

第三章 J2EE应用

3.4 J2EE应用部署

- J2EE应用部署概述
- J2EE应用中的模块及应用部署
- J2EE应用开发生命周期
- 打包和部署的角色分配

第三章 J2EE应用

3.4.1 J2EE应用部署概述

- 应用组件需要用基于XML的部署描述文件来描述，而且部署描述文件中还包含了容器需要的其它信息，以便容器能够有效地管理组件。
- 在J2EE应用中，一般会把组件打包形成档案文件。
- 对于应用开发和部署的过程，J2EE规范定义了如下的步骤：
应用组件开发、把应用组件合成模块、把模块组合成应用、应用部署。

第三章 J2EE应用

3.4.2 J2EE应用中的模块

J2EE应用中的模块主要包括

- Web模块
- EJB模块
- 客户应用模块

第三章 J2EE应用

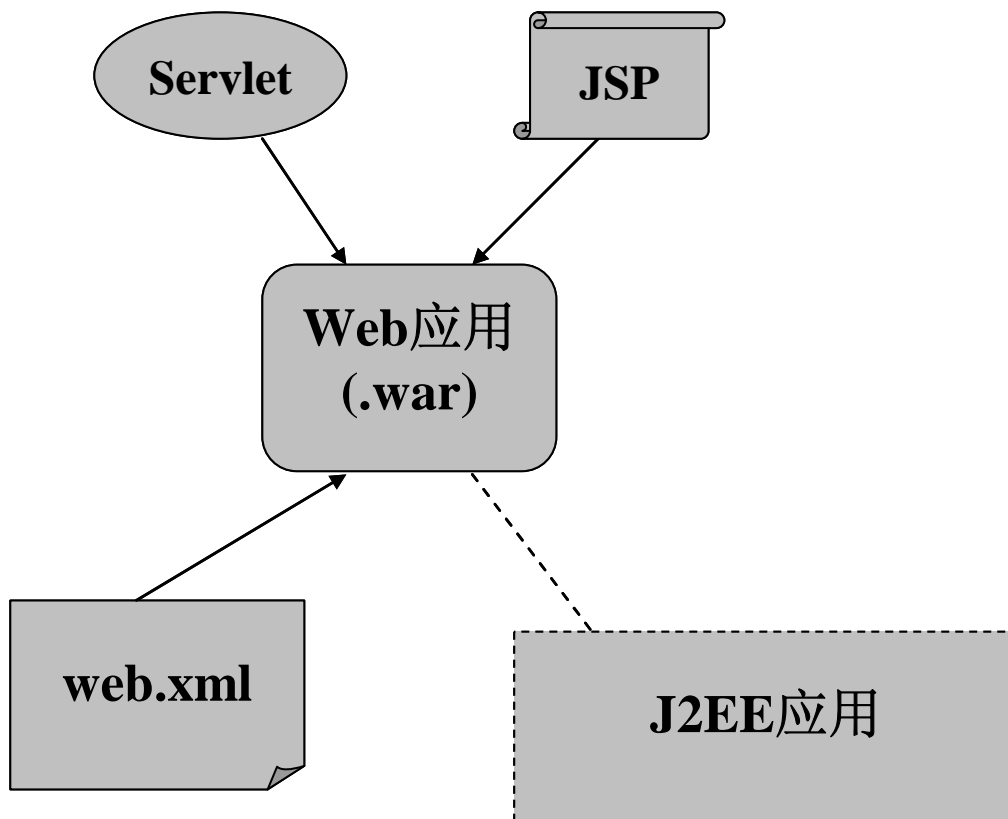
3.4.2.1 Web模块

(1) 概述

- Web模块是一个**可部署**的单元，它里面包括了Servlet、JSP页面、JSP标签库、JAR库文件、HTML/XML文档和其它的公用资源，如图像、Applet等。
- 一个Web模块打包形成一个Web档案文件，即**war**文件。
- war文件包含了一个**WEB-INF**目录，其中以**web.xml**文件形式包含了部署。

第三章 J2EE应用

(2) Web模块的打包过程



第三章 J2EE应用

(3) Web应用的目录层次

一个Web应用程序的目录由两部分组成

- 私有的名为**WEB-INF**的目录

其中的资源访问客户不能够下载，这个目录中包括

- 一个web.xml文件，即部署描述器。

- classes子目录，包含.class文件。

- lib子目录，用来包含应用程序使用的任何Jar文件。

- 包含**公有**资源的目录，包括其它所有的子目录。

第三章 J2EE应用

(4) 创建、部署和执行Web应用程序的过程

- 开发Web组件**代码**。
- 开发Web应用程序**部署描述符**。
- 编译Web应用程序**组件**和组件引用的**助手类**。
- 有选择地将应用程序**打包**成可部署的单位。
- 将应用程序**部署**到Web容器中。
- 访问引用该Web应用程序的 **URL**。

第三章 J2EE应用

(5) Web归档文件.WAR

- 为了创建.WAR文件，应该进入应用文件放置的目录，运行下述的命令：

```
jar -cf myWebApp.war *
```

- 用下述的命令来查看WAR文件的内容：

```
jar -tvf myWebApp.war
```

- WAR文件代表的是Web应用程序，而不仅是一个类的归档文件。
- 在开发阶段不适合使用WAR文件，到了应用程序部署的最后阶段，这时才开始打包并进入最后的产品阶段。

第三章 J2EE应用

(6) 把客户请求映射到应用程序上

- 在Web应用中，每一个Web应用程序与一个上下文相联系，并且Web应用程序中所有的资源都相对于该上下文而存在。

例如：一个应用上下文为myWebApp，则访问它的URL为：

`http://localhost:8080/myWebApp/index.html`

- 每个Web应用程序都必须被映射到一个唯一的URL路径前缀。这种路径前缀为Web应用程序中的所有资源定义了一个分离的名字空间。

第三章 J2EE应用

(7) 把客户请求映射到Servlet上

可以用URL路径把请求映射到servlet上，这个映射需要按下述步骤来进行：

- 定义到Servlet的别名。在部署描述文件Web.xml中加入别名定义。

例如：将myWebApp中的servlet ShoppingCart映射为一个短的别名Shopping

```
<servlet>  
  <servlet-name>Shopping</servlet-name>  
  <servlet-class>ShoppingCart</servlet-class>  
</servlet>
```

第三章 J2EE应用

- 把URL路径映射到servlet别名。例如，把上例中的Servlet别名Shopping映射为URL路径/start，需要在Web.xml中作如下的定义。

```
<servlet>
  <servlet-name>Shopping</servlet-name>
  <servlet-class>ShoppingCart</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>Shopping</servlet-name>
  <url-pattern>/start/*</url-pattern>
</servlet-mapping>
```

第三章 J2EE应用

(8) 部署配置(web.xml)

所有的Web应用程序的部署描述文件应包含的**DOCTYPE**声明如下:

```
<!DOCTYPE web-app  
PUBLIC "-//Sun Microsystems,Inc.//DTD Web Application 2.2//EN"  
"http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">。
```

第三章 J2EE应用

实例：假设存在一个web应用程序，根目录为test。

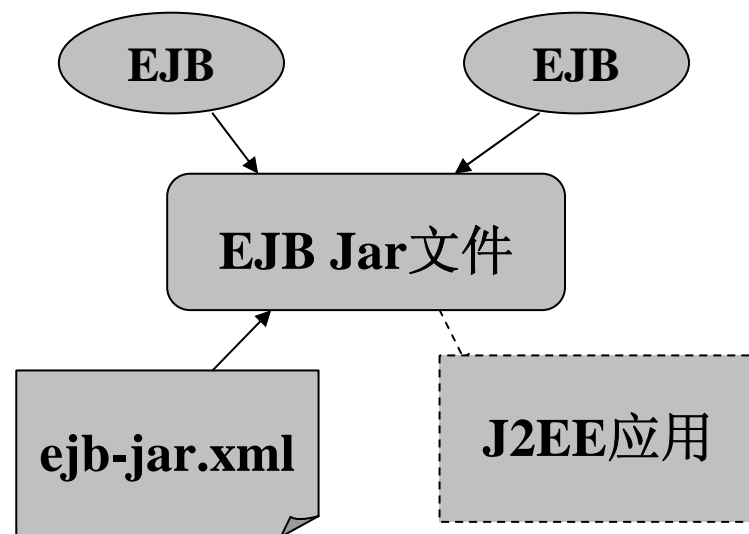
打包和部署步骤包括：

- (1) 启动windows自带的Dos环境，进入test目录；
- (2) 在命令行.../test>后输入jar -cf myWebApp.war *；
- (3) 把myWebApp.war部署到Tomcat服务器的webApp目录下；
- (4) 在浏览器地址栏输入<http://localhost:8080/webApp/>入口。

第三章 J2EE应用

3.4.2.2 EJB模块

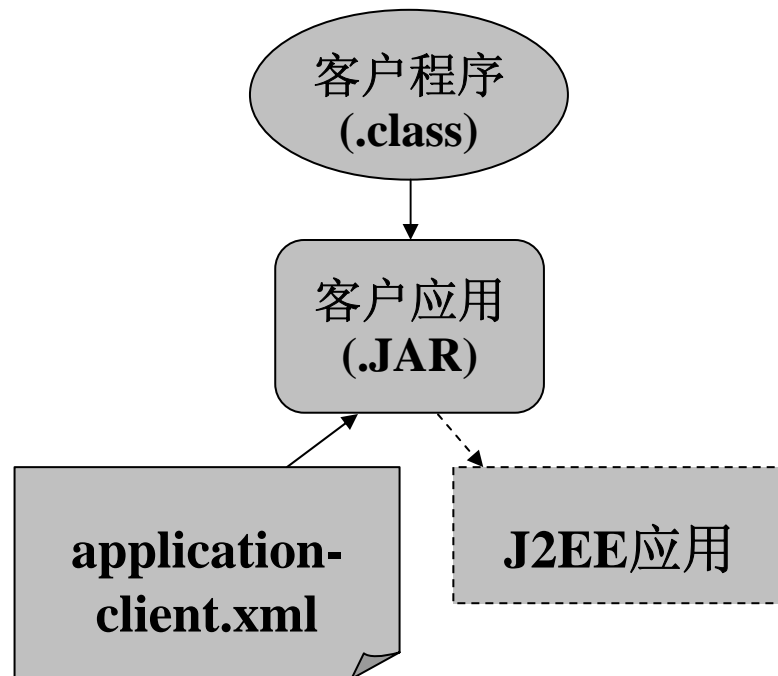
- EJB可被打包成一个压缩Jar文件，其中包含了EJB组件、相关的Jar文件和资源，以及一个**ejb-jar.xml**部署描述文件。
- JAR文件的内容具体列举如下：
EJB类文件（.class）和接口、其它相关类的类文件和接口、部署描述文件。



第三章 J2EE应用

3.4.2.3 客户应用模块

用作J2EE应用客户端的一个Java应用可以打包成一个Java归档文件.jar，其中包含一个application-client.xml部署描述文件。也称这个应用客户端为Java模块。



第三章 J2EE应用

3.4.2.4 把模块组合成J2EE应用

Web模块在应用的部署描述文件中进行配置时的语法如下:

```
<module>
```

```
<web>
```

```
  <web-uri>myWebApp.war</web-uri>
```

```
  <context-root>myWebApp/</context-root>
```

```
</web>
```

```
</module>
```


第三章 J2EE应用

将EJB模块加入应用的语法如下：

```
<module>
```

```
<ejb>pathToEJB.jar</ejb>
```

```
</module>
```

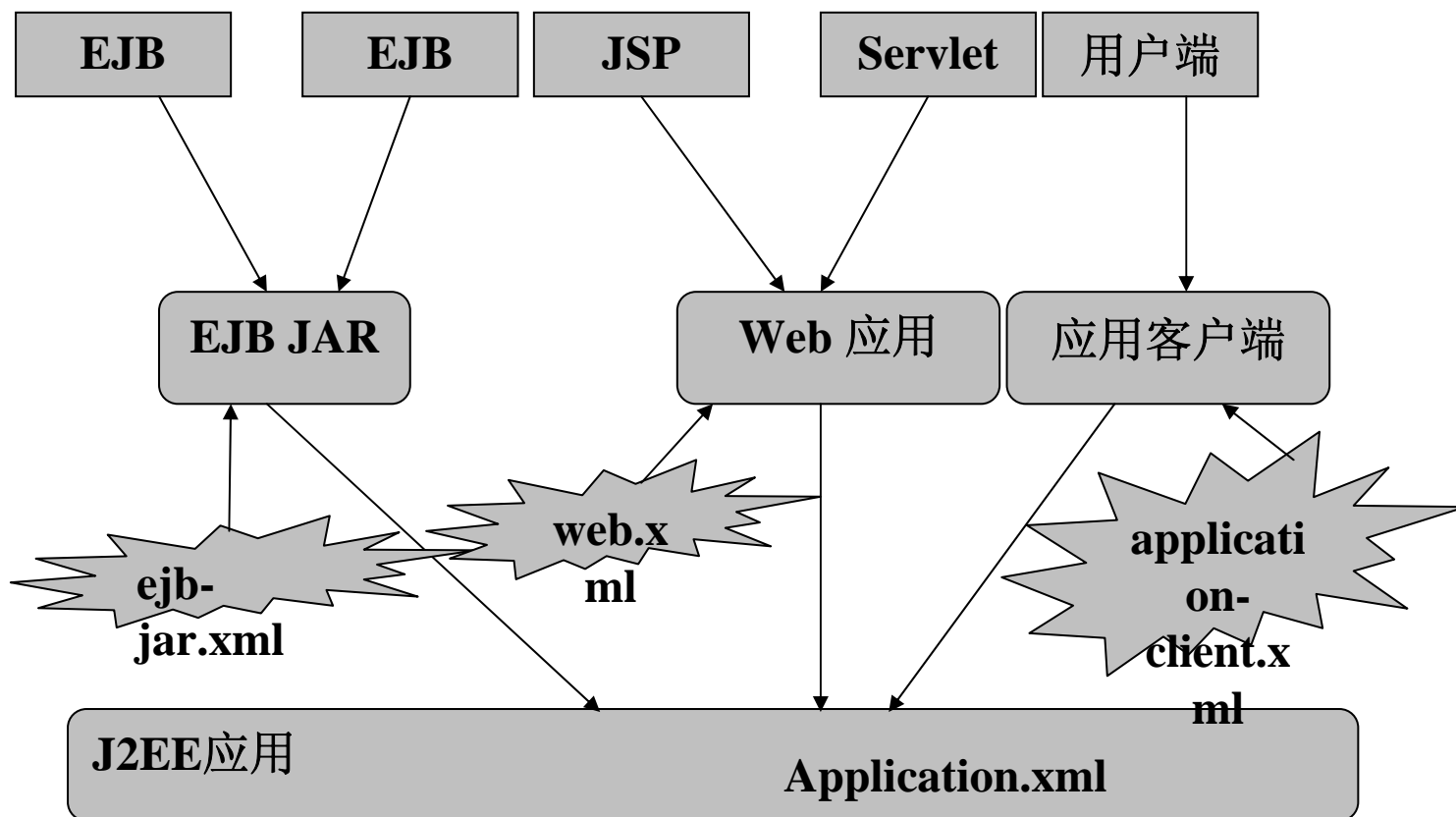
Java模块加入J2EE应用的语法如下：

```
<module>
```

```
<java>pathToClient.jar</java>
```

```
</module>
```

第三章 J2EE应用



第三章 J2EE应用

application.xml

- J2EE应用的部署描述文件**application.xml**。
- 指明了J2EE应用的如下属性：
在图形工具中表示J2EE应用的**图标**。
应用名。
应用的**持久描述器**。
应用的**模块**。
应用使用的**安全角色**。

第三章 J2EE应用

<application>

<icon>

<small-icon>path/to/icon.gif</small-icon>

<large-icon>path/to/icon.gif</large-icon>

</icon>

<display-name>theName</display-name>

<description>a description</description>

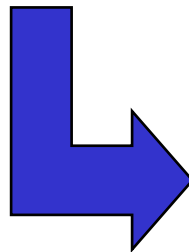
<module>

<ejb>path/to/EJB.jar</ejb>

<!-- 或者 -->

<java>path/to/client.jar</java>

<!-- 或者 -->



<web>

<web-uri>path/to/webapp.war</web-uri>

<context-root>theDir/</context-root>

</web>

</module>

<security-role>

<description>description</description>

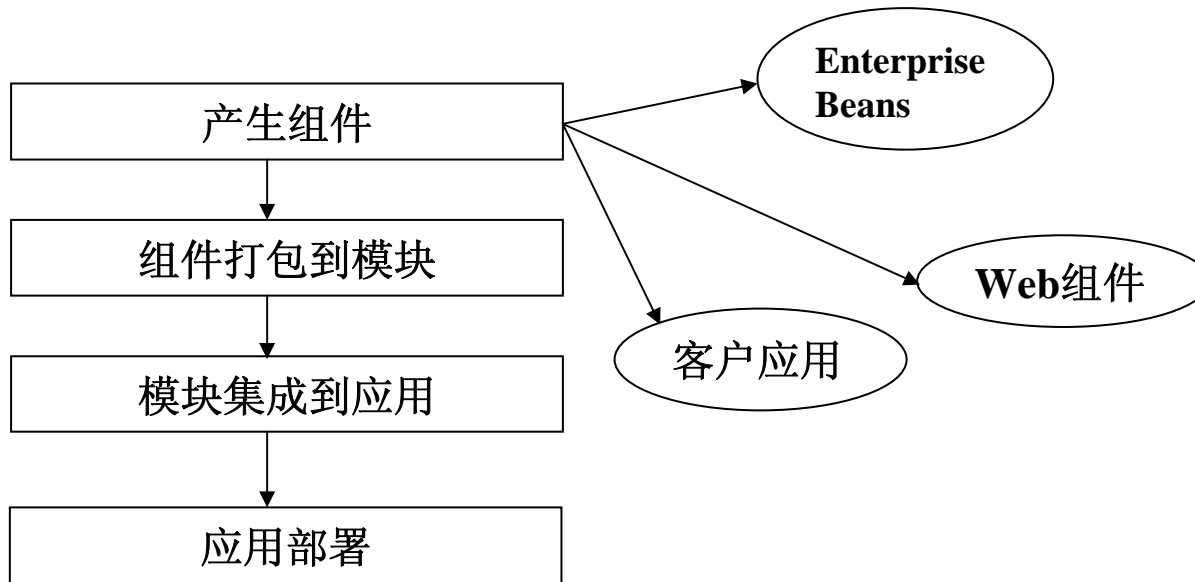
<role-name>name</role-name>

</security-role>

</application>

第三章 J2EE应用

3.4.3 J2EE应用开发生命周期



第三章 J2EE应用

3.4.4 打包和部署的角色分配

