要在应用程序中控制Excel的运行,首先必须在编制自动化客户程序时包含
Comobj.hpp
#include "Comobj.hpp"
C++ Builder把Excel自动化对象的功能包装在下面的四个Ole Object Class函数中,应用人员可以很方便地进行调用。
设置对象属性:void OlePropertySet(属性名,参数……);
获得对象属性:Variant OlePropertyGet(属性名,参数……);
调用对象方法:1) Variant OleFunction(函数名,参数……);
2) void OleProcedure(过程名,参数……);
在程序中可以用宏定义来节省时间:
#define PG OlePropertyGet
#define PS OlePropertySet
#define FN OleFunction
#define PR OleProcedure
举例:
ExcelApp.OlePropertyGet("workbooks").OleFunction("Add");
可写为
ExcelApp.PG("workbooks").FN("Add");
C++ Builder中使用OLE控制Excel2000,必须掌握Excel2000的自动化对象及Microsoft Word Visual Basic帮助文件中的关于Excel的对象、方法和属性。对象是一个Excel元素,属性是对象的一个特性或操作的一个方面,方法是对象可以进行的动作。
首先定义以下几个变量:
Variant ExcelApp,Workbook1,Sheet1,Range1;
1、Excel中常用的对象是:Application,Workbooks,Worksheets等。
★创建应用对象★
Variant ExcelApp;
ExcelApp = Variant::CreateObject ("Excel.Application");
或者
ExcelApp = CreateOleObject ("Excel.Application");
Variant WorkBook1;
WorkBook1 = ExcelApp.PG("ActiveWorkBook");
★创建工作表对象★
Variant Sheet1;
Sheet1 = WorkBook1.PG("ActiveSheet");
★创建区域对象★
Variant Range;
Range = Sheet1.PG("Range","A1:A10");
或者使用
Excel.Exec(PropertyGet("Range")<<"A1:C1").Exec(Procedure ("Select"));
2、常用的属性操作:
★使Excel程序不可见★
ExcelApp.PS("Visible", (Variant)false);
★新建EXCEL文件★
◎ 新建系统模板的工作簿
ExcelApp.PG("workbooks").FN("Add") //默认工作簿
ExcelApp.PG("workbooks").FN("Add", 1) //单工作表

```
ExcelApp.PG("workbooks").FN("Add", 2) //图表
ExcelApp.PG("workbooks").FN("Add", 3) //宏表
ExcelApp.PG("workbooks").FN("Add", 4) //国际通用宏表
ExcelApp.PG("workbooks").FN("Add", 5) //与默认的相同
ExcelApp.PG("workbooks").FN("Add", 6) //工作簿且只有一个表
或者使用ExcelApp的Exec方法
Excel.Exec(PropertyGet("Workbooks")).Exec(Procedure("Add"));
◎ 新建自己创建的模板的工作簿
ExcelApp.PG("workbooks").FN("Add", "C:\\Temp\\result.xlt");
★打开工作簿★
ExcelApp.PG("workbooks").FN("open", "路径名.xls")
★保存工作簿★
WorkBook1.FN("Save"); //保存工作簿
WorkBook1.FN("SaveAs", "文件名");//工作簿保存为,路径注意用"\\"
★退出EXCEL★
ExcelApp.FN ("Quit");
ExcelApp = Unassigned;
或者
ExcelApp.Exec(Procedure("Quit"));
★操作工作表★
◎ 选择选择工作表中第一个工作表
Workbook1.PG("Sheets", 1).PR("Select");
Sheet1 = Workbook1.PG("ActiveSheet");
◎ 重命名工作表
Sheet1.PS("Name", "Sheet的新名字");
◎ 当前工作簿中的工作表总数
int nSheetCount=Workbook1.PG("Sheets").PG("Count");
★操作行和列★
◎ 获取当前工作表中有多少行和多少列：
Sheet1.PG("UsedRange").PG("Columns").PG("Count"); //列数
Sheet1.PG("UsedRange").PG("Rows").PG("Count"); //行数
◎ 设置列宽
ExcelApp.PG("Columns", 1).PS("ColumnWidth", 22);
或者
Range = ExcelApp.PG("Cells", 1, 3);
Range.PS("ColumnWidth", 22);
◎ 设置行高
ExcelApp.PG("Rows", 2).PS("RowHeight", 25);
或者
Range = ExcelApp.PG("Cells", 2, 1);
Range.PS("RowHeight", 25);
◎ 在工作表最前面插入一行
Sheet1.PG("Rows", 1).PR("Insert");
◎ 删除一行
ExcelApp.PG("Rows", 2).PR("Delete"); //将第2行删除
// 本文作者：ccrun ，如转载请保证本文档的完整性，并注明出处。
// 欢迎光临 C++ Builder 研究 www.ccrun.com
// 摘自:http://www.ccrun.com/doc/go.asp?id=529
```

★操作单元格★
◎ 设置单元格字体
```
Sheet1.PG("Cells", 1, 1).PG("Font").PS("Name", "隶书"); //字体
Sheet1.PG("Cells", 2, 3).PG("Font").PS("size", 28); //大小
```
◎ 设置所选区域字体
```
Range.PG("Cells").PG("Font").PS("Size", 28);
Range.PG("Cells").PG("Font").PS("Color", RGB(0, 0, 255));
```
其中参数的设置:
```
Font Name：“隶书” //字体名称
Size：12 //字体大小
Color：RGB(*,*,*) //颜色
Underline：true/false //下划线
Italic: true/false //斜体
```
◎ 设置单元格格式为小数百分比
```
Sheet1.PG("Cells", 1, 1).PS("NumberFormatLocal", "0.00%");
```
◎ 设定单元格的垂直对齐方式
```
Range = ExcelApp.PG("Cells", 3, 4);
// 1=靠上 2=居中 3=靠下对齐 4=两端对齐 5=分散对齐
Range.PS("VerticalAlignment", 2);
```
◎ 设定单元格的文本为自动换行
```
Range = ExcelApp.PG("Cells", 3, 4);
Range.PS("WrapText", true);
```
★单元格的合并★
◎ Range = Sheet1.PG("Range", "A1:A2"); //A1和A2单元格合并
```
String strRange = "A" + IntToStr(j) + ":" + "C" + IntToStr(j); //比
如：A1:C5
Range1=Sheet1.PG("Range", strRange.c_str()); //可以用变量控制单元
格合并
Range1.FN("Merge", false);
```
★读写单元格★
◎ 指定单元格赋值
```
String strValue = "abcdefg";
Sheet1.PG("Cells", 3, 6).PS("Value", strValue.c_str());
Sheet1.PG("Cells", j, 1).PS("Value", "总记录:" + String(j-6));
```
或者使用
```
Excel.Exec(PropertyGet("Cells")<<1<<3).Exec(PropertySet("Value")
<<15);
```
◎ 所选区域单元格赋值
```
Range.PG("Cells").PS("Value", 10);
```
◎ 所选区域行赋值
```
Range.PG("Rows",1).PS("Value", 1234);
```
◎ 工作表列赋值
```
Sheet1.PG("Columns",1).PS("Value", 1234);
```
◎ 读取取值语句:
```
String strValue = Sheet1.PG("Cells", 3, 5).PG("Value");
```
★窗口属性★
◎ 显示属性
```
ExcelApp.PS("Windowstate", 3); //最大化显示
```

1---------xlNormal //正常显示
2---------xlMinimized //最小化显示
3---------xlMaximized //最大化显示
◎ 状态栏属性
ExcelApp.PS("StatusBar", "您好,请您稍等。正在查询！");
ExcelApp.PS("StatusBar", false); //还原成默认值
◎ 标题属性:
ExcelApp.PS("Caption", "查询系统");
3、操作图表
★添加图表
```
Variant Chart;
Chart = ExcelApp.Exec(PropertyGet("Charts")).Exec(Function("Add"));
ExcelApp.Exec(PropertySet("Visible") << true);
Chart.Exec(PropertySet("Type") << -4100);
```
★滚动图表
```
for(int nRotate=5; nRotate <= 180; nRotate += 5)
{
Chart.Exec(PropertySet("Rotation") << nRotate);
}
for (int nRotate = 175; nRotate >= 0; nRotate -= 5)
{
Chart.Exec(PropertySet("Rotation") << nRotate);
}
```
另外，为保证程序能正常运行，需要在程序中判断目标机器是否安装了Office;
```
try
{
ExcelApp = Variant::CreateObject ("Excel.Application");
}
catch(...)
{
ShowMessage("运行Excel出错,请确认安装了Office");
return;
}
#include "comobj.
//---------------------------------------------------------------------------
// 对指定Excel文件中的指定列进行排序
// strExcelFileName：excel文件名
// nCol： 指定的列号
// nSortStyle：1:升序，2:降序
void SortExcelColumn(String strExcelFileName, int nCol, int nSortStyle)
{
Variant vExcelApp, vWorkbook, vRange;
vExcelApp = Variant::CreateObject("Excel.Application");
vExcelApp.OlePropertySet("Visible", false);
vExcelApp.OlePropertyGet("WorkBooks").OleProcedure("Open",
strExcelFileName.c_str());
vWorkbook = vExcelApp.OlePropertyGet("ActiveWorkbook");
vExcelApp.OlePropertyGet("Columns", nCol).OleProcedure("Select");
vExcelApp.OlePropertyGet("ActiveSheet").OlePropertyGet("Cells", 1,
```

```
nCol).OleProcedure("Select");
vRange = vExcelApp.OlePropertyGet("Selection");
vRange.Exec(Function("Sort")<<vExcelApp.OlePropertyGet("Selection")
<<nSortStyle);
vWorkbook.OleProcedure("Save");
vWorkbook.OleProcedure("Close");
vExcelApp.OleFunction("Quit");
vWorkbook = Unassigned;
vExcelApp = Unassigned;
ShowMessage("ok");
}
void __fastcall TForm1::Button1Click(TObject *Sender)
{
// 对C:\123\123.xls文件中第一个Sheet的第四列进行升序排序
SortExcelColumn("C:\\123\\123.xls", 4, 1);
}
```