# 资源过滤（属性替换）

【参考：<008--Maven__Maven的生命周期.docx>】

Maven属性默认只有在POM中或者settings.xml中才会被解析，如果自定义了一个db.properties文件放在src/main/resources/目录下，并且进行了属性引入，Maven是不会解析的，此时就需要让Maven解析属性引用。

资源文件的处理其实是maven-resources-plugin插件做的事，它默认的行为只是将项目主资源文件复制到代码编译输出目录中，将测试资源文件复制到测试代码编译输出目录中。不过只要做一些配置，该插件就能解析资源文件中的属性引用，即开启资源过滤。

要使用资源过滤，我们需要在项目的POM中指定两件事：

1. 构建配置的filters元素中的属性文件列表；
2. 指定资源目录；

看一个场景：

* 项目中，数据库的连接、用户名、密码等信息属于保密信息，不应该被配置在pom.xml中，而是应该配置在各个环境(dev/test/prod)的服务器的settings.xml文件中。
* 项目中有一个统一的文件，专门存放配置信息，这个文件可以叫config.properties，也可以叫db.properties，文件名随意。
* 其他地方需要引用到配置信息。

下面看一个例子：

机密信息放在settings.xml文件中：%MAVEN\_HOME%/config/settings.xml

[settings.xml]

|  |
| --- |
| <profile>  <id>properties\_profile</id>  <properties>  <user.password>root</user.password>  </properties>  </profile> |

统一的配置文件入口：src/main/resources/properties/db.properties

[db.properties]

|  |
| --- |
| user.username=chenming  user.pwd=${user.password} |

具体使用配置文件地方：src/main/resources/xml/service.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <service>  <name>${user.username}</name>  <password>${user.pwd}</password>  </service> |

如果什么配置都不加，Maven不会对db.properties和service.xml文件中的属性引用进行资源过滤。

此时需要我们配置资源过滤：

[pom.xml]

|  |
| --- |
| <build>  <filters>  <filter>src/main/resources/properties/db.properties</filter>  </filters>  <resources>  <resource>  <directory>src/main/resources/</directory>  <filtering>true</filtering>  </resource>  </resources>  </build> |

* filters：

配置资源过滤，告诉Maven要解析db.properties文件的属性

* resource：

告诉Maven这些目录需要解析引用。