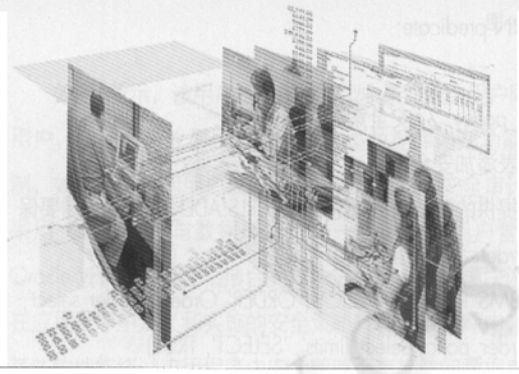
Powerbuilder应用实现中的若干问题探讨

Discussion about Several Problems in PowerBuilder Application Implementation



摘　要：文章论述了PowerBuilder应用在实现过程中的几个难以解决的问题和相应处理方法。同时，深入地分析了PowerBuilder的不足及弥补方法。

关键词：PowerBuilder　OLE　DLL　多线程

苏子林（烟台师范学院交通学院 264025）

PowerBuilder（以下简称PB）是美国著名的数据库及应用开发工具生产厂商Sybase推出的产品。作为一种可视化的，面向对象的开发工具，PB已被广大数据库应用开发人员所使用，并获得好评[1]。但是PB在其它应用方面，有很多不足，如通信方面、监控方面和与其它应用的集成方面等等。根据作者多年的PB编程实践，提出以下几个问题及弥补方法：

1 集成编程问题

由于PB不支持指针，没有专门实现多线程应用的函数和对象，因此在通信、监控或其他应用实现时，由于考虑其数据库管理编程的简洁高效，不得不采用与其他编程语言集成开发，如：VC、VB等。PB与其他语言的集成一般采用动态数据交换(DDE)、对象的连接与嵌入(OLE)或自动化技术(Automation)。自动化技术就是将某一个应用程序的功能（可能是一部分）当作一系列的对象来向其他的外部应用程序暴露（Expose），这个外部应用程序就可以使用这些对象[2]。在采用OLE或Automation 技术进行集成编程时，需要创建类型为OLEObject的变量，来作为OLE对象的代理，最后要调用destroy()函数删除此变量，否则程序会丢失内存。但是当OLE对象出现错误时，代理自动变为无效。再调用destroy()函数删除，程序会死机，但没有任何提示。因此代码必须写成：

```
if isvalid(lo_ole) then destroy(lo_ole) // lo_ole为创建的OLE对象代理变量。
```

2 多线程的实现问题

PB中没有专门用于线程管理的函数和对象，也无法直接调用Windows操作系统中的API函数进行多线程应用开发，但是提供了异步访问EAServer组件的共享对象登记、实例创建和撤销登记函数，即SharedObjectRegister()、SharedObjectGet()和SharedObjectUnregister()。客户端应用登记共享对象成功后，就启动一个单独的运行会话，并创建了一个共享对象，即启动了一个独立的线程。SharedObjectGet()函数创建一个共享对象的引用实例，即得到这个线程的句柄，这样PB客户端应用就可以通过这个句柄调用共享对象的函数，从而实现与EAServer组件的异步通信。实践证明共享对象线程可以不访问EAServer，而是进行其他耗时多的处理，而且PB应用可以登记多个共享对象，即启动多个独立的线程。这样就可以实现多线程编程。但是主线程必须采用Post方法触发后台线程的函数或事件；后台线程中必须调用Yield()函数使得主线程中的界面刷新或处理用户操作。

3 调用动态连接库问题

PB无法直接访问系统的底层资源，必须调用操作系统中的应用编程接口(API)函数，如获得系统的当前目录。此时必须声明下列外部函数：

```
FUNCTION ulong GetCurrentDirectory(ulong nBufferLength,ref string lpBuffer) LIBRARY "kernel32.dll"
```

然后书写下列程序得到当前路径（不得超过254个字符）：

string ls_temp / ls_temp=space(254) / GetCurrentDirectory(254, ls_temp)

如果其中没有语句ls_temp=space(254)来创建包含254个空格的字符串，给ls_temp分配内存，程序运行时会很不稳定，经常死机。笔者认为可能是PB的动态内存分配功能有问题。

4 数据窗口的打印问题

在PB程序中如果直接调用函数dwcontrol.Print()来打印数据窗口的内容，对于一部分打印机会出现乱码，采用下列代码则可以完全解决：

ulong li_job / li_job = PrintOpen() / PrintDataWindow (li_job, dwcontrol) / PrintClose(li_job)

出现这种问题的原因是dwcontrol.Print()函数采用与屏幕上相同的字体和布局在打印机上打印，而PrintDataWindow ()函数采用打印机设置的字体和布局打印数据窗口内容。如果某些打印机不支持数据窗口的字体，则会出现乱码。

5 数据窗口的事务问题

PB使用一个叫做逻辑工作单元(Logical Unit of Work简称LUW)的数据库事务处理概念，一个事务是由一条或多条SQL语句构成的一个LUW。在事务内部，所有的SQL语句作为一个逻辑整体，或者全部执行成功，或者全部执行失败[3]。在PB中有COMMIT、CONNECT、DISCONNECT和ROLLBACK四条语句进行事务处理。在事务对象(Transaction Object)的自动提交(AutoCommit)属性为假时，需要通过COMMIT或ROLLBACK语句结束事务。数据窗口的Update方法象SQL语句一样会开始一个事务，需要通过COMMIT或ROLLBACK语句结束事务。否则，系统的资源就会被锁定，直到当前SQL语句所在的窗口被关闭。在支持多客户端的网络编程时，如果在数据窗口的Update方法后，没有通过COMMIT或ROLLBACK语句结束事务，那么只有数据窗口被关闭后，系统才会释放事务锁定的数据库资源。也就是当多个客户端程序同时访问一个数据窗口时，会出现只有一个客户端程序可用，其他程序出现类似死机的等待状态。因此在PB应用程序，特别是在实时监控应用程序中，必须通过COMMIT或ROLLBACK语句及时结束事务。

6 数据窗口的 SetTrans 方法引起的不稳定问题

数据窗口有SetTrans和SetTransObject两个设置事务对象的方法。它们都可以使得数据窗口与数据库建立连接，从而通过数据窗口对数据库数据进行操作。SetTrans方法从指定的事务对象中，把相关数据拷贝到数据窗口的内部事务对象中。数据窗口利用其内部事务对象，自动进行数据库的连接和断开、事务的提交和回滚，如：数据窗口在每次检索(Retrieve)和更新(Update)后，都自动结束事务并断开数据库连接。SetTransObject方法把数据窗口与某个事务对象建立联系后，数据窗口控件就可以使用该对象访问数据库，每次执行检索(Retrieve)和更新(Update)时，数据窗口控件不需要与数据库建立连接，完成操作后也不会断开与数据库的连接，而且不自动进行事务的提交和回滚[4]。因此采用SetTransObject方法比SetTrans的应用程序有更高的运行效率，而且稳定性好。采用SetTrans方法的应用程序在网络条件不好时会经常死机，但不会出现数据窗口事务的死锁问题；采用SetTransObject方法时，务必添加事务处理的脚本，否则会出现资源锁定问题。综上所述，建议采用SetTransObject方法，并适当采用COMMIT或ROLLBACK进行数据窗口的事务处理，尽量减少数据库管理系统(DBMS)对数据资源的锁定时间。

应用程序的建立问题。PB应用程序可以附带动态库(DLL)或PB动态库(PBD)，但只能是其中之一。考虑应用程序的跨平台性能、文件大小和编译速度，一般选择PB动态库，反之则选择DLL[5]。对于较大的数据库应用程序，一般建议选择PB动态库。另外，在可执行文件的生成过程中，不要按“STOP”按钮，否则会破坏程序代码。有时出现错误被迫停止时，也会破坏程序代码，即再次编译，即使没有程序错误，也会出现很多编译错误。因此，建议在程序编译之前一定要备份程序代码。

7 结束语

文章提出了PB应用实现过程中的七个问题，并提出了相应的弥补方法。其中关于数据窗口的问题同样适合于数据存储对象(DataStore)。正确处理这些问题，有利于提高应用程序的运行效率，并大大提高应用程序的稳定性。这在监控和通信应用程序的开发方面尤为重要。

参考文献

1 赵斌、吉根林，Powerbuilder7.0应用程序发布方法的研究，计算机工程与应用,2002.23。

2 刘涛、潘存云，在Powerbuilder中利用自动化技术调用Excel处理与打印表格，计算机与信息技术,2002.10。

3 崔巍、林小茶等，PowerBuilder8.0高级应用技术，清华大学出版社,2002,61—70。

4 王丰锦等，PowerBuilder8.0 对象与控件技术详解，中国水利水电出版社,2002,108—110。

5 陈桂友、孙同景、雷印胜等，PowerBuilder数据库开发技术，机械工业出版社,2002,305—310。