

使用 VB.NET 对 Excel 批量操作在工程资料上的应用

刘利

南京南瑞继保电气有限公司

DOI:10.32629/pe.v3i6.18052

[摘要] 大中型机电安装项目施工与调试周期常超一年,期间的工程资料制作中需编制大量表格资料,且存在较多重复性工作。人工处理批量Excel表格时,查找、校验和纠正常见问题不仅耗时久,还易出现疏漏。采用VB.NET对Excel文件进行批处理,能显著提升资料制作效率,提高正确率并避免疏漏。

[关键词] VB.NET Excel; 批处理; 工程资料

中图分类号: C67 文献标识码: A

Application of batch operation on engineering data using VB.NET in Excel

Li Liu

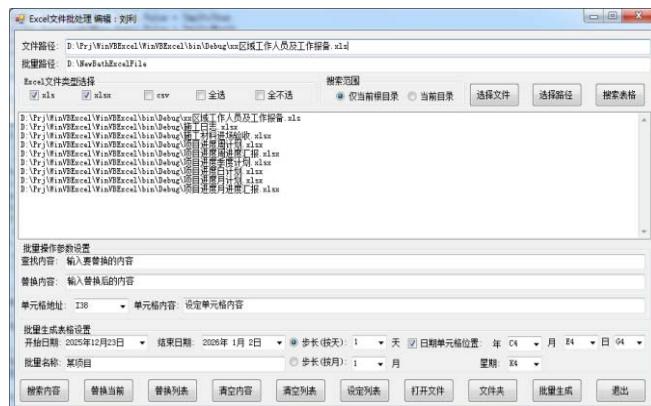
Nanjing Nanrui, Jiangsu Nanjing 211102, China

[Abstract] The construction and commissioning cycle of large and medium-sized electromechanical installation projects often exceeds one year, during which a large amount of table data needs to be prepared, and there are many repetitive tasks. When manually processing batch Excel spreadsheets, searching, verifying, and correcting common problems not only takes a long time, but also easily leads to omissions. Using VB.net for batch processing of Excel files can significantly improve data production efficiency, increase accuracy, and avoid omissions.

[Key words] VB.NET; Excel Batch Engineering data

引言

大中型机电安装项目施工与调试周期常超一年,施工过程中工程资料也很重要的一个环节,工程涉及到很多的表格资料。参与了几个项目的资料制作后,了解了资料制作的流程,很多工作都需要重复和反复修改。有些资料是一段时间连续的资料,如每日施工日志、明日施工计划、进出人员报备等;有些是隔一段时间的资料,如周报、月报、半年报等。部分是日期不确定的资料,如材料的进场报验、分部分项和隐蔽工程等的验收。软件的界面如下:



如某一批表格的施工名称或信息错误,需要批量修改。人工修改完表格后,经常还会遗漏未改到的地方。如果再逐个打开表格检查,会浪费时间,效率低下,且容易遗漏部分表格。

根据前面项目出现的常见问题,很多都可以通过批量处理来解决。使用VB.NET进行表格批量处理,不仅能提高效率,还能对表格内容进行检查和纠正,显著减少这些问题的发生。

1 搜索Excel文件

1.1 搜索应用路径下Excel文件

Excel工具软件比较小,可以方便地进行复制,可以将软件复制到对应的模板文件夹下。打开工具时,能够自动搜索工具所在文件目录下的Excel表格,包括.xls、xlsx和csv文件,将搜索到的文件显示在列表框中,方便用户选择,同时省去了通过浏览选择文件夹的步骤。

用Dim StrStAPath As String=System.Windows.Forms.Application.StartupPath即可获取应用所在的路径。在窗体加载时调用搜索程序,根据文件类型,搜索到的文件直接加载到列表框中。定义一个字符串数组Dim StrExcelFiles ()As String,并将数组内容赋值给当前目录下的所有Excel文件,StrExcelFiles =Directory.GetFiles(StrStAPath, "*.xls"),使用For Each循环将数组中的数据添加到列表框中。语句如下:

```
For Each TmpStrF In TmpStrExcelFiles
    LstBEPPathExcelFile.Items.Add(TmpStrF)
Next
```

1.2 Excel文件和路径获取

文件路径默认是当前的应用工具所在的文件夹,通过选择按钮选择一个Excel文件便可以重新确定路径,选择文件直接使用OpenFileDialog控件打开文件选择对话框,并通过Filter参数对文件类型进行过滤,选择要操作的文件。

后面的选择路径则是用于批量生成新文件的保存目录。文件目录的选择通过FolderBrowserDialog控件打开文件夹浏览得到。

1.3 搜索路径下的Excel文件

首先获取要搜索的路径,要得到文件路径(TextExcelFileName.Text)下的"\的位置,才能得到要搜索的路径,定义IntFPLen=InStrRev(StrXlsFilePath,"\"),这样前面的就是文件的路径:SrchExcelFilePath StrXlsFilePath.Substring(0, IntFPLen),如果是当前目录下的文件搜索直接和应用所在目录下的搜索方式相同,直接StrExcelFiles=Directory.GetFiles(SrchExcelFilePath, "*.xls")就可以得到Excel文件(Excel后缀为".xls"或".xlsx",搜索前者包含后者)。如果选择搜索目录下所包含子目录下的Excel文件,则需要进一步进行文件夹下的子文件搜索,StrFileDir=System.IO.Directory.GetDirectories(SrchExcelFilePath),得到文件夹StrFileDir后,然后对文件夹中的文件再进行一次Excel文件搜索即可。有多个层次则需要使用递归方式检查每个目录下的文件。

2 VB.NET调用Excel

2.1 Excel引用

使用Visual Studio中的VB.NET可以非常方便的调用Excel,新建工程后,只需要通过菜单栏的项目下单击添加引用,在COM选项卡添加对"Microsoft Excel xx.0 Object Library"的引用,然后在程序中通过编写Imports Microsoft.Office.Interop进行引用。

引用后使用Dim AppExcel As Excel.Application, XlsWbk As Excel.Workbook, XlsSht As Excel.Worksheet,就可以将定义的Excel表格变量对应的Excel应用、工作簿和表单关联,来实现表格的操作。

2.2 打开Excel文件并赋值给对象

打开前需要进行一些Excel属性赋值以提高效率,关闭Excel的报警功能AppExcel.DisplayAlerts=False,打开路径中的Excel文件,并让后台运行,AppExcel.Workbooks.Open(TxtExcelFile),AppExcel.Visible = False,再将表格赋值给变量XlsWbk=AppExcel.Workbooks(1),其中1代表第一张表格。

2.3 打开文件

操作完成后,需要检查文件可以直接选择列表框中的文件,打开文件进行检查。打开文件前先判断是否存在, System.IO.File.Exists(TextExcelFileName.Text),文件存在可以使用

```
System.Diagnostics.Process.Start(TextExcelFileName.Text)
```

t)打开路径下的文件。

3 批量处理

3.1 设定指定单元格内容

资料上需要添加人员签字等内容到表格指定的单元格中,在界面上设定单元格的位置(需符合单元格名称格式),在后面设置单元格要设定的内容,打开表格后用XlsSht.Range(TextExcelCellPos.Text).Value = TextSetCellContext.Text完成赋值。要进行整个列表中的表格都要设置,只需要对列表中的文件进行循环操作就可以完成列表中所有表格的指定单元格设置,而不需要一张一张的表格去打开再设置。需要清空单元格,只需将设定内容为空即可。

3.2 替换文件中的内容和搜索内容

Excel文件中的部分内容可能输错了,如公司或材料名称有问题,需要将所有文件中的涉及整个名称的字符替换,输入要替换的字符,再输入替换后的字符,Excel文件打开后,得到Excel表单的数量ExcelShtCnt=XlsWbk.Sheets.Count,循环在这些表格中使用Replace()函数进行内容替换,即:

```
For i=1 To ExcelShtCnt
    XlsSht=XlsWbk.Worksheets(i)
    XlsSht.Cells.Replace(What:=StrFind, Replacement:=StrReplace)
```

Next

替换单个表格文件,只需要选择当前列表中的文件,文件名称就会显示在路径中,直接替换表格中的路径文件即可。由于一般不是替换单个表格中的内容,可以将列表中的表格进行循环批量替换即可。需要删除某些内容,只需要将替换后的内容为空即可。

替换执行的时候,是否存在替换,先进行查找再进行替换,否则替换完成了就无法找到替换前的字符,查找使用XlsSht.UsedRange.Cells.Find(What:=StrFind, LookAt:=Excel.XlLookAt.xlPart).Cells.Count得到一个表格中的文件的数量,判断查找到的数量大于0,则存在替换。存在替换则执行完成后在弹出Msgbox对话框中增加替换的信息。

同时利用此Find方法还可以实现对单个或列表框中的表格进行字符串的查找,以判断是否存在对应的字符串。可以用在某项工作,某个任务,或某批次的材料忘记什么时间进场的,可以通过对表格列表进行搜索内容,就可以搜索到。

3.3 批量生成文件

施工日志,周计划,月计划,周报,月报,进度报表等之类的需要周期性的文件,调用模板按照规定的周期批量生成,模板文件是以文件路径选择的,周期则可以按照天为单位和月为单位进行选择,后面输入对应的数量,则数量和单位就是生成表格的间隔,如周报则是单位为天,数量为7,季报则是单位为月,数量为3。

开始的日期使用DateTimePicker控件可以轻松的选择开始

日期和结束日期,开始日期默认值在应用启动的时候设置为今天,也可以进行选择,结束日期也用DateTimePicker控件进行选择,按照项目的资料需要持续时间设置结束日期。然后使用DtProjectStartDate=DTPPrjStartDate.Value得到起始日期。先初始化增量DtIncDay=DateAdd(DateInterval.Day, 0, DtProjectStartDate),结合设置的日期循环计算增量,如果是按照日为单位,得到每加一次的日期增量IncStep=IncStep+IntStepDay,得到增量后的日期DtIncDay=TmpDtPrjStartDate.AddDays(IncStep)。比较增量后的日期和工程的结束日期,如果大于则结束循环。

表格生成后,在表格中需要显示日期以区别不同的表格,显示还需要指定日期和星期所在表格位置。文件的保存路径保存至批量路径下,保存的名称设置中包含批量名称,如项目名称以及文件对应的日期。使用DayOfWeek.Monday得到的星期是数字,数字通过转换为一二三。文件名称采用日期加项目名称、模板名称的方式来命名新生成的文件,参数设置好之后,单击批量生成按钮,就在批量路径下生成表格。不同类型的文件选择在不同的路径下,也有利于不同类型文件的管理。

3.4 退出应用

完成对Excel操作后,恢复报警功能AppExcel.DisplayAlerts=True,再释放对象,才能退出应用,使用If Not(XlsSht Is Nothing)Then XlsSht=Nothing,释放XlsSht对象,同样方法释放XlsWbk,再关闭Excel文件,关闭的时候进行文件保存,AppExcel.ActiveWorkbook.Close(SaveChanges:=True),关闭后退出应用,AppExcel.Quit(),再释放AppExcel,AppExcel=Nothing,最

后使用GC.Collect()释放资源,整个过程按顺序执行才能将Excel应用进程退出。

4 结论

借助通用批处理工具,其价值贯穿项目全生命周期:在前期资料准备阶段,可实现文件批量整理、格式统一、数据初步校验等操作,大幅减少重复手动劳动;在面对中期新增需求时,能快速适配批量修改、数据补充、流程适配等场景,灵活响应变更;进入后期资料检查环节,可通过批量核查、合规性校验等功能,精准定位问题。这一工具的应用,不仅显著提升了各阶段的工作效率,更通过标准化、自动化的处理方式,有效规避了人工操作易出现的疏漏,确保了项目资料的准确性与一致性。

参考文献

[1] 丁士锋,张鹏伟.Excel VBA标准教程[M].化学工业出版社,2011.

[2] 北京土木建筑学会主编,机电安装工程施工资料表格填写范例[M].机械工业出版社,2010.

[3] 丘仲潘,宋智军.Visual Basic2010从入门到精通[M].电子工业出版社,2011.

[4] 代洪卫,黄志安.工程施工常用表格与制度范本选用指南[M].湖南大学出版社,2010.

[5] 张明.EXCEL 2007 VBA办公范例应用[M].清华大学出版社,2007.

作者简介:

刘利(1983--),男,汉族,湖南娄底人,本科,职称:工程师,从事研究方向:能源管理,轨道交通、工业自动化和新能源。