
文章

[Michael Lei](#) · 四月 19, 2022 阅读大约需 13 分钟

用Caché ObjectScript 生成EXCEL

有很多方法可以使用InterSystems生成excel文件，其中一些是ZEN报告、IRIS报告（Logi报告或正式称为JReports），或者我们可以使用第三方Java库，可能性几乎是无限的。

但是，如果你想只用Caché ObjectScript创建一个简单的电子表格呢？(没有第三方应用程序)

在我的案例中，我需要生成包含大量原始数据的报告（财务人员喜欢这些数据），但是我的ZEN/IRIS失败了，给了我一个我想称之为“零字节的文件”，基本上说java的内存用完了，并导致报告服务器上的重载。

这可以用Office Open XML（OOXML）来完成。Office Open XML格式是由一个ZIP包内的一些XML文件组成的。因此，基本上我们需要生成这些XML文件，并将其压缩重命名为.xlsx。就这么简单。

这些文件遵循一套简单的惯例，称为开放包装惯例。你需要声明各部分的内容类型，以及告诉消费应用程序应该从哪里开始。

为了创建一个简单的电子表格，我们至少需要5个文件。

- workbook.xml
- worksheet.xml
- [ContentTypes].xml
- styles.xml
- rels
 - .rels
 - workbook.xml.rels

workbook.xml

工作簿是各种工作表的容器。工作簿是你引用样式部分、共享字符串表以及适用于整个电子表格文件的任何其他信息的地方。

```
ClassMethod GenerateWorkbookXML(){
    set status =$$$OK
    set xmlfile = tempDirectoryPath_"workbook.xml"
    try{
        set stream = ##class(%Stream.FileCharacter).%New()
        set sc=stream.LinkToFile(xmlfile)
        do stream.WriteLine("<?xml version='1.0' encoding='UTF-8' standalone='yes'?>")
        do stream.WriteLine("<workbook xmlns='http://schemas.openxmlformats.org/spreadsheetml/2006/main' xmlns:r='http://schemas.openxmlformats.org/officeDocument/2006/relationships'>")
        do
            stream.WriteLine("<sheets> <sheet name='\"_workSheetName_\"' sheetId='1' r:id='rId1'/>")
            do stream.WriteLine("</sheets> </workbook>")

            do stream.%Save()
        }catch{
            set status=$$$NO
        }
    }
```

```

        kill stream
        return status
    }

```

rels/workbook.xml.rels

我们只需要创建一个id为rId1的关系，这样它就会与workbook.xml部分的引用相匹配

```

ClassMethod CreateRelsXML(){
set status =$$$OK

    set isunix=$zcvt($p($zv," ",3,$l($p($zv," (")," ")), "U")["UNIX"]
    if isunix {
        set ext="/"
    }else{
        set ext="\ "
    }
    set xmlfile = fileDirectory_"_rels"_ext_"workbook.xml.rels"
    set stream = ##class(%Stream.FileCharacter).%New()
    set sc=stream.LinkToFile(xmlfile)
    do stream.WriteLine("<?xml version='1.0' encoding='UTF-8' standalone='yes'?>")
    do stream.WriteLine("<Relationships xmlns='http://schemas.openxmlformats.org/package/2006/relationships'>")
        do stream.WriteLine("<Relationship Id='rId1' Type='http://schemas.openxmlformats.org/officeDocument/2006/relationships/worksheet' Target='worksheet.xml'/>")
        do stream.WriteLine("<Relationship Id='rId2' Type='http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles' Target='styles.xml' />")
    do stream.WriteLine("</Relationships>")
    try{
        do stream.%Save()
    }catch{
        set status=$$$NO
    }
    kill stream
    set xmlfile = fileDirectory_"_rels"_ext_"_rels"
    set stream = ##class(%Stream.FileCharacter).%New()
    set sc=stream.LinkToFile(xmlfile)

    do stream.WriteLine("<?xml version='1.0' encoding='UTF-8' standalone='yes'?>")
    do stream.WriteLine("<Relationships xmlns='http://schemas.openxmlformats.org/package/2006/relationships'>")
        do stream.WriteLine("<Relationship Id='rId1' Type='http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument' Target='workbook.xml'/>")
    do stream.WriteLine("</Relationships>")
    try{
        do stream.%Save()
    }catch{
        set status=$$$NO
    }
    kill stream
    return status
}

```

[ContentTypes].xml

静态文件（目前，它应该是一个动态文件，取决于工作表的数量）将工作簿的工作表和样式链接在一起。每个Office Open XML文件必须声明ZIP包中使用的內容类型。这是用[ContentTypes].xml文件完成的。

```

ClassMethod GenerateContentTypesXML(){
    set status =$$$OK
    set xmlfile = tempDirectoryPath_"[Content_Types].xml"
    set stream = ##class(%Stream.FileCharacter).%New()
    set sc=stream.LinkToFile(xmlfile)
    try{
        do stream.WriteLine("<?xml version='1.0' encoding='UTF-8' standalone='yes'?">")
    )
        do stream.WriteLine("<Types xmlns='http://schemas.openxmlformats.org/package/2006/content-types'>")
        do stream.WriteLine("<Default Extension='rels' ContentType='application/vnd.openxmlformats-package.relationships+xml' />")
        do stream.WriteLine("<Override PartName='/workbook.xml' ContentType='application/vnd.openxmlformats-officedocument.spreadsheetml.sheet.main+xml' />")
        do stream.WriteLine("<Override PartName='/worksheet.xml' ContentType='application/vnd.openxmlformats-officedocument.spreadsheetml.worksheet+xml' />")
        do stream.WriteLine("<Override PartName='/styles.xml' ContentType='application/vnd.openxmlformats-officedocument.spreadsheetml.styles+xml' />")
        do stream.WriteLine("</Types>")
        do stream.%Save()
    }catch{
        set status=$$$NO
    }
    kill stream
    return status
}

```

styles.xml

所有的格式化都在这里，目前我已经添加了一些静态样式，（计划将其转换为更多的动态工作簿特定的样式）。

Excel Styles	ID	Style	Excel Format
	1	default	Text
	2	#, [Red]-#	Number
	3	###, [Red]-###	Number
	4	yyyy/mm/dd	Date
	5	hh:mm	Date
	6	Header and Center Aligned	Text
	7	Header 2 Left Aligned	Text
	8	Good (Green Highlight)	General
	9	Bad (Red Highlight)	General
	10	Neutral (Orange Highlight)	General
	11	yyyy/mm/dd hh:mm	Date

```

ClassMethod CreateStylesXML(){
    set status =$$$OK
    set xmlfile = tempDirectoryPath_"styles.xml"
    try{
        set stream = ##class(%Stream.FileCharacter).%New()
        set sc=stream.LinkToFile(xmlfile)
        do stream.WriteLine("<?xml version='1.0' encoding='UTF-8' standalone='yes'?">")
        do stream.WriteLine("<styleSheet xmlns='http://schemas.openxmlformats.org/sp

```

```

readsheetml/2006/main"" xmlns:mc=""http://schemas.openxmlformats.org/markup-compatibility/2006"" mc:Ignorable=""x14ac x16r2 xr"" xmlns:x14ac=""http://schemas.microsoft.com/office/spreadsheetml/2009/9/ac"" xmlns:x16r2=""http://schemas.microsoft.com/office/spreadsheetml/2015/02/main"" xmlns:xr=""http://schemas.microsoft.com/office/spreadsheetml/2014/revision"">)
    do stream.WriteLine("<numFmts count=""4"">")
    do stream.WriteLine("<numFmt numFmtId=""166"" formatCode=""#,##0;[Red]\-#,##0""/>")
    do stream.WriteLine("<numFmt numFmtId=""168"" formatCode=""#,##0.00;[Red]\-#,##0.00""/>")
    do stream.WriteLine("<numFmt numFmtId=""169"" formatCode=""dd\mm\yyyy;@""/>")
    do stream.WriteLine("<numFmt numFmtId=""170"" formatCode=""dd/mm/yyyy\ hh:mm""/></numFmts>")
    do stream.WriteLine("<fonts count=""5"" x14ac:knownFonts=""1"">")
    do stream.WriteLine("<font><sz val=""10""/><color theme=""1""/><name val=""Calibri""/><family val=""2""/><scheme val=""minor""/></font>")
    do stream.WriteLine("<font><sz val=""10""/><color rgb=""FF006100""/><name val=""Calibri""/><family val=""2""/><scheme val=""minor""/></font>")
    do stream.WriteLine("<font><sz val=""10""/><color rgb=""FF9C0006""/><name val=""Calibri""/><family val=""2""/><scheme val=""minor""/></font>")
    do stream.WriteLine("<font><sz val=""10""/><color rgb=""FF9C5700""/><name val=""Calibri""/><family val=""2""/><scheme val=""minor""/></font>")
    do stream.WriteLine("<font><b><sz val=""10""/><color theme=""1""/><name val=""Calibri""/><family val=""2""/><scheme val=""minor""/></font></fonts>")
    do stream.WriteLine("<fills count=""5"">")
    do stream.WriteLine("<fill><patternFill patternType=""none""/></fill>")
    do stream.WriteLine("<fill><patternFill patternType=""gray125""/></fill>")
    do stream.WriteLine("<fill><patternFill patternType=""solid""><fgColor rgb=""FFC6EFCE""/></patternFill></fill>")
    do stream.WriteLine("<fill><patternFill patternType=""solid""><fgColor rgb=""FFFFC7CE""/></patternFill></fill>")
    do stream.WriteLine("<fill><patternFill patternType=""solid""><fgColor rgb=""FFFFEB9C""/></patternFill></fill></fills>")
    do stream.WriteLine("<borders count=""1""><border><left/><right/><top/><bottom/><diagonal/></border></borders>")
    do stream.WriteLine("<cellStyleXfs count=""4"">")
    do stream.WriteLine("<xf numFmtId=""0"" fontId=""0"" fillId=""0"" borderId=""0""/>")
    do stream.WriteLine("<xf numFmtId=""0"" fontId=""1"" fillId=""2"" borderId=""0"" applyNumberFormat=""0"" applyBorder=""0"" applyAlignment=""0"" applyProtection=""0""/>")
    do stream.WriteLine("<xf numFmtId=""0"" fontId=""2"" fillId=""3"" borderId=""0"" applyNumberFormat=""0"" applyBorder=""0"" applyAlignment=""0"" applyProtection=""0""/>")
    do stream.WriteLine("<xf numFmtId=""0"" fontId=""3"" fillId=""4"" borderId=""0"" applyNumberFormat=""0"" applyBorder=""0"" applyAlignment=""0"" applyProtection=""0""/></cellStyleXfs>")
    do stream.WriteLine("<cellXfs count=""12""><xf numFmtId=""0"" fontId=""0"" fillId=""0"" borderId=""0"" xfId=""0""/>")
    do stream.WriteLine("<xf numFmtId=""49"" fontId=""0"" fillId=""0"" borderId=""0"" xfId=""0"" quotePrefix=""1"" applyNumberFormat=""1""/>")
    do stream.WriteLine("<xf numFmtId=""166"" fontId=""0"" fillId=""0"" borderId=""0"" xfId=""0"" applyNumberFormat=""1""/>")
    do stream.WriteLine("<xf numFmtId=""168"" fontId=""0"" fillId=""0"" borderId=""0"" xfId=""0"" applyNumberFormat=""1""/>")
    do stream.WriteLine("<xf numFmtId=""169"" fontId=""0"" fillId=""0"" borderId=""0"" xfId=""0"" applyNumberFormat=""1""/>")
    do stream.WriteLine("<xf numFmtId=""20"" fontId=""0"" fillId=""0"" borderId=""

```

```

"0" xfid="0" applyNumberFormat="1"/>")
    do stream.WriteLine("<xf numFmtId=""49"" fontId=""4"" fillId=""0"" borderId=""
"0" xfid="0" applyNumberFormat="1" applyFont="1"/>")
    do stream.WriteLine("<xf numFmtId=""49"" fontId=""4"" fillId=""0"" borderId=""
"0" xfid="0" applyNumberFormat="1" applyFont="1" applyAlignment="1"><alignme
nt horizontal=""center""/>")
    do stream.WriteLine("</xf>")
    do stream.WriteLine("<xf numFmtId=""49"" fontId=""1"" fillId=""2"" borderId=""
"0" xfid="1" applyNumberFormat="1"/>")
    do stream.WriteLine("<xf numFmtId=""0"" fontId=""2"" fillId=""3"" borderId=""
0" xfid="2"/>")
    do stream.WriteLine("<xf numFmtId=""0"" fontId=""3"" fillId=""4"" borderId=""
0" xfid="3"/>")
    do stream.WriteLine("<xf numFmtId=""170"" fontId=""0"" fillId=""0"" borderId=""
""0" xfid="0" applyNumberFormat="1"/></cellXfs>")
    do stream.WriteLine("<cellStyles count=""4""><cellStyle name=""Bad"" xfid=""2
"" builtinId=""27""/>")
    do stream.WriteLine("<cellStyle name=""Good"" xfid=""1"" builtinId=""26""/><c
ellStyle name=""Neutral"" xfid=""3"" builtinId=""28""/>")
    do stream.WriteLine("<cellStyle name=""Normal"" xfid=""0"" builtinId=""0""/><
/cellStyles><dxfs count=""0""/>")
    do stream.WriteLine("<tableStyles count=""0"" defaultTableStyle=""TableStyleM
edium2"" defaultPivotStyle=""PivotStyleLight16""/>")
    do stream.WriteLine("<extLst><ext uri=""{EB79DEF2-80B8-43e5-95BD-54CBDDF9020C
}"" xmlns:x14=""http://schemas.microsoft.com/office/spreadsheetml/2009/9/main"">")
    do stream.WriteLine("<x14:slicerStyles defaultSlicerStyle=""SlicerStyleLight1
""/></ext><ext uri=""{9260A510-F301-46a8-8635-F512D64BE5F5}"" xmlns:x15=""http://sche
mas.microsoft.com/office/spreadsheetml/2010/11/main"">")
    do stream.WriteLine("<x15:timelineStyles defaultTimelineStyle=""TimeSlicerSty
leLight1""/></ext></extLst>")
    do stream.WriteLine("</styleSheet>")
    do stream.%Save()
}catch{
    set status=$$$$NO
}
kill stream
return status
}

```

worksheet.xml

这是我们的数据所在的地方。工作表的第一行将有列的标题。接下来的行将只有第一列的数据。

我们将在这里定义每一列的列宽，如果不是默认的，列将被设置为自动适应。

worksheet.xml 示例

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<worksheet xmlns="https://schemas.openxmlformats.org/spreadsheetml/2006/main" xmlns:r
="https://schemas.openxmlformats.org/officeDocument/2006/relationships">
<sheetData>
<row>
    <c t="inlineStr">
        <is>
            <t>Name</t>
        </is>
    </c>
    <c t="inlineStr">
        <is>
            <t>Amount</t>

```

```
</is>
</c>
</row>
<row>
  <c t="inlineStr">
    <is>
      <t>Jhon Smith</t>
    </is>
  </c>
  <c>
    <v>1000.74</v>
  </c>
</row>
<row>
  <c t="inlineStr">
    <is>
      <t>Tracy A</t>
    </is>
  </c>
  <c>
    <v>6001.74</v>
  </c>
</row>
</sheetData>
</worksheet>
```

Excel 示例

A	B
Name	Amount
Jhon Smith	1000.74
Tracy A	6001.74

工作表中的公式可以用函数<f>标签来完成

```
<c >
<f>B2*0.08</f >
</c >
<c >
<f>B2+C2</f >
</c>
```

and finally we zip them, rename it to.xlsx (using unix zip)

```
set cmd ="cd "_fileDirectory_" && find . -type f | xargs zip .."_ext_xlsxFile
```

生成excel文件.

以下代码生成excel 文件.

```
set file = "/temp/test.xlsx"
set excelObj = ##class(XLSX.writer).%New(file)
do excelObj.SetWorksheetName("test1")
set status = excelObj.BeginWorksheet()
set row = 0
set row = row+1
;----- excelObj.Cells(rowNumber,columnNumber,style,content)

set status = excelObj.Cells(row,1,1,"Header1")
set row = row+1
set status = excelObj.Cells(row,1,2,"Content 1")
set status = excelObj.EndWorksheet()
W !,excelObj.fileName
```

写Excel类请看这里 [xlsx.writer.xml.zip](#)

[#ObjectScript](#) [#Caché](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%94%A8cach%C3%A9-objectscript-%E7%94%9F%E6%88%90excel>