[返回主页](https://www.cnblogs.com/csyzlm/)

[**Maven 下载、安装与配置**](https://www.cnblogs.com/csyzlm/p/11660710.html)

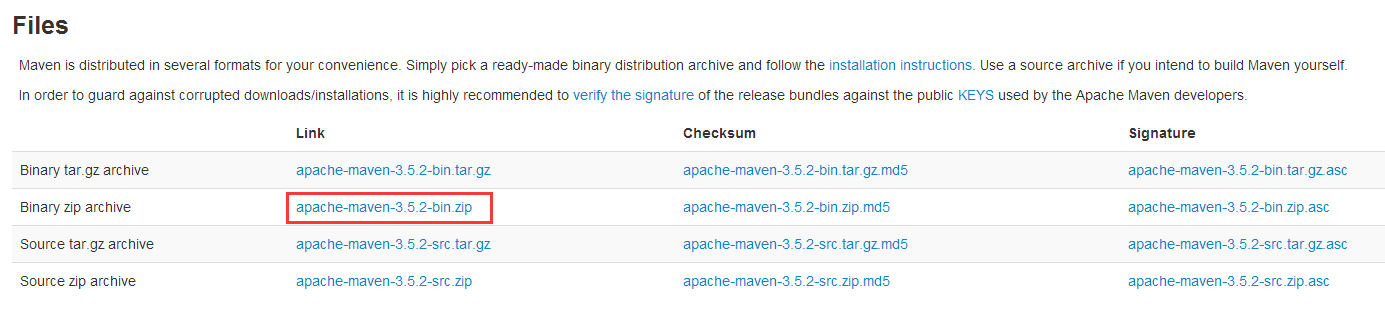
**目录**

* [一、需要准备的东西](https://www.cnblogs.com/csyzlm/p/11660710.html#_label0)
* [二、下载与安装](https://www.cnblogs.com/csyzlm/p/11660710.html#_label1)
* [三、修改maven配置文件](https://www.cnblogs.com/csyzlm/p/11660710.html#_label2)
* [四、IDEA配置maven](https://www.cnblogs.com/csyzlm/p/11660710.html#_label3)
* [五、Eclipse下配置maven](https://www.cnblogs.com/csyzlm/p/11660710.html#_label4)
* [附：完整的Settings.xml文件](https://www.cnblogs.com/csyzlm/p/11660710.html#_label5)

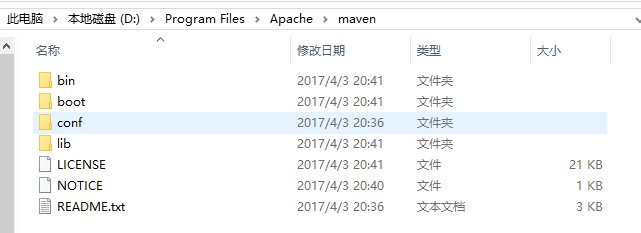
确定电脑上已经成功安装JDK

1. 前往<https://maven.apache.org/download.cgi>下载最新版的Maven程序：

注意：Maven3.2.\*版本需要JDK1.6的支持，Maven3.3.\*以上需要JDK1.7以上的支持



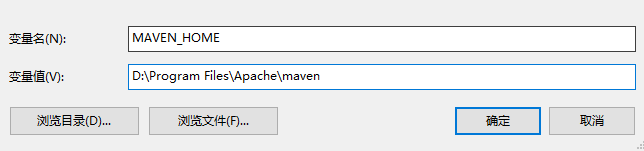
2. 将文件解压到D:\Program Files\Apache\maven目录下:（尽量保证安装目录下无中文及空格）



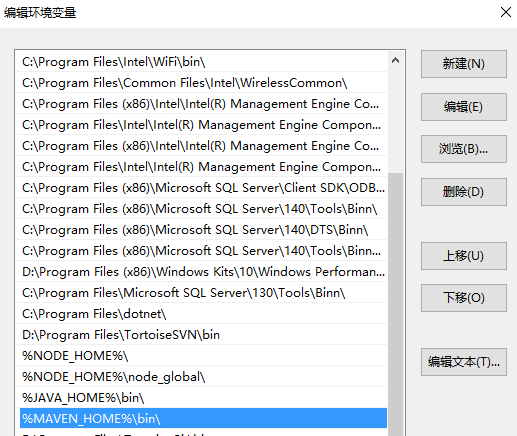
3.配置Maven环境变量

在我的电脑-------属性-------高级系统设置---------环境变量---------系统变量--------新建

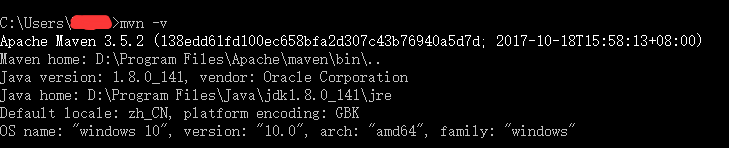
新建环境变量MAVEN\_HOME和M2\_HOME，赋值D:\Program Files\Apache\maven



4. 编辑环境变量Path，追加;%MAVEN\_HOME%\bin\;



5. 至此，maven已经完成了安装，我们可以通过DOS命令输入“mvn -v”检查一下我们是否安装成功：

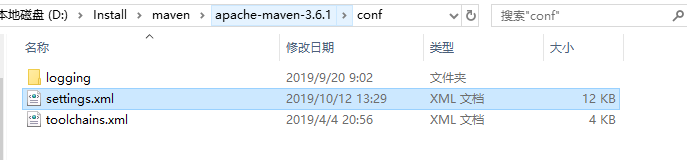


[Top](https://www.cnblogs.com/csyzlm/p/11660710.html#_labelTop)

**三、修改maven配置文件**

通常我们需要修改解压目录下conf/settings.xml文件

此处注意：所有的修改一定要在注释标签外面，不然修改无效。Maven很多标签都是给的例子，都是注释掉的。



**1. 本地仓库位置修改**

在**<localRepository>**标签内添加自己的本地仓库路径

<!--自定义maven仓库路径-->

<localRepository>D:\Install\maven\maven-repository</localRepository>

**2. 修改maven默认的JDK版本（默认为1.4）**

在<**profiles**>标签下添加一个<**profile**>标签，修改maven默认的JDK版本。

[复制代码](javascript:void(0);)

<profile>

　　<id>jdk-1.8</id>

　　<activation>

　　　　<activeByDefault>true</activeByDefault>

　　　　<jdk>1.8</jdk>

　　</activation>

　　<properties>

　　　　<maven.compiler.source>1.8</maven.compiler.source>

　　　　<maven.compiler.target>1.8</maven.compiler.target>

　　　　<maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>

　　</properties>

</profile>

[复制代码](javascript:void(0);)

**3. 添加国内镜像源**

添加**<mirrors>**标签下**<mirror>**，添加国内镜像源，这样下载jar包速度很快。一般使用阿里云镜像库即可。

[复制代码](javascript:void(0);)

<!-- 阿里云仓库 -->

<mirror>

<id>alimaven</id>

<mirrorOf>central</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/repositories/central/</url>

</mirror>

[复制代码](javascript:void(0);)

[Top](https://www.cnblogs.com/csyzlm/p/11660710.html#_labelTop)

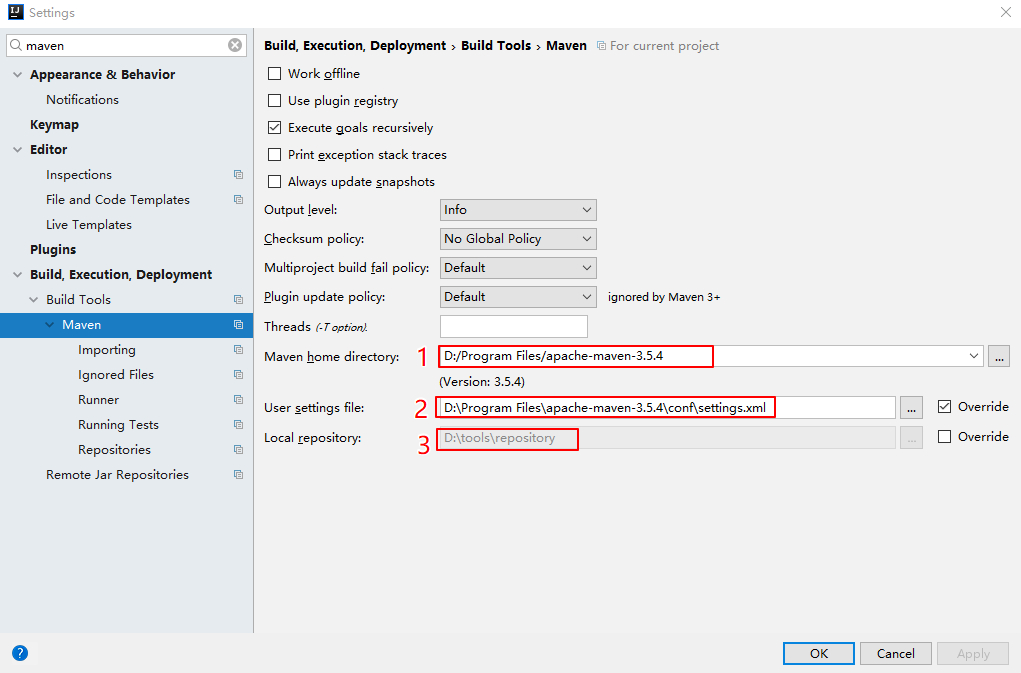
**四、IDEA配置maven**

*目前常用的开发工具如idea，eclipse都自身集成了一个版本的Maven。但是通常我们使用自己已经配置好的Maven。*

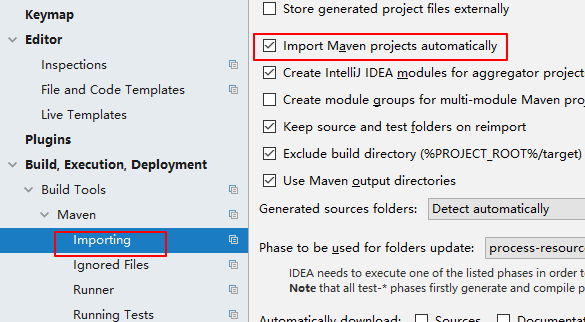
1：此处修改为自己解压的Maven目录

2：勾选**Override**，修改为自己目录下的**settings.xml**目录

3：修改为自己的本地仓库地址，一般会自动识别。

****

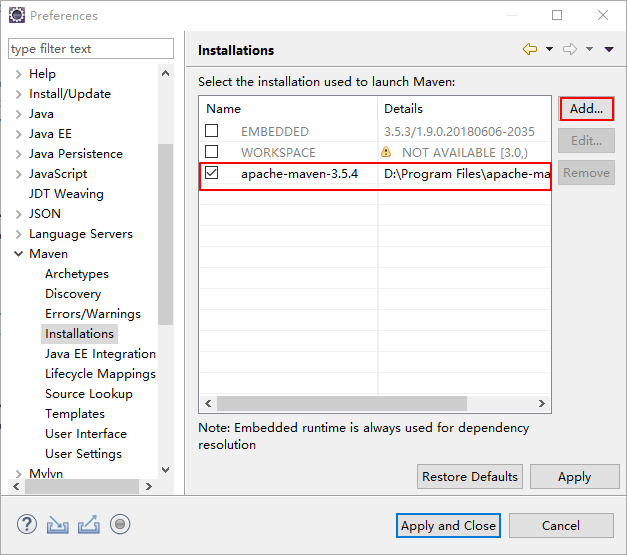
4.勾选此处，当修改pom文件时，Maven就能帮我们自动导包了。



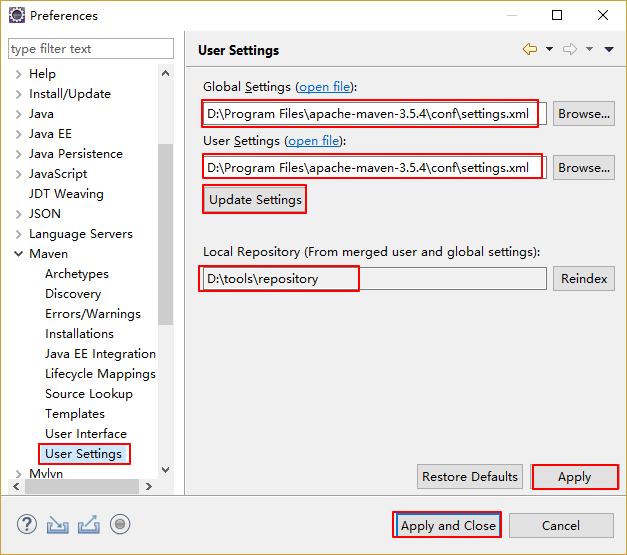
[Top](https://www.cnblogs.com/csyzlm/p/11660710.html#_labelTop)

**五、Eclipse下配置maven**

1.将eclipse使用的Maven修改为自己的。点击add后选择自己Maven的安装目录即可。添加好之后记得勾选。



 2.将所有的settings修改为自己Maven目录下的**conf/settings.xml**.点击**Update Settings**按钮，下面的Local Respository会自动识别出来。



[Top](https://www.cnblogs.com/csyzlm/p/11660710.html#_labelTop)

**附：完整的Settings.xml文件**

[复制代码](javascript:void(0);)

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one

or more contributor license agreements. See the NOTICE file

distributed with this work for additional information

regarding copyright ownership. The ASF licenses this file

to you under the Apache License, Version 2.0 (the

"License"); you may not use this file except in compliance

with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,

software distributed under the License is distributed on an

"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

KIND, either express or implied. See the License for the

specific language governing permissions and limitations

under the License.

-->

<!--

| This is the configuration file for Maven. It can be specified at two levels:

|

| 1. User Level. This settings.xml file provides configuration for a single user,

| and is normally provided in ${user.home}/.m2/settings.xml.

|

| NOTE: This location can be overridden with the CLI option:

|

| -s /path/to/user/settings.xml

|

| 2. Global Level. This settings.xml file provides configuration for all Maven

| users on a machine (assuming they're all using the same Maven

| installation). It's normally provided in

| ${maven.conf}/settings.xml.

|

| NOTE: This location can be overridden with the CLI option:

|

| -gs /path/to/global/settings.xml

|

| The sections in this sample file are intended to give you a running start at

| getting the most out of your Maven installation. Where appropriate, the default

| values (values used when the setting is not specified) are provided.

|

|-->

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

<!-- localRepository

| The path to the local repository maven will use to store artifacts.

|

| Default: ${user.home}/.m2/repository

<localRepository>/path/to/local/repo</localRepository>

-->

<localRepository>D:\tools\repository</localRepository>

<!-- interactiveMode

| This will determine whether maven prompts you when it needs input. If set to false,

| maven will use a sensible default value, perhaps based on some other setting, for

| the parameter in question.

|

| Default: true

<interactiveMode>true</interactiveMode>

-->

<!-- offline

| Determines whether maven should attempt to connect to the network when executing a build.

| This will have an effect on artifact downloads, artifact deployment, and others.

|

| Default: false

<offline>false</offline>

-->

<!-- pluginGroups

| This is a list of additional group identifiers that will be searched when resolving plugins by their prefix, i.e.

| when invoking a command line like "mvn prefix:goal". Maven will automatically add the group identifiers

| "org.apache.maven.plugins" and "org.codehaus.mojo" if these are not already contained in the list.

|-->

<pluginGroups>

<!-- pluginGroup

| Specifies a further group identifier to use for plugin lookup.

<pluginGroup>com.your.plugins</pluginGroup>

-->

</pluginGroups>

<!-- proxies

| This is a list of proxies which can be used on this machine to connect to the network.

| Unless otherwise specified (by system property or command-line switch), the first proxy

| specification in this list marked as active will be used.

|-->

<proxies>

<!-- proxy

| Specification for one proxy, to be used in connecting to the network.

|

<proxy>

<id>optional</id>

<active>true</active>

<protocol>http</protocol>

<username>proxyuser</username>

<password>proxypass</password>

<host>proxy.host.net</host>

<port>80</port>

<nonProxyHosts>local.net|some.host.com</nonProxyHosts>

</proxy>

-->

</proxies>

<!-- servers

| This is a list of authentication profiles, keyed by the server-id used within the system.

| Authentication profiles can be used whenever maven must make a connection to a remote server.

|-->

<servers>

<!-- server

| Specifies the authentication information to use when connecting to a particular server, identified by

| a unique name within the system (referred to by the 'id' attribute below).

|

| NOTE: You should either specify username/password OR privateKey/passphrase, since these pairings are

| used together.

|

<server>

<id>deploymentRepo</id>

<username>repouser</username>

<password>repopwd</password>

</server>

-->

<!-- Another sample, using keys to authenticate.

<server>

<id>siteServer</id>

<privateKey>/path/to/private/key</privateKey>

<passphrase>optional; leave empty if not used.</passphrase>

</server>

-->

</servers>

<!-- mirrors

| This is a list of mirrors to be used in downloading artifacts from remote repositories.

|

| It works like this: a POM may declare a repository to use in resolving certain artifacts.

| However, this repository may have problems with heavy traffic at times, so people have mirrored

| it to several places.

|

| That repository definition will have a unique id, so we can create a mirror reference for that

| repository, to be used as an alternate download site. The mirror site will be the preferred

| server for that repository.

|-->

<mirrors>

<!-- mirror

| Specifies a repository mirror site to use instead of a given repository. The repository that

| this mirror serves has an ID that matches the mirrorOf element of this mirror. IDs are used

| for inheritance and direct lookup purposes, and must be unique across the set of mirrors.

|

<mirror>

<id>mirrorId</id>

<mirrorOf>repositoryId</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://my.repository.com/repo/path</url>

</mirror>

-->

<!-- 阿里云仓库 -->

<mirror>

<id>alimaven</id>

<mirrorOf>central</mirrorOf>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/repositories/central/</url>

</mirror>

</mirrors>

<!-- profiles

| This is a list of profiles which can be activated in a variety of ways, and which can modify

| the build process. Profiles provided in the settings.xml are intended to provide local machine-

| specific paths and repository locations which allow the build to work in the local environment.

|

| For example, if you have an integration testing plugin - like cactus - that needs to know where

| your Tomcat instance is installed, you can provide a variable here such that the variable is

| dereferenced during the build process to configure the cactus plugin.

|

| As noted above, profiles can be activated in a variety of ways. One way - the activeProfiles

| section of this document (settings.xml) - will be discussed later. Another way essentially

| relies on the detection of a system property, either matching a particular value for the property,

| or merely testing its existence. Profiles can also be activated by JDK version prefix, where a

| value of '1.4' might activate a profile when the build is executed on a JDK version of '1.4.2\_07'.

| Finally, the list of active profiles can be specified directly from the command line.

|

| NOTE: For profiles defined in the settings.xml, you are restricted to specifying only artifact

| repositories, plugin repositories, and free-form properties to be used as configuration

| variables for plugins in the POM.

|

|-->

<profiles>

<!-- profile

| Specifies a set of introductions to the build process, to be activated using one or more of the

| mechanisms described above. For inheritance purposes, and to activate profiles via <activatedProfiles/>

| or the command line, profiles have to have an ID that is unique.

|

| An encouraged best practice for profile identification is to use a consistent naming convention

| for profiles, such as 'env-dev', 'env-test', 'env-production', 'user-jdcasey', 'user-brett', etc.

| This will make it more intuitive to understand what the set of introduced profiles is attempting

| to accomplish, particularly when you only have a list of profile id's for debug.

|

| This profile example uses the JDK version to trigger activation, and provides a JDK-specific repo.

<profile>

<id>jdk-1.4</id>

<activation>

<jdk>1.4</jdk>

</activation>

<repositories>

<repository>

<id>jdk14</id>

<name>Repository for JDK 1.4 builds</name>

<url>http://www.myhost.com/maven/jdk14</url>

<layout>default</layout>

<snapshotPolicy>always</snapshotPolicy>

</repository>

</repositories>

</profile>

-->

<!--

| Here is another profile, activated by the system property 'target-env' with a value of 'dev',

| which provides a specific path to the Tomcat instance. To use this, your plugin configuration

| might hypothetically look like:

|

| ...

| <plugin>

| <groupId>org.myco.myplugins</groupId>

| <artifactId>myplugin</artifactId>

|

| <configuration>

| <tomcatLocation>${tomcatPath}</tomcatLocation>

| </configuration>

| </plugin>

| ...

|

| NOTE: If you just wanted to inject this configuration whenever someone set 'target-env' to

| anything, you could just leave off the <value/> inside the activation-property.

|

<profile>

<id>env-dev</id>

<activation>

<property>

<name>target-env</name>

<value>dev</value>

</property>

</activation>

<properties>

<tomcatPath>/path/to/tomcat/instance</tomcatPath>

</properties>

</profile>

-->

<profile>

<id>JDK-1.8</id>

<activation>

<activeByDefault>true</activeByDefault>

<jdk>1.8</jdk>

</activation>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>

</properties>

</profile>

</profiles>

<!-- activeProfiles

| List of profiles that are active for all builds.

|

<activeProfiles>

<activeProfile>alwaysActiveProfile</activeProfile>

<activeProfile>anotherAlwaysActiveProfile</activeProfile>

</activeProfiles>

-->

</settings>

[复制代码](javascript:void(0);)