基于 INI 文件的 PowerBuilder 事务对象参数读取与加密①

Fetching and Encrypting of PowerBuilder Transaction Object Parameters Based on INI File

马　丽　(西华师范大学 商学院 四川 南充 637002)

摘　要： 在数据库应用系统开发中，INI 文件常用于存储应用系统连接数据库服务器时所需的事务对象参数。为了保证数据库的安全，还必须对 INI 文件中的事务对象参数进行加密处理。针对上述问题，本文研究并给出了基于 INI 文件的 PowerBuilder 事务对象参数读取与加密实现方法。实际应用结果表明了本文方法的有效性。

关键词： PowerBuilder 事务对象参数 INI 文件 读取 加密

1　引言

PowerBuilder[1-3]作为优秀的 C/S(Client /Server，客户机/服务器)架构数据库应用系统开发工具之一，支持面向对象技术并具有良好的可视化开发功能[4]，多年来已在电信、金融、保险、铁路、石油等各类行业的软件项目开发中得到广泛应用。在 C/S 模式中，应用系统客户端软件通过事务对象参数来连接数据库服务器。而在通常的应用开发中，一般都使用 INI 文件来保存这些事务对象参数。但由于 INI 文件能够被任何文本编辑器打开，缺乏必要的安全保证，故可能会导致非法用户盗用事务对象参数，从而进入数据库服务器窃取和破坏数据。因此，如何对 INI 文件中保存的事务对象参数进行加密处理，也就成为数据库应用系统开发中的重要工作。针对上述问题，本文研究了 INI 文件的 PowerBuilder 事务对象参数读取方法，并着重探讨了对事务对象参数的加密处理，并给出了相应的实现程序。

2　PowerBuilder事务对象参数

PowerBuilder 提供了一个默认的全局事务对象 SQLCA(SQL Communications Area)来完成客户端程序与数据库服务器之间的连接。SQLCA 有 15 个属性参数，其中 10 个用于连接数据库，另外 5 个则用于接受数据库返回的操作状态信息(成功或失败)。在应用系统运行时，相关参数需要赋给正确的值，才能连通数据库服务器。表 1 对与本文有关的 5 个参数进行了具体介绍。

对事务对象参数加密的最简单方法，是将其写入应用系统的“open”事件代码，然后编译成可执行的.exe 文件。但是这给应用系统的维护工作带来很大麻烦。因为在应用系统使用过程中，用户名、密码甚至数据库管理系统都有可能变化，尤其是数据库用户密码会经常更换，以保证数据库服务器的安全。一旦事务对象参数发生变动，就只能对应用系统源程序作相应改动，再重新编译、安装。因此，在实际开发中通常使用 INI 文件来保存事务对象参数。

表 1　PowerBuilder 事务对象参数

参数名	数据类型	参数描述
DBMS	String	要连接的数据库管理系统标识符
ServerName	String	数据库服务器名称
LogID	String	数据库用户名
LogPass	String	数据库用户密码
DBParm	String	数据库连接参数

① 收稿时间:2008-08-05

3　基于INI文件的事务对象参数读取

3.1 INI 文件的应用背景

随着现代企业的飞速发展，企业数据的急剧膨胀，许多企业正在着手进行全面信息化建设，对企业信息系统进行统一规划。而规划中的重点内容之一，就是对企业数据库的建设。这其中存在以下情况：

(1)企业目前使用了多种数据库，导致数据过于分散，不便于企业进行各系统之间的数据共享，需要将数据集中和统一到一种数据库；

(2)企业目前使用的数据库是小型数据库，越来越难以满足对当前企业海量数据的管理，需要更换成大中型数据库；

(3)各种数据库不定期地需要升级；

(4)数据库用户的增删；

(5)数据库用户口令的更换。

因此，开发人员在开发 C/S 模式的数据库应用系统时，如果没有考虑到系统的可移植性，将数据库连接参数值直接写在系统源程序中，则会给系统开发和维护工作造成极大的麻烦——开发人员不得不先修改系统源程序中的数据库连接参数值，然后再将源程序编译生成新系统，最后替换旧系统。这显然是个繁琐且易出错的过程。而采用 INI 文件保存数据库连接参数，则可以在不涉及修改系统源程序的前提下，满足多用户(数据库用户的增删、数据库用户口令的更换)访问多种数据库(数据库更换/升级)的需求。

3.2 INI 文件的读取

用于数据库连接参数存储的 INI 文件也称为初始化文件或配置文件，它是一种有固定格式的文本文件，其内容包括节、关键字、值，并以.ini 作为文件扩展名。以图 1 所示的 INI 文件 start.ini 为例：

```
[database]
DBMS = "O84 Oracle9i (9.0.1)"
ServerName = "oradb"
LogPass = "admin"
LogId = "sys"
DBParm = "ConnectAs='SYSDBA'"
```

图 1　INI 文件示例

start.ini 用来保存应用系统连接数据库服务器 Oracle 9i 时所需事务对象参数。其中，[database] 说明了一个节，“database”是该节的名称。等号左边的 DBMS、ServerName、LogPass、LogId 和 DBParm 是数据库连接参数的名称，等号右边则是各个参数所对应的值。一个 INI 文件中可以有多个节，每一节中保存一类参数。

PowerBuilder 提供了 ProfileString、SetProfileString 两个函数来读写 INI 文件的内容。其中，ProfileString 执行读操作(返回值是字符串)，SetProfileString 执行写操作，这两个函数的格式为：

①(ProfileString(filename，section，key，default)

②(SetProfileString(filename，section，key，value)

其中，filename 指定 INI 文件的名称；section 指定 INI 文件中的一个节的名字；key 给出在某一节中设置的参数名；default 给出参数默认值，如果指定的文件、指定的节或指定的项目不存在，则返回该默认值；value 则是向 INI 文件写入的参数值。INI 文件的读取流程如图 2 所示。

读取 INI 文件的 PowerBuilder 应用程序代码如下：

```
//从 INI 文件读取数据库管理系统参数
SQLCA.DBMS = ProfileString(ls_startini,"database","DBMS","")
//从 INI 文件读取数据库服务器参数
```

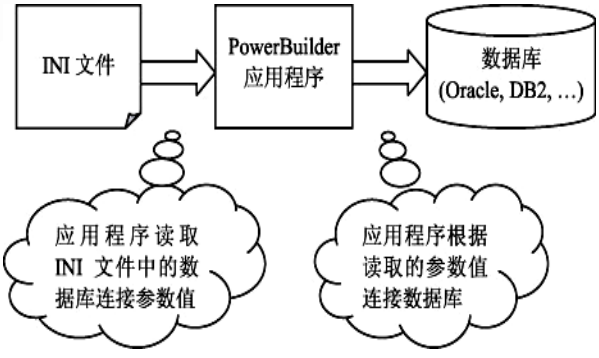

图 2　INI 文件读取流程

```
SQLCA.ServerName = ProfileString(ls_startini,"database","ServerName
```

```
","")
//从 INI 文件读取数据库用户密码
SQLCA.LogPass=ProfileString(ls_startini,"datab
ase","LogPass","")
//从 INI 文件读取数据库用户 ID
SQLCA.LogID=ProfileString(ls_startini,"database
","LogID","")
//从 INI 文件读取数据库连接参数
SQLCA.DBParm =
ProfileString(ls_startini,"database","DBParm","")
SQLCA.AutoCommit = False
//连接数据库
Connect Using SQLCA;
```

其中，SQLCA 是 PowerBuilder 提供的默认全局事务对象，系统在连接数据库前，必须先为事务对象的数据库连接属性(包括 DBMS、ServerName、LogPass、LogID、DBParm 等)赋值；ls_startini 是开发人员定义的字符串变量，其值为 INI 文件 start.ini 在当前计算机上的存储路径；“Connect Using SQLCA;”则是 PowerBuilder 用来连接数据库的语句。

需要说明的是，以上程序代码只是用于连接 Oracle 9i 数据库，而 PowerBuilder 还为用户提供了对其他多种数据库(例如 DB2、Sybase、MS SQL Server、Access 等)的访问接口，不同的数据库接口所需设置的事务对象属性也不同。对于开发人员而言，最好的了解办法就是在 PowerBuilder 开发环境中先连接好后台数据库，然后参看 Database Profile 画板的“Preview”属性页，其内容即为 PowerBuilder 自动生成的数据库连接代码(见图 3)：

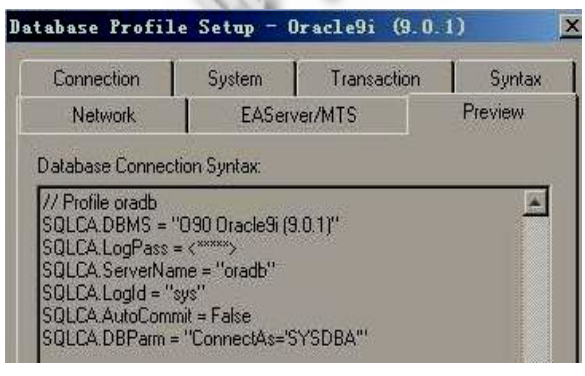

图 3　Database Profile 画板的“Preview”属性页

从“Preview”属性页可以看到连接当前数据库所用到的事务对象属性，然后据此编写相应的 INI 文件。

在创建好 INI 文件后，每当后台数据库发生变动(如更换数据库、更换数据库用户或密码、数据库升级)时，开发人员只需要对 INI 文件中的相关参数及其值进行修改，就可以实现对数据库应用系统连接后台数据库的调整，从而保证系统对数据的正常读取。

4　基于INI文件的事务对象参数加密

对 INI 文件中事务对象参数加密的基本思路是：先使用加密函数将事务对象参数转变为密文，写入 INI 文件；应用系统启动时，从 INI 文件中读取密文，进行解密处理后生成事务对象参数的明文，然后据此连接数据库服务器。具体的加密处理流程如图 4 所示：

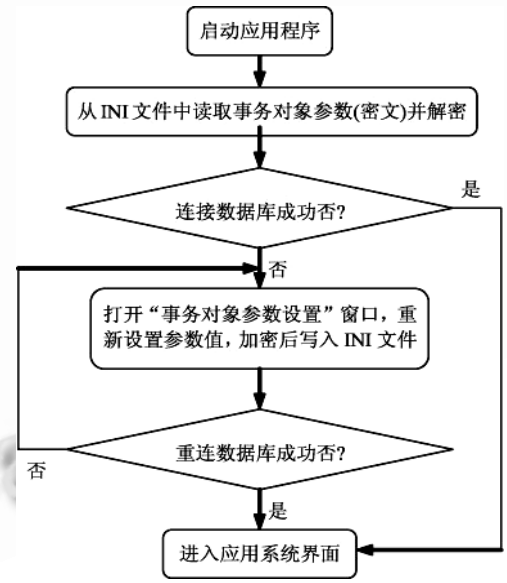

图 4　加密处理流程图

对加密/解密函数的调用方法大致有两种方式。第一种方式是动态链接库(DLL)，即采用 Visual C++、Delphi 等开发工具编写并生成 DLL，然后在 PowerBuilder 应用程序运行时链接调用 DLL，例如文献[5]在 VC 环境中生成的 PBDLL 动态链接库。第二种方式则是本文采用的 PowerBuilder 用户自定义函数。下面给出一个由用户自定义函数完成的基于对称密钥算法体制的加密算法。该算法的优点是程序代码简洁，加密效果好，且支持对汉字的加密/解密处理，在实际

开发中的应用结果令人满意。相应的 PowerBuilder 用户自定义函数代码如下：

```
//------------------------------------//
//函数名称:    f_lockparm              //
//功能描述:    对事务对象参数进行加密处理  //
//作用范围:     public                 //
//参数说明:ls_connectparm 用户输入的事务对象参数 //
//返 回 值:ls_lockwords 经过加密后形成的密文//
//------------------------------------//

    string ls_lockwords
    integer i
    for i = 1 to len(ls_connectparm)
    ls_lockwords=ls_lockwords+char(asc(mid(ls_connectparm,i,1)) + i)
    next
    return ls_lockwords

//------------------------------------//
//函数名称:          f_unlockparm      //
//功能描述:    对事务对象参数进行解密处理  //
//作用范围:       public               //
//参数说明:ls_lockwords 经过加密后形成的密文//
//返 回 值:ls_connectparm 用户输入的事务对象参数 //
//------------------------------------//

    string ls_connectparm
    integer i
    for i = 1 to len(ls_lockwords)
       ls_connectparm=ls_connectparm + char(asc(mid(ls_lockwords,i,1)) - i)
    next
    return ls_connectparm
```

经函数 f_lockparm 加密后的 start.ini 文件内容如下：

```
[database]
DBMS=P;3$TxhkuoDu-6H>A@D=
ServerName=ptdhg
LogPass=bfpms
LogId=t{v
DBParm=Dqqrji{I|G2_faSRR9
```

可见加密后的 INI 文件能有效隐藏和保护事务对象参数值。

在实际应用加密/解密函数时，加密处理是在应用系统“事务对象参数设置”窗口(如图 5 所示)中“保存设置”按钮的“clicked”事件代码中实现的：

```
/* 以下程序中，ls_startini 是开发人员定义的字符串型变量，其值为 INI 文件 start.ini 在当前计算机上的存储路径 */
    string ls_connectparm
    ls_connectparm=sle_dbms.text    // 读 取 DBMS 文本框的数据库管理系统参数
    SetProfileString(ls_startini,"database","DBMS",f_lockparm(ls_connectparm))  //调用加密函数 f_lockparm 完成对数据库管理系统参数的加密
    ls_connectparm=sle_servername.text    //读取 ServerName 文本框的数据库服务器参数
```

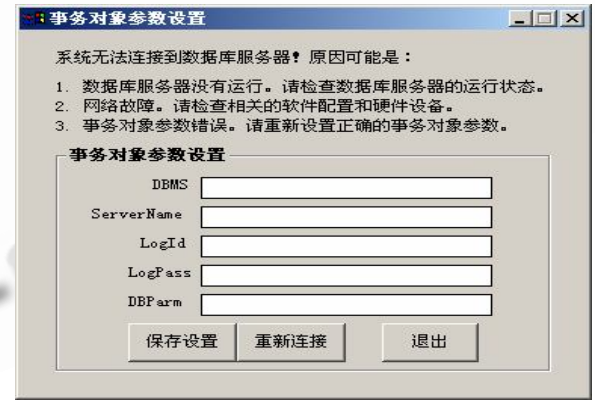

图 5 “事务对象参数设置”窗口

```
    SetProfileString(ls_startini,"database","ServerName",f_lockparm(ls_connectparm))  //调用加密函数 f_lockparm 完成对数据库服务器参数的加密
    ls_connectparm=sle_logid.text    // 读 取 LogId 文本框的用户 ID
    SetProfileString(ls_startini,"database","LogId",f_lockparm(ls_connectparm))  //调用加密函数 f_lockparm 完成对用户 ID 的加密
    ls_connectparm=sle_logpass.text    //读取
```

LogPass 文本框的用户密码

SetProfileString(ls_startini,"database","LogPass",f_lockparm(ls_connectparm))　//调用加密函数 f_lockparm 完成对用户密码的加密

ls_connectparm=sle_dbparm.text　//读取 DBParm 文本框的数据库连接参数

SetProfileString(ls_startini,"database","DBParm",f_lockparm(ls_connectparm))　//调用加密函数 f_lockparm 完成对数据库连接参数的加密

解密处理则在应用程序的“open”事件代码中实现：

```
string ls_lockwords
//从 INI 文件读取数据库管理系统参数的密文
ls_lockwords=ProfileString(ls_startini,"database","DBMS","")
//调用解密函数 f_unlockparm 完成对数据库管理系统参数密文的解密
SQLCA.DBMS = f_unlockparm(ls_lockwords)
//从 INI 文件读取数据库服务器参数的密文
ls_lockwords=ProfileString(ls_startini,"database","ServerName","")
//调用解密函数 f_unlockparm 完成对数据库服务器参数密文的解密
SQLCA.ServerName=f_unlockparm(ls_lockwords)
//从 INI 文件读取用户密码的密文
ls_lockwords=ProfileString(ls_startini,"database","LogPass","")
//调用解密函数 f_unlockparm 完成对用户密码密文的解密
SQLCA.LogPass=f_unlockparm(ls_lockwords)
//从 INI 文件读取用户 ID 的密文
ls_lockwords=ProfileString(ls_startini,"database","LogId","")
//调用解密函数 f_unlockparm 完成对用户 ID 密文的解密
SQLCA.LogId = f_unlockparm(ls_lockwords)
//从 INI 文件读取数据库连接参数的密文
ls_lockwords=ProfileString(ls_startini,"database","DBParm","")
//调用解密函数 f_unlockparm 完成对数据库连接参数密文的解密
SQLCA.DBParm=f_unlockparm(ls_lockwords)
```

5　结束语

对事务对象参数进行 INI 文件读取和加密处理，是使用 PowerBuilder 开发数据库应用系统的过程中非常重要的一个环节。通过在实际软件项目开发中运用本文的事务对象参数读取和加密实现方法，证明本文方法简洁实用。未来的研究工作将考虑进一步对加密算法密钥增加更复杂的易位、置换处理，以得到更好的加密效果。

参考文献

1 崔巍.PowerBuilder7.0 数据库应用系统开发教程.北京:清华大学出版社,2000.

2 郭宝利.PowerBuilder 8.0 完全解析.北京:电子工业出版社, 2002.

3 柯建勋,张涛,邵亮.PowerBuilder 9.0 进阶开发篇.北京:清华大学出版社,2003.

4 陈利剑,曾一.基于 PowerBuilder 的面向对象可视化.计算机工程与应用,2003,39(1):138－140.

5 赵学锋,张金隆,蔡淑琴,等.基于 PowerBuilder 的数据库连接参数加密实现.计算机系统应用,2004,13(8):49－51.