

## 第17章 配置客户机/服务器和Web网络环境

由于Oracle支持许多不同的配置和平台，所以还没有一套对每个客户机 /服务器或瘦客户机环境有效的规范说明。不过，有许多可用于网络环境配置的常规过程。有关基本网络配置和UNIX配置的详细情况，请参见第 13章和第 14章。本章描述客户机 /服务器和瘦客户机环境的网络配置准则。

### 17.1 客户机/服务器处理概述

使用客户机/服务器配置可以把CPU和对应用程序的处理负载分布在两个机器上。第一个机器叫做客户机，它支持发出数据库请求的应用程序。数据库所驻留的后端叫做服务器。当数据库服务器致力于查询而不是应用程序时，客户机可以承受 CPU负载的大部分。图 17-1示出了这种资源需求的分布。

与图 17-1所示的配置稍有不同的是把应用程序的逻辑移到存储在数据库中的 PL/SQL软件包。在这种配置中，数据库服务器承受应用处理和数据访问的 CPU负载，而客户机致力于数据的图形显示。这种环境的网络配置与图 17-1相同，但应用程序执行对 PL/SQL软件包的调用而不是执行insert、update和delete命令。

当客户机向服务器发出一个数据库请求(通过SQL\*Net或Net8)时，服务器接收并执行给它的SQL语句。SQL语句的结果要加上返回的所有错误信息，一起返回到客户机。

若要使用客户机/服务器结构，客户机和服务器必须能相互进行通信。这意味着两个机器间有一个硬件连接。每个机器都必须支持使它们能交换数据的通信协议。可以使用 SQL\*Net V2多协议交换或 Net8连接管理器来解决各种协议之间的兼容性问题。

本章将描述两种配置：客户机 /服务器配置和基于 Web(瘦客户机)的配置。首先，设想一个客户机运行Microsoft Windows和SQL\*Net V2，它通过网络接口卡 (NIC, network interface card)连接到以太网网络上。希望与客户机进行通信的服务器也连接到这个网络上。为使这个例子通用，假定服务器运行UNIX并且这两个机器都运行TCP/IP协议。这种常规配置将用作贯穿本章客户机/服务器部分的一个示例。

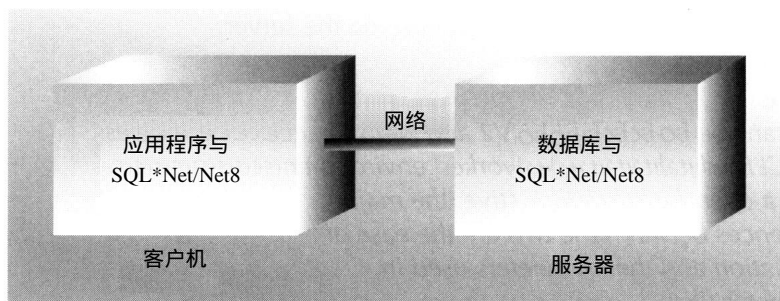


图17-1 客户机/服务器结构

图17-2形象地示出了这个配置。

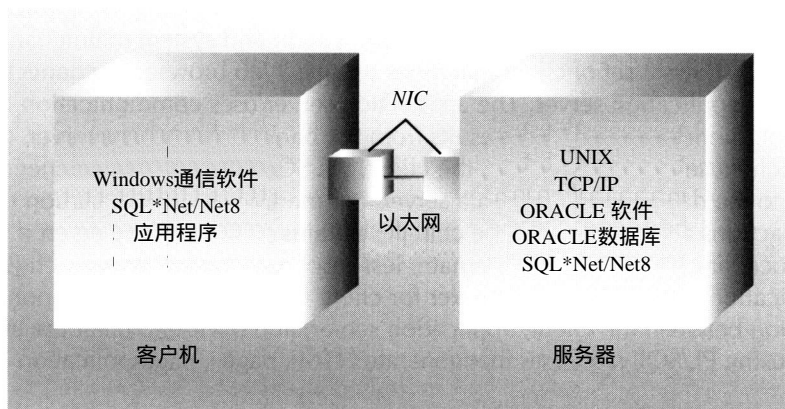


图17-2 客户机/服务器配置样例

客户机与服务器通信必须进行以下几步操作：

- 服务器必须配置成通过网络接受通信。
- 服务器必须识别哪个数据库允许网络登录。
- 服务器必须运行SQL \* Net / Net8。
- 客户机必须配置成通过网络进行通信。
- 客户机必须有充足的内存和磁盘资源可供使用。
- 客户机必须已安装有SQL\*Net / Net8并指定一个连接字符串。

如果跳过任何一个步骤，客户机应用程序将不能与服务器上的数据库进行通信。

注意 可以使用SQL\*Net V2或Net8来访问网络环境中的Oracle8数据库。从客户机/服务器的观点看，两者之间的主要差别是安装的难易和参数文件中所使用的参数。

## 17.2 瘦客户机配置概述

瘦客户机是客户机/服务器结构的一种扩展。传统上，两层的客户机/服务器结构往往会导致客户机的维护费用偏高。大部分情况下，客户机维护费用高是由于客户机操作系统的复杂性造成的。如果不需要客户机服务于应用程序且不把重点放在客户机的联网能力上，就可以创建一个非常“瘦的”、廉价且便于维护的客户机。任何一台能与网络连接的机器都可以作为客户机使用。客户机不再需要是一台具有大量资源的机器。近来，许多公司都把重点放在互联网或内联网上，所以现在都要求瘦客户机能包括一个像 Microsoft Internet Explorer或 Netscape Communicator那样的互联网浏览器。

如果客户机非常简单，它以前所担负的任务就必须在其他地方执行。通常，这些任务由网络上其他地方的一个应用服务器来完成。这样，一个瘦客户机结构一般要三台机器（三层）而不是两台。把应用任务集中起来还有助于减少应用程序和系统的维护费用。图 17-3展示了一组客户机，它们用一个 Web浏览器连接到一个中层应用服务器上。应用服务器使用通信线程来提供查询并从后端数据库服务器获得结果。

在客户机/服务器环境中，客户机通过SQL\*Net或Net8直接连接到位于服务器的数据库上，并在全部事务完成之前一直保持连接状态。即使没有出现处理操作，客户机也要保持连接。占用资源不太多的方案是使用一个应用服务器来担任客户机请求和服务器响应的中介器。使

用生成HTML页面的PL/SQL软件包来写客户机、应用服务器和后端数据库之间的交互操作。应用服务器接收从客户机通过Web浏览器或其他界面软件发来的信息请求，并连接到服务器上。接着对请求进行处理并从服务器返回信息。然后，应用服务器与服务器断开并把请求的信息返回客户机。图17-3示出了这种操作。

带Web浏览器的客户机

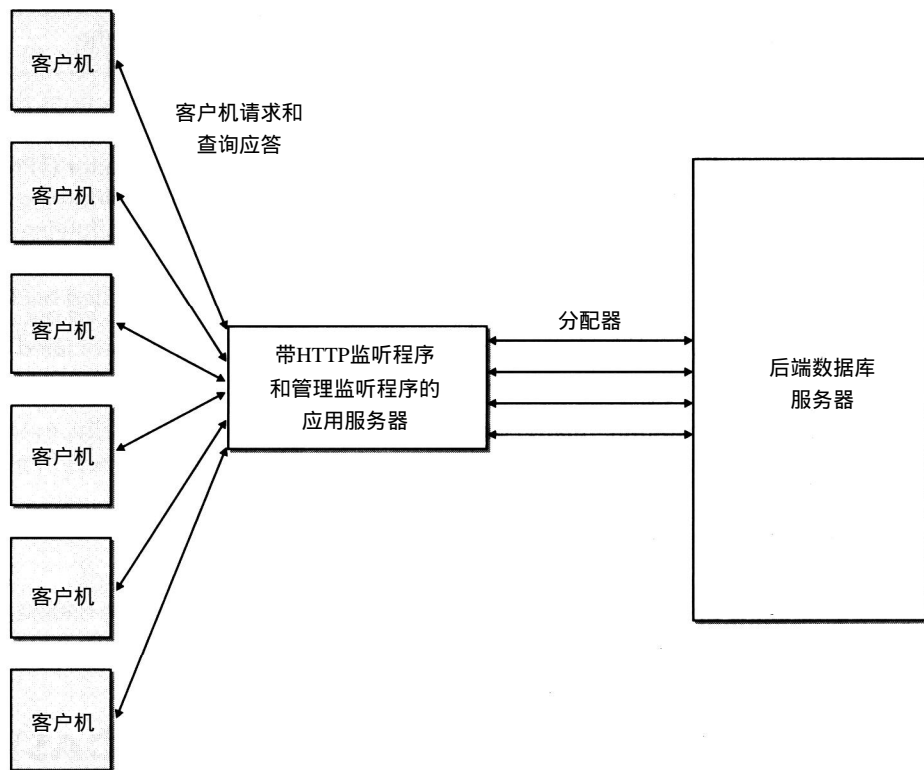


图17-3 瘦客户机体系结构

使用这种方法时，客户机不与数据库保持连接，应用服务器只有在进行请求并接收请求的结果时才与数据库保持连接。空闲进程并不占用网络资源，网络流量显著减少，可以更快地进行查询处理并且响应时间得到了改善。如果许多客户机都在频繁请求，可以对这种配置进行更改，使应用服务器能预先与数据库连接，以降低每次处理客户机查询请求时建立连接所需的开销时间量。多线程服务器和连接管理器用于保持与服务器的固定预先连接；这样连接就可以重复使用。

三层应用一般使用一个事务处理监控器 (TPM) 实用程序。TPM运行在Application Server上，支持对数据库的大容量事务处理。TPM通常有以下几个特征：

- 原子性 全部事务不是全部提交就是回滚。
- 一致性 可以把多个事务(例如一个贷方和一个相关的借方)作为一个集合来提交或回滚。
- 隔离性 即使最终结果变化一致，多个相关的事务也好像是串联执行。看不到中间步骤。
- 持久性 一旦一个事务被提交，改变就在数据库内固定下来。

在一个三层环境中，事务可以引用多个数据库。Oracle支持这样的分布式事务。有关分布式事务管理的详细情况，请参见第16章。

## 使用Oracle应用服务器

Oracle提供一个叫做Oracle Application Server(OAS)的附加应用服务器。OAS使用中央对象请求中介器(ORB)处理客户机和服务器之间的交互作用。使用可执行程序之间跨平台通信的工业标准(叫作Common Object Request Broker Architecture, CORBA)来处理对象请求。

与客户机/服务器技术一样,部分组件需要使用OAS。其中的一些组件由Oracle公司供应,而另外一些则不是(用星号\*表示)。

- 互联网浏览器\*
- 用于监听请求的HTTP监听器(Oracle提供Spyglass HTTP监听器)。
- 执行应用程序并返回HTML内容的磁带。磁带是Oracle提供的软件或第三方卖主软件,它使用Oracle Call Interface(OCI)调用或PL/SQL与数据库通信的第三方软件。
- 需要磁带的请求调度程序。
- Web Request Broker(WRB, Web请求中介器)。
- Oracle数据库(建立自己的数据库)。

### 安装和配置OAS

进行这种写操作时,没有专门为Oracle8i发布的OAS版本。因此必须使用Oracle8.0.5数据库来安装当前的OAS版本。一旦安装和配置好OAS,就可以用它来为Oracle8i应用程序服务。

你既可以使用基本的Oracle Application Server版本,也可以使用OAS Enterprise Edition版本(缺省),OAS Enterprise Edition版本提供的功能更多。本节的例子基于OAS Enterprise Edition版本4.0.7。若要安装OAS,必须确定以下几方面:

- 安装的版本。
- 安装类型。
- 安装站点名称和引导端口号。
- Node Manager Listener端口号。
- Node Manager的用户名和口令。
- Administration Utility Listener端口号。
- Oracle Web Listener名称和端口号。

在选定了安装的OAS版本后,系统就提示你要执行的安装类型: Typical、Complete或Custom。Typical选项将把所有组件安装在具有磁带Jweb、JCORBA、LiveHTML和PLSQL的一个节点上。将要安装的OAS组件是Web Request Broker、Oracle Web Listener和OAS Options。可以使用Complete选项把全部磁带和全部组件安装在一个或多个机器上。使用Custom选项时,可以把一个或多个组件安装在一个或多个机器上,以便在组合中提供装载平衡,这些组合是: OAS Options、OAS Listener,或者OAS Options和OAS Listener。若要添加组件,以前的安装必须保存在机器上。本例是选择Typical安装。

然后,安装进程将询问有关安装站点的信息。将提示一个站点名称和引导端口号。缺省站点名是website40,缺省引导端口为2649。下面的屏幕是询问Node Manager Listener信息并询问一个端口号(缺省为8888)、用户名(admin)和用户口令没有缺省设置。

下一个询问的信息是Administration Utility Listener信息。屏幕上有一个提示通知你:这是Administration Utility Listener的TCP端口号,为Administration Utility Listener输入的用户名和口令也将用于这个监听器。缺省端口是8889。



最后一个询问的信息是 Oracle Web Listener 名(www)和端口号(80)。一旦把这些信息全部收集在一起,就可以安装产品。完成安装时必须重新引导机器,然后完成配置。图 17-4示出了安装OAS时创建的服务。

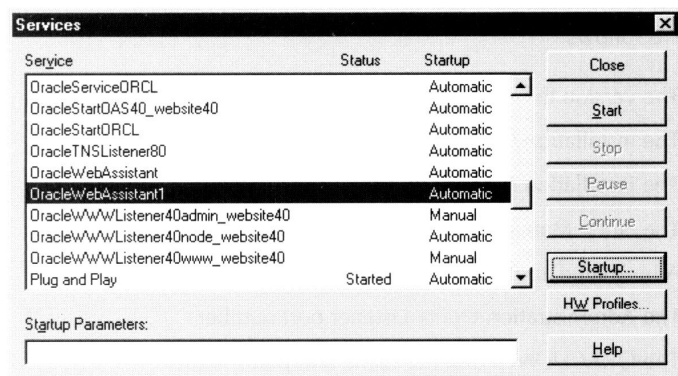


图17-4 安装OAS时创建的服务

一旦重新引导了机器,就可以使用 Web浏览器连接到 Welcome页面,这个 Welcome页面采用下述形式的Web地址:

```
http: // your_machine_name> : 8888
```

连接后,系统就向你提示这个站点的用户名和口令。这是在安装时提供的 Node Manager用户名和口令(admin原来是缺省值)。一旦输入正确的用户名和口令,就显示图 17-5所示的 Welcome屏幕。

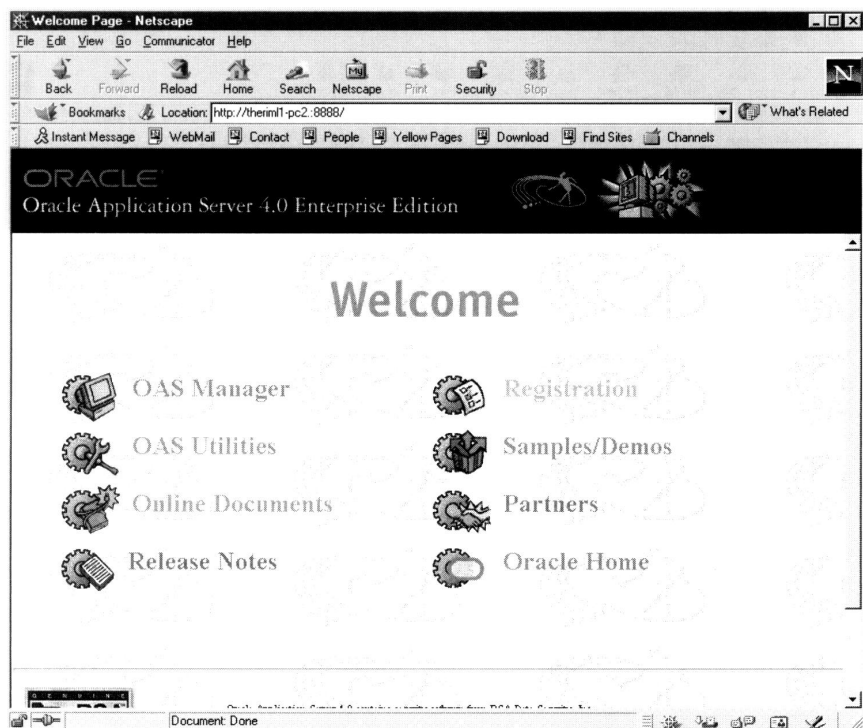


图17-5 OAS Welcome屏幕

若要完成这种配置，必须通过单击 OAS Welcome 屏幕上的 OAS Manager 选项来生成 OAS Manager 页面。OAS Manager 页面有两个列。Web 站点名位于左边，表示 Web 站点状态的一个表显示在右边。从右边框中选中 All 并单击绿色的 Start 按钮。完成 Start 选项后，具有 Application Server 状态的扩展 Web 站点映像就如图 17-6 所示。

在本章后面的几节中，将描述传统的客户机/服务器计算，此后是对客户机/服务器结构的瘦客户机扩充例子。

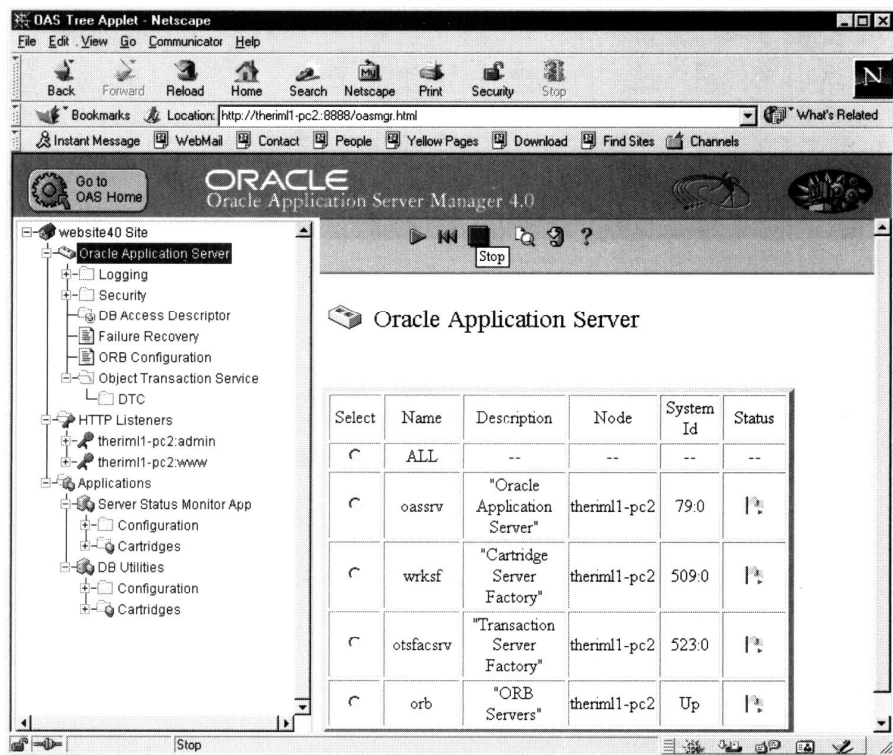


图17-6 选项被启动的OAS Manager页面

## 17.3 配置服务器

服务器的配置情况已在第 14 章中描述。服务器必须标识可以和哪个主机通信，规定哪个数据库可以使用并运行 SQL\*Net/Net8 监听器进程。

### 17.3.1 标识可用主机

主机就是能通过网络与另一个服务器通信的服务器。每个主机可以维护能与其进行通信的那些主机的清单。这个清单通常在 /etc/hosts 文件中。文件名中的 /etc 部分表示它位于 /etc 目录中。这个文件含有主机的互联网地址，加上主机名。并可以有选择地包括每一个主机的别名。下面列出 /etc/hosts 文件的样本部分：

```
130.110.238.109 nmhost
130.110.238.101 txhost
130.110.238.102 azhost arizona
```

在这个例子中，列出了三个主机。前两项把主机名 (nmhost和txhost)赋予互联网地址上。最后一项把一个主机名(azhost)和别名(arizona)赋予一个互联网地址。

UNIX工具可以使用一个域名服务器 (DNS)，在这种情况下，主机 IP地址可以不全部列入 /etc/hosts文件中。如果DNS正在使用中，就可以使用 nslookup命令来查询已知主机名的 IP地址 (反过来也是一样)。下面清单示出了查找DNS的IP地址的样例：

```
> nslookup txhost
Server: txhost.company.com
Address: 130.110.238.101
```

### 17.3.2 标识可用的服务

服务器进程监听客户机的连接请求。服务器进程直接把那些请求送到合适的 UNIX端口，然后就可以产生连接。SQL\*Net V2和Net8的服务已在第13、14章描述过。本章的例子将重点放在Net8的安装上。

UNIX服务器上的服务通常列在一个称作 /etc/services的文件中。文件名的 /etc部分表示它位于/etc目录下。与/etc/hosts文件一样，这个文件一般由 UNIX系统管理员进行维护。不必把全部服务都列在/etc/services文件中，但是这样做有助于防止通信端口的冲突。

尽管可以使用任意想使用的名称，但 Net8服务被称作Net8的监听程序。如果在一个服务器上配置不止一个监听程序，就必须为每个监听程序指定一个不同的服务名和不同的端口号。在 Net8中，被赋予的标准端口号是端口 1521。不过，通常也可以使用端口 1526。如果需要定义两个以上的服务，可以选择端口号；但是一定要检查主机上的服务文件，以确保不选择已赋予其他进程或程序的端口号。

可以使用多个监听程序来均衡负载输入的请求。/etc/services文件中的Net8条目如下所示。这个条目有两部分；服务名和端口号。一旦通过监听程序的端口进行连接，操作系统就把这个对话的通信传送到另一个端口，避免出现资源冲突。

```
listener 1521
```

### 17.3.3 标识可用数据库

所有在UNIX主机上运行并能通过网络访问的数据库都必须列入一个叫作 /etc/oratab的文件中。文件名的/etc部分表示它位于/etc目录中。这个文件由数据库管理人员维护。在客户机 / 服务器应用程序中使用的每一个数据库都必须列入 /etc/oratab文件。

下面显示的是/etc/oratab文件样本：

```
loc:/orasw/app/oracle/product/8.1.5.1:Y
cc1:/orasw/app/oracle/product/8.1.5.1:N
old:/orasw/app/oracle/product/8.1.5.0:Y
```

注意 oratab文件的位置依平台而定。例如，在一些 UNIX平台上，oratab文件位于 /var/opt/oracle中。

在/etc/oratab文件中，每个条目有三个部分，用冒号 (:)分开。条目的三个部分是实例名 (Oracle\_SID)、Oracle软件根目录(Oracle\_HOME)和一个用于指示实例是否在主机启动时启动的标志(这是个必需有的标志，但与SQL\*Net 通信无关)。这些都已在第14章中描述过。

这个例子示出了三个实例的条目：LOC、CC1和OLD(每个条目中的第一个部分)。前两个实例有相同的ORACLE\_HOME；第三个实例使用Oracle核心程序的旧版本(如条目中第二个部分所示)。LOC和OLD在服务器启动时自动启动，CC1必须人工启动(通过每个条目第三个部分中的Y和N标志设置)。

在Windows NT Server上，数据库将列在注册表的HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE\ALL\_HOMES\ID<sub>x</sub>键下。图17-7显示具有高亮显示的ALL\_HOMES条目的注册表。ALL\_HOMES值包括使用参数HOME\_COUNTER的Oracle号，最后一个ORACLE\_HOME将由带一个数字的参数LAST\_HOME识别。在图17-7中，ALL\_HOMES的值是3，而LAST\_HOME的值是2。

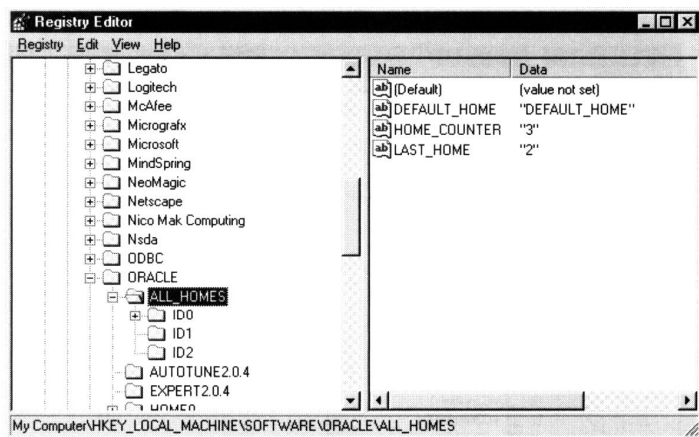


图17-7 注册表中的ALL\_HOME条目

每个独立的ORACLE\_HOME都有一个带不同数字(例如ID0和ID1)的ID。在图17-7中，条目是ID0、ID1和ID2。在列出的每个ID<sub>x</sub>中，都有一个带目录位置的PATH参数。例如，图17-8示出了ID1条目。这个条目的NAME是ora8i，PATH值是D:\ora8i。

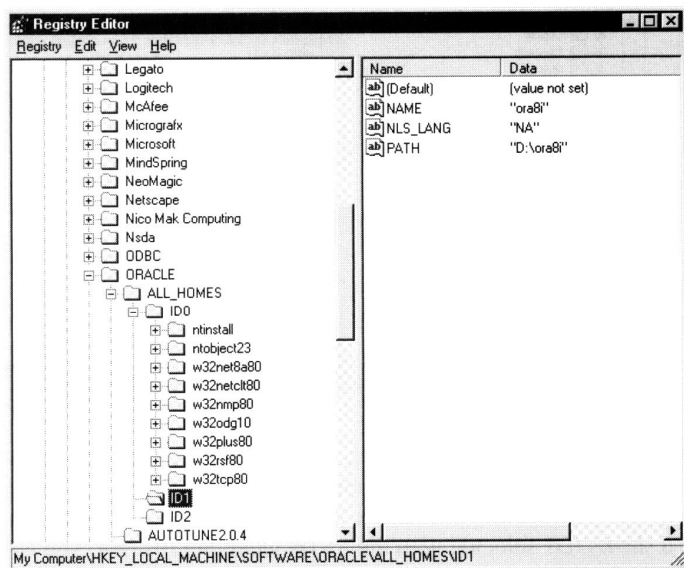


图17-8 显示ID1条目的Windows NT注册表



### 17.3.4 启动Net8

在Windows到UNIX例子中，如果数据库应用程序要使用 Net8，就必须在服务器上启动监听器进程。这个进程可以利用的所有参数在第 14章中已列出。

下列命令启动list1监听器的监听器进程：

```
> lsnrctl start list1
```

若要检查监听器的状态，请执行 `lsnrctl status service_name` 命令，如下所示：

```
> lsnrctl status list1
```

有关对 `lsnrctl` 命令选项的详细描述，请参见第 14章。对于 Windows NT，可以用下述三种方式之一来启动和关闭监听器：

- 1) 从 Services 屏幕启动或关闭监听器。
- 2) 使用 MS DOS 窗口并像为 UNIX 发布命令一样发布一些命令。
- 3) 从 MS DOS 提示符或一个批文件发布合适的命令。

下面的清单示出了第三个选项的一个例子，这个例子交互执行 `net start` 命令：

```
net start oracle<version_name>TNSListener
net stop oracle<version_name>TNSListener
```

`version_name` 变量是一个值，它存储在注册表的 ID 部分的 NAME 参数中(见图 17-8)。

## 17.4 配置客户机

在任何通信开始之前，客户机必须物理地连接到网络上，并且必须已安装好网络通信软件。客户机中的 NIC 必须由通信软件支持。有关所支持的通信软件包的列表，请参见《Net8 管理员指南》中客户机操作系统部分。每当升级通信软件包或客户机的 Net8 版本时，一定要检查这个列表。

### 17.4.1 标识可用的主机

就像服务器必须识别其可用主机一样，客户机也必须指定能与其连接的主机。其典型做法是，通过在结构上与服务器的 `/etc/hosts` 文件相同的文件来使用一个域名服务器 (DNS)。

`hosts` 文件含有主机的互联网地址和主机名。可以有选择地包括每个主机名的别名。下面列出了 `hosts` 文件的样本部分：

```
130.110.238.109 nmhost
130.110.238.101 txhost
130.110.238.102 azhost  arizona
```

在这个例子中，列出了三个主机。前两条目是把主机名 (`nmhost` 和 `txhost`) 赋予一个互联网地址。最后一条目是把主机名 (`azhost`) 和一个别名 (`arizona`) 赋予一个互联网地址。如果网络工具使用 DNS，就可以不用本地 `host` 文件列出主机 IP 地址。如果正在使用一个瘦客户机结构且不使用 DNS，则应用服务器的名称及其 IP 地址将在这个文件中。实际上，这个样本中定义的服务器都可以是一个应用服务器。因此，若要使识别更容易，只要在注释前加上一个磅符号 (#)，就可以把该注释添加到该条目的末尾。例如：

```
130.110.238.101 txhost      # HR Application Server
```

### 17.4.2 客户机说明

若Windows客户机能支持这些要求，必须满足下面这些最低条件。

#### 1. 硬件

- IBM、Compaq或与80286处理器(或更高) 100% 兼容的PC。
- 用于存储操作系统文件、Net8、通信软件和应用软件的足够硬盘空间。
- 安装时使用的磁盘驱动器。
- 网络通信用的NIC。

#### 2. 内存

- 运行网络软件、Net8和应用软件的足够内存，推荐至少为 16MB。

#### 3. 软件

- Microsoft Windows
- Net8 for Windows，以及TCP/IP协议适配器。
- 网络通信软件。

Net8 TCP/IP for Windows软件占用100KB磁盘空间和大约 120KB内存。调整客户机时要记住：应用前端程序、操作系统和通信软件需要的资源要比 Net8所需的资源大得多。

如果处理器比较慢，则程序通常能运行（只是比希望的慢一点）。如果可利用的内存不足，就根本不能运行应用程序。由于这个原因，PC机上至少要有16MB的内存，越大越好。

## 17.5 中间层应用服务器

Application Server装有一个或多个应用程序可执行的代码。如果没有使用本章前面所述的Oracle Application Server附加选项，当客户机请求连接时就需要运行一个程序，以便把这个客户机连接到应用程序上。并且需要应用程序中的另一个程序把该应用程序与数据库连接起来。某些第三方销售商能供应执行这些任务的软件。销售商供应的软件可以为每个与应用程序连接的客户机提供各自的用户名和口令，并且只有一个用户名和口令执行与数据库的连接。

作为客户，应使用提供给你的个人用户名和口令向应用程序证明自己；该应用程序本身又使用一个完全不同的用户名和口令以及单独的 Net8连接而连接到数据库上，以执行你请求的操作。

### 17.5.1 运行Net8

现在机器已安装好并准备通信，你要做的所有工作是确保用来与远程数据库进行通信的工具得到正确配置。如第11章所述，当从Oracle的一个版本移植到另一个版本时，需要经常修改工具。在数据库服务名的语法方面，工具之间也有一些不同。例如，有些工具要求整个服务名用小写输入和用双引号括起来(尽管这是个别情况)。每当网络硬件、通信软件、操作系统或Net8版本改变时，都一定要完整地测试连接。

### 17.5.2 Oracle与防火墙

许多组织都采用防火墙来保护他们的站点，以避免有意或无意地泄露其商业信息。防火

墙是运行在服务器上的一个软件程序，根据预先制定的某种保密准则格式接受或拒绝连接。只要用户名和口令正确，就能通过防火墙成功访问数据库。在其他配置中，可能需要安全字卡(safeword card)或其他同步传递密钥问答系统。一旦对防火墙的询问成功地响应，就允许访问防火墙机器上的信息；或者在大多数情况下，访问位于防火墙后面的另一个机器。

若要访问防火墙后面的一个 Oracle 数据库，需要满足以下准则：

- 除非网络管理员已为远程主机设置了一个别名，否则 tnsnames.ora 文件必须完全指定这个主机。
- 必须恰当配置基本网络。服务器必须能够相互 ping，防火墙必须允许这种通信量形式，远程监听器必须正在运行和可用。
- 必须知道远程站点的监听器端口号，把这个端口号存放在 tnsnames.ora 文件中。

如果你在公司总部工作并且必须访问现场办公室防火墙后面的数据库（指定为 fld.com 位置上的机器名 server1），HOST 文件中的条目可能是：

```
server1.fld.com
```

在 FLD1 实例的 tnsnames.ora 文件中，会有一个如下所示的条目；

```
fld1 =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)  
        (HOST = server1.fld.com) (PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SID = fld1)  
    )  
  )
```

如果服务器名、端口号或实例名发生变化，必须要确保你的站点和远程站点之间保持一致。