



[\[公告\]](#)[一去、二三里](#)[个人中心](#)[发博文](#)[消息](#)



一去、二三里的博客

<http://blog.sina.com.cn/liang19890820>

首页 | 博文目录 | 图片 | 关于我

[发博文](#)[页面设置](#)[个人中心](#)

个人资料 [\[管理\]](#)



一去、二三里

[Qing](#)[微博](#)

博客等级: **18**
博客积分: **78**
博客访问: **407,827**
关注人气: **378**
获赠金笔: **12**
赠出金笔: **0**
荣誉徽章: 

相关博文

- php单引号与双引号用法引号嵌套去零度_PHP
- QT中数据源数据导出Excel的问题梧桐下的约定
- 在Qt中使用ActiveX控件（转）梧桐下的约定
- SVN地址改变后，如何将原先的地址悠然渐晨
- 出离心的标准是什么？莲花妙音
- S Q L操作符流浪的孤云
- qt解析xml文件小唐
- 南海-东海.海鲜游记「顺德.南记渔大食堂公社生产队长
- Qt用DOM解析XML语言的一个例子忧伤岁月
- 对DataTable或objectlist使用LIN远方的大鼓

更多>>

正文 字体大小: 大 中 小

Qt之操作Excel (2013-12-06 13:57:20) [\[编辑\]](#)[\[删除\]](#)

标签: qt qaxobject qt操作excel qtvba excelvba 分类: Qt [+ 转载](#)

Visual Basic for Applications (VBA) 是一种Visual Basic的一种宏语言，主要能用来扩展Windows的应用程式功能，特别是Microsoft Office软件。也可说是一种应用程式视觉化的Basic Script。1994年发行的Excel 5.0版本中，即具备了VBA的宏功能。

在VBA的参考手册中就可以看到具体函数、属性的用法，Qt操作Excel主要通过 QAxObject + Excel VBA来实现！

关于Qt对Excel的操作，网上的资料挺多的，但大多数都是比较基础的，关于插入工作表（至最后一行）、删除工作表、合并/拆分单元格、设置单元格背景色、设置单元格边框色、设置单元格字体（类型、大小、加粗、斜体、下划线、颜色等）、以及设置单元格对齐方式等用法都没有怎么提到，今天就总结一下有关Qt对Excel的操作。

Qt操作Excel，无论后缀是xls还是xlsx都可以。

如下，是我下载的一个Excel VBA参考手册，内容不算太全！



The screenshot shows the 'Microsoft Excel Visual Basic 参考' (Microsoft Excel Visual Basic Reference) window. It displays a tree view of the Excel object model. The 'Application' object is the root, containing various sub-objects like 'Workbooks', 'Worksheets', 'Charts', 'DocumentProperties', 'VBProject', 'CustomViews', 'CommandBars', 'HTMLProject', 'PivotCaches', 'Styles', 'Borders', 'Font', 'Interior', 'Windows', 'Panels', 'AddIns', 'Answer', 'AutoCorrect', 'Assistant', 'AutoRecover', 'CellFormat', 'COMAddIns', 'Debug', 'Dialogs', 'CommandBars', 'ErrorCheckingOptions', 'LanguageSettings', 'Names', 'Windows', and 'Panels'. Each sub-object is represented by a yellow box with its name and a brief description.

Excel读取

为了便于测试，假设已存在一个excel文件，操作内容已经被红色标记出来。如下所示：

- 2550mAh电
- 日本雾霾之战：民众与政府的博弈
- 南非猎豹冒死捕杀豪猪被扎满嘴刺
- 别光顾着把枪口对准柴静
- 马来西亚男子与眼镜王蛇接吻(图)
- 【早评】降息利好将会在后市中逐
- 哪些人能从柴静的“穹顶之下”赚
- “深圳机场撞人事件”不是一个人
- 政采剔除国外品牌：早该说了！
- 【DIY】做一枚宫灯迎元宵



[查看更多>>](#)

谁看过这篇博文

	碧落	3月1日
	杭州@聚散...	3月1日
	yuxuange	2月28日
	小灰狼	2月27日
	自然麻雀	2月26日
	独自等待	2月20日
	baggiokang	2月13日
	sunnyhuil...	2月11日
	斌斌first	2月11日
	残情磊	2月11日
	小白	2月11日
	Writead	2月11日

主要读取内容：

- 标题
- 工作表数目
- 工作表名称
- 起始行
- 起始列
- 行数
- 列数
- 单元格内容

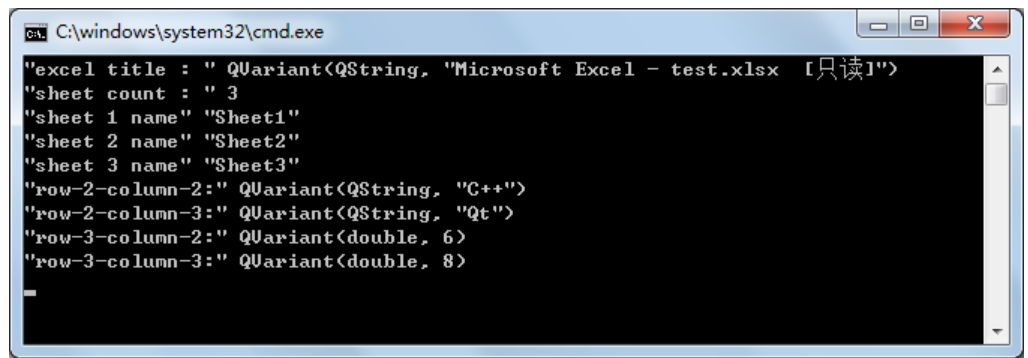
代码如下：

```
QAxObject excel("Excel.Application");
excel.setProperty("Visible", true);
QAxObject *work_books = excel.querySubObject("WorkBooks");
work_books->dynamicCall("Open (const QString&)", QString("E:/test.xlsx"));
QVariant title_value = excel.property("Caption"); //获取标题
QDebug()<<QString("excel title : ")<<title_value;
QAxObject *work_book = excel.querySubObject("ActiveWorkBook");
QAxObject *work_sheets = work_book->querySubObject("Sheets"); //Sheets也可换用Worksheets

int sheet_count = work_sheets->property("Count").toInt(); //获取工作表数目
QDebug()<<QString("sheet count : ")<<sheet_count;
for(int i=1; i<=sheet_count; i++)
{
    QAxObject *work_sheet = work_book->querySubObject("Sheets(int)", i); //Sheets(int)也可换用
    Worksheets(int)
    QString work_sheet_name = work_sheet->property("Name").toString(); //获取工作表名称
    QString message = QString("sheet ") + QString::number(i, 10) + QString(" name");
    qDebug()<<message<<work_sheet_name;
}

if(sheet_count > 0)
{
    QAxObject *work_sheet = work_book->querySubObject("Sheets(int)", 1);
    QAxObject *used_range = work_sheet->querySubObject("UsedRange");
    QAxObject *rows = used_range->querySubObject("Rows");
    QAxObject *columns = used_range->querySubObject("Columns");
    int row_start = used_range->property("Row").toInt(); //获取起始行
    int column_start = used_range->property("Column").toInt(); //获取起始列
    int row_count = rows->property("Count").toInt(); //获取行数
    int column_count = columns->property("Count").toInt(); //获取列数
    for(int i=row_start; i
    {
        for(int j=column_start; j
        {
            QAxObject *cell = work_sheet->querySubObject("Cells(int,int)", i, j);
            QVariant cell_value = cell->property("Value"); //获取单元格内容
            QString message = QString("row-")+QString::number(i, 10)+QString("-column-")+QString::number(j,
            10)+QString(":");
            qDebug()<<message<<cell_value;
        }
    }
}
```

效果如下：



Excel增、删、改

主要操作：

- 设置标题
- 插入工作表（至最后一行）
- 设置工作表名称
- 删除工作表
- 设置单元格内容
- 设置单元格字体（类型、大小、加粗、斜体、下划线、颜色等）
- 设置单元格对齐方式
- 设置单元格高度、宽度
- 设置单元格背景色、边框色
- 合并/拆分单元格
- 清空单元格

代码如下：

```
QAxObject excel("Excel.Application");
excel.setProperty("Visible", true);
QAxObject *work_books = excel.querySubObject("WorkBooks");
work_books->dynamicCall("Open(const QString&", "E:\\test.xlsx");
excel.setProperty("Caption", "Qt Excel");
QAxObject *work_book = excel.querySubObject("ActiveWorkBook");
QAxObject *work_sheets = work_book->querySubObject("Sheets"); //Sheets也可换用Worksheets

//删除工作表（删除第一个）
QAxObject *first_sheet = work_sheets->querySubObject("Item(int)", 1);
first_sheet->dynamicCall("delete");

//插入工作表（插入至最后一行）
int sheet_count = work_sheets->property("Count").toInt(); //获取工作表数目
QAxObject *last_sheet = work_sheets->querySubObject("Item(int)", sheet_count);
QAxObject *work_sheet = work_sheets->querySubObject("Add(QVariant)", last_sheet->asVariant());
last_sheet->dynamicCall("Move(QVariant)", work_sheet->asVariant());

work_sheet->setProperty("Name", "Qt Sheet"); //设置工作表名称

//操作单元格（第2行第2列）
QAxObject *cell = work_sheet->querySubObject("Cells(int,int)", 2, 2);
cell->setProperty("Value", "Java C++ C# PHP Perl Python Delphi Ruby"); //设置单元格值
cell->setProperty("RowHeight", 50); //设置单元格行高
cell->setProperty("ColumnWidth", 30); //设置单元格列宽
cell->setProperty("HorizontalAlignment", -4108); //左对齐 (xlLeft)：-4131 居中
(xlCenter)：-4108 右对齐 (xlRight)：-4152
cell->setProperty("VerticalAlignment", -4108); //上对齐 (xlTop) -4160 居中 (xlCenter)：-4108 下
对齐 (xlBottom)：-4107
cell->setProperty("WrapText", true); //内容过多，自动换行
//cell->dynamicCall("ClearContents()"); //清空单元格内容

QAxObject* interior = cell->querySubObject("Interior");
interior->setProperty("Color", QColor(0, 255, 0)); //设置单元格背景色（绿色）

QAxObject* border = cell->querySubObject("Borders");
border->setProperty("Color", QColor(0, 0, 255)); //设置单元格边框色（蓝色）

QAxObject *font = cell->querySubObject("Font"); //获取单元格字体
```

```
font->setProperty("Name", QStringLiteral("华文彩云")); //设置单元格字体
font->setProperty("Bold", true); //设置单元格字体加粗
font->setProperty("Size", 20); //设置单元格字体大小
font->setProperty("Italic", true); //设置单元格字体斜体
font->setProperty("Underline", 2); //设置单元格下划线
font->setProperty("Color", QColor(255, 0, 0)); //设置单元格字体颜色（红色）

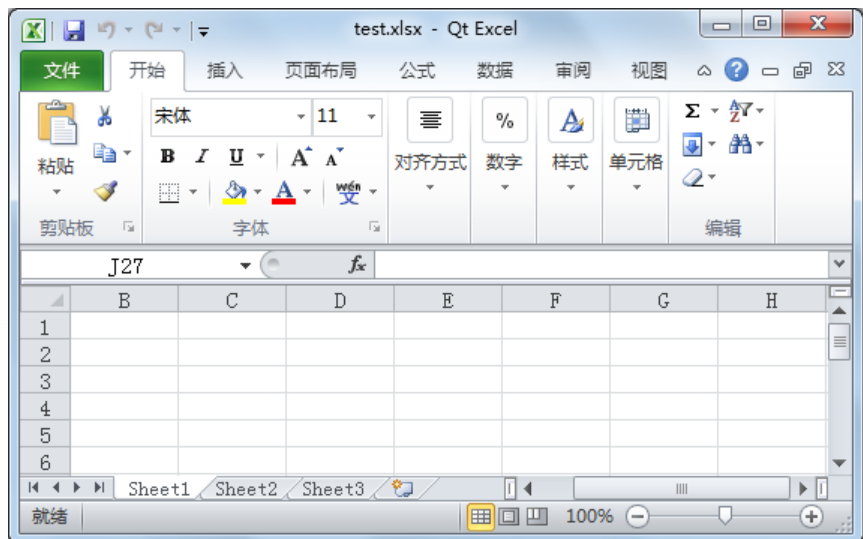
//设置单元格内容，并合并单元格（第5行第3列-第8行第5列）
QAxObject *cell_5_6 = work_sheet->querySubObject("Cells(int,int)", 5, 3);
cell_5_6->setProperty("Value", "Java"); //设置单元格值
QAxObject *cell_8_5 = work_sheet->querySubObject("Cells(int,int)", 8, 5);
cell_8_5->setProperty("Value", "C++");

QString merge_cell;
merge_cell.append(QChar(3 - 1 + 'A')); //初始列
merge_cell.append(QString::number(5)); //初始行
merge_cell.append(":");
merge_cell.append(QChar(5 - 1 + 'A')); //终止列
merge_cell.append(QString::number(8)); //终止行
QAxObject *merge_range = work_sheet->querySubObject("Range(const QString&)", merge_cell);
merge_range->setProperty("HorizontalAlignment", -4108);
merge_range->setProperty("VerticalAlignment", -4108);
merge_range->setProperty("WrapText", true);
merge_range->setProperty("MergeCells", true); //合并单元格
//merge_range->setProperty("MergeCells", false); //拆分单元格

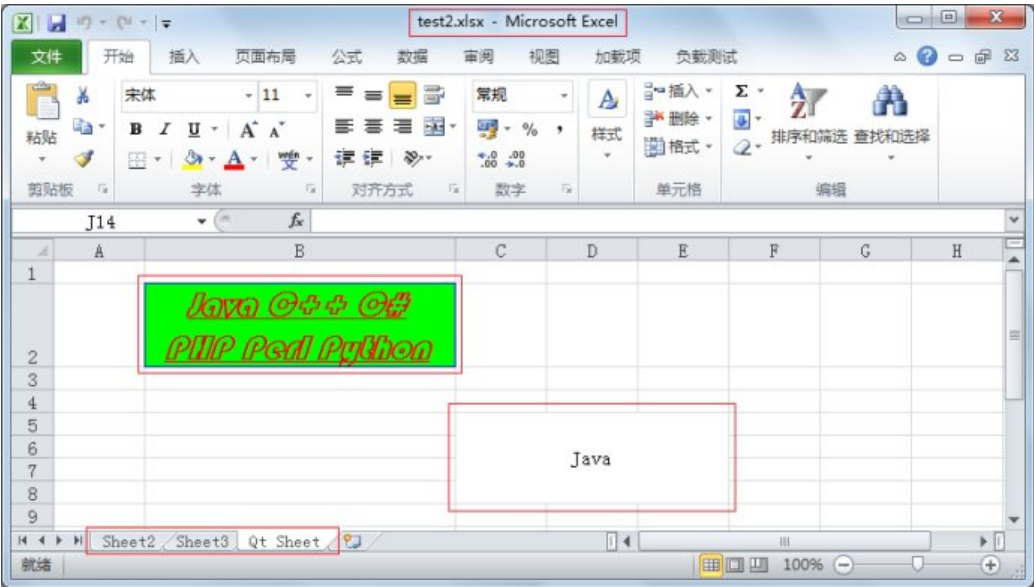
//work_book->dynamicCall("Save()"); //保存文件（为了对比test与下面的test2文件，这里不做保存操作）
work_book->dynamicCall("SaveAs(const QString&)", "E:\\test2.xlsx"); //另存为另一个文件
work_book->dynamicCall("Close(Boolean)", false); //关闭文件
excel.dynamicCall("Quit(void)"); //退出
```

效果如下：

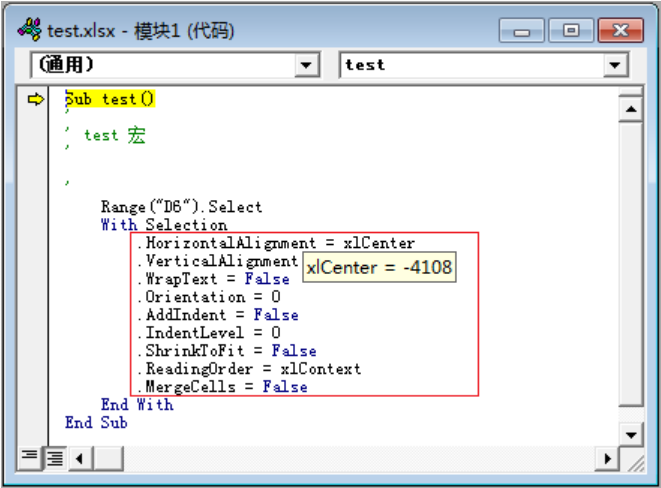
操作前：



操作后：



到这里很多人也许都在纳闷，单元格的宏怎么获取的？比如对齐方式（居中对齐），为什么值是-4108，而不是其他值呢？当然那不是我随便写的，自己可以录制宏，然后跟踪。
看下图：



如上所讲，已经基本可以满足常用的操作，如有更多专业需求，请参考Excel VBA...
• Excel Object Model Reference.

注：
技术在于交流、沟通，转载请注明出处并保持作品的完整性。
作者：`☆奋斗ing♥孩子` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90101gv2p.html。

5

喜欢

分享:          

阅读(4825) | 评论(5) | 收藏(6) | 已有7人转载 | 喜欢 | 打印

已投稿到:  排行榜

前一篇: Qt之日志输出窗口
后一篇: Qt之生成pdf

评论

重要提示: 警惕虚假中奖信息

[发评论]



2013-12-6 15:33

回复 (1)

miaozhengwei1986



2014-1-11 21:44

回复 (1)

leileileilei2009

我按照楼主的方法，能获取工作表数目、起始行列、行数、列数等信息，可是就是无法获取单元格数据，返回的单元格数据总是为空（QVariant(,)）。也就是以下操作没有获取到数据：

```
QAxObject *cell = work_sheet->querySubObject("Cells(int,int)", i, j);
QVariant cell_value= cell->property(" 查看更多▼")
```

2014-11-26 21:20

回复 (0)

发评论

一去、二三里：

☐ 分享到微博

☐ 匿名评论

验证码： 请点击后输入验证码 收听验证码

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇

Qt之日志输出窗口

后一篇 >

Qt之生成pdf