

在 UNIX 平台上数据库应用程序的安装及一些技巧

周 星 (上海市云台路 30 弄 200126)

邱雯瑾 (上海市桂花苑 14 号 200233)

在完成一个大型数据库应用程序的开发工作后,往往需要考虑该应用程序的安装问题。即如何在一个全新的环境中顺利、方便、安全地安装该应用程序。本文将通过在 UNIX 环境中安装一个 ORACLE 应用程序为例,探讨这方面的一般方法及一些技巧。

为了安装 ORACLE 应用程序,一般需要在数据库中创建以下数据对象:

- 数据表(TABLE)
- 数据视图(VIEW)
- 存储程序(STORE PROCEDURE)
- 包(PACKAGE)
- 序列(SEQUENCE)

然后需要为这些数据对象创建别名(SYNONYM)和授权(GRANT)。

最后往往还需要将一些基本的配置(CONFIGURE)数据输入到数据库中。

整个安装程序分为两部分:第一部分是创建一个 ORACLE 数据库用户;第二部分是让这个新用户登录然后进行应用程序的安装。

创建新用户是一件非常简单的事,这里不再叙述。在下文中我们假定这个新用户的用户名和口令分别为 \$INS-U 和 \$INS-P(值得注意的是,我们极力推荐创建一个新用户来安装应用程序,因为这将会对安装过程带来极大的便利)。

第二部分的安装程序是一个 UNIX 命令文件(script),它会调用其他的 UNIX 命令文件和一些 ORACLE SQL 文件。其主体只是调用了若干个子函数,如下所示:

```
f-main()
|
|
|   f-create-table
|   f-create-sequence
|   f-create-sp
|   f-create-synonym
|   f-grant
```

```
f-load-data
|
```

其中, f-create-table、f-create-sequence、f-create-sp 分别创建该应用程序的数据表及视图、序列对象和存储过程及包。这些工作是相对简单的,只需调用相应的 SQL 文件即可:

```
f-create-table()
|
|   # -----
|   # Display the message
|   # -----
|   echo
|   echo ----- Creating tables ...
|   echo
|   # -----
|   # Call the sql file to create all the tables
|   # -----
|   sqlplus $INS-U/ $INS-P @./sql/create-table.sql
```

其中 create-table.sql 是一个包含有创建数据表命令的 SQL 文件。其他的子函数也是完全类似的。

在创建别名和授权时,我们采用了一个非常精巧和实用的方法。通常来说,创建别名和授权,必须一一列出所有的数据对象名并为每一个数据对象编写一条命令。例如:

```
CREATE PUBLIC SYNONYM PKG1 FOR INS-U. PKG1;
CREATE PUBLIC SYNONYM TBL1 FOR INS-U. TBL1;
```

无疑这种方法是非常麻烦的,因为必须保证这个文件和所有创建数据对象的文件保持一致。当改动任何创建数据对象的文件时,必须同时修改该文件。另外,数据对象的拥有者是硬编码的(这里是 INS-U),不具有通用性。

我们的思路是利用 SQL 技巧通过对 ORACLE 系统

数据表的查询动态地生成这个文件。这样就避免了硬编码和对其他创建数据对象 SQL 文件的依赖。同时由于这个 SQL 语句是非常简洁明了的,也便于我们的维护工作。该 SQL 语句如下所示:

```
SELECT 'CREATE PUBLIC SYNONYM ' || OBJECT-NAME ||
      ' FOR ' || OWNER || '.' || OBJECT-NAME || ';'
FROM   ALL-OBJECTS
WHERE  OWNER = USER
AND    OBJECT-TYPE IN ( 'TABLE' ,
                        'VIEW' ,
                        'SEQUENCE' ,
                        'PROCEDURE' ,
                        'PACKAGE'
                      )
ORDER BY OBJECT-TYPE ,
         OBJECT-NAME
;
```

大家可以看出,这个 SQL 语句将挑选出所有属于当前用户的对象,并凑成相应创建别名的语句返回用户。换句话说,这个 SQL 语句将返回和前面所示的那个“硬编码”文件内容完全一样的结果。其完整的 SQL 文件如下:

```
SET HEADING OFF
;SET FEEDBACK OFF ;
SPOOL &1 ;
SELECT 'CREATE PUBLIC SYNONYM ' || OBJECT-NAME ||
      ' FOR ' || OWNER || '.' || OBJECT-NAME || ';'
FROM   ALL-OBJECTS
WHERE  OWNER = USER
AND    OBJECT-TYPE IN ( 'TABLE' ,
                        'VIEW' ,
                        'SEQUENCE' ,
                        'PROCEDURE' ,
                        'PACKAGE'
                      )
ORDER BY OBJECT-TYPE ,
         OBJECT-NAME
;
SPOOL OFF ;
START &1 ;
COMMIT ;
EXIT ;
```

它的工作原理是:首先动态生成这个包含创建别名命令的 SQL 文件(&1,即调用该 SQL 文件是所传入的第一个参数),然后直接执行这个 SQL 文件。同时通过阅读这个 SQL 文件,就可以确切地知道我们执行了哪些命令。

相应的在 UNIX 命令文件中,只需直接调用这个 SQL 文件即可:

```
f-create-synonym()
|
|
# -----
# Display the message
# -----
echo
echo ----- Creating the synonym ...
echo
# -----
# Compose the spooling file name
# -----
DbSynLst = $ INS-LOG-PATH/SYN. 'date + % y% m% d. %
H%M%S' .SQL
echo The actions executed will be logged in the file $ DbSynLst.
# -----
# Call sqlplus to drop the database objects
# -----
sqlplus $ INS-U/ $ INS-P @./sql/create-synonym. sql $ Db-
SynLst
|
```

对数据对象的授权也可以采用完全一样的技巧。

最后,装载配置数据我们可以直接利用 sqlldr 来实现,如下所示。

```
sqlldr userid = $ SR/ $ J-PSD@ $ J-DB control = IRIDIUM-
CONFIG. CTL
```

这样,这个数据库应用程序的安装工作就基本完成了。另外值得提请读者注意的一点是如何保留整个安装过程的原始记录(LOG)。在整个安装过程中将会有许多的信息显示在屏幕上,我们希望做到的是用一种简单的方法将这些信息保存到一个 UNIX 文件中以作参考。

我们采用的方法如下:

```

# -----
# Main process
# -----
# -----
# Compose the log file name
# -----
DbSetupLog = $ LOG-PATH/DBSETUP. 'date + % y% m%
d. % H%M%S'. LOG
echo The actions executed is logged into $ DbSetupLog
# -----
# Process
# -----
f-main 2>&1 | tee $ DbSetupLog
# -----
# Check the warning information
# -----
warningnum = 'grep -i Warning $ DbSetupLog | wc -l'
if [ $ warningnum -gt 0 ]
then
banner WARNINGS!!!
fi

```

其中 f-main 是在第一个例子中的那个主函数。在这里编写一个函数是使用 tee 命令的关键。同时大家也可以看到,将这些信息记录在一个文件中也便于我们检测安装过程中可能出现的错误和警告。举例来说,如果在安装存储过程时出现一些编译错误,那么 ORACLE 会显示 "Warning: store procedure created with compilation errors.", 这样只要在该 LOG 文件中找到有关的关键字,就可以提醒安装者一些可能存在的错误。

最后,顺便指出的是如果需要彻底清除或者部分清除已经安装的应用程序(uninstall)的话,可以通过直接删除这个应用程序安装用户(\$ INS-U)来达到彻底清除的目的,也可以利用创建别名时所采用的技巧来动态生成一些 SQL 文件从而达到部分清除的目的。

(来稿时间:1998 年 1 月)