服务器知识手册（六）

狄登峰 主编

需要全本请访问 www.ertongbook.com

目　　录

Web 服务器.....1
J2EE 概述 1
Cluster 之架设(NAT 形式) 22
使用 Translator 模式构建更好的网站 29
Webalizer 安装使用指南 43
Web 服务器安全指南 49
构造公司内部 Web、FTP 服务器 55
ApacheWeb 让 JSP"动"起来 58
构建高负载 Web 服务器 68
通过内核 httpd 实现 web 服务加速 69
iptables 应用之动态 DNS 74
利用 JAVA 语言实现支持视频点播的 WEB 服务器 78
Apache 服务器的用户认证 90
Apache 的 httpd.conf 文件常用指令解释 97
安全维护 IISASP 站点的高级技巧 100
OPENSSL 服务-安全信息传输 107

Web 服务器

J2EE 概述

今天，越来越多的开发者想要编写企业级的分布式的事务处理应用程序，而这些应用程序必须可以发挥速度、安全性和服务器端技术的可靠性。如果你已经在这一领域从事工作，你应该了解在现在这个高速发展、要求苛刻的电子商务和信息技术的世界中，企业级的应用程序必须具有以下特点：花费更少的金钱、具有更快的速度、占用更少的资源。

为了减少费用，快速设计和开发企业级的应用程序，Java2Platform,EnterpriseEdition(J2EE)技术提供了一个基于组件的方法来设计、开发、装配和部署企业级应用程序。J2EE 平台提供了一个多层结构的分布式的应用程序模型，该模型具有重用组件的能力、基于扩展标记语言(XML)的数据交换、统一的安全模式和灵活的事务控制。你不仅可以比以前更快地发表对市场的新的解决方案，而且你的独立于平台的基于组件的 J2EE 解决方案不再受任何提供商的产品和应用程序编程界面(APIs)的限制。提供商和买主都可以自己选择最合适于它们的商业应用和所需技术的产品和组件。

1、分布式的多层应用程序

J2EE 平台使用了一个多层的分布式的应用程序模型。应用程序的逻辑根据其实现的不同功能被封装到组

件中，组成 J2EE 应用程序的大量应用程序组件根据在其所属的多层的 J2EE 的环境中所处的层被安装到不同的机器中。图 1-1 表示了两个多层的 J2EE 应用程序根据下面的描述被分为不同的层。在图 1-1 中涉及的 J2EE 应用程序的各个部分将在 J2EE 组件中给出详细描述。

1、运行在客户端机器的客户层组件。

2、运行在 J2EE 服务器中的 Web 层组件。

3、运行在 J2EE 服务器中的商业层组件。

4、运行在 EIS 服务器中的企业信息系统(EIS)层软件。

尽管从图 1-1 中可以看到 J2EE 应用程序既可以是三层结构，也可以是四层结构，但是我们通常将 J2EE 应用程序的多层结构考虑为三层结构。这是因为它们分布在三个不同的位置：客户端机器、J2EE 服务器机器和在后端的传统的机器。三层结构的应用程序可以理解为在标准的两层结构的客户端/服务器模式的客户端应用程序和后端存储资源中间增加了一个多线程的应用程序服务器。

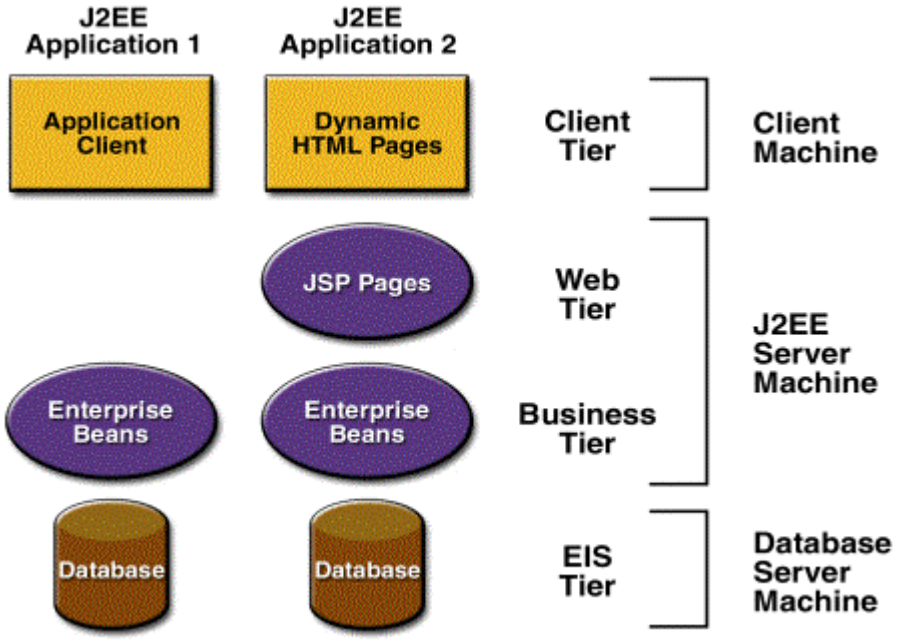

图 1-1：多层结构的应用程序

2、J2EE 组件

J2EE 应用程序由组件组成。一个 J2EE 组件就是一个自带功能的软件单元，它随同它相关的类和文件被装配到 J2EE 应用程序中，并实现与其它组件的通信。J2EE 规范是这样定义 J2EE 组件的：

客户端应用程序和 applet 是运行在客户端的组件。

JavaServlet 和 JavaServerPages(JSP)是运行在服务器端的 Web 组件。

EnterpriseJavaBean(EJB)组件(enterprisebean)是运行在服务器端的商业软件。

J2EE 组件由 Java 编程语言写成，并和用该语言写成的其它程序一样进行编译。J2EE 组件和"标准的"Java

需要全本请访问 www.ertongbook.com

类的不同点在于：它被装配在一个 J2EE 应用程序中，具有固定的格式并遵守 J2EE 规范，它被部署在产品中，由 J2EE 服务器对其进行管理。

3、J2EE 客户端

一个 J2EE 客户端既可以是一个 Web 客户端，也可以是一个应用程序客户端。

3.1Web 客户端

一个 Web 客户端由两部分组成：由运行在 Web 层的 Web 组件生成的包含各种标记语言(HTML、XML 等等)的动态 Web 页面和接受从服务器传送来的页面并将它显示出来的 Web 页面。

一个 Web 客户端有时被称之为瘦客户端。瘦客户端一般不做象数据库查询、执行复杂的商业规则及连接传统应用程序这样的操作。当你使用一个瘦客户端时，象这样的重量级的操作被交给了在 J2EE 服务器执行的 enterprisebean。这样就可以充分发挥 J2EE 服务器端技术在安全性、速度、耐用性和可靠性方面的优势。

3.2Applets

从 Web 层接收的一个 Web 页面可以包含内嵌的 applet。一个 applet 是一个用 Java 编程语言编写的小的客户端应用程序，它在安装在 Web 浏览器中的 Java 虚拟机中运行。然而，为了在 Web 浏览器中成功地运行 applet，客户端系统很可能需要 Java 插作和安全策略文件。

Web 组件是用来建立一个 Web 客户端程序的首选的 API，因为这样在客户端系统中就不需要插件和安全策略文件。同样的，使用 Web 组件可以有效地改善应用程序设计，因为它们提供了一个将应用程序设计和 Web 页面设计有效分离的途径。Web 页面的设计者可以不必关

心 Java 编程语言的语法就能很好地完成自己的工作。

3.3 应用程序客户端

一个 J2EE 应用程序客户端运行在客户端机器上，它使得用户可以处理需要比标记语言所能提供的更丰富的用户界面的任务。具有代表性的是用 Swing 或抽象窗口工具包(AWT)API 建立的图形用户界面(GUI)，但是一个命令行界面也是当然可能的。

应用程序客户端直接访问运行在商业层的 enterprisebean。然而，如果应用程序需要授权，一个 J2EE 应用程序客户端可以打开一个 HTTP 连接来与一个运行在 Web 层的 servlet 建立通信。

4、JavaBeans 组件体系结构

服务器层和客户层也可以包含以 JavaBean 组件体系结构(JavaBeans 组件)为基础的组件来管理在一个应用程序客户端或 applet 与运行在 J2EE 服务器上的组件之间的数据流动以及服务器端组件与数据库之间的数据流动。在 J2EE 规范中 JavaBeans 组件不被认为是 J2EE 组件。

JavaBeans 组件具有实例变量以及用来访问实例变量中的数据的 get 方法和 set 方法。作这种用途的 JavaBeans 组件在设计和执行时相当简单，但是它必须遵守 JavaBeans 组件体系结构的命令和设计惯例。

5、J2EE 服务器通信

图 1-2 显示了客户层组成的多种方式。客户端可以直接和运行在 J2EE 服务器中的商业层进行通信。如果是一个运行在浏览器中的客户端，也可以通过运行在 Web 层中的 JSP 页面和 Servlet 进行这种通信。

你的 J2EE 应用程序是采用瘦客户端还是胖客户端。

要作出这样的决定，你应该明白将胖客户端是将功能留在客户端，使它与用户更接近，而瘦客户端是将功能的实现尽可能地交给服务器。由服务器处理更多的功能，就更容易分发、部署和管理应用程序；而将更多的功能留在客户端对于有经验的用户也许是个明智的选择。

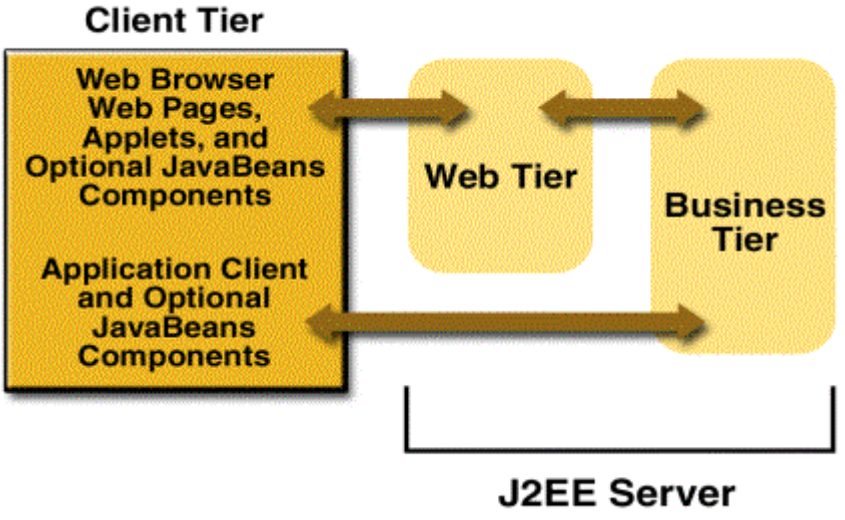

图 1-2 服务器通信

5.1Web 组件

J2EE 的 Web 组件既可以是 servlet 也可以是 JSP 页面。Servlets 是一个 Java 编程语言类，它可以动态地处理请求并作出响应。JSP 页面是一个基于文本的文档，它以 servlet 的方式执行，但是它可以更方便建立静态内容。

在装配应用程序时，静态的 HTML 页面和 applet 被绑定到 Web 组件中，但是它们并不被 J2EE 规范视为 W

eb 组件。服务器端的功能类也可以被绑定到 Web 组件中，与 HTML 页面一样，它们也不被 J2EE 规范视为 Web 组件。

正如图 1-3 中所示，和客户层一样，Web 层也可以包含一个 JavaBeans 组件以管理用户的输入并将输入发送到运行在商业层的 enterprisebean 进行处理。

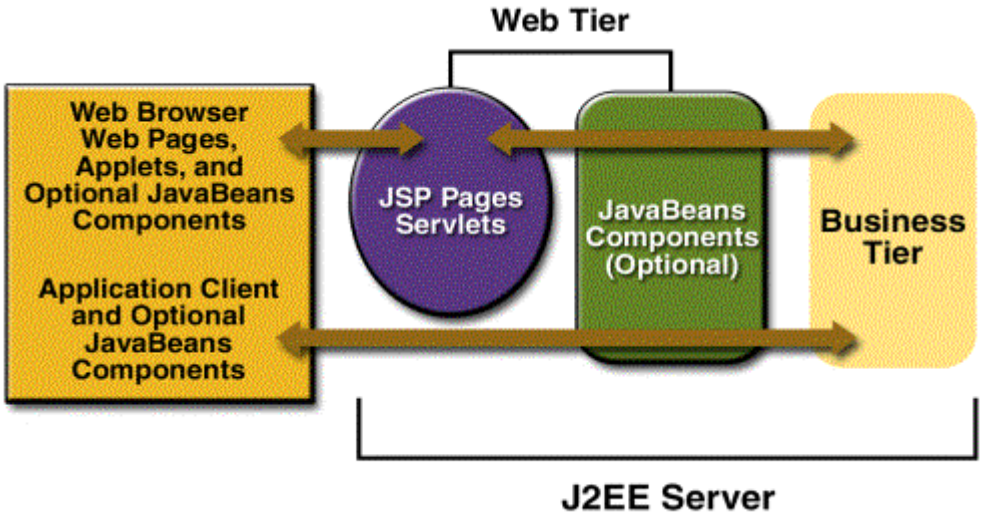

图 1-3Web 层和 J2EE 应用程序

5.2 商业组件

商业代码，表示了例如银行、零售和财政这样的特定的商业领域的相适应的逻辑。它由运行在商业层的 enterprisebean 处理。图 1-4 显示了一个 enterprisebean 如何从客户端接受数据，对它进行处理(如果需要)，并将其发送到企业信息系统层以作存储。一个 enterprisebean 也可以从存储器获取数据，对它进行处理(如果需要)，并将其发送到客户端应用程序。

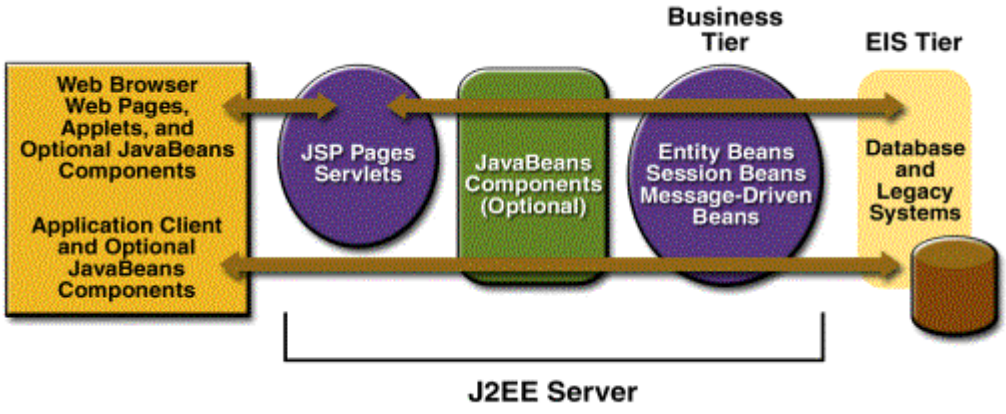

图 1-4 商业层和 EIS 层

有三种类型的 enterprisebeans：sessionbeans、entity beans 和 message-drivenbeans。一个 sessionbean 描述了与客户端的一个短暂的会话。当客户端的执行完成后，session bean 和它的数据都将消失。与些相对应的是一个 entitybean 描述了存储在数据库的表中的一行的持久稳固的数据。如果客户端终止或者服务结束，底层的服务会负责 entitybean 数据的存储。

一个 message-drivenbean 结合了一个 sessionbean 和一个 Java 信息服务(JMS)信息监听者的功能，它允许一个商业组件异步地接受 JMS 消息。这份指南只介绍 entitybean 和 sessionbean。有关 message-drivenbean 的介绍，请参看 Java 消息服务指南：

http://java.sun.com/products/jms/tutorial/index.html

6、企业信息系统层

企业信息系统层处理企业信息系统软件并包含诸如企业资源计划(ERP)、主机事务处理、数据库系统和其它传统系统这样的底层系统。J2EE 应用程序组件可能需要

访问企业信息系统，例如是获得一个数据库连接。

6.1J2EE 容器

通常，瘦客户端的多层应用程序是很难编写的，这是因为这得包括许多行复杂的代码以处理事务、状态管理、多线程、资源池和其它复杂的底层详细资料。基于组件并与平台无关的 J2EE 体系结构使得 J2EE 应用程序易于编写，这是因为商业逻辑被封装到可重用的组件中，此外，J2EE 服务器以容器的形式为每一个组件类型提供底层服务。因为我们不需要自己开发这些服务，这使我们可以全力以赴地着手处理商业问题。

6.2 容器服务

容器是一个组件和支持组件的底层平台特定功能之间的接口，在一个 Web 组件、enterprisebean 或者是一个应用程序客户端组件可以被执行前，它们必须被装配到一个 J2EE 应用程序中，并且部署到它们的容器。

装配的过程包括为 J2EE 应用程序中的每一个组件以及 J2EE 应用程序本身指定容器的设置。容器设置定制了由 J2EE 服务器提供的底层支持，这将包括诸如安全性、事务管理、Java 命名目录接口(JNDI)搜寻以及远程程序连接。下面是其中的主要部分：

1、J2EE 的安全性模式可以让你对一个 Web 组件或 enterprisebean 进行配置以使得只有授权用户访问系统资源。

2、J2EE 的事务模式可以让你指定方法之间的关系以组成一个单个的事务，这样在一个事务中的所有方法将被视为一个单一的整体。

3、JNDI 搜寻服务为企业中的多种命名目录服务提供一个统一的接口，这使得应用程序组件可以访问命名

需要全本请访问 www.ertongbook.com

目录服务。

4、J2EE 远程连接模式管理客户端和 enterprisebean 之间的底层通信。在一个 enterprisebean 被建立后，客户端在调用其中的方法时就象这个 enterprisebean 就运行在同一个虚拟机上一样。

实际上，J2EE 体系结构提供了可配置的服务意味着在相同的 J2EE 应用程序中的应用程序组件根据其被部署在什么在地方在实际运行时会有所不同。例如，一个 enterprisebean 可能在一个产品环境中拥有包含访问数据库数据的某种级别的安全性设置，而在另一个产品环境中是另一个访问数据库的级别。

容器还管理诸如一个 enterprisebean 和 servlet 的生存周期、数据库连接资源池以及访问在 J2EEAPIs 中介绍的 J2EE 平台 API 这样不能配置的服务。尽管数据持久化是一个不能配置的服务，但是 J2EE 体系统结构允许你在你想要获得比默认的容器管理持久化所能提供更多的控制时，通过在你的 enterprisebean 执行中包含适当的代码以重载容器管理持久化。例如，你可以使用 bean 管理持久化以实现你自己的 finder(查找)方法或者是建立一个定制的数据库缓冲区。

6.3 容器类型

部署时会将 J2EE 应用程序组件安装到 J2EE 容器中，就象图 1-5 中所示那样。

1、J2EE 服务器：是 J2EE 产品的运行部分。一个 J2EE 服务器提供 EJB 容器和 Web 容器。EnterpriseJavaBeans(EJB)容器：管理 J2EE 应用程序的 enterprisebean 的执行。Enterprisebean 和它的容器运行在 J2EE 服务器中。

2、Web 容器：管理 J2EE 应用程序的 JSP 页面和 s

ervlet 组件的执行。Web 组件和它的容器也运行在 J2EE 服务器中。

3、客户端应用程序容器：管理应用程序客户端组件的运行。应用程序客户端和它的容器运行在客户端中。

4、Applet 容器：管理 applet 的执行。由运行在客户端的一个 Web 浏览器和 Java 插件一同组成。

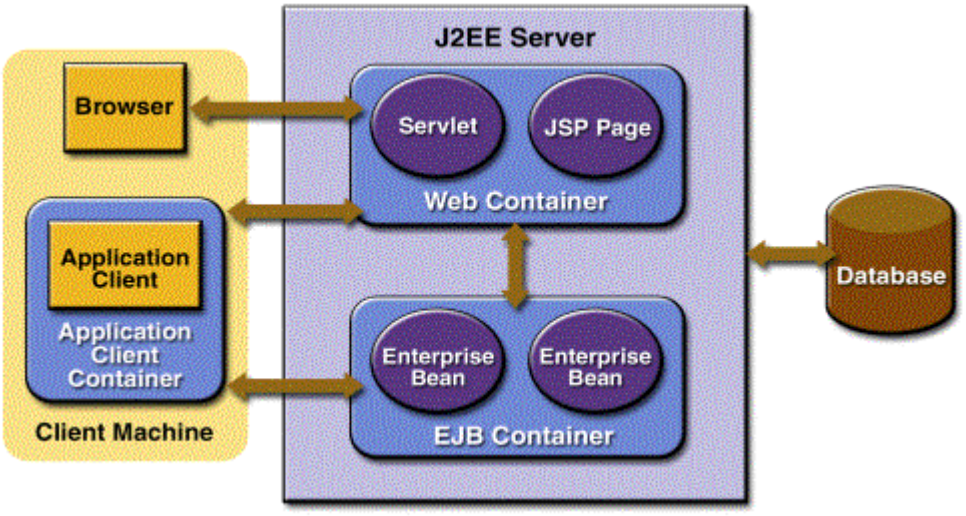

图 1-5J2EE 服务器和容器

7、封装

J2EE 组件被分别打包并绑定到一个 J2EE 应用程序中以供部署。每一个组件、它的诸如 GIF、HTML 文件和服务器端功能类这样的相关文件以及一个部署说明组成了一个模块并被添加到 J2EE 应用程序中。一个 J2EE 应用由一个或几个 enterprisebean 组件模块、Web 组件模块或应用程序客户端组件模块组成。根据不同的设计需求，最终的企业解决方案可以是一个 J2EE 应用程序，

也可以由两个或更多的 J2EE 应用程序组成。

一个 J2EE 应用程序以及它的每一个模块有它自己的部署说明。一个部署说明就是一个具有.xml 扩展名的 XML 文件，它描述了一个组件的部署设置。例如，一个 enterprisebean 模块的部署说明，描述了一个 enterprisebean 的事务属性和安全性授权。因为部署说明信息是公开的，因此它可以被改变而不必修改 bean 的源代码。在运行时，J2EE 服务器将读取这个部署说明并遵照执行。

一个 J2EE 应用以及它的所有模块被提交到一个 EnterpriseARchive(EAR)文件中。一个 EAR 文件就是一个具有.ear 扩展名的标准的 JavaArchive(JAR)文件。在 J2EE SDK 应用程序部署工具的 GUI 版本中，你首先建立一个 EAR 文件，并在其中添加 JAR 文件和 WebArchive(WAR)文件。然而，如果你使用的是命令行的打包工具，你必须首先建立 JAR 和 WAR 文件，然而才是建立 EAR 文件。有关 J2EESDK 工具的详细介绍请参看工具。

1、每一个 EJBJAR 文件包含一个部署说明、enterprisebean 文件以及相关的文件。

2、每一个应用程序客户端 JAR 文件包含一个部署说明、应用程序客户端的类文件以及相关的文件。

3、每一个 WAR 文件包含一个部署说明、Web 组件文件以及相关的资源。

使用模块和 EAR 文件使得运用同一组件以装配许多不同的 J2EE 应用程序成为可能。不需要额外的编程工作，你唯一要做的是在 J2EEEAR 文件中添加各种 J2EE 模块。

8、开发角色

可重用的模块使得将应用程序的开发过程和部署过

程分成不同的角色成为可能。这样，不同的人或公司可以在这一过程的各个部分承担不同的任务。

最先的两个角色承担购买和安装 J2EE 产品和工具的任务。在购买和安装好软件之后，应用程序组件提供商就开始开发 J2EE 组件，应用程序装配者负责装配，而应用程序部署者负责部署。在一个大的组织中，每个这样的角色可能对应于不同的个人和小组。作这样的分工是因为前一个角色将会输出一个可移植的文件，而这将是后一个角色的输入。例如，在应用程序组件提供商阶段，一个 enterprisebean 软件开发者提交 EJBJAR 文件。而在应用程序装配者阶段，另一些开发者将这些 EJBJAR 文件组合到一个 J2EE 应用程序中并将其保存为一个 EAR 文件。在应用程序部署者阶段，消费者站点的系统管理员使用 EAR 文件将 J2EE 应用程序安装到 J2EE 服务器中。

不同的角色不一定总得由不同的人来执行。例如，你是在一个小公司工作或者你是从事一个样本程序的原型开发，你也可能需要执行每一个阶段的任务。

9、J2EE 产品提供者

J2EE 产品提供者是设计并提供实现 J2EE 规范所定义的 J2EE 平台、API 和其它功能的公司。这一般是指操作系统、数据库系统、应用程序服务器或 Web 服务器的卖主，它们依照 J2EE 的规范实现 J2EE 平台。

9.1 工具提供者

工具提供者是指建立组件提供者、装配者和部署者所使用的开发、装配和打包的工具的公司或个人。有关 J2EESDK1.3 可用工具的详细信息参见工具。

9.2 应用程序组件提供者

应用程序组件提供者是指建立 J2EE 应用程序所使用的 Web 组件、enterprisebean、applet 或应用程序客户端的公司或个人。

9.3EnterpriseBean 开发者

一个 enterprisebean 开发者执行下面的任务并递交一个包含 enterprisebean 的 EJBJAR 文件：

1、编写并编译源代码

2、详细说明部署描述

3、将.class 文件和部署描述绑定到一个 EJBJAR 文件中

9.4Web 组件开发者

一个 Web 组件开发者执行下面的任务并提交一个包含 Web 组的 WAR 文件：

1、编写并编译 servlet 源代码

2、编写 JSP 和 HTML 文件

3、详细说明 Web 组件的部署描述

4、将.class、.jsp、.html 和部署描述绑定到 WAR 文件中

9.5J2EE 应用程序客户端开发者

一个应用程序客户端开发者执行下面的任务并提交一个包含 J2EE 应用程序客户端的 JAR 文件：

1、编写并编译源代码

2、详细说明客户端的部署描述

3、将.class 文件和部署描述绑定到 JAR 文件中

9.6 应用程序装配者

应用程序装配者是从组件提供者接受应用程序组件 JAR 文件并将其装配到一个 J2EE 应用程序 EAR 文件中的公司或个人。装配者或部署者可以直接编辑部署说明

或者使用可以根据交互式的正确地添加 XML 标志的工具。一个软件开发者执行下列任务并递交一个包含 J2EE 应用程序的 EAR 文件：

1、将上一阶段建立的 EJBJAR 文件和 WAR 文件装配到一个 J2EE 应用程序(EAR)文件中。

2、详细说明有关 J2EE 应用程序的部署说明。

3、检验 EAR 文件中的组件是否遵守 J2EE 规范。

9.7 应用程序部署者和系统管理员

应用程序部署者和系统管理员可以是公司或个人，他们配置和部署 J2EE 应用程序，管理 J2EE 应用程序在其中运行的计算机和网络这些低层结构，并对运行环境进行监控。他们的任务可能包括这样的一些事：设置事务控制、安全属性并指定数据库连接。

在配置时，部署者按照由应用程序组件提供者提供的指示以解决外部的支持、指定安全设置并定义事务属性。在安装时，部署者将应用程序组件装入服务器中并生成特定容器的类和接口。

一个部署者/系统管理员执行下列任务以安装和配置一个 J2EE 应用程序：

1、将上一阶段生成的 J2EE 应用程序(EAR)文件添加到 J2EE 服务器中

2、根据运行环境通过修改 J2EE 应用程序的部署说明对 J2EE 应用程序进行配置。

3、检验 EAR 文件的内容是否遵守 J2EE 规范

4、部署(安装)J2EE 应用程序 EAR 文件到 J2EE 服务器中

10、相关执行软件

J2EESDK 是非经营的 J2EE 平台的操作定义，它由

需要全本请访问 www.ertongbook.com