

# Windows 平台下的 Pb6.0 应用程序的发行技术

广西师范大学计算机科学系 胡振宇 林士敏

本文讨论Windows平台下的PowerBuilder 6.0应用程序的发行中的一些关键性技术问题。主要包括建立Pb6.0应用程序的可执行文件时,创建资源文件和动态库问题,配置Pb6.0应用程序应提供的文件,安装程序Setup的制作方法和制作工具InstallShield的使用等。所论技术在广西高校计算机等级考试管理软件MCT的开发项目中得到验证和成功使用。

就PowerBuilder 本身来讲,它仅是一个开发工具,当开发出应用程序以后如何移植到终端用户的机器上,却是要开发者自己解决的问题。本文论述Windows平台下用PowerBuilder 6.0开发应用程序的发行的关键技术问题及解决办法。所论发行技术在广西高校计算机等级考试管理软件MCT的开发项目中得到验证和成功使用。

## 建立可执行文件

在PowerBuilder 6.0开发环境中开发并调试应用程序后,进入发行阶段的第一步是要建立可执行文件。PowerBuilder 使开发者可用两种方式建立可执行文件。第一种方式产生所谓的原生码(Native)的可执行文件。第二种方式产生真正的机器码可执行文件。对于每一种方式,又可根据是否使用动态库分为两种方法。当不使用动态库时,产生一个比较大的包括所有应用程序元素的可执行文件。当使用动态库时生成称为PowerBuilder 动态库PBD(原生码时)或DLL(机器码时)的文件和一个较小的可执行文件。

创建的PowerBuilder 原生(Native)的可执行文件实际上在它前面包含一个小引导程序。此引导程序引起PowerBuilder DLL的加载。PowerBuilder 放在原生EXE文件中的代码及PBD中的代码,是编译的PowerBuilder 代码即P代码(Pcode),而不是编译的Windows 代码。引导程序使PowerBuilder 解释EXE文件和PBD文件中的P代码。

虽然编译的机器码EXE文件是真正的可执行文件,但编译的DLL文件不能由其他应用程序使用,同时仍然要求对EXE和DLL文件提供运行环境。这是因为Pb6.0与以前的版本一样,大部分功能包含在DLL中,而不能包含在应用程序中。

### 1. 可执行文件

可选择创建原生(Native)的可执行文件或创建机器码可执行文件。当创建原生(Native)的可执行文件时,PowerBuilder把编译的P代码复制到EXE文件中,其顺序为应用程序指定的库搜索路径。当创建机器码可执行文件时,PowerBuilder则把编译的机器码复制到EXE文件。P代码与机器代码的主要的不同是,P代码是编译的PowerBuilder 代码(Pcode),而不是编译的Windows 代码,必须由PowerBuilder 虚拟机解释执行,而机器代码则可直接执行。

在生成可执行文件时,PowerBuilder 遍历库搜索路径。它仅复制显式引用的且不在将要成为动态库的库中的对象。这意味着不复制图形对象(位图、图标和指针)、窗口、数据管道及动态引用的Datawindow到可执行文件中,而要在一个资源文件中列出,以供使用。最后生成的原生(Native)可执行文件中包含Windows 引导代码,接着是对象的P代码,然后是任意动态指定的资源代码。而生成的机器码可执行的文件中,仅有可执行的文件及动态指定的资源代码。

### 2. 资源文件

资源文件实际上是一个后缀名为PBR的ASCII文本文件。资源文件的内容有图形文件(BMP, RLE)、图标文件(ICO)、光标文件(CUR)、数据窗口对象等。编译生成可执行文件时如果指定了编译选项,除了图形外,所有的对象都将编译成动态库文件,包括动态地指定DataWindow、窗口等等。但用户仍然需要在资源文件中指定图形对象(位图、图标,等等)。当应用程序编译为可执行文件时,通过搜索PBR文件上附加的资源名称,

将这些资源（即图形文件）编译到执行文件中，这样，用户计算机上就不再需要另外存储这些图形文件。通过PBR文件的方式将额外的资源包含在应用程序可执行文件中，可以减少交付给用户的文件数目。

创建资源文件的规则：

- 每行只列出一个资源的名称。
- 把所有期望拷贝到执行文件中资源列出。
- 如果是一个数据窗口，必须指定在哪个PBL中。其格式为库文件名.pbl（数据窗口对象名）
- 如果资源与应用程序在同一个目录中，只要写上资源名称即可。如果资源文件与应用程序不在同一目录中，则需要给出完整的路径和名称。

### 3. 动态库

PBD或DLL在概念上类似于Windows DLL，并且包含了PowerBuilder库文件的所有编译代码。PBD文件仅与同一个编译的PowerBuilder可执行文件相关联，并且必须是访问应用程序的库搜索路径的一部分。在生成可执行文件时可选择是否将PowerBuilder库文件编译成相应的动态库（PBD或DLL）。把代码放入动态库可使用户将自己的应用程序分割成更小的相关段。在运行时不是把完整的PBD文件整个加到内存中，而仅在需要时才加载PBD中单个对象。DLL文件也使用相同的机制。

### 4. 机器码与P代码的比较

应该承认，机器码比P代码的执行速度快，但在绝大多数情况下，机器码与P代码一样快。

机器码在PowerScript循环结构、浮点与整数运算以及调用函数等方面可大大提高性能，因此，在循环敏感型及算术敏感型脚本中最有利。但机器码比P代码占用的内存大得多，如果应用程序是数据库敏感型，或需要考虑可执行文件的大小，则应该使用P代码。机器码可执行文件在如下诸多场合并不能提高系统性能：使用本地数据库；DataWindow的处理；调用PowerScript系统函数（因为这些函数已经在VM虚拟机中被编译了）；以及绘制图形用户界面（GUI）等。在PowerScript内执行十进制运算和字符串处理对于P代码和机器码来说实际上是一样的。如果考虑使用便携交叉平台，那么P代码可能更灵活些。因为P代码独立于机器，而机器码则依赖于指定的平台。一些应用程序在编译成P代码时能很好地运行，但编译成机器码时却出现无法解释的崩溃和其他反常行为。

### 配置Pb6.0应用程序

要使开发的应用程序能脱离PowerBuilder开发环境运行，除了提交应用程序可执行文件（EXE）、应用程序的动态库（PBD或DLL）外，还需为应用程序配置以下资源：

#### 1. Pb6.0应用程序运行时的支持文件

Pb6.0应用程序运行时的支持文件DLL应放在与应用程序相同的目录中或放在搜索路径中的目录中。

#### 2. PowerBuilder数据库接口文件

如果应用程序使用了数据库（大多数情况下如此），就必须建立恰当的数据库接口文件。一般情况下都采用ODBC（Open Database Connectivity）数据源，因此应将PowerBuilder的ODBC数据库接口文件PBODB60.DLL和PBODB60.INI两个文件放在应用程序目录或出现在系统的查找路径上。

#### 3. ODBC数据源驱动文件

如果用户使用ODBC数据源，就必须安装和配置恰当的ODBC驱动文件。对于WIN32，应配置下面的ODBC驱动文件：

CTL3D32.DLL, ODBC32.DLL, DS16GT.DLL, ODBC32GT.DLL, DS32GT.DLL, ODBCAD32.EXE, MSVCRT20.DLL, MSVCRT.DLL, MSVCRT40.DLL, ODBC32.DLL, ODBC32.DLL, ODBC16GT.DLL, ODBCINT.DLL

#### 4. DBMS驱动文件

如果用户使用本地数据库，还要提供由DBMS供应商提供的数据库驱动文件。使用PowerBuilder自带的SQL AnyWhere数据库时应配置以下的数据库驱动文件：

WTR50T.DLL, DBL50T.DLL, WL50ENT.DLL, WOD50T.DLL, DBENG50.EXE

#### 5. OLE系统文件

如果应用程序中使用了OLE控件，那么终端用户的机器上应该有OLE系统文件，若它们不存在的话，应将下面文件拷贝到终端用户的WINDOWS\SYSTEM目录中：

COMPOBJ.DLL, OLE2NLS.DLL, CTL3DV2.DLL, OLE2PROX.DLL, OLE2.DLL, STORAGE.DLL, OLE2CONV.DLL, TYPELIB.DLL, OLE2DISP.DLL, STDOLE.TLB

### 安装程序的制作

在Windows平台上开发的应用程序不能象在DOS环境下经过简单的COPY就可发行和安装。这些应用程序一般都具有WINDOWS风格的界面和简单灵活的操作，需要大量的文件支持。更为重要的是，WINDOWS95/98对文件的管理方式与DOS相比有着本质的不同。许多应用程序要经过对注册表的适当设置，才能正确运行。一个应用程序应放在终端用户的机器上的什么地方、应安装哪些部件、应该由哪些文件来支持、应对注册表进行怎样的

设置等等一系列问题,如果都由终端用户来考虑、解决显然是行不通的。而开发者却往往不能亲自到用户的机器上进行安装(公开发行的商业软件更是如此)。因此,制作一个安装程序(Setup)就成为Windows平台应用程序发行的必不可少的重要一步。

### 1. Setup 的基本功能

Setup程序的唯一目的就是安装应用程序,应该由它来完成对安装过程的处理。Setup程序应用具备以下基本功能:

(1)传送文件。将需要安装的应用程序及有关的支持文件传送到目标机器上,并处理传送过程中的错误。此乃Setup程序最基本的功能。

(2)询问终端用户有关必要的信息,如用户名、产品序列号、目标路径等,并响应用户输入。

(3)给用户以改变想法的机会,如改变安装路径、取消安装等。

(4)注册表访问。如设置应用程序查找路径键、创建应用程序信息键等。

(5)卸载功能。对安装的文件及对注册表的设置应有卸载登记功能,以使用户在取消安装或在不需要时顺利卸载应用程序并改回注册表。

(6)安装完毕应给用户以完成提示。

### 2. 应用程序的安装规则

当把应用程序安装到机器上时,有许多规则应当遵从,其中大多数是常识性的。它们使用户的应用程序维护更加容易,而且较少产生由文件复制而引起的问题。

(1)应避免把任何文件拷贝到系统目录中,除非正在升级由外部的应用程序使用的常用文件(例如,ODBC或本机数据库驱动程序)。这有助于避免由于改变了这些区域的文件数而降级操作系统的性能。

(2)如果文件是一个仅在单个用户的应用程序之间共享的通用文件,就应在Program Files的子目录Common Files中建立一个入口。

(3)在Windows95/98和NT中,应把用户自己的应用程序放在Program Files目录下自己的目录中。

(4)对于Windows 95/98应把支持文件(HLP、DLL等)分布在称为System用户自己的应用程序的一个子目录中。然后将它的位置注册在应用程序的HKEY\_LOCAL\_MACHINE\Software入口处的App Paths子键的注册表中。

(5)应使用注册表存储有关应用程序的所有必需信息,而不应该修改autoexec.bat、config.sys、win.ini或应用程序的INI文件。

(6)在操作系统的启动引擎中制作一个入口,提供对

用户应用程序的访问。对于Windows95/98,应把应用程序的一个快捷图标放在桌面或在开始菜单的程序文件夹中。

### 3. 安装程序制作工具 InstallShield

InstallShield是InstallShield软件公司开发的著名的安装程序制作软件。在PowerBuilder6.0中,PowerSoft提供了一个免费的InstallShield版本--InstallShield5 Free Edition。这是一个商业化产品,是InstallShield5 Professional的普及版。InstallShield产品提供了易于使用的向导和一个集成环境来帮助创建应用程序安装程序。

### 4. 修改注册表

仅将PowerBuilder应用程序及其所需的支持文件拷贝到用户的机器上,还不能使应用程序很好地运行,其主要原因就是前面所提到的Windows95/98是通过注册表来管理文件的,因此还需要对注册表进行修改。Pb应用程序要求对注册表的修改主要有三处:应用程序信息键,应用程序路径键和应用程序卸载键。如果使用本地数据库和ODBC数据源,还必须对注册表中数据库和ODBC数据源两键进行修改。

如果由InstallShield制作安装程序,则应用程序信息键、应用程序路径键和应用程序卸载键均由InstallShield在创建Setup.rul脚本时自动生成相应的函数调用语句,位于函数ProcessBeforeDataMove()的定义中。若要修改某键的键值设置(比如应用程序路径键),可用函数RegDBSetItem()进行。

本地数据库和ODBC数据源两键的创建和设置可在函数SetupRegistry()的定义中进行。首先用函数RegDBSetDefaultRoot()将默认的根键设置为HKEY\_CURRENT\_USER。其次对于每一个要创建的键,先要使用函数RegDBKeyExist()检查该键是否存在。若不存在,则调用函数RegDBCreateKeyEx()创建之,然后调用函数RegDBSetKeyValueEx()设置其键值;否则,直接调用函数RegDBSetKeyValueEx()设置其键值。■

### 参考文献

- (1) Simon Gallagher, Simon Herbert 著,康博创作室译《PowerBuilder 6.0 程序设计大全》,机械工业出版社
- (2) William B. Heys 著,王芝 等译《PowerBuilder 6 开发指南》,电子工业出版社
- (3) 方泊汾 著《PowerBuilder 5.0 实用教程》,科学出版社