maven 安装和配置

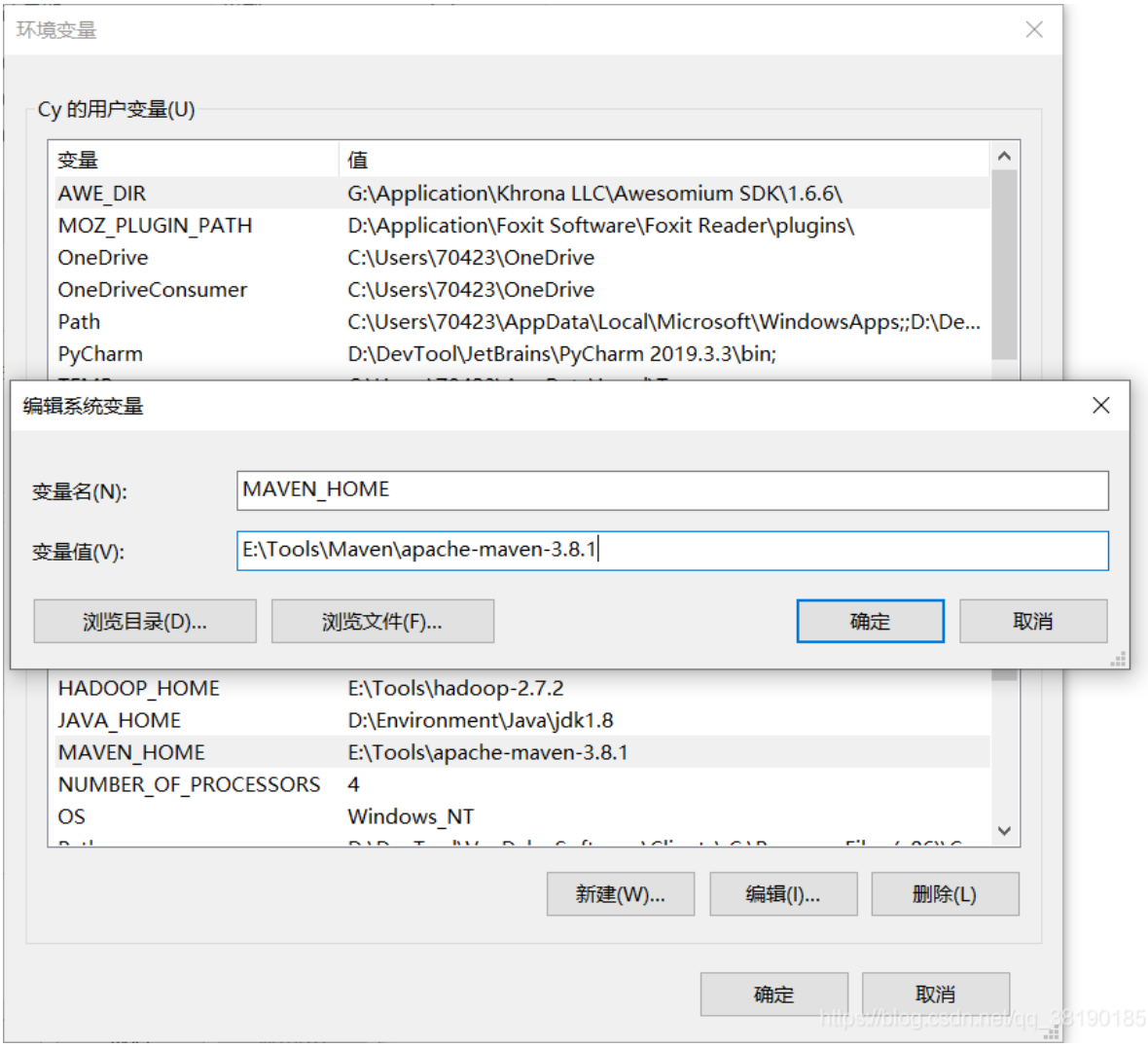
安装 maven

官网下载安装maven： https://maven.apache.org/download.cgi

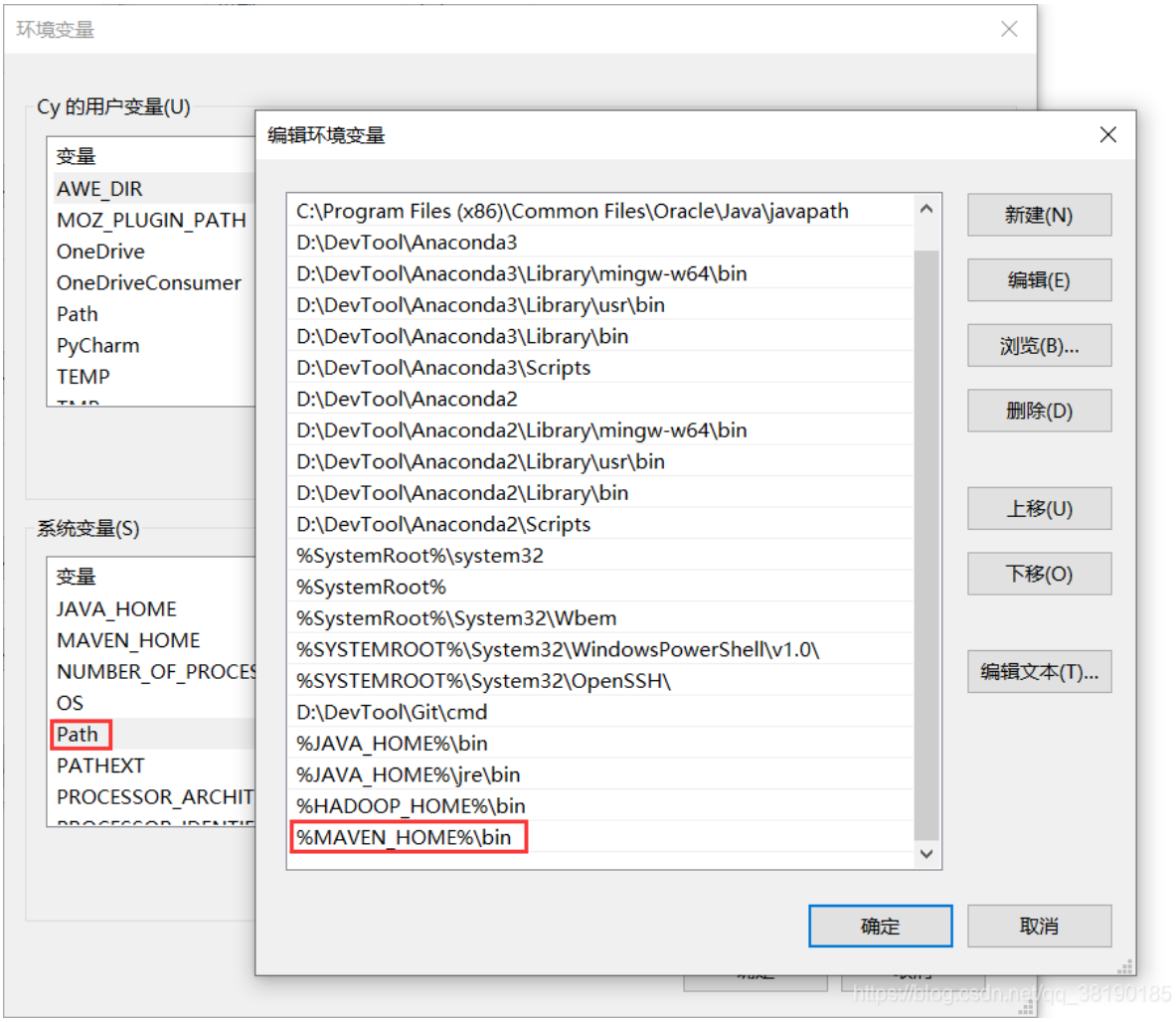
> “
>
> Binary是可执行版本，已经编译好可以直接使用。
> Source是源代码版本，需要自己编译成可执行软件才可使用。
> tar.gz和zip两种压缩格式,其实这两个压缩文件里面包含的内容是同样的,只是压缩格式不同
> tar.gz格式的文件比zip文件小很多,用于unix操作系统。
> zip格式用于Windows操作系统,但在Windows系统使用WinRar工具一样能够解压缩tar.gz格式

配置环境变量

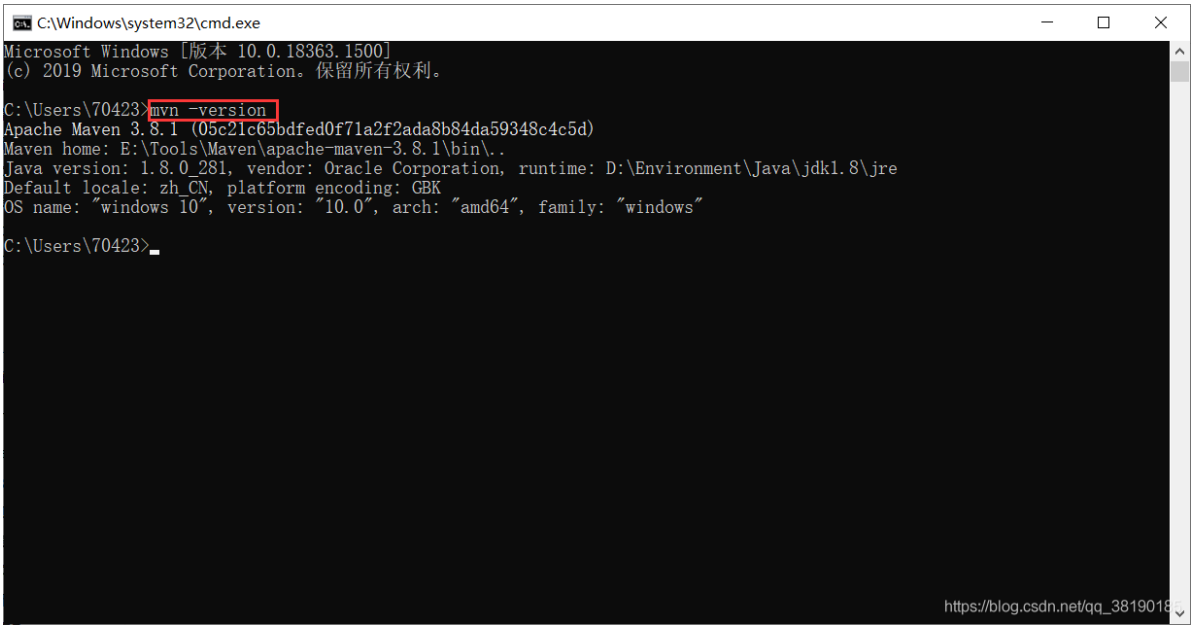
在【系统变量】新建变量MAVEN_HOME = E:\Tools\Maven\apache-maven-3.8.1（以自己的安装路径为准）



编辑系统变量的Path，添加变量值%MAVEN_HOME%\bin



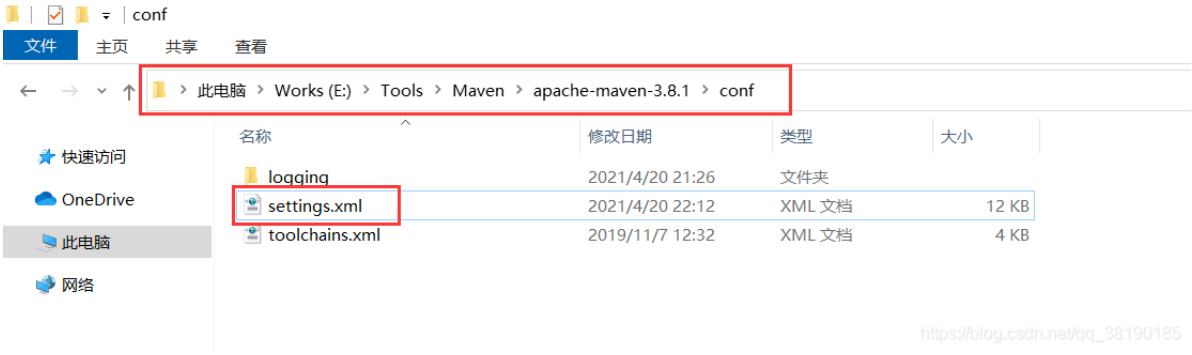
然后win+R运行cmd，输入mvn -version，如图所示则配置成功



配置本地仓库

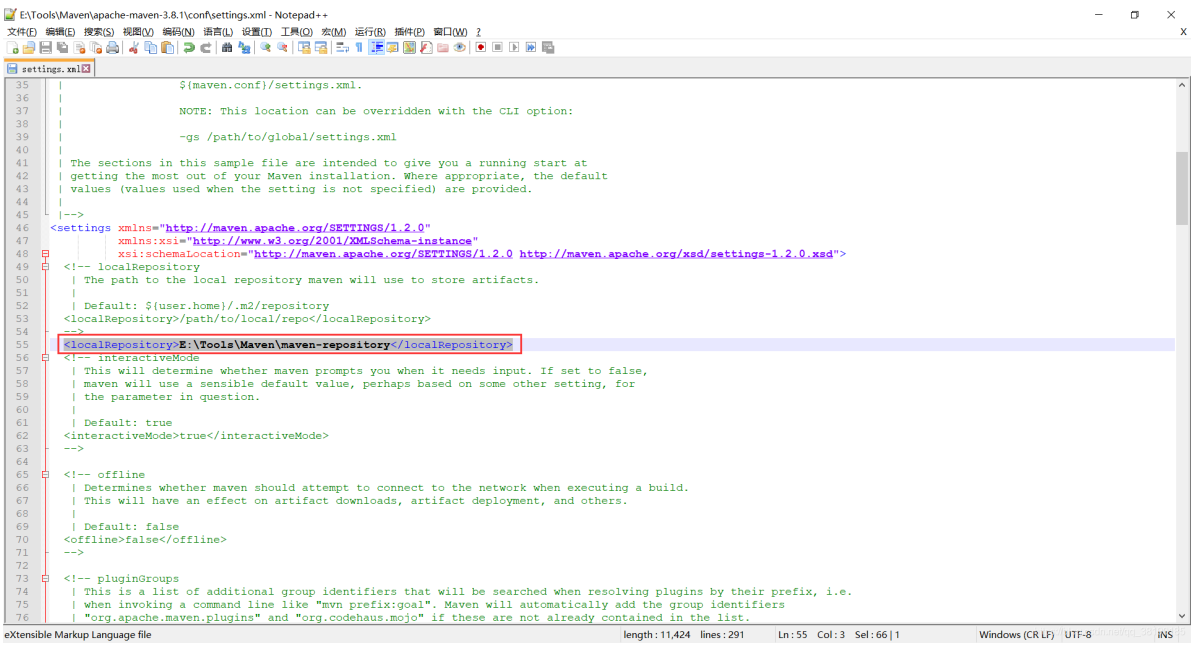
1. 在E:\Tools\Maven\路径下新建maven-repository文件夹，用作maven的本地库。

2. 在路径E:\Tools\Maven\apache-maven-3.8.1\conf下找到settings.xml文件



3. 找到节点localRepository，在注释外添加

```
1    <localRepository>E:\Tools\Maven\maven-repository</localRepository>
```

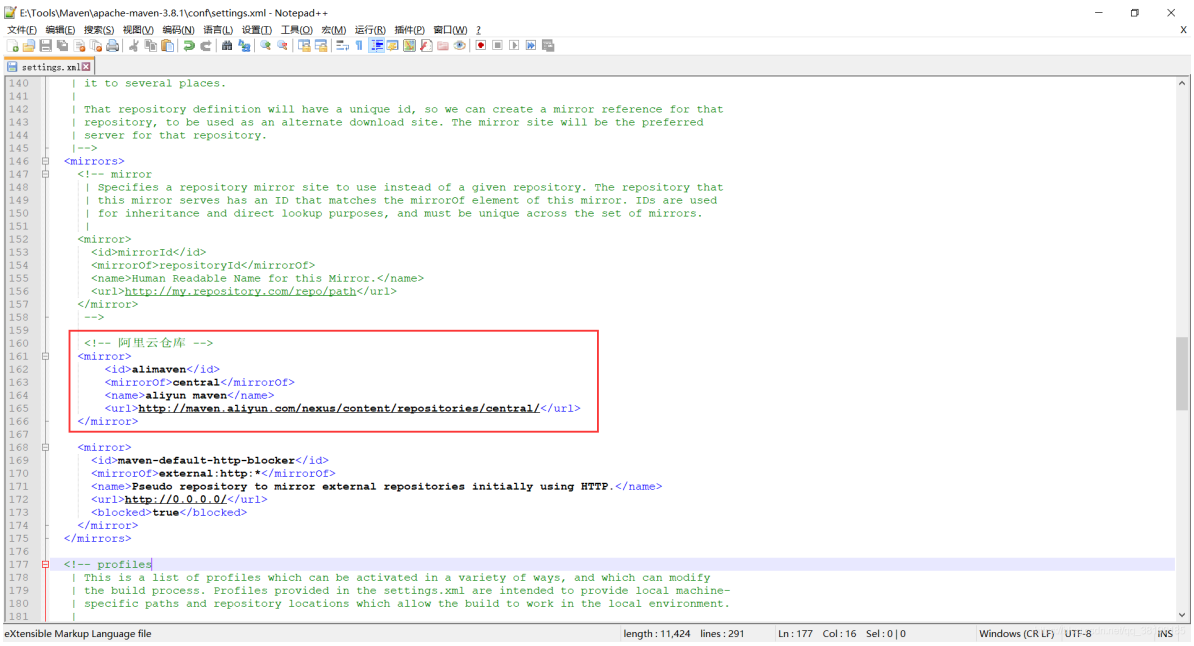


> i
>
> localRepository节点用于配置本地仓库，本地仓库其实起到了一个缓存的作用，它的默认地址是C:\Users\用户名.m2。
> 当我们从maven中获取jar包的时候，maven首先会在本地仓库中查找，如果本地仓库有则返回；如果没有则从远程仓库中获取包，并在本地库中保存。
> 此外，我们在maven项目中运行mvn install，项目将会自动打包并安装到本地仓库中。

配置镜像

1. 在settings.xml配置文件中找到mirrors节点
2. 添加如下配置（注意要添加在和两个标签之间，其它配置同理）

```
<!-- 阿里云仓库 -->
<mirror>
    <id>alimaven</id>
    <mirrorOf>central</mirrorOf>
    <name>aliyun maven</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
</mirror>
```



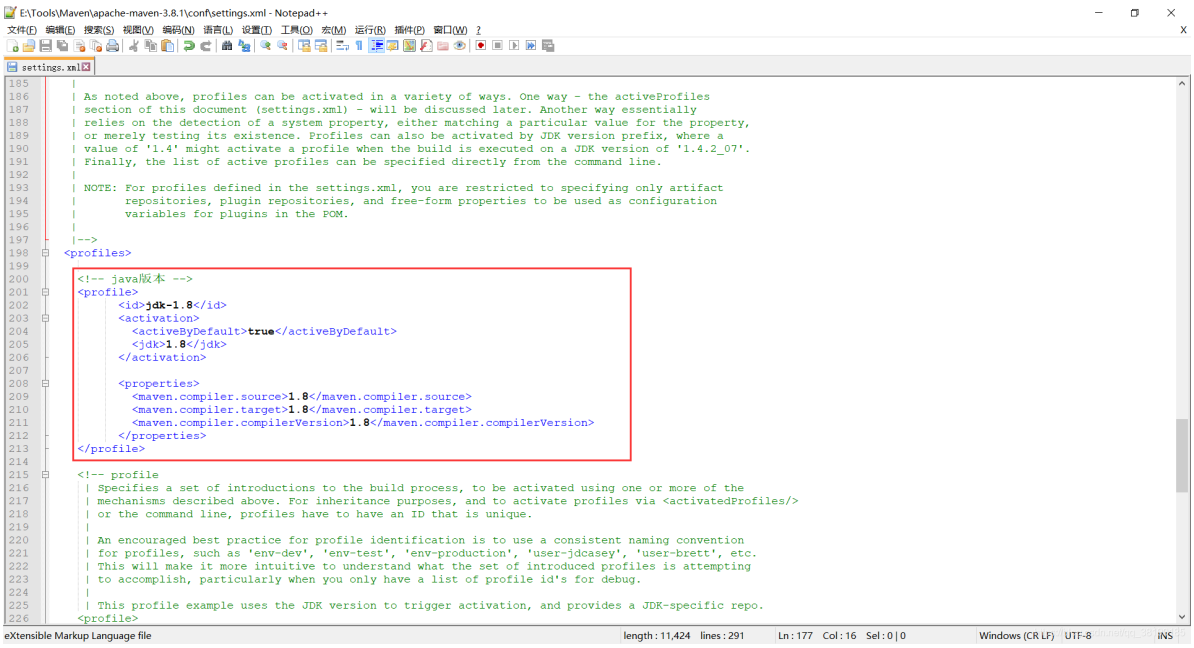
配置JDK

1. 在settings.xml配置文件中找到profiles节点
2. 添加如下配置

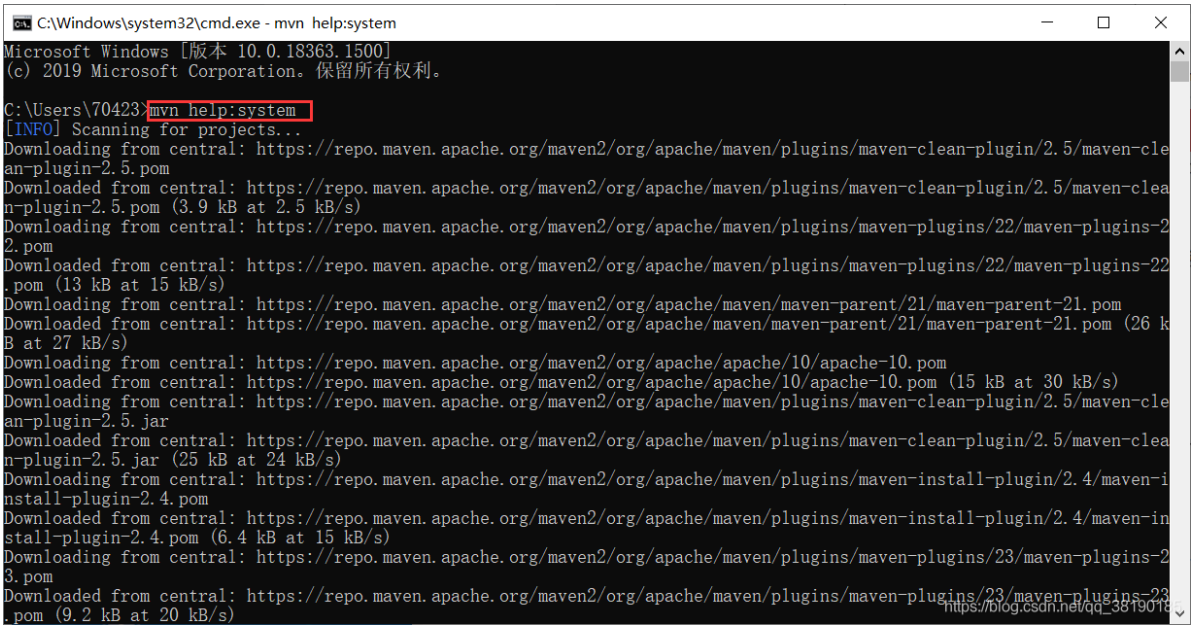
```
<!-- java版本 -->
<profile>
      <id>jdk-1.8</id>
      <activation>
        <activeByDefault>true</activeByDefault>
        <jdk>1.8</jdk>
      </activation>

      <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>
```



配置完成， win+R 运行 cmd ， 输入 mvn help:system 测试， 配置成功则本地仓库（E:\Tools\Maven\maven-repository）中会出现一些文件。



> “
>
> 首次执行 mvn help:system 命令，Maven相关工具自动帮我们到Maven中央仓库下载缺省的或者Maven中央仓库更新的各种配置文件和类库（jar包)到Maven本地仓库中。
> 下载完各种文件后， mvn help:system 命令会打印出所有的Java系统属性和环境变量，这些信息对我们日常的编程工作很有帮助。

附录

完整的settings.xml配置文件，可以直接复制使用，只需要修改相应的路径即可

```
<?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/1.2.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.2.0 http://maven.apache.org/xsd/settings-1.2.0.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>E:\Tools\Maven\maven-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>
```

```
84
85    <!-- proxies
86     | This is a list of proxies which can be used on this machine to connect
      to the network.
87     | Unless otherwise specified (by system property or command-line
      switch), the first proxy
88     | specification in this list marked as active will be used.
89     |-->
90    <proxies>
91      <!-- proxy
92       | Specification for one proxy, to be used in connecting to the
      network.
93       |
94      <proxy>
95        <id>optional</id>
96        <active>true</active>
97        <protocol>http</protocol>
98        <username>proxyuser</username>
99        <password>proxypass</password>
100       <host>proxy.host.net</host>
101       <port>80</port>
102       <nonProxyHosts>local.net|some.host.com</nonProxyHosts>
103     </proxy>
104     -->
105   </proxies>
106
107   <!-- servers
108    | This is a list of authentication profiles, keyed by the server-id used
      within the system.
109    | Authentication profiles can be used whenever maven must make a
      connection to a remote server.
110    |-->
111   <servers>
112     <!-- server
113      | Specifies the authentication information to use when connecting to a
      particular server, identified by
114      | a unique name within the system (referred to by the 'id' attribute
      below).
115      |
116      | NOTE: You should either specify username/password OR
      privateKey/passphrase, since these pairings are
117      |       used together.
118      |
119     <server>
120       <id>deploymentRepo</id>
121       <username>repouser</username>
122       <password>repopwd</password>
123     </server>
124     -->
125
126     <!-- 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>

    <mirror>
```

```
        <id>maven-default-http-blocker</id>
        <mirrorOf>external:http:*</mirrorOf>
        <name>Pseudo repository to mirror external repositories initially using HTTP.</name>
        <url>http://0.0.0.0/</url>
        <blocked>true</blocked>
      </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>

      <!-- java版本 -->
      <profile>
            <id>jdk-1.8</id>
            <activation>
              <activeByDefault>true</activeByDefault>
              <jdk>1.8</jdk>
```

```
206             </activation>
207
208             <properties>
209               <maven.compiler.source>1.8</maven.compiler.source>
210               <maven.compiler.target>1.8</maven.compiler.target>
211
    <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>
212             </properties>
213       </profile>
214
215       <!-- profile
216        | Specifies a set of introductions to the build process, to be
    activated using one or more of the
217        | mechanisms described above. For inheritance purposes, and to
    activate profiles via <activatedProfiles/>
218        | or the command line, profiles have to have an ID that is unique.
219        |
220        | An encouraged best practice for profile identification is to use a
    consistent naming convention
221        | for profiles, such as 'env-dev', 'env-test', 'env-production',
    'user-jdcasey', 'user-brett', etc.
222        | This will make it more intuitive to understand what the set of
    introduced profiles is attempting
223        | to accomplish, particularly when you only have a list of profile
    id's for debug.
224        |
225        | This profile example uses the JDK version to trigger activation, and
    provides a JDK-specific repo.
226       <profile>
227         <id>jdk-1.4</id>
228
229         <activation>
230           <jdk>1.4</jdk>
231         </activation>
232
233         <repositories>
234           <repository>
235             <id>jdk14</id>
236             <name>Repository for JDK 1.4 builds</name>
237             <url>http://www.myhost.com/maven/jdk14</url>
238             <layout>default</layout>
239             <snapshotPolicy>always</snapshotPolicy>
240           </repository>
241         </repositories>
242       </profile>
243
244       -->
245
246       <!--
247        | Here is another profile, activated by the system property 'target-
    env' with a value of 'dev',
```

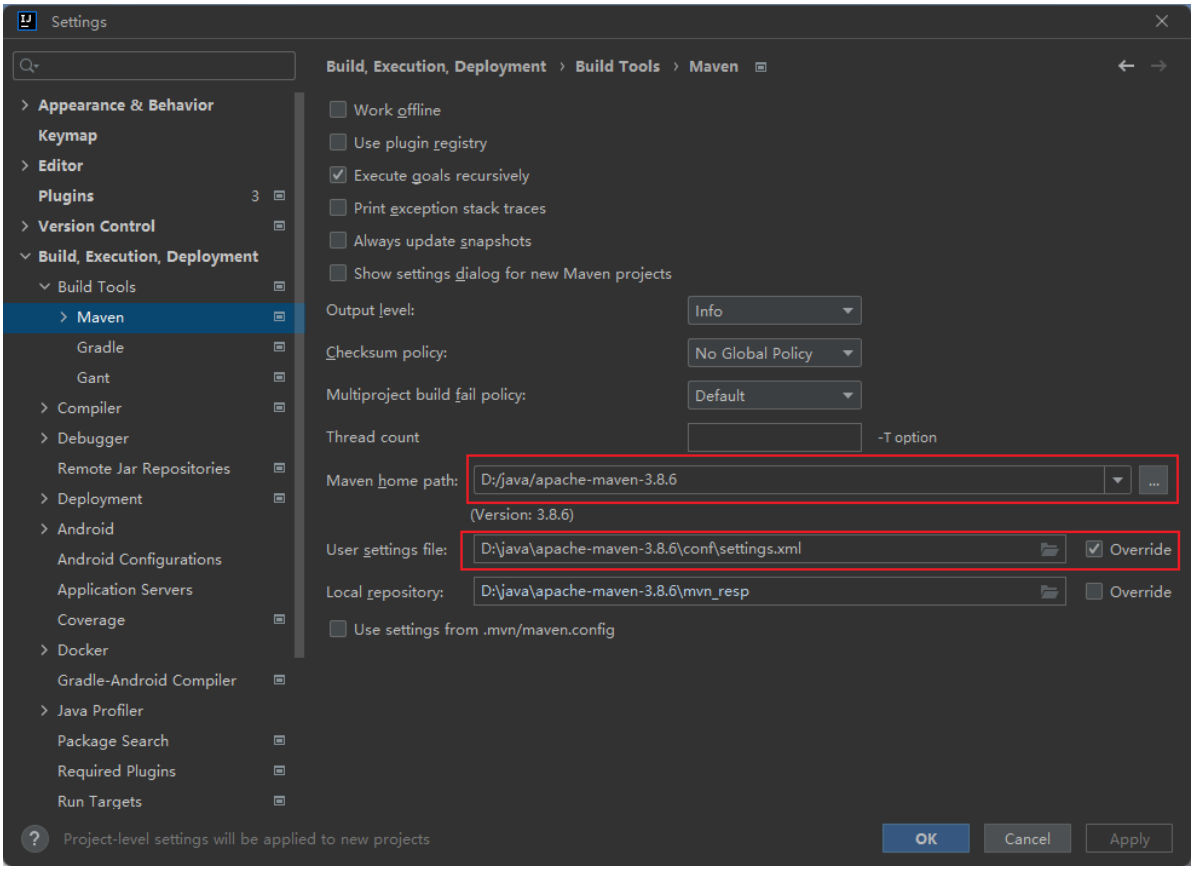
```
      | which provides a specific path to the Tomcat instance. To use this, your plugin configuration
      | might hypothetically look like:
      |
      | ...
      | <plugin>
      |   <groupId>org.myco.myplugins</groupId>
      |   <artifactId>myplugin</artifactId>
      |
      |   <configuration>
      |     <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>
    -->
  </profiles>

  <!-- activeProfiles
   | List of profiles that are active for all builds.
   |
  <activeProfiles>
    <activeProfile>alwaysActiveProfile</activeProfile>
    <activeProfile>anotherAlwaysActiveProfile</activeProfile>
  </activeProfiles>
  -->
</settings>
```

IDEA配置maven



- 配合插件 `maven Helper` 使用