

PB 中实现动态 Word 报表的完整解决方案

Integrated Solution to Realize Dynamic Microsoft Word Reports in PB

郭彦红 滕桂法 李阅历 (河北农业大学信息科学与技术学院 河北保定 071001)

摘要:为了弥补在 PB 中采用 DW 对象实现报表方法的不足,利用 Blob 数据类型和 OLE 技术,本文介绍了在 PB 中根据实际输出数据动态实现 Word 报表的完整过程,包括如何将 Word 模板保存到数据库,以及 Word 临时报表的生成、根据实际数据的修改、保存、预览、打印和最终删除,并提供了分块 PB 代码。该代码在监狱信息化集成系统审批表的处理中进行了成功应用,完成了对审批流程中各级审批表的动态预览和打印操作。对此代码只要稍做修改,就可以应用到其他系统中去。

关键词:报表 PB Blob OLE 技术

1 引言

PowerBuilder(简称 PB)是 Sybase 公司的一个功能强大、易学易用的数据库应用程序开发工具。在报表的实现上,通过 PB 提供的数据窗口对象(DW),可以很轻松的实现关于报表的各种操作。但是对于复杂格式的报表来说,使用 PB 的数据窗口对象来实现也有一些令人头疼的地方:

- 报表大小和控件大小比例不易准确把握;
- 设计好的表格布局调整起来工作量大;
- 表格线粗细不等;
- 难以实现真正意义上的垂直居中。

而微软的 Word 是大家都熟悉的文档编辑工具,其功能强大,文档格式丰富,编辑修改方便。如果在实现报表的时候将 PB 与 Word 结合起来,将充分发挥二者的优点,并互补不足。本文正是提供了这样一个在 PB 中实现动态 Word 报表的完整解决方案。实现了将 Word 模板保存在数据库中,在需要的时候生成 Word 格式的临时报表,并能根据不同的输出数据对报表内容进行修改,对保存后的报表提供预览和打印,打印完毕之后再将该临时报表删除。

2 实现技术

2.1 Blob 数据类型及其处理

将 Word 模板保存到数据库中以及从数据库中读取 Word 模板时,属于对大二进制数据的处理,需要使

用 Blob 数据类型的变量,对 Blob 数据进行处理需要使用专门的 SQL 语句。

2.2 OLE 技术

在 PB 中连接 Word 应用程序,使用 Word 文件实现报表,有 OLE(对象链接与嵌入)控件和 OLE 对象两种方式:

(1) 使用 OLE 控件方式。这种方式对代码限制严格,对一般的用户来说不易使用。并且操作结果显示速度较慢,占用时间较长。

(2) 定义和使用 OLE 对象方式。这种方式代码使用灵活,从 VBA(Visual Basic for Applications)代码转换后即可应用,处理速度快。但必须定义一个完整的操作过程,包括定义 OLE 对象,创建 OLE 对象,连接 OLE 应用程序,打开文件,修改文件,保存文件,关闭文件,释放 OLE 对象,如果这中间出现问题,可能造成进程停留在内存中,对以后的运行速度产生影响。因此必须经过严格的测试,才能够实际应用。

本文报表的生成和打印将两种方式结合起来,打印预览时使用 OLE 控件,而对报表的生成、修改和打印使用 OLE 对象。

3 方案设计及代码

本方案主要包括两大部分:Word 报表模板的保存和 Word 临时报表的生成和处理。

3.1 Word 报表模板的保存

由于 Word 文档是用户熟悉的一种文档格式,为了防止由于用户误操作对模板造成不必要的修改或删除,应将模板保存于数据库中,实现方案如下:

(1) 建立报表模板。根据实际报表格式建立一个 Word 文档 report.doc 作为报表的模板,这里假设将其存放于应用程序所在文件夹,如果存放于其它文件夹中,只要修改下面代码中对应的文件路径即可。

(2) 将建立的 Word 模板存入数据库。首先在数据库中建立一个表 report_model,表结构如表 1 所示。本文数据库管理系统使用 SQL Server,如果是其他类型的数据库,只要将字段类型修改为对应的字段类型即可。

表 1 report_model 表结构

字段名	数据类型	长度	备注
ID	Int		自增 1
报表名称	varchar	50	
模板	image		

然后将(1)建立的报表模板存入该表的模板字段中,主要代码如下所示。处理二进制数据需要使用 Blob 数据类型的变量。因 FileRead 函数一次最多只能读出 32765 个字节,因此如果所读文件的长度大于 32765 个字节,就需要分段读出。

```
integer li_FileNum, loops, i
long flen
blob b,tot_b
string ls_path, ls_docname
ls_path = GetCurrentDirectory( )
ls_docname = ls_path + "\report.doc" //如果报表
模板存放于其他文件夹中只要修改此处文件路径即可
IF NOT FileExists(ls_docname) THEN
    MessageBox(" 错误", " 找不到报表模
板,请与系统管理员联系!")
RETURN
END IF
flen = FileLength(ls_docname)
li_FileNum = FileOpen( ls_docname, Stream-
Mode!, Read!, LockReadWrite!)
IF li_FileNum > 0 THEN
    IF flen > 32765 THEN
```

```
IF Mod( flen, 32765) = 0 THEN
    loops = flen/32765
ELSE
    loops = (flen/32765) + 1
END IF
ELSE
    loops = 1
END IF
END IF
FOR i = 1 to loops
    FileRead( li_FileNum, b)
    tot_b = tot_b + b
NEXT
FileClose( li_FileNum)
UPDATEBLOB report_model //将模板写入数据表
SET 模板 = :tot_b
WHERE id = 1; //假设要存入报表的 id 为 1,以下
同。
IF Sqlca.Sqlcode < >0 THEN
    MessageBox(" 保存报表模板失败",Sqlca.
Sqlerrtext)
RETURN
END IF
```

3.2 Word 临时报表的生成和处理

实现 Word 临时报表的生成,根据不同实际输出数据的修改、报表的保存、预览、打印和删除,实现方法和步骤如下:

(1) 将报表模板写入 Word 临时文件 temp.doc 从数据库中取出模板,写入 Word 临时文件 temp.doc,存放于应用程序所在文件夹。因 FileWrite 函数一次最多只能写入 32765 个字节,因此如果所写数据长度大于 32765 个字节,就需要分段写入。代码示例如下,其中 ls_tempfile 需定义为窗口实例变量(string ls_tempfile)。

```
blob b, blb_tmp, blb_word
integer li_FileNum, loops, i
long flen
string ls_path
//从数据库中读出模板
SELECTBLOB report_model. 模板 INTO :blb_tmp
```

```

FROM report_model
WHERE report_model.id =1;
ls_path = GetCurrentDirectory( )
ls_tempfile = ls_path + "\temp.doc"
flen = Len( blb_tmp)
li_FileNum = FileOpen( ls_tempfile, Stream-
Mode!, Write!, LockReadWrite!, Append!)
IF flen > 32765 THEN
    IF Mod( flen, 32765) = 0 THEN
        loops = flen/32765
    ELSE
        loops = ( flen/32765) + 1
    END IF
ELSE
    loops = 1
END IF
//写入临时文件 temp.doc
FOR i = 1 to loops
    blb_word = BlobMid( blb_tmp, (i - 1)
* 32765 +1, 32765)
    FileWrite( li_FileNum, blb_word)
NEXT
FileClose( li_FileNum)

```

(2) 打开临时文件 temp.doc。创建 OLE 对象, 连接 Word 应用程序, 打开生成的 Word 临时文件 temp.doc, 代码示例如下, 其中 o1 需定义为窗口实例变量 (OLEObject o1)。

```

int li_ret
o1 = CREATE OLEObject
li_ret = o1. ConnectToNewObject( " word. applica-
tion" )
IF li_ret < > 0 THEN
    DESTROY o1
    MessageBox(" OLE 错误", " OLE 无法连接! 错
误号:" + String( li_ret) )
    RETURN
END IF
o1. Documents. Open( ls_tempfile)

```

(3) 将实际输出数据写入临时文件 temp.doc。修改打开的 Word 临时文件, 为了便于填入数据, 创建

Word 模板时应尽量将要填入的数据做成表格形式, 例如下面的代码中在表格的第二行第二列处插入数据“张三”。在填入数据时, 可以用 Insertbefore() 或 Insertafter() 方法, 二者的区别是 Insertbefore() 在所选项区域前插入指定文字, 而 Insertafter() 将指定文本插入某一区域的后面。

```
String ls_xm = "张三"
```

```
o1.ActiveDocument.Tables[1].Cell[1,2].Range.
InsertBefore( ls_xm)
```

如果不能做成表格, 可以在需要写入数据的地方制作标签, 将数据写入标签处, 例如下面的代码中使用了 Word 文件中的标签“xm”, 并将“张三”插入在该标签处。

```
o1.ActiveDocument.Bookmarks.Item(" xm"). Se-
lect
```

```
o1.Selection.TypeText("张三")
```

(4) 保存临时文件 temp.doc。将修改完成后的 Word 临时文件保存, 代码示例如下:

```
o1.ActiveDocument.Save( )
```

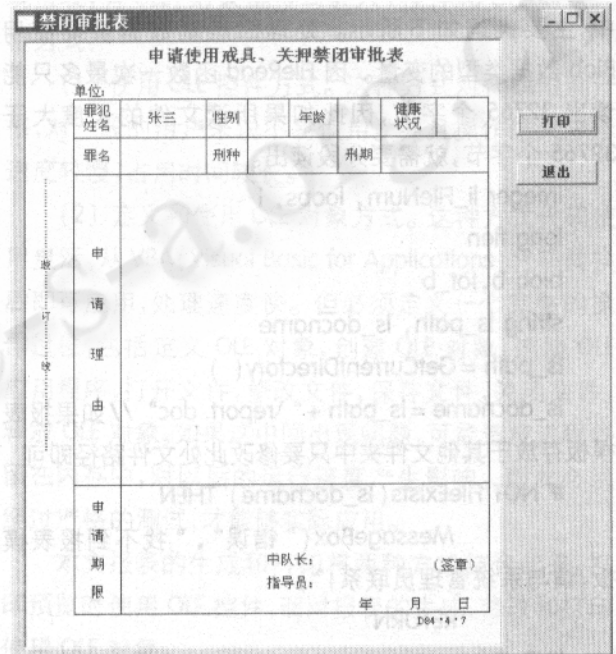


图 1 报表预览

(5) 报表预览。将临时文件插入 OLE 控件对象中, 可以进行预览。假设当前窗口中 OLE 控件名为 ole_1。代码示例如下, 图 1 为报表预览示例。

(下转第 125 页)

(上接第 122 页)

ole_1.InsertFile(ls_tempfile)

(6) 打印报表

代码示例如下:

o1.ActiveDocument.Printout()

(7) 关闭并删除临时文件 temp.doc

打印完毕,关闭临时文件,释放 OLE 对象,删除临时文件:

o1.ActiveDocument.Close

o1.Quit()

o1.DisconnectObject()

DESTROY o1

FileDelete(ls_tempfile)

3.3 代码应用说明

上面没有明确指出该代码应放在哪一对象的哪一事件或方法函数中,这些代码既可以集成在一起应用,也可以根据需要分成多个部分单独使用。

4 结论

将 PB 强大的开发功能与 Word 文档丰富的编辑功能结合起来,可以充分发挥二者的长处,在应用程序不变的情况下,用户可以对报表模板进行简单的格式修改,从而大大提高应用程序的可维护性。

参考文献

- 1 唐俊勇、李新军,“PB 平台下结合 OLE 技术实现数据导入 Excel[J]”,《计算机应用》,2005,25(4).
- 2 莫家庆、盘炜生,“PB 基于 OLE 操纵 Excel 在教材管理中的应用[J]”,《肇庆学院学报》,2004,25(2).