

Maven 构建配置文件

构建配置文件是一系列的配置项的值，可以用来设置或者覆盖 Maven 构建默认值。使用构建配置文件，你可以为不同的环境，比如说生产环境（ Production ）和开发（ Development ）环境，定制构建方式。

配置文件在 pom.xml 文件中使用 activeProfiles 或者 profiles 元素指定，并且可以通过各种方式触发。配置文件在构建时修改 POM，并且用来给参数设定不同的目标环境（ 比如说，开发（ Development ）、测试（ Testing ）和生产环境（ Production ）中数据库服务器的地址 ）。

构建配置文件的类型

构建配置文件大体上有三种类型:

类型	在哪定义
项目级（ Per Project ）	定义在项目的POM文件pom.xml中
用户级（ Per User ）	定义在Maven的设置xml文件中 (%USER_HOME%/.m2/settings.xml)
全局（ Global ）	定义在 Maven 全局的设置 xml 文件中 (%M2_HOME%/conf/settings.xml)

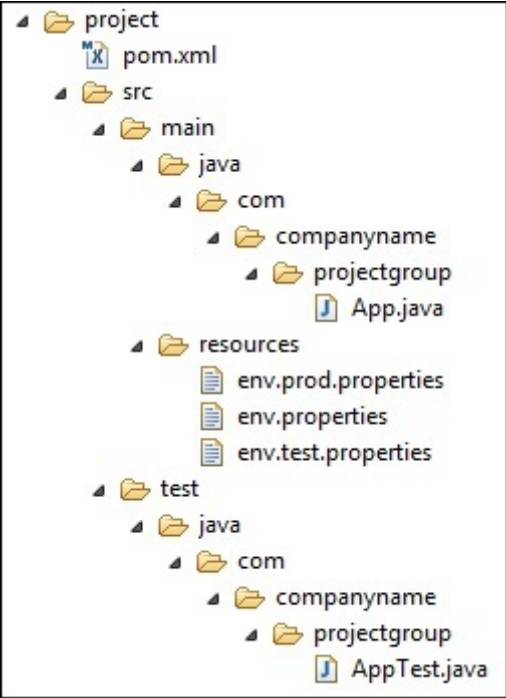
配置文件激活

Maven的构建配置文件可以通过多种方式激活。

- 使用命令控制台输入显式激活。
- 通过 maven 设置。
- 基于环境变量（ 用户或者系统变量 ）。
- 操作系统设置（ 比如说， Windows系列 ）。
- 文件的存在或者缺失。

配置文件激活实例

假定项目结构如下：



其中在src/main/resources文件夹下有三个用于测试文件：

文件名	描述
env.properties	如果未指定配置文件时默认使用的配置。
env.test.properties	当测试配置文件使用时的测试配置。
env.prod.properties	当生产配置文件使用时的生产配置。

注意：这三个配置文件并不是代表构建配置文件的函数，而是用于本次测试的目的；比如，我指定了构建配置文件为 prod 时，项目就使用 envprod.properties文件。

注意：下面的例子仍然是使用 AntRun 插件，因为此插件能绑定 Maven 生命周期阶段，并通过 Ant 的标签不用编写一点代码即可输出信息、复制文件等，经此而已。其余的与本次构建配置文件无关。

1、配置文件激活

profile 可以让我们定义一系列的配置信息，然后指定其激活条件。这样我们就可以定义多个 profile，然后每个 profile 对应不同的激活条件和配置信息，从而达到不同环境使用不同配置信息的效果。

以下实例，我们将 maven-antrun-plugin:run 目标添加到测试阶段中。这样我们可以在不同的 profile 中输出文本信息。我们将使用 pom.xml 来定义不同的 profile，并在命令控制台中使用 maven 命令激活 profile。

pom.xml 文件如下：

```
<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>com.jsoft.test</groupId>
<artifactId>testproject</artifactId>
<packaging>jar</packaging>
```

```
<version>0.1-SNAPSHOT</version>
<name>testproject</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>
<profiles>
<profile>
<id>test</id>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-antrun-plugin</artifactId>
<version>1.8</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>run</goal>
</goals>
<configuration>
<tasks>
<echo>Using env.test.properties</echo>
<copy file="src/main/resources/env.test.properties" tofile="${project.build.outputDirectory}/env.properties" overwrite="true"/>
</tasks>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
<profile>
<id>normal</id>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-antrun-plugin</artifactId>
<version>1.8</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>run</goal>
</goals>
<configuration>
```

```
<tasks>
<echo>Using env.properties</echo>
<copy file="src/main/resources/env.properties" tofile="${project.build.outputDirectory}/env.pro
properties" overwrite="true"/>
</tasks>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
<profile>
<id>prod</id>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-antrun-plugin</artifactId>
<version>1.8</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>run</goal>
</goals>
<configuration>
<tasks>
<echo>Using env.prod.properties</echo>
<copy file="src/main/resources/env.prod.properties" tofile="${project.build.outputDirectory}/en
v.properties" overwrite="true"/>
</tasks>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
</profiles>
</project>
```

注意：构建配置文件采用的是 `<profiles>` 节点。

说明：上面新建了三个 `<profiles>`，其中 `<id>` 区分了不同的 `<profiles>` 执行不同的 AntRun 任务；而 AntRun 的任务可以这么理解，AntRun 监听 test 的 Maven 生命周期阶段，当 Maven 执行 test 时，就除了发 AntRun 的任务，任务里面为输出文本并复制文件到指定的位置；而至于要执行哪个 AntRun 任务，此时**构建配置文件**起到了传输指定的作用，比如，通过命令行参数输入指定的 `<id>`。

执行命令：

```
mvn test -Ptest
```

提示：第一个 test 为 Maven 生命周期阶段，第 2 个 test 为**构建配置文件**指定的 <id> 参数，这个参数通过 **-P** 来传输，当然，它可以是 prod 或者 normal 这些由你定义的<id>。

运行的结果如下：

```
D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject
λ mvn test -Ptest
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building testproject 0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 3 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ testproject ---
[INFO] Surefire report directory: D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\surefire-reports

-----
T E S T S
-----
Running com.jssoft.test.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (default) @ testproject ---
[WARNING] Parameter tasks is deprecated, use target instead
[INFO] Executing tasks

main:
[echo] Using env.test.properties
[copy] Copying 1 file to D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\classes
[INFO] Executed tasks
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.100s
[INFO] Finished at: Tue May 09 04:50:30 CST 2017
[INFO] Final Memory: 12M/491M
[INFO] -----
```

可以看出成功的触发了AntRun的任务。并且是对应**构建配置文件**下的 <id> 为 test 的任务。

再测试其余两个命令，结果如下：

```
D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject
λ mvn test -Pprod
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building testproject 0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 3 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ testproject ---
[INFO] Surefire report directory: D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\surefire-reports

-----
T E S T S
-----
Running com.jsoft.test.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.004 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (default) @ testproject ---
[WARNING] Parameter tasks is deprecated, use target instead
[INFO] Executing tasks

main:
  [echo] Using env.prod.properties
  [copy] Copying 1 file to D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\classes
[INFO] Executed tasks
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.096s
[INFO] Finished at: Tue May 09 04:52:34 CST 2017
[INFO] Final Memory: 12M/491M
[INFO] -----
```

```

D:\开发工程\Github\5_java_example\maventest\test4\test4\testproject
λ mvn test -Pnormal
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building testproject 0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 3 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\开发工程\Github\5_java_example\maventest\test4\test4\testproject\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ testproject ---
[INFO] Surefire report directory: D:\开发工程\Github\5_java_example\maventest\test4\test4\testproject\target\surefire-reports

-----
T E S T S
-----
Running com.jsoft.test.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (default) @ testproject ---
[WARNING] Parameter tasks is deprecated, use target instead
[INFO] Executing tasks

main:
  [echo] Using env.properties
  [copy] Copying 1 file to D:\开发工程\Github\5_java_example\maventest\test4\test4\testproject\target\classes
[INFO] Executed tasks
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.011s
[INFO] Finished at: Tue May 09 04:53:19 CST 2017
[INFO] Final Memory: 12M/491M
[INFO] -----

```

2、通过Maven设置激活配置文件

打开 %USER_HOME%\.m2 目录下的 settings.xml 文件，其中 %USER_HOME% 代表用户主目录。如果 setting.xml 文件不存在就直接拷贝 %M2_HOME%\conf/settings.xml 到 .m2 目录，其中 %M2_HOME% 代表 Maven 的安装目录。

配置 setting.xml 文件，增加 <activeProfiles>属性：

```

<settings 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/xsd/settings-1.0.0.xsd">
...
<activeProfiles>
<activeProfile>test</activeProfile>
</activeProfiles>
</settings>

```

执行命令：

```
mvn test
```

提示 1：此时不需要使用 -Ptest 来输入参数了，上面的 setting.xml 文件的 <activeprofile> 已经指定了 test 参数代替了。

提示 2：同样可以使用在 %M2_HOME%/conf/settings.xml 的文件进行配置，效果一致。

执行结果：

```
D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject
λ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building testproject 0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 3 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ testproject ---
[INFO] Surefire report directory: D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\surefire-reports

-----
T E S T S
-----
Running com.jsoft.test.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.002 sec
Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (default) @ testproject ---
[WARNING] Parameter tasks is deprecated, use target instead
[INFO] Executing tasks

main:
[echo] Using env.test.properties
[copy] Copying 1 file to D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\classes
[INFO] Executed tasks
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.179s
[INFO] Finished at: Tue May 09 05:03:46 CST 2017
[INFO] Final Memory: 12M/491M
[INFO] -----
```

3、通过环境变量激活配置文件

先把上一步测试的 setting.xml 值全部去掉。

然后在 pom.xml 里面的 <id> 为 test 的 <profile> 节点，加入 <activation> 节点：

```
<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>com.jsoft.test</groupId>
<artifactId>testproject</artifactId>
<packaging>jar</packaging>
<version>0.1-SNAPSHOT</version>
<name>testproject</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>
<profiles>
<profile>
<id>test</id>
<activation>
<property>
<name>env</name>
<value>test</value>
</property>
</activation>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-antrun-plugin</artifactId>
<version>1.8</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>run</goal>
</goals>
<configuration>
<tasks>
<echo>Using env.test.properties</echo>
<copy file="src/main/resources/env.test.properties" tofile="${project.build.outputDirectory}/env.properties" overwrite="true"/>
</tasks>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
<profile>
<id>normal</id>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-antrun-plugin</artifactId>
<version>1.8</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>run</goal>
```

```
</goals>
<configuration>
<tasks>
<echo>Using env.properties</echo>
<copy file="src/main/resources/env.properties" tofile="${project.build.outputDirectory}/env.pro
properties" overwrite="true"/>
</tasks>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
<profile>
<id>prod</id>
<build>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-antrun-plugin</artifactId>
<version>1.8</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>run</goal>
</goals>
<configuration>
<tasks>
<echo>Using env.prod.properties</echo>
<copy file="src/main/resources/env.prod.properties" tofile="${project.build.outputDirectory}/en
v.properties" overwrite="true"/>
</tasks>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</profile>
</profiles>
</project>
```

执行命令：

```
mvn test -Denv=test
```

提示 1：上面使用 -D 传递环境变量，其中 env 对应刚才设置的 <name> 值，test 对应<value>。

提示 2：在 Windows 10 上测试了系统的环境变量，但是不生效，所以，只能通过 **-D** 传递。

执行结果：

```

D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject
λ mvn test -Denv=test
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building testproject 0.1-SNAPSHOT
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 3 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ testproject ---
[WARNING] Using platform encoding (GBK actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ testproject ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ testproject ---
[INFO] Surefire report directory: D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\surefire-reports

-----
T E S T S
-----
Running com.jssoft.test.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (default) @ testproject ---
[WARNING] Parameter tasks is deprecated, use target instead
[INFO] Executing tasks

main:
[echo] Using env.test.properties
[copy] Copying 1 file to D:\开发工程\GitHub\5_java_example\maventest\test4\test4\testproject\target\classes
[INFO] Executed tasks
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.044s
[INFO] Finished at: Tue May 09 05:08:42 CST 2017
[INFO] Final Memory: 12M/491M
[INFO] -----

```

4、通过操作系统激活配置文件

activation 元素包含下面的操作系统信息。当系统为 windows XP 时，test Profile 将会被触发。

```

<profile>
<id>test</id>
<activation>
<os>
<name>Windows XP</name>
<family>Windows</family>
<arch>x86</arch>
<version>5.1.2600</version>
</os>
</activation>
</profile>

```

现在打开命令控制台，跳转到 pom.xml 所在目录，并执行下面的 mvn 命令。不要使用 -P 选项指定 Profile 的名称。Maven 将显示被激活的 test Profile 的结果。

```
mvn test
```

5、通过文件的存在或者缺失激活配置文件

现在使用 activation 元素包含下面的操作系统信息。当 target/generated-sources/axistools/wsdl2java/com/companyname/group 缺失时，test Profile 将会被触发。

```
<profile>
<id>test</id>
<activation>
<file>
<missing>target/generated-sources/axistools/wsdl2java/
com/companyname/group</missing>
</file>
</activation>
</profile>
```

现在打开命令控制台，跳转到 pom.xml 所在目录，并执行下面的 mvn 命令。不要使用 -P 选项指定 Profile 的名称。Maven 将显示被激活的 test Profile 的结果。

```
mvn test
```

参考：<https://www.cnblogs.com/EasonJim/p/6828743.html>

← Maven 构建生命周期

Maven 仓库 →

 点我分享笔记