

第四部分 高级工具

第14章 部署和配置应用程序

在完成了应用程序原型，调查了用户的意见，重新设计和确保了应用程序的安全，通过了编译、测试和调试之后，剩下要做的事情是什么呢？

在完成应用程序之后所要做的事情，是将应用程序交付给用户的过程中真正关键的问题，这也是出于自己爱好或为练习而编程与开发软件产品之间的区别。你已经在开发工具中运行了应用程序，现在是建立 Application Server和在因特网上运行的时候了。

14.1 部署应用程序

部署软件产品就是将它分发给用户。它包括将软件打包并安装到用户的工作环境或网络上。

首先你必须决定部署模式：客户机 / 服务器模式或三方部署模式。三方部署模式利用 Application Server作为客户和数据库服务器的第三方。

客户机/服务器部署模式需要将一个安装程序包安装到每一个客户机上，包括 Oracle Developer客户机的运行时间引擎、可执行的应用程序和其他各种各样运行应用程序所需要的文件。

三方发送模式需要在中心服务器上安装 Oracle Developer Server驱动程序、应用程序可执行文件及相关文件、作为 Oracle Application Server一部分的HTTP接收器或者其他Web服务器。

三方部署模式的好处是非常明显的，它便于更多用户使用应用程序。因为只需要安装一次，故不必制作安装系统，也不必四处奔走为所有的用户安装。此外，客户机仅需要网络浏览器或Java程序浏览器即可运行应用程序，而不需整个 Oracle Developer和与应用程序相关的软件。这可节省计算机空间去做其他事情，而且不必在客户机维护该软件，从而减少系统的维护和管理费用。

14.2 复杂的Web

可以想象一下，如果发送一个应用软件所要做的就是告诉用户一个它的名字，那该多好。这就是因特网的承诺以及 Oracle Developer提供的多层部署平台 Oracle Developer Server的能力。

Web浏览器是连接网络的一个应用程序，通常是在因特网或企业内部网上利用 HTML主页浏览文件。Sun Microsystems加上Java这种面向网络的程序设计语言，生成一种真正有效的部署软件的方法。这样就可以通过浏览 HTML网页、轻点按钮运行Java程序来使用应用程序。

Oracle Developer Server是一个工作平台，它帮助你通过 Web浏览器部署应用程序。并且

它使应用程序的部署变得更简单、更廉价,让你创造出更多优秀实用的 Web应用程序。Oracle Developer Server可以充分利用因特网的优点和客户/服务器的计算功能。使用基于 Web的应用程序可以达到脱离工作平台的高度,通过浏览器可以在任何一部机器上运行你的应用程序,而不需要在不同的操作系统下重新编译。可以得到多种功能的表单和报表,包括交互式表单、HTML和Adobe Acrobat报表、下钻报表和图形、动态报表和报表队列监控。

图14-1说明了Oracle Developer Server Forms的组织结构。客户工作站运行一个可运行Java的Web浏览器。用户对HTML页面输入一个URL进入应用程序开始页面(例如http://www.talbotfarms.com/hr)。当用户在浏览器中运行一个表单文件时,该浏览器就下载 Oracle Developer Forms Java Client小应用程序(关于用户界面说明的一个小程序代码)并运行它。该程序向Forms Server Listener发出请求,要求服务器运行应用程序。Listener连接到Forms Runtime Server上,并传送运行时间参数,包括用户名和数据库服务器的口令。

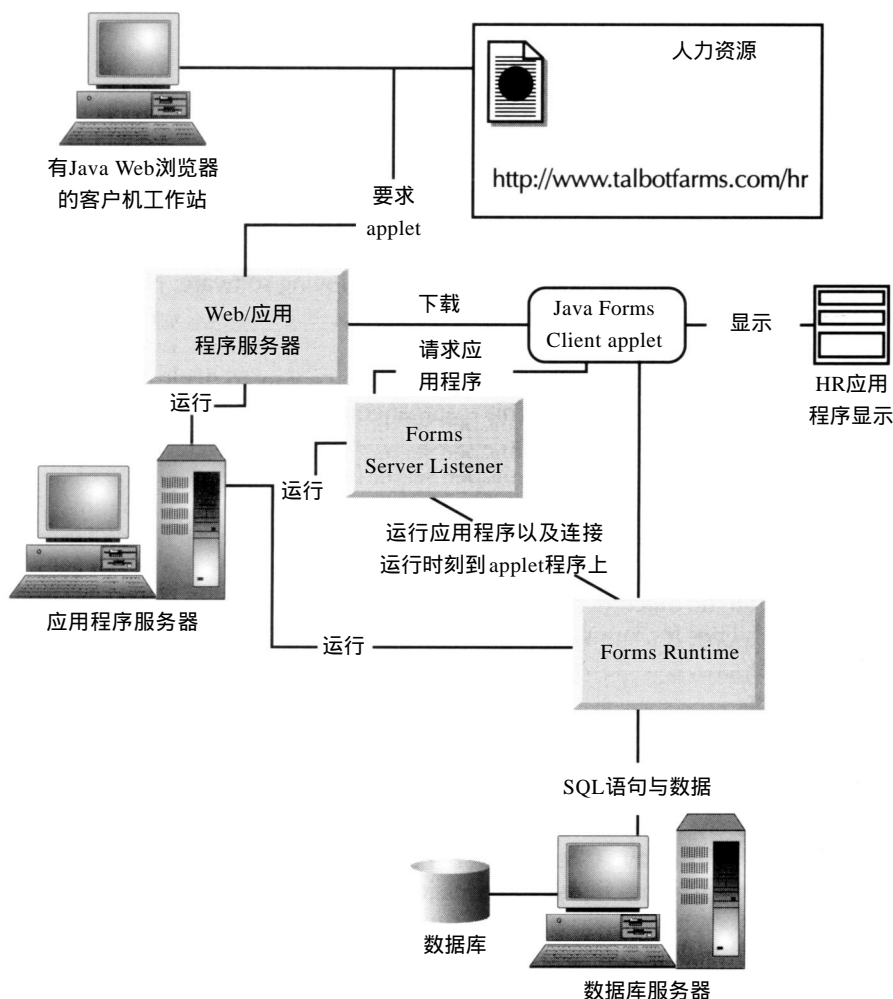


图14-1 Oracle Developer Server——Forms Server的结构

这时Listener又将连接信息回传给小应用程序,与Forms Runtime Server建立直接联系。小应用程序和Forms Runtime Server可以来回直接通讯,由小应用程序为表单提供显示功能,而

应用程序的其他事情由服务器负责。而 Listener则空闲下来接收其他应用程序发出的请求。Forms Runtime Server运行应用程序，通过SQL与Database Server进行通讯并获得数据，然后将数据传给小应用程序以供显示。

Reports则有一点不同。有两种方式进入 Reports Server：Reports Web Cartridge和Reports Web CGI。Reports Web Cartridge与Oracle Application Server相结合；CGI通过Web服务器与 Report Server连接。图 14-2说明了 Report Server系统的结构。

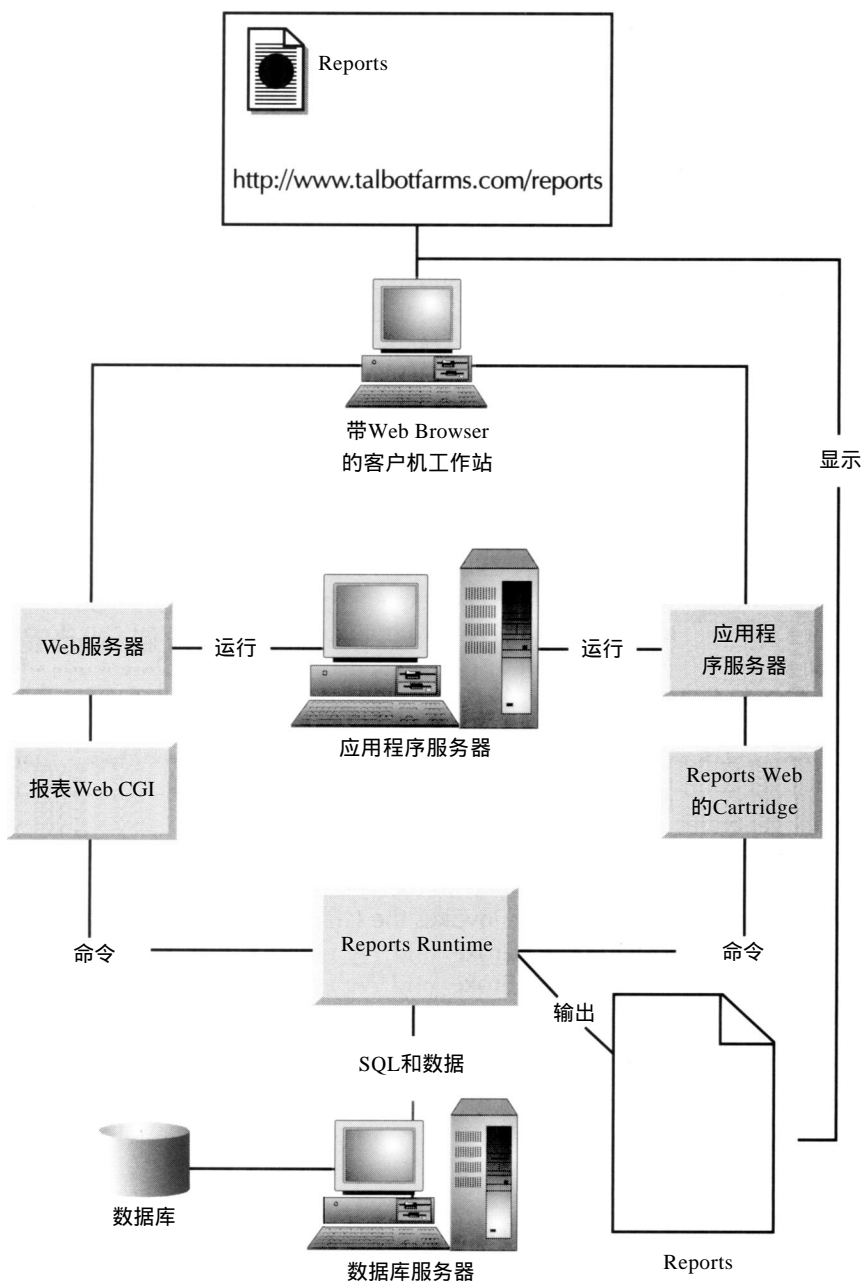


图14-2 Oracle Developer Server——Reports Server体系结构

客户机工作站运行 Web 浏览器，用户进入 HTML 网页可以获得报表 (例如 <http://www.talbotfarms.com/reports>)，轻点按钮或以超文本链接方式发出请求报表的信息。如果你用盒式磁带机方法进入，那么 Application Server 就会利用 Report Cartridge 建立一个 Report Server 命令行。Application Server 接受命令，同时 (等待输出) 传给 Report Server。Server 又通过运行时系统运行报告产生输出文件，然后 Web Cartridge 将它传回 URL 告诉浏览器输出文件在哪里。或者你已经选用了 CGI，那么 Web 服务器中的 CGI 脚本也可产生和提交命令行，然后告诉浏览器输出文件在哪里。浏览器可以 HTML、HTML CSS 或 Adobe Acrobat PDF 方式显示报表。

Graphics 的组织结构则更为简单一些，如图 14-3 所示。

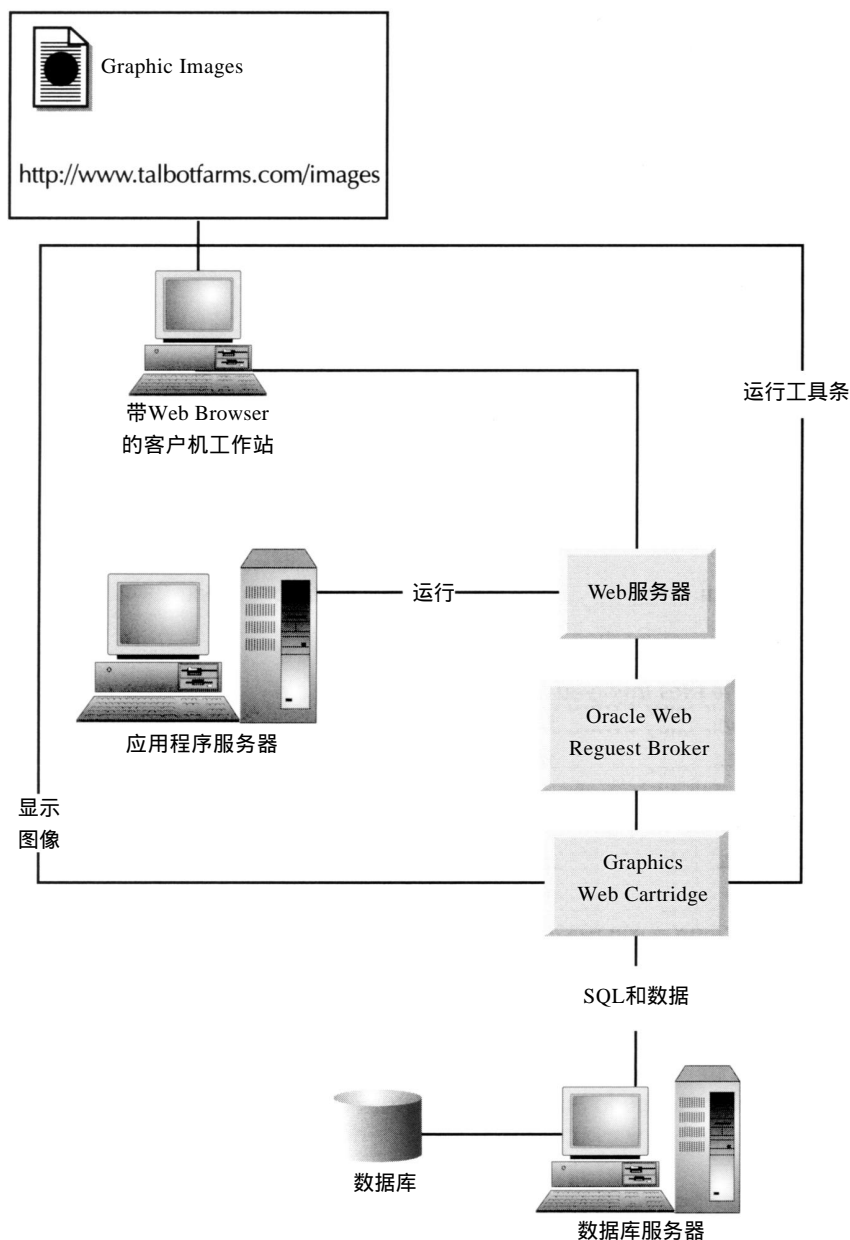


图14-3 Oracle Developer Server—Graphics Server体系结构

客户机运行网络浏览器，用户进入他可以获得图形显示的 HTML网页(例如http://www.talbotfarms.com/images)。这时你可以任选一种方式：或是通过特殊图像显示链接，或是通过 Oracle Graphics Builder Web界面的工具栏显示链接，工具栏为浏览器使用者提供了一个与数据库连接的方式，并显示一组可供选择的可用图像。这两种方式的区别在于利用本地 URL你必须自己设定URL的全部参数，而工具栏可为你自动产生全部参数，同时利用本地 URL在客户网页显示上你会受到更多的限制。不管你选用哪种方式，URL都要为显示图形向 Web服务器发出请求，由Oracle Web Request Broker 或Graphics Sever Cartridge产生显示。Cartridge通过SQL进入数据库服务器为显示获取数据，然后 Broker和Web服务器将图像传给网页。

14.2.1 安装Oracle Developer Server

安装Developer Server组件需要完成以下任务：

利用Oracle Installer提供的服务器部署选项，将 Oracle Developer Sever从它的CD-ROM上安装到中间 Application Server上。

注意 Oracle公司销售Oracle Developer Sever不同于Oracle Developer开发系统。

建立Forms Listener和Reports Server。

配置与Oracle Developer Server相连的Web服务器参数。

配置从服务器下载给客户机浏览器的 HTML文件和其他环境选项。

用JInitiator软件配置客户机。

注意 涉及到不同服务器的多个文件放在Oracle安装目录层中，如果还不知道其中约定，那么Oracle Home就被认为是当前层的根目录，就像Oracle_Home一样，在后面所有的内容里会出现根目录。不论在哪里看 Oracle_Home，都会替代本地根目录，就像Microsoft Windows NT系统的G:\Orant一样。

在开始安装之前，必须先建立 Web服务系统。从理论上说，在多终端环境中使用任何网络服务系统都可运行 Oracle Developer应用程序，但是采用 Oracle Application Server可获得更好的集成环境和效能。每个 Web服务系统在建立虚拟目录时都有自己的原理，请参考服务器的指令文件。

注意 Oracle Application Server是一个在多终端环境中部署应用程序的常用应用程序服务器系统。请参考 LLC的《Oracle Web Application Server by Dynamic Information Systems》(Oracle Press, 1998)一书。下面部分将重点放在 Oracle Developer Server的 Windows NT版和UNIX版的不同之处。

要在Web服务器上按照表 14-1建立虚拟目录，在这里你可任取一个所喜欢的名字。

表14-1 为Oracle Developer Server设置的Web Server Virtual目录

/codebase/	物理层目录包含Java类文件，例如 Oracle_Home\forms60\java。如果你用了Forms Server中的平衡装载特性，那么在服务器池中的每个服务器中都要建立相同的目录。它允许盒式磁带机将服务器名称参数化，用最小装载服务器在运行时装入
/jars/	该物理层目录包含Oracle JAR(Java Archive)文件，通常与/codebase一样。在平衡装载服务器池的所有服务器上都要定义这个目录
/html/	该物理层目录用于 Web服务器寻找静态的和盒式磁带机 HTML文件

(续)

/cache/	该物理层目录包含 Report Runtime的输出文件(仅用于 Report Server)
/temp_image/	该物理层目录, 如 Oracle_Home\Forms60\Temp_Image, 包含 Forms Server 向浏览器发送图像或位图时所产生的 JPEG 临时图像, 在 FORMS60_MAPPING 环境变量(参见“配置环境”部分)中输入虚拟目录名, 并在 FORMS60_OUTPUT 环境变量中输入物理层目录名
/web_reports/	该物理层目录包含 Forms Server 的输出报表, 运行一个表单所产生的报表

1. 安装软件

在 Internet 上部署应用程序的第一步就是在中间层 Application Server 上安装 Oracle Developer Server。可以为你的 Application Server 选择一台机器, 它只需要运行 Developer Server 即可。也可以将系统安装在常规的 Web 服务器上, 但是会发现, 加重常规 Web 服务器的负担会影响应用程序的效能, 除非它可以装载大量的应用程序或你只有很少的用户。

如果有很多用户或很多应用程序, 就会发现将进程分散到几个 Application Server 上是很重要的。Oracle Developer Server 包含一个平衡装载系统, 它可以使你将 Forms 应用程序进程装载到多台服务器上。

注意 有关建立平衡装载的信息请参考 Developer Server 安装文件。有关利用 Oracle Application Server 初始化安装的信息请参考附录 B。Graphics Server 只在相关情况下工作, 所以这一部分不讨论 Graphics Server 的安装。

1) 在你用来作为 Application Server 的机器上运行 Oracle Developer Server 安装文件, 将会看到图 14-4 所示对话框。

2) 选择 Deployment Installation 单选按钮, 会看到图 14-5 所示对话框。

3) 选择 Server Install 单选按钮, 你就完成了安装, 它将所有 Oracle Developer Server 的相关内容都安装到了 Application Server 上。

2. 配置 Forms Server

Forms Listener 和 Reports Server 是两个必须手动开始的不同进程。当服务器开始后, 你可将 Reports Server 配置成自动开始。

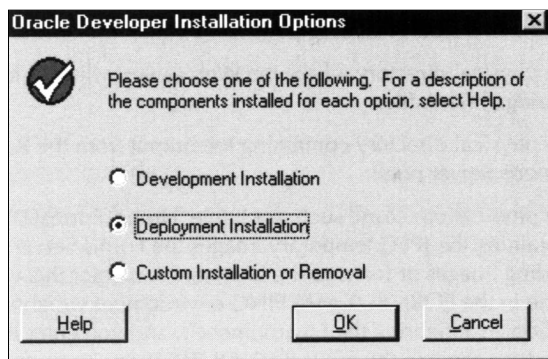


图14-4 安装文件对话框

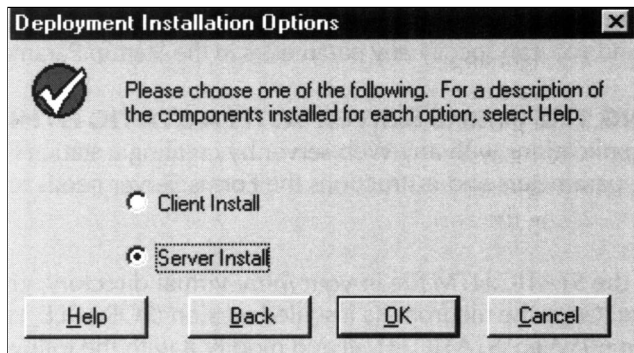


图14-5 Deployment Installation Options对话框

1) 从Windows NT任务栏启动Forms Listener：

Start|Programs|Oracle Developer 6.0|Forms Server Listener.(在UNIX环境下，发出命令 ifsrv60-listen port=5555 &.)

2) 查看Task Manager Processes的内容,验证是否成功地启动了接收器(在任务栏右键显示Task Manager)for IFSRV60.EXE。(在UNIX环境下，键入ps -ef | grep ifsrv60)。

3) 在\$ Oracle_Home\Net80\Admin目录下修改TNSNAMES.ORA，让它包含跟随TNS名称的规格，此时<hostname>为服务器名称或IP地址，如191.191.191.5：

```
repsvr.world=(ADDRESS=(PROTOCOL=tcp)(HOST=<hostname>)(PORT=1949))
```

4) 启动Reports Server作为NT服务(在UNIX背景使用命令“rwmts60 name=repsvr.world &”)处理，使用你添加到TNSNAMES.ORA中的TNS名字：

```
rwmts60 -install repsvr.world tcpip
```

5) 用Start|Settings|Control Panel|Services打开服务窗口，选择 Report Server服务，单击Start按钮。如果需要使用Startup按钮使得服务器无论在哪里开始，都可自动启动服务，可以在Startup Parameters区设定任意参数。

3. 利用静态HTML页配置Forms Server

通过建立静态HTML文件可在任意Web服务器下运行表单应用程序，该文件包含Forms Server需要的所有参数和说明。

在/html虚拟目录下建立STATIC.HTM文件，可任取名称。从安装版拷贝文件(\$ORACLE_HOME\Tools\Devdem60\Web\STATIC.HTM)，并用下面的内容替换修改：

codebase：设置Web服务器时为Java编码定义的虚拟目录。

archive：为建立应用程序下载的JAR文件的路径和文件，包括Oracle提供的ifsrv60.jar。

serverPort：Forms Server Listener端口(默认值为9000)。

serverArgs：表单参数和用户参数。

serverApp：指定字体和位图路径设置的应用程序类的名称（参看盒式磁带机文档），“default”意味着使用默认的字体和图标路径。

下面是一个经修改过的运行LEDGER应用程序模板的例子：

```
<HTML>
<HEAD><TITLE>Talbot Farms Ledger</TITLE></HEAD>

<BODY><CENTER>
  <BR>Welcome to Talbot Farms.
  <BR>Loading application...
</P>

<APPLET CODEBASE="/codebase/"
  CODE="oracle.forms.engine.Main"
  ARCHIVE="/jars/f60all.jar"
  HEIGHT=20 WIDTH=20>
<PARAM NAME="serverPort" VALUE="9000">
<PARAM NAME="serverArgs" VALUE="Module=Ledger userid=Talbot/george@orc1">
<PARAM NAME="serverApp" VALUE="default"
</APPLET>
</BODY>
</HTML>
```

这一页将模块名称设置为Ledger参数，所以运行附属程序applet就运行LEDGER.FMX应用程序。需要为每个想运行的应用程序建立一个静态页。

你可能想尝试一种新的 Oracle 界面和感觉, 为 Oracle Developer Server 运行应用程序提供不同的界面, 那么为 HTML 添加以下代码:

```
<PARAM NAME="lookAndFeel" VALUE="oracle">
<PARAM NAME="colorScheme" VALUE="teal">
```

可以使用 Oracle 指定的 Windows 组件中不同的颜色对 colorScheme 值来改变颜色。

(1) 用 JInitiator 配置客户机

Oracle JInitiator 是一个 Oracle 公司的 Java 插件, 用于在 HTML 页中运行 Java 小应用程序或 JavaBeans。它为优化使用 Oracle Developer 应用程序建立一个特定的 Java 虚拟机器。可以让用户在运行时从他们的客户机上下载, 也可以预先在客户机上安装。

在 Oracle Developer 安装处 (Oracle_Home) 找到 jinit.exe 文件, 将它拷贝到客户机上, 并运行该程序。这是一个可执行的用于安装 Java 插件的安装保护程序。

(2) 在因特网上使用图标和字体的应用程序类

在因特网上部署表单, 在网络上使用的 GIF 图像和字体是一个难题。如果想定义特殊的字体和图标, 可使用标准的 Form Builder 来设定。但是如果在一个题目中有若干次变化, 就可能需要使用应用程序类。这种情况多发生在不同 GUI 平台上 (如 Windows、Motif 和 Macintosh) 需要特殊字体的情况中。

一个应用程序类就是特殊字体的一组字体映像的集合和指定图标的图标路径。可以在几个依赖于 HTML serverAPP 参数的配置文件或静态 HTML 基本文件中之间转换 (有关参数的详细内容见 “利用静态 HTML 页配置 Forms Server” 一节)。运行时, Forms Server 首先使用已命名的应用程序类的设置, 如果在 serverApp 参数中没有命名应用程序类, 那么 Forms Server 就使用默认的字体和图标。默认字体建立到表单组件的环境中, 默认图标从与基本 HTML 页相同的目录中获取。

建立应用程序类需要编辑 REGISTRY.DAT 文件, 可以从以下目录中找到:

```
Oracle_Home\forms60\java\oracle\forms\uiClient\v1_4\util
```

1) 对想添加的每个字体的映像图, 在 REGISTRY.DAT 文件中加一行, 包括应用程序类名称、字体参数名称、字体参数值, 如下面的 Windows_Form 应用程序类:

```
Windows_Form.appFontNames=Courier,Terminal,Arial
Windows_Form.javaFontNames=MonoSpaced,Dialog,Helvetica
```

2) 当没有字体与应用程序字体相配时, 为了使表单显示指定的默认 Java 字体, 应在 Std_Form 应用程序类增加以下几行:

```
Windows_Form.DefaultFontname=Helvetica
Windows_Form.DefaultSize=12
Windows_Form.DefaultStyle=BOLD
```

3) 要指定一个图标目录, 应在 Windows_Form 应用程序类中增加以下一行:

```
Windows_Form.iconpath=/temp_image
```

在这种情况下, 该值是一个虚拟目录 /temp_image/, 你已经将它定义在了 Web 服务器上, 参看表 14-1。也可以指定一个 URL 或一个与 HTML 页相关的目录, 而虚拟目录是最佳的选择, 因为在移动它而没有改变 REGISTRY.DAT 文件时, 可以在 Web 服务器上改变它。

(3) 配置 Reports Server CGI

- 配置 Reports Server CGI 是简单的:

在Oracle_Home\bin下找到rwcgi60.exe文件，将它拷贝到 Web服务器 CGI脚本目录中，如Oracle_Home\ows\40\bin(如是UNIX环境，拷贝rwcgim60文件)。

4. 建立Reports Server和Forms Server作为NT服务

现在已经安装和配置了服务器，但是还需要将它们设置 NT服务(或UNIX后台进程)。这里最好能够用管理员身份注册到 NT系统上，参阅NT系统管理员帮助。

1) 首先为Reports Server建立一个TNS项。编辑TNSNAMES.ORA文件，如果是Oracle 7 SQL*Net，则该文件在Oracle_Home\Network\Admin目录下；如是Oracle 8 Net8，则该文件在Oracle_Home\Net8\Admin目录下，将TNS项添加在文件底部：

```
repserver.world=(ADDRESS=(PROTOCOL=tcp)(HOST=<server>)(PORT=1949))
```

这里，<server>是运行Reports Server的服务器名称或 TCP/IP地址，如 WAVERLY 或 127.100.100.100。只要你添加了适当的区域，就可以使用任何 TNS名称，如reports.world或news_of_the.world。

2) 为Forms Server创建NT服务。应当以管理员身份注册进入 Windows NT。在任务栏中以Start|Programs|MSDOS Prompt打开DOS shell，执行下面命令：

```
ifsrv60 -install forms60
```

3) 在相同的DOS shell下为Reports Server创建NT服务，命令如下：

```
rwmts60 -install repserver.world tcpip
```

服务器名称repserver.world，是你在TNSNAMES.ORA文件中指定的TNS名称。

4) 运行Service(服务)小应用程序：选择 Start|Settings|Control Panel，然后双击Services图标，找到Forms Server 60入口，并点击Startup。在得到的对话框中(见图14-6)，找到Log On As部分，选择 This Account，填写当前用户名和口令，即在当前帐号下运行服务。也可选择 System account为系统帐号(这是个好选择)或一些其他的帐号，可参考管理程序。最后单击 Automatic，让NT在引导时自动启动 Forms Server。

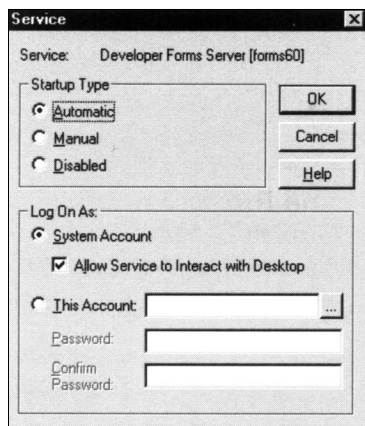


图14-6 Service对话框

提示 在系统帐号环境下和其他标准服务一样运行这种服务，通常是个好方法。可以简化安全性和配置问题。

然而，也可能会因为安全或其他原因不能为特定的NT服务器使用这种方法。

5) 在Service对话框中重复最后一步，建立 Report Server60登录项。

注意 你也许需要阅读有关创建NT服务的帮助文件，以便解决在建立Reports Server时遇到的更广泛的问题。因为有更多的需要考虑的问题超出了基本安装范围。

5. 配置环境

一旦建立了服务器，就必须为它们配置各种环境选项。在 Window NT环境下就意味着编辑系统注册表。

1) 在任务栏中选择Start|Run，输入下面命令并单击OK：

```
regedit
```

2) 找到注册表树下的 HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE目录，单击 ORACLE，显示所有 Oracle关键字(见图14-7)。然后建立表14-2中列出的关键字。

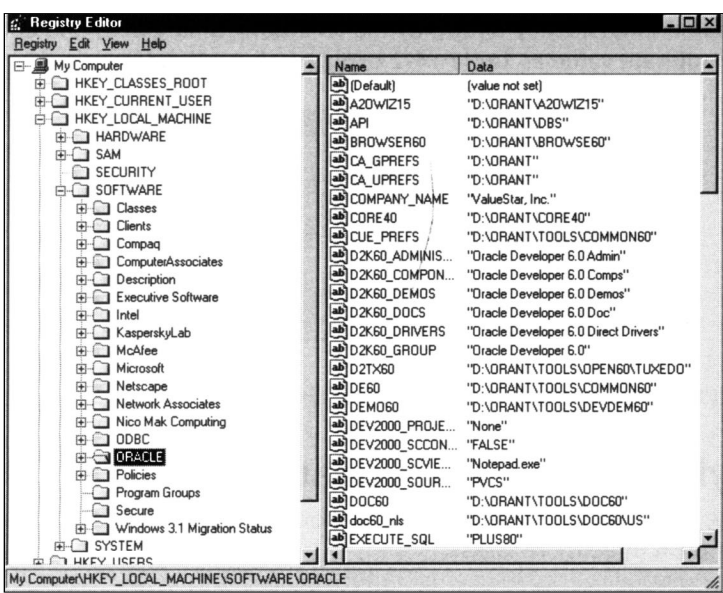


图14-7 显示Oracle关键字

表14-2 Oracle Developer Server的注册关键字

名称	类型	值
FORMS60_MAPPING	字符串值	如果在应用程序中使用了图像和图标，那么必须在 Web服务器上建立临时目录和虚拟目录，当把它们传送给客户浏览器时， Forms Server这些目录来临时存储 JPEG图像。键入目录名称，如 “/image,” (结尾处无斜线)
FORMS60_OUTPUT	字符串值	一个对应FORMS60_MAPPING虚拟目录的物理目录，如G:\Orant\Forms60\temp_image
FORMS60_PATH	字符串值	在服务器上以分号隔开的物理目录列表，Forms Server为可执行的FMX表单访问它们。Oracle 安装程序应当在初始化时自动建立这个列表，所以在这里只需要进入指定目录
FORMS60_REPFORMAT	字符串值	使用字符串 “HTML” 告诉 Forms Server产生类似于 HTML页的报表，用 “ PDF” 则产生 Adobe Acrobat报表
FORMS60_TIMEOUT	双数值	键入 Forms Server等待输入所允许的最长分钟数，应为 3–1440之间的整数，可从 15分钟开始
REPORTS60_WebLOC	字符串值	Web服务器查找报表输出文件的虚拟目录，如/web_reports/
REPORTS60_WebLOC_TRANSLATED	字符串值	如果Reports Server和Web服务器不能共享一个文件系统时， Reports Server用来转移 Reports Runtime输出的物理目录，如 G:\ORANT\web_reports
REPORTS60_PATH	字符串值	Reports Server查找报表定义文件的物理目录列表

对表中的每个关键字，都要执行以下步骤：

3) 右击关键字，显示面板，选择 New|String Value或 New|DWORD Value，如表 14-1所示。Editor创建有关重新命名的新关键字和位置(见图 14-8)。

4) 按照表 14-2所示的关键字重新命名。

5) 双击关键字，显示 Edit String对话框，按照表 14-2的说明填写适当的值，然后点击 OK(见图 14-9)。



图14-8 新关键字及其位置

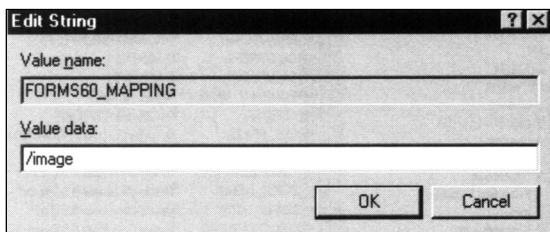


图14-9 Edit String对话框

6) 用File|Exit退出注册编辑器 Registry Editor。

14.2.2 部署因特网应用程序

要完成使用 Web浏览器部署应用程序，还必须完成下面三个任务：

在服务器上安装表单、报表和显示的应用程序代码。

在服务器上安装HTML访问页。

为用户提供访问页的URLs。

编译应用程序的输出由可执行文件组成 (FMX、RDF和OGD文件)。必须用与 Developer Server所使用的相同的操作系统的生成器或编译器来编辑这些可执行文件。例如，如果准备在 Windows NT环境下运行 Developer Server，那么必须用 Windows NT的生成器或编译器来编辑应用文件，而不能用 Windows 98或Sun Solaris。

要安装表单的代码 (FMX可执行文件)，将它们放入所希望的任意目录下，在动态或静态的HTML页上用完整的路径和文件名使用它们。要安装报表代码 (RDF可执行文件)，也可以将它们放入服务器的任意目录下，但是它在服务器注册表中的变量应为 REPORTS60_PATH。

注意 可执行的Graphics必须放在可知位置，允许 Graphics Server以图形显示方式的下拉列表提供任务栏。

将动态或静态的 HTML基本页放入标准虚拟目录中，如 /html，然后为用户提供适当的 URL。例如 Talbot 农场有一个 WorkerSkill表单静态基本页，叫 WorkerSkill.html，Talbot用户在 URL上写下面命令即可运行表单：

```
http://www/talbotfarms.com/html/WorkerSkill1.html
```

14.3 客户机/服务器计算领域

在多层服务器上部署应用程序具有巨大的优势。然而，如果因为某些原因，环境要求单一服务器，那么还可以像客户机/服务器应用程序一样部署应用程序，在每个想要运行应用程

序的客户机上安装 Oracle Developer Deployment 软件。本部分将介绍建立客户机 / 服务器软件包的步骤。

用户必须从数据库或系统管理员处获得一些专门的信息，比如主机和网络文件服务器的连接字串和数据库服务器名称。必须为用户提供文件和使用材料，必须告诉用户什么信息是他们必需的，还需要仔细告诉用户在所指定的环境下所要求的附加软件（包括 Oracle Developer 运行时和 Oraclr SQL*Net 软件），以及所需要的特定版本。

14.3.1 将应用程序打包

在建立应用程序安装系统时，有两方面的情况决定你的努力成果：安装过程中个人参与的程度和安装过程的复杂性。

通常，个人参与程度和应用程序本身的性质直接相关。你自己安装的应用程序通常是个人性的，或是某个单位的工具程序，或是某个工程的应用程序，而公司社团的应用程序倾向于要求一些个人参与在其他的应用上使用的功能。

安装过程的复杂性包含几个方面。第一，可能需要将应用程序分成几个部分：客户工作站和服务器、一些协作工作站、一些分散的服务器、一个 Web 服务器等等。第二，可能需要有条件地安装程序包的各个部分。例如用户可以选择只安装程序包的一部分，或者在安装过程中检测到的指定配置的那一部分，或者检测到的需要更新的那些部分，或者需要重新安装的部分。第三，可能需要对程序包进行一些处理：解压缩；检查哪些程序是已存在的，哪些是新的；编译；生成选择文件；或其他一些对用户来说增加了复杂性的安装方法。

如果产品在安装过程中没有个人参与并有相当程度的复杂性，那么应该使用专业的安装软件程序包。这些程序包提供了所需要的全部工具，对最终用户和其他管理者即购买产品的人来说，再复杂的安装也会显得简单和专业化。

否则，你必须留心每一个细节，必须了解每个地方所需要做的和已经做的所有事情。程序打包的总原则就是世界上最古老的原则：力求简单，仅安装用户需要的文件。不要用一些东西将磁盘塞得乱七八糟，如压缩说明文件，产品不同部分的 23 个 README 文件，用户可自己定制安装的 600 个特殊图标位图。将所有这些放在以备特殊需要时使用的磁盘上。

此外，必须决定是否将源文件和目标文件一起提供，这个决定通常取决于你和用户之间的合作性质。如果需要让用户防止他们的错误，不要提供源程序。如果想要保护你对该系统的专有权，不要提供源程序。如果用户不需要或不想要源程序，不要提供源程序。

这里是 Oracle Developer 应用程序打包组建清单，表 14-3 告诉你哪些组件是必须打包的，哪些组件是可以选择的。应该仔细地将这些文件的用途和特性写在 README 文档中，并将它装入软件包中去。此外还可以将各种各样的文件与系统文件一起打包。例如图标、位图、声音文件及视频文件等等。

表14-3 Oracle Developer文件组件

产 品	文 件	源程序或 可执行程序	可选择的 或需要的	说 明
应用程序系统	PRF	源文件	可选择的	如果这些与 Oracle 用 ORAINST 建立的默认值不同，那么用应用程序建立一个模块优先文件或建立一个系统运行所需要的包含运行时间选项的文件

(续)

产 品	文 件	源程序或 可执行程序	可选择的 或需要的	说 明
	EXE	可执行文件	需要的	被允许的Oracle运行时间可执行文件，一些增加的可执行工具。用户可使用ORAINST安装，它们在Oracle公司提供的安装光盘上
	DLL	可执行文件	需要的	通过ORAINST安装的被允许的Oracle运行时间动态连接库，和其他一些应用程序所需要的库（如用户退出、外语功能等）
表单	FMB	源文件	可选择	表单源文件。只在想要用户能够修改应用程序时才装载
	FMX	可执行文件	需要的	没有源程序的可执行代码
菜单	MMB	源文件	可选择的	菜单文件源代码。只在想要用户可以改变应用程序菜单时装载
	MMX	可执行文件	需要的	没有源程序的可执行菜单代码
PL/SQL	PLL	源文件	可选择的	PL/SQL库源程序。只在允许用户改变程序单元时装载
	PLX	可执行文件	需要的	库中的可执行程序单元
报表	RDF	源文件	可选择的	报表源文件。只在允许用户改变报表时装载
	REP	可执行文件	需要的	可执行报表
图形	OGD	源文件	可选择的	图形显示源文件。只在允许用户改变显示时装载
	OGR	可执行文件	需要的	可执行显示文件

注意 可执行文件必须和用户的操作系统兼容，这意味着必须在该操作系统下编译源文件。例如不能在UNIX系统中编译一个表单，又将生成的可执行文件交给Windows 98用户去执行。

还需要考虑需要提供哪些文件用于升级更新。你可以提供一个完整的系统，包括所有的文件，也可以只装载改动了的新文件。如果选择后一种方法，那么将需要扩大系统测试涵盖所有可能的不同配置，还必须人工组成 Oracle Developer Server的Project Builder的内容，仅提供你所指定的文件。必须保证文件的子集对系统的所有不同配置都能够服务。

注意 作为一个实用而合法的软件，必须用Oracle Installer来安装客户机/服务器的配置选项。有关安装配置软件的详细内容请参考附录B。

可以点击图标并输入适当的信息，使用 Form Compiler(表单编译器)产生FMX文件。然而还有更简单的方法，通过 Project Builder，只需单击一下鼠标就可容易地编译所有文件。你需要按照第3章中“Oracle Developer工程管理器”部分讨论的那样建立工程文件。首先在 Project Navigator中找到想要编译的目标代码节点，选择该节点，然后从这个结点的弹出菜单（点鼠标右键）中选择Build All菜单项，或在主菜单中选择Tools | Build All菜单项(见图14-10)。

可以只产生用于文件和压缩的 FMB和MMB文件的文本。Oracle Developer支持扩展编辑以缩短生成所需要的时间。如果你对一些 PL/SQL做了小小的改动，然后重新生成表单，那么Oracle Developer不会重新编译表单中的所有PL/SQL，而只重新编译改动的触发器或程序单元。为此，Oracle Developer将附加信息存在FMB文件中，如所有的PL/SQL源文本，中间描述代

码和已编译的代码。只有在生成表单或编译触发器或程序单元时，才生成中间描述代码和已编译的代码。变换成 FMT文件然后又变换回 FMB文件回缩小 FMB 文件。因为 FMT文件不存储中间描述代码和已编译的代码，所以，生成的 FMX文件仅包含已编译的代码，这就是要使用的文件。

在Form Builder中，可以使用File|Administration|Covert菜单项生成FMT、MMT表单和菜单的源文件文本。这个菜单项显示在Covert对话框中(见图14-11)。

要生成FMT表单源文件，将Type设为Form，浏览表单文件(FMB)，并将Direction域设置为Binary-to-Text。要生成MMT菜单源文件，将Type设为Menu，浏览菜单文件(MMB)，并将Direction域的内容设为Binary-to-Text。当单击Convert时，Builder就会生成表单源文件和菜单源文件。

注意 可以用Form Compiler生成表单和菜单的文本文件。选择“Write script file?”复选框，将告诉你生成的是文本文件而不是目标代码。

在Report Builder中有两种方式产生REP文件(报表定义为单一目标格式)。可用File|Administration|Generate生成这个文件，也可以通过File|Administration|Convert和Convert对话框将数据库或报表定义的RDF版本转化为REP版本。将Source Type设为RDF，在源文件域浏览报表RDF文件，将Destination Type设为REP，并为Destination域填写文件名。可以用Convert对话框创建生成REX文件(报表定义为单一源文本格式)，通过选择REX的Destination Type取代REP(见图14-12)。



图14-10 使用Project Builder编译文件

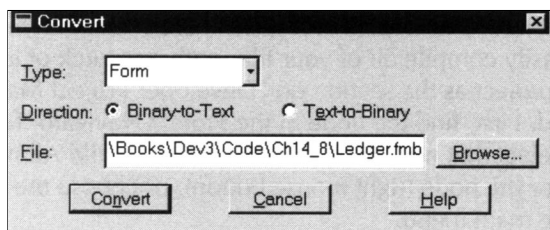


图14-11 Covert对话框

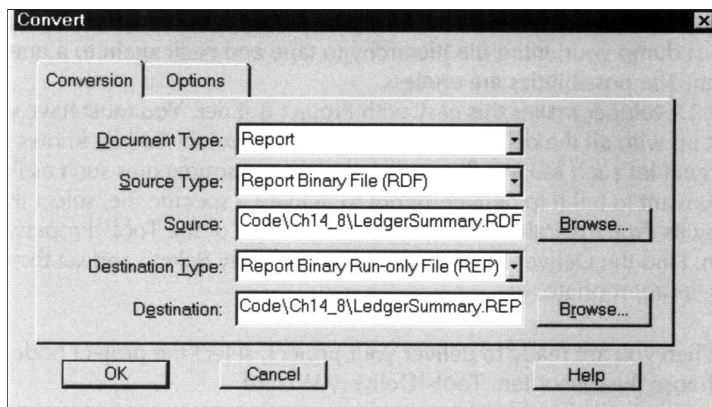


图14-12 选择Destination Type选项

可用File|Administration|Generate菜单项生成OGR文件(单一目标代码图形显示)。

14.3.2 客户机/服务器的部署

可以用很多不同的方法为客户部署你的程序包。但在不久以前,还只能根据目标操作系统或其他系统选择软盘或磁带。

目前,CD-ROM得到了偏爱,特别是对 Oracle或Oracle Developer这样大型的商业产品。CD-ROM作为主要工具,随着其价格的很快下降和技术上的进步,你应该认真考虑这种选择。

不久以前,因特网和其他在线服务和商用公告栏已经成为获取软件包的资源。可以通过连接到适当的服务器和FTP或将程序包下载到工作站上,然后安装。这是发送低价软件、免费试用软件和升级的普遍方法。如果你的公司全部连接到了因特网或本地网或广域网络,那么可以用这种机制有效地发送家用应用软件。

更新的Web 技术使你下载程序更加容易并可运行,如Java,在前面部分有详细介绍。

大多数商业安装产品是通过这些不同的媒介发送的。例如用软盘,在盘上系统提供压缩和打包文件,并把建立和压缩文件的过程作为在工作站上安装的一部分。可以用标准的磁盘压缩工具简单而廉价地打包文件,如PKWARE公司提供的各种各样的PKZIP压缩工具。若使用类似于磁带机的大容量媒介,则只需将大量文件分层次地装入磁带机并重新返回到一个新的工作站上。这种能力是无止境的。

Oracle Developer利用Project Builder将工作简单化。必须用所有能够识别的可发送文件建立工程。Project Builder知道传送类似于FMX文件的目标文件,而不发送像FMB文件这样的源文件。如果想要发送或不发送一个指定文件,可以利用弹出菜单或用Tools|Property Palette菜单选项,选定该文件并显示它的Property Palette。Property Palette中找到Deliver文件属性,安装需要设置值为Yes或No。

1) 当准备发表工程时,选择这个工程的结点并选择菜单项Tools|Delivery Wizard(见图14-13)。

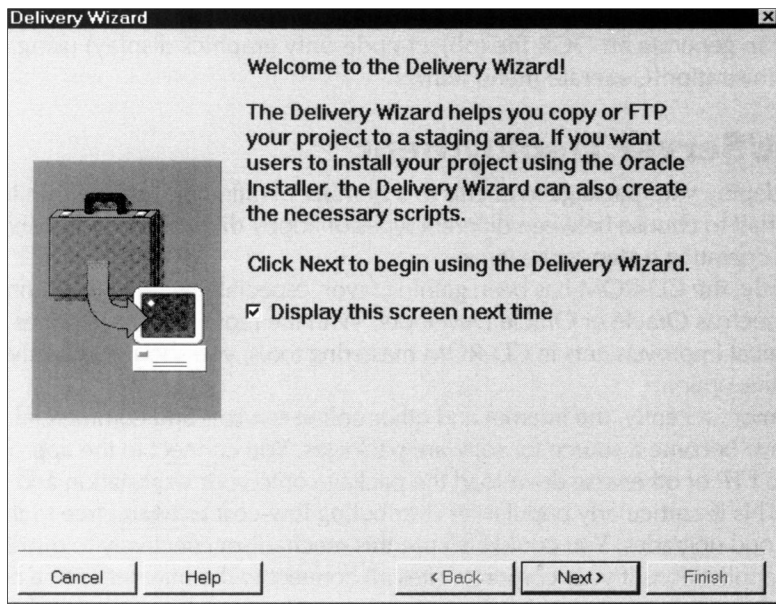


图14-13 Delivery Wizard对话框

这个Wizard将工程中可以发表的部分拷贝或FTP到公布区域。还可以让向导创建脚本，这样可以通过 Oracle Installer将工程文件和运行时间安装结合起来，这使你非常容易地部署 Oracle Developer运行时和应用程序代码。

使用Delivery Wizard有两种方式：发送一个完整系统或发送一个更新系统。下一屏让你选择是发送目标文件中所有可发送文件还是只发送至上次发送后改变的文件。

2) 单击Next。如果要发送一个完整系统，那么选择“Deliver all files”，否则选择“Deliver files that have changed since the last delivery。”(见图14-14)。

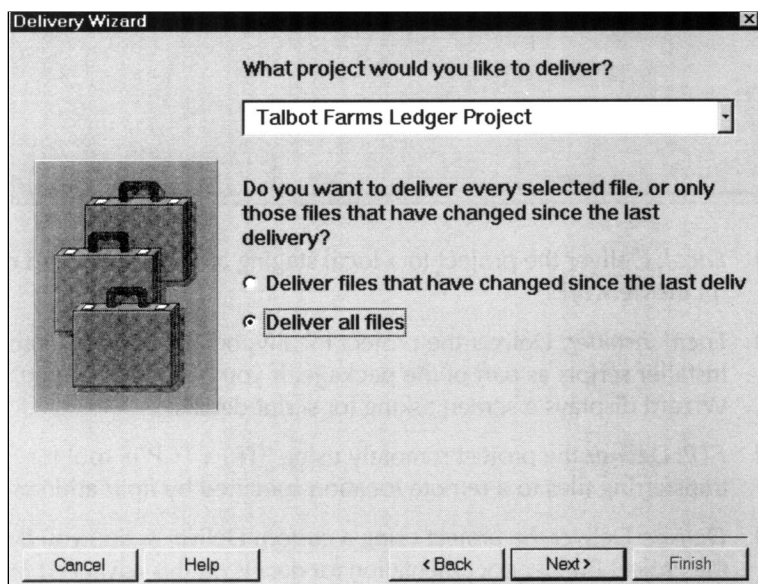


图14-14 选择Deliver all files选项

3) 单击Next。选择想要建立的发送类型，然后输入分级区域的路径或FTP地址(见图14-15)。

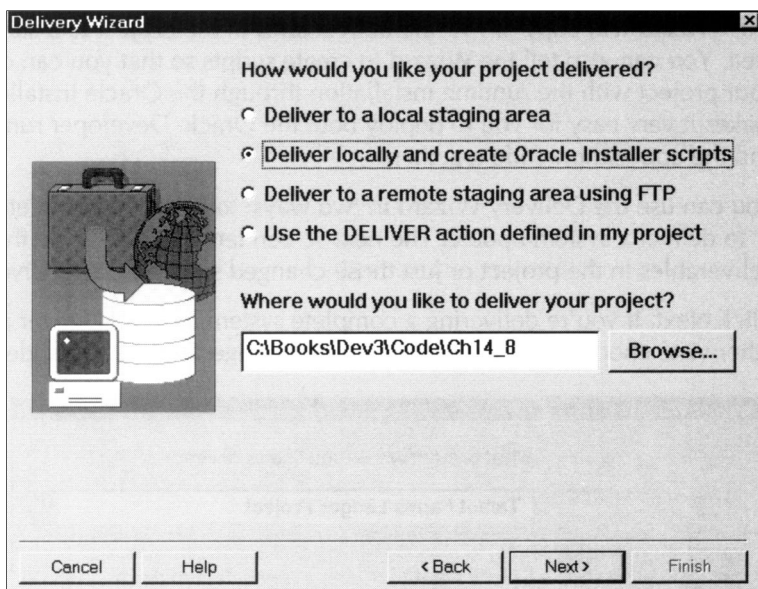


图14-15 输入分级区域的路径或FTP地址

本地的(Local)：将工程文件发送到你的本地硬盘或网络的合适区域。

本地安装程序(Local installer)：本地发送工程文件并创建 Oracle Installer脚本作为程序包的一部分。如果你选择了这一选项，那么 Wizard就会显示下一步要求更多的细节。

FTP：应用FTP远程发送工程程序，一个TCP/IP传送工具远程发送文件到由ftp://address指定的位置。

Deliver：使用定义的Deliver方式发送工程程序(参见Project Builder文件)。

4) 单击Next。在文件屏幕所列出的文件中选出将要发送的工程文件。通常你将会发送所有文件，那么单击Next即可(见图14-16)。

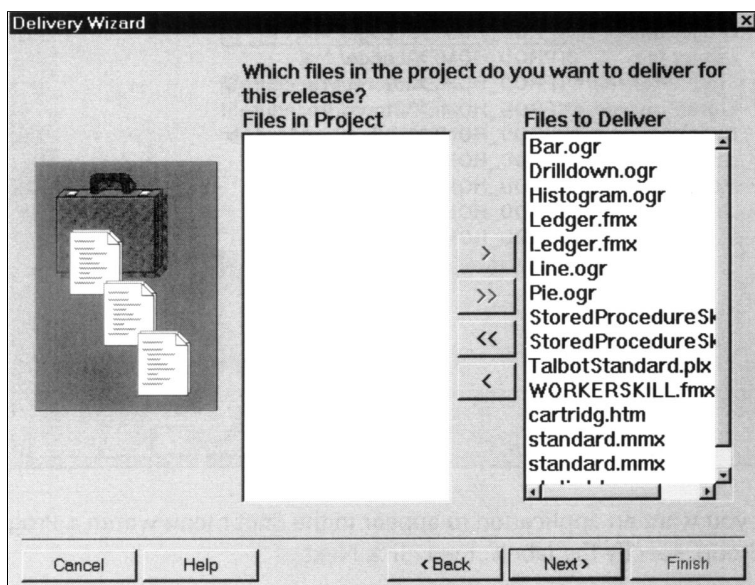


图14-16 选择Next，发送所有文件

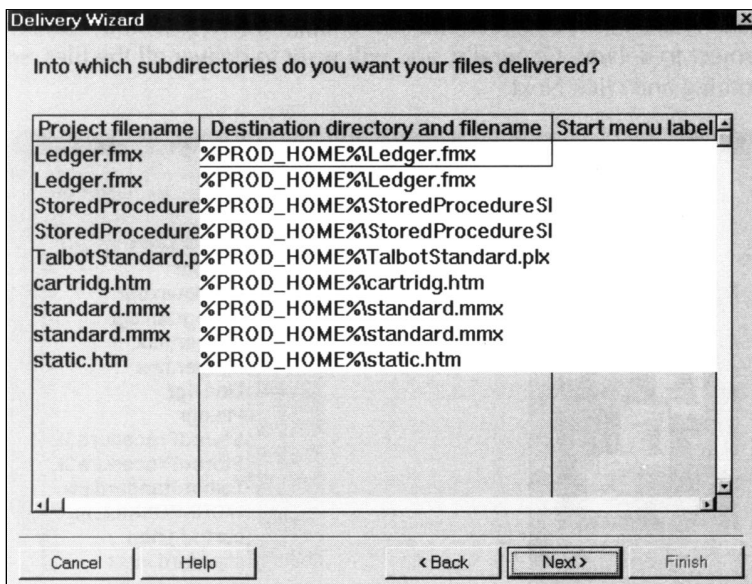


图14-17 第一个屏幕窗口

现在 Wizard 进入产生 Oracle Installer 脚本过程。第一个屏幕窗口显示想要安装文件的位置。通常应该将所有的工程文件安装在一个单独的目录下。不这样做,就意味着大量的文件将会随处放置(见图 14-17)。

5) 如果想要应用程序出现在 Program Group 的 Start 菜单中,那么指定标签,然后单击 Next。

6) 单击 Finish, 发送工程文件(见图 14-18)。

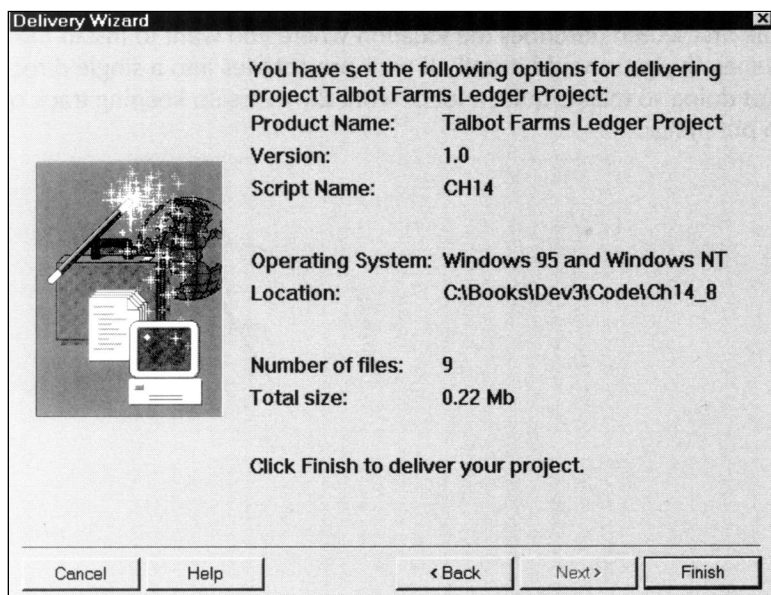


图14-18 选择Finish发送工程文件

现在已经完成了发送工程文件的工作。下一步是将文件压缩为一个 ZIP 文件或其他形式发送到软盘、网络或 CD-ROM 上。下面讲述发送部署以后如何进一步管理。

14.4 配置管理

当进入了应用程序的生命周期阶段时,使应用程序在配置管理系统的控制下是非常重要的。如果脱离了配置管理系统的管理或控制,那么就会受到损害。在部署过程的每一步都应当重视这种形式的控制,包括控制所提出的要求和设计方案。

Oracle Developer 结合了多种版本控制和配置管理工具,如 Intersolv PVCS、Rational Clear Case 或 Starbase 中的 Starteam。你可从 Oracle Developer Server 的每个单一组件的生成器中使用这些工具,但是你必须用 Project Builder 界面来管理文件配置。可以使用 File|Administration 菜单的 Check-In、Check-Out, Source Control Options 选项或这些选项的弹出菜单中的 Check In 和 Check Out 选项。

界面产生 ASCII 文件,存储在版本控制档案中。可以通过 File|Administration|Source Control Options 菜单选项和 Source Control Option 对话框创建档案,在对话框中指定档案目录的位置和名字。

File|Administration|Check-In 菜单选项显示 Check-In 对话框,它让你选择一个模块放入档案。File|Administration|Check-Out 菜单选项显示 Check-Out 对话框,它让你选择一个模块从档

案中取出。如果想修改模块，必须选择 Lock This File 复选框，否则得到的是只读模块文件。

Oracle Developer 的配置管理生命周期有以下几个过程：

- 1) 创建档案目录。
- 2) 选择模块的初始版本。
- 3) 选择想要修改的模块。
- 4) 选择已经修改过要返回归档的模块，这些模块使用新的版本号。

使用配置管理软件窗口，可以管理配置管理系统中更高级的特性，如跨越几个模块发表版本基准。也可以用 File|Administration|Source Control Options 菜单选项和 Source Control Options 对话框产生不同的报表(报表显示两个文件的不同)。

也可通过输出文本文件来利用其他配置管理系统，如 Source Integrity on Windows 或 UNIX 中的 rcs 和 sccs，虽然不能直接通过 Oracle Developer 记录和检查文件。如果系统提供了类似于 PVCS 的命令行，就可以在 Project Builder 中增加一些基本特性。这个小小的缺点不会有太大的影响。

在大多数配置管理系统中版本号相当复杂。对大多数应用软件来说，一个简单的两部分组成的版本号是有效的：<Release>.<Version>。<Release>号随产品主要更新而变化，<Version>号随每个升级而变化。例如 1.0 是产品最初的版本，1.5 是第 5 个较小的升级，而 2.3 是 Release 2 主要版本后的第 3 个较小版本。“可能是(Might be)”描述了所有的情况，因为你不可能依靠版本号这种简单的逻辑。这里涉及到管理。你将发现跳过较小版本的很多理由，在两个主要版本号下有并行版本，还有许多其他混乱但是合理的情况。3000 年后这将为上百个考古学家和历史学家提供工作机会，因为他们要找出计数系统的原始软件。

现在你已经学习了使用 Oracle Developer 的部署和使用的所有基本组件，最后一章将学习应用程序与 Oracle Developer 产品之外的工具相结合的先进工具。