-8 所示。

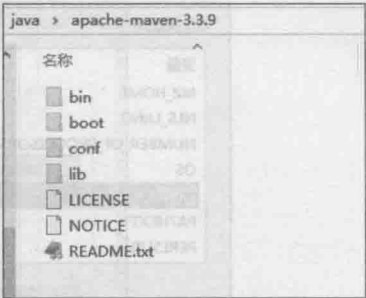


图 2-7 Maven 安装目录

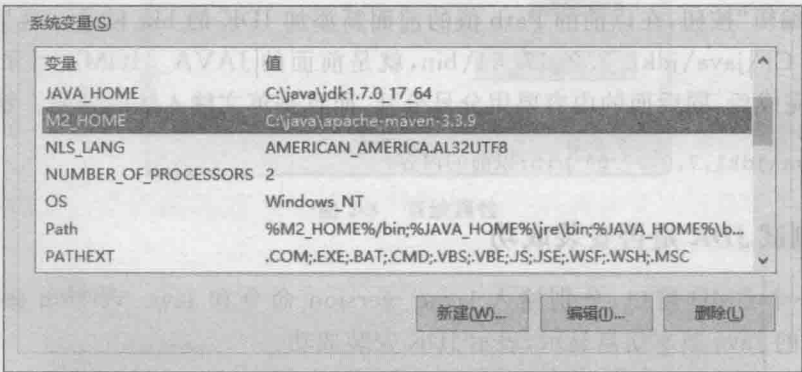


图 2-8 设置 M2_HOME 环境变量

(2) 追加 Path。类似前面 JDK Path 环境变量的配置,请参考 JDK 配置,效果如图 2-9 所示。

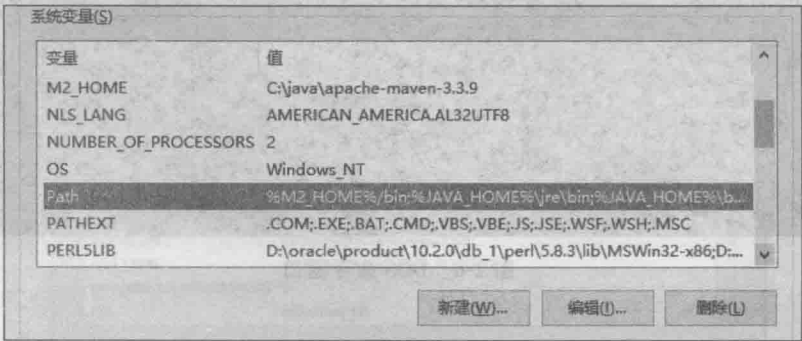


图 2-9 Maven Path 环境变量

其中,%M2_HOME%/bin 是作为 Maven 的 Path 添加的。

(3) 测试安装是否成功。打开一个 CMD 窗口(一定要重新打开一个,不能使用配置