

Maven的核心特征是依赖管理。管理依赖关系变得困难的任务一旦我们处理多模块项目（包含数百个模块/子项目）。Maven提供了一个高度的控制来管理这样的场景。

传递依赖发现

这是很通常情况下，当一个库说A就依赖于其他库说B的情况下，另一个项目C想用A，则该项目需要使用库中B。

在Maven帮助下以避免这样的要求来发现所有需要的库。Maven通过读取依赖项项目文件（pom.xml中），找出它们的依赖等。

我们只需要在每个项目POM定义直接依赖关系。Maven自动处理其余部分。

传递依赖，包括库的图形可能会快速增长在很大程度上。可能出现情况下，当有重复的库。Maven提供一些功能来控制传递依赖程度

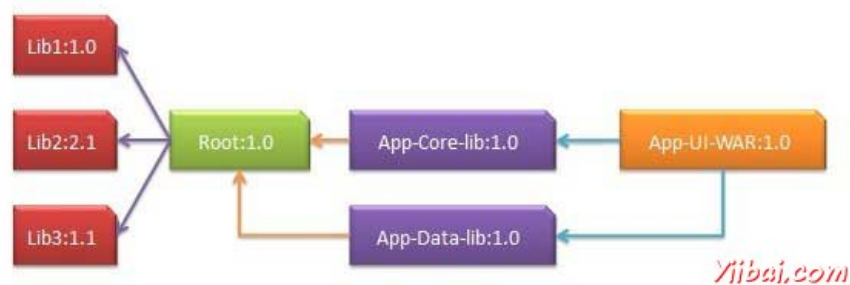
依赖范围

传递依赖发现可以使用各种依赖范围如下文所述受到限制

Scope	描述
compile	This scope indicates that dependency is available in classpath of project. It is default scope.
provided	This scope indicates that dependency is to be provided by JDK or web-Server/Container at runtime.
runtime	This scope indicates that dependency is not required for compilation, but is required during execution.
test	This scope indicates that the dependency is only available for the test compilation and execution phases.
system	This scope indicates that you have to provide the system path.
import	This scope is only used when dependency is of type pom. This scope indicates that the specified POM should be replaced with the dependencies in that POM's section.

依赖关系管理

通常情况下，我们已经一套项目在一个共同的项目下。在这种情况下，我们可以创造让所有的公共依赖一个共同的POM，然后进行分项目POMS为这个POM父。下面的例子将帮助你理解这个概念



以下是上述的依赖图的细节

- APP-UI-WAR依赖于App-Core-lib和 App-Data-lib。
- Root 是 App-Core-lib 和 App-Data-lib 的父类。
- Root 定义LIB1，LIB2，Lib3作为其依赖部分依赖关系。

maven仓库

三者之间的关系是，当我们在项目中依赖一个jar包时，Maven程序会先去本地仓库中找，如果没找到就回去私服找，如果还是没有，最后就回去中央仓库找。其过程如下图：

