

基于数据报表实现 Excel 和 Word 自动化

张仁平 卜淮原 (重庆后勤工程学院研究生队 400016)

张 建 (重庆创新生物工程有限公司 400016)

摘要: 本文结合工程应用中的例子有针对性地阐述了用 Delphi 实现 Excel 自动化的主要原理、方法, 以及应用过程中的处理技巧和容易忽视或应该注意的问题, 并通过从 Excel 向 Word 复制数据简要说明实现 Word 自动化的方法, 最后对创建的对象进行释放。

关键词: Delphi Excel Word 自动化(Automation)

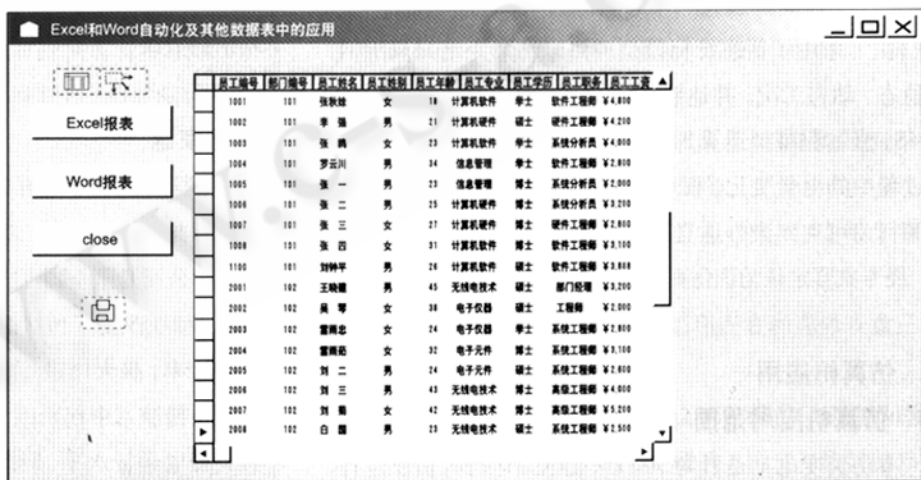
1 问题的提出

Excel 和 Word 作为 Microsoft 的最重要的软件产品之一, 是迄今为止全球最优秀、最成熟的办公自动化辅助工具, 而 Inprise 公司推出的新一代可视化开发工具 Delphi, 也以其可视化组件多、面向对象编程功能强、代码执行速度快和简单易用等特点, 正赢得了越来越多编程人员的喜爱, 作为强大的数据库开发工具, 如何满足用户对输出数据(主要为报表数据)作进一步处理呢? 用 Delphi 实现 Excel 和 Word 的自动化显然是该问题较好的解决方案。当然, 对报表数据作进一步处理的工作量不能转嫁于我们的用户, 而应该在系统开发阶段根据用户的要求进行解决, 即在 Delphi 中操纵 Excel 和 Word, 完成诸如数据报表, 报表标题的颜色、字体等修饰和数据文件的存盘(根据用户在 SaveDialog1 对象中输入的文件名)等共性问题。有关用 Delphi 操纵 Excel 和 Word, 完成数据报表这个问题, 也是许多朋友非常关心的问题。

2 Excel 和 Word 自动化及其在数据报表中的应用

为了更好地理解和把握相关原理、方法和技术, 下面采用结合具体的实现过程来进行说明。

2.1 建立窗体文件(如下图所示)。



2.2 单元文件的主要控件代码及其说明

(1) “Excel 报表”按钮主要完成 Excel 自动化和数据报表输出两大功能, 下面分五个部分进行说明。

① 在 uses 语句中加入 ComObj 单元。在 COMObj 单元中, 包括了操作 OLE 自动化对象的函数集和命令集。Delphi 通过 COMObj 例程的 VarDispInvoke 方法, 在 IDispatch 接口调用其 GetIDsOfNames 函数, 获得调用方法 CreateOleObject 的 ID 代码, 然后根据返回的 ID 代码调用 IDispatch 的 Invoke 方法, 最后启动 Excel, 并返回调用方法的名称, 存放于变量 MyExcelObject 中。

② 理解并创建 Excel 的自动化对象。在 Excel 的自动化对象中, 其层次可分为 Application → Workbooks → Sheets、WorkSheets、Charts-----。用 CreateOleObject 方法调用了大量的 OLE 函数, 返回一个包装了 IDispatch 接口的变量, 即为 Excel 的 Application 对象, 并使其可视化,

其代码如下:

```
var
  Workbook: Variant;// 工作簿变量, 声明 Automation 对象
  Worksheet: Variant;// 工作表变量
  RowRange: Variant;// 工作表行变量
  ColumnRange: Variant;// 工作表列变量
  Range: Variant;// 工作表选择操作区域变量
  i,j: Integer;// I 代表行数或记录数, J 代表列或字段数
  M:Array of integer;// 存放数据表的字段宽度, 用于控制
  工作表的列宽
```

```
begin
  MyExcelObject := CreateOleObject('Excel.Application');//
  该变量声明为全局变量
  MyExcelObject.Application.Workbooks.Add;// 添加工作簿
  MyExcelObject.Application.Visible := True;// 使
  MyExcelObject 对象可视
  Workbook := MyExcelObject.Application.Workbooks [1];
  Worksheet := Workbook.WorkSheets.Item [1];
```

③ 创建电子表格上面在完成创建了一个 Excel 的 Application 对象的基础上, 又添加一个新工作簿, 下面代码实现了将数据表插入电子表格的功能。

```
i := 0;
Table1.Active:=false;
Table1.DatabaseName:='C:\Excel';
Table1.TableName:='员工表.DB';
Table1.Active:=true;
while not Table1.Eof do// 循环直到数据表的最后一条记录
begin
  Inc(i);// 记录计数器加 1
  for j := 0 to Table1.FieldCount - 1 do
    Worksheet.Cells [i, j+1] := Table1.Fields [j]
.AsString;// 将数据表中的数据插入电子表格
  Table1.Next;
end;
```

从上面代码所实现的功能不难看出, 只要在 Delphi 中能实现的动态报表, 就能通过调用 Excel 自动化的方法来实现, 并且更灵活、方便。如果我们想要改变电子表格中的某一列的字体、宽度和颜色等属性, 可用以下函数, 当然也可以在 Excel 环境中完成类似的功能。

```
ColumnRange := MyExcelObject.Workbooks [1] .
  Worksheets ['Data1'] .Columns;
```

```
ColumnRange.Columns [1] .ColumnWidth := 5;
ColumnRange.Columns.Item [1] .Font.Bold := True;
ColumnRange.Columns [1] .Font.Color := clRed ;
end;
```

如果要控制电子表格所有列, 则首先要获得数据宽度, 下面采用根据数据字段的 DataType 属性, 将字段分为字符型和非字符型两类, 对字符型的字段用其 DataSize 属性获得数据宽度, 对非字符型的字段用其最大宽度作为电子表格对应列的宽度, 最后将获得的数据宽度放在一维数组 M 中。

```
SetLength(m,Table1.FieldCount);// 动态设置数组
for j := 0 to Table1.FieldCount - 1 do
  if Table1.Fields [j] .DataType=ftString then// 判断字段
  属性
    M [j]:=Table1.Fields [j] .DataSize;// 数据表的字段
    数起始数为 0
  else
    begin
      M [j]:=0;
      for i:=0 to Table1.RecordCount-1 do//用冒泡法求出该
      字段数据的最大宽度
        begin
          if i=0 then// 开始时将数据表指向第一条记录
            Table1.First;
            if M [j] <Length(Trim(Table1.Fields [j]
.AsString)) THEN // 将非字符型数据(实型、整型等)转
            换成字符型数据, 并通过 Length 函数和 Trim 函数求得其
            有效长度, 然后进行比较。
              M [j]:=Length(Trim(Table1.Fields [j]
.AsString));
            Table1.Next;
          end;
        end;
      ColumnRange.Columns [j+1] .ColumnWidth :=M
      [j];// 注意工作表的列起始数为 1, 而数据表的字段数
      起始数为 0, 故应为 j+1, 注意其区别。
    end;
```

以上所谈对列的操作完全适用于对行的操作, 虽然说对行的操作一般用的很少, 但有时还是需要这方面的修饰(比如常用的数据报表的标题等)。其源代码如下:

```
RowRange := MyExcelObject.Workbooks [1] .
```

Worksheets ['Data1'] .Rows;

RowRange.Rows [1] .RowHeight := 20;

RowRange.Rows [1] .Font.Color := clBlue;

④ 为电子表格画边框通过以上工作创建完成的电子表格还没有边框,或者说边框的宽度为0,可通过 Excel 的菜单中选择“文件\打印预览”进行预览。一般来说要加边框,需要选定某一区域,在 Excel 环境中操作也是如此,然后设置单元格的 LineStyle 属性。

```
Range:=MyExcelObject.Workbooks [1] .Worksheets
['Data1'] .Range ['A1:I25'];
```

```
Range.Borders.LineStyle:=1;
```

例中采用了选取工作表 Data1 的一部分数据的方法,若你在事先不能确定范围,则可全选,在打印预览中会发现在数据表以外的单元格并不会出现,这自然是我们最期待的一种情形,具有很强的灵活性。其实现的方法更简单,仅有一句 RowRange.Borders.LineStyle:=1;或 ColumnRange.Borders.LineStyle:=1;。

⑤ 在信息系统数据报表中的应用“学以致用”是我们学习的出发点和归宿点,在信息系统报表中,除了数据表外,至少还有报表的标题。标题作为数据报表的一个组成部分,其插入方法除了上面提到的 Worksheet.Cells 方法外,还可用以下方法,用该方法的优点便于对标题进行修饰处理,如颜色等。

```
Range:=MyExcelObject.Workbooks [1] .Worksheets
['Data1'] .Range ['E1'];
```

```
Range.Formula:=' 公司员工基本情况表';
```

```
Range.Font.Color := clRed;
```

另外,由于标题一般不需要边框,因此,用以上两种画电子表格边框的方法都不适合。第一种方法不够灵活,第二种方法又不符合要求。当然,也没有必要花太多精力去另求它路,只是立足于第二种方法,并对其作稍加修改即可满足以上要求。其做法是逐行画边框(当然包括边框和内框,相当于在 Excel 环境中操作格式\单元格菜单,再点击边框,然后选中外边框和内部两项操作),比如只对第二行画边框,其方法为: RowRange.Rows [2] .Borders.LineStyle:=1;其余的行就“同理可得”了。

(2) “Word 报表”按钮除完成“Excel 报表”按钮的功能外,还完成了 Word 自动化和将报表数据从 Excel 向 Word 复制两项功能。其实,Word 的自动化的原理、方法与 Excel 自动化很相似,采用此方法的原因主要是在 Word 中操作数据不如 Excel 方便。若不想 Excel 出现,只须将

MyExcelObject 对象的 Visible 属性值修改为 false 即可。其实现的源程序略。

(3) 调用对象的释放。(源程序略)

3 结束语

本文介绍了有关用 Delphi 实现 Excel 和 Word 自动化及在数据报表中的应用。用 Delphi 实现 Excel 和 Word 自动化这个技术领域,可以形象地把她比作“一块很肥沃的处女地”。说她是“处女地”,是因为操纵 Excel 和 Word 自动化,并满足不同用户的各种报表要求方面的应用还很少见;说她很肥沃是因为她具有很广的应用价值,从理论上讲,使用 Delphi 可以完全操纵 Excel 和 Word 的所有功能,当然是基于你对 Excel 和 Word 对象的内部属性和方法的理解程度上。如果想查看 Excel 和 Word 中使用的常数,请参阅---\ Borland \ Delphi4 \ Imports 目录下的 Excel_TLB.pas 和 Word_TLB.pas 文件。如果该目录不存在此文件,可先在---\ Microsoft Office \ Office 目录中找到类型库文件 Excel9.olb 和 Msword9.olb,然后在 Delphi 的菜单中选择 Project \ Import Type Library, 自动创建 Delphi 的类型库译本,存放于---\ Borland \ Delphi4 \ Imports 目录下。用 Delphi 实现 Excel 和 Word 自动化这个问题涉及的范围很广,需要做的工作也很多,如果有对此问题感兴趣的朋友,可以来信共同讨论。通信地址:重庆长江二路 174 号后勤工程学院研究生队(400016); E-mail: Renping_Zhang@263.net。■

参考文献

- 1 郑城荣等, *Delphi 运行时间库 RTL 和组件库 VCL 技术参考*, 人民邮电出版社, 1999
- 2 Charlie Calvert *Delphi4 编程技术内幕*, 机械工业出版社, 1999
- 3 Steve Teixeira & Xavier Pacheco, *Delphi4 开发大全*, 人民邮电出版社, 1999