

Publishing from JMP

Catherine Hosage Norman, Ph.D.

Publishing

- Produce or release for distribution
- Implies print but also implies layout
- Typical approach is to copy & paste into Word
 - repetitive task
- Table to word, first layout, then copy & paste
- Run report and then cut & paste graphics
- Both require formatting in Word

Programmatic Approach

- Involves using JMP objects and functions to manipulate Office Open XML products via:
 - Microsoft XML SDK
 - PowerShell
 - Or any .NET language
 - PowerTools 2.2- library of command line calls

Office Open XML

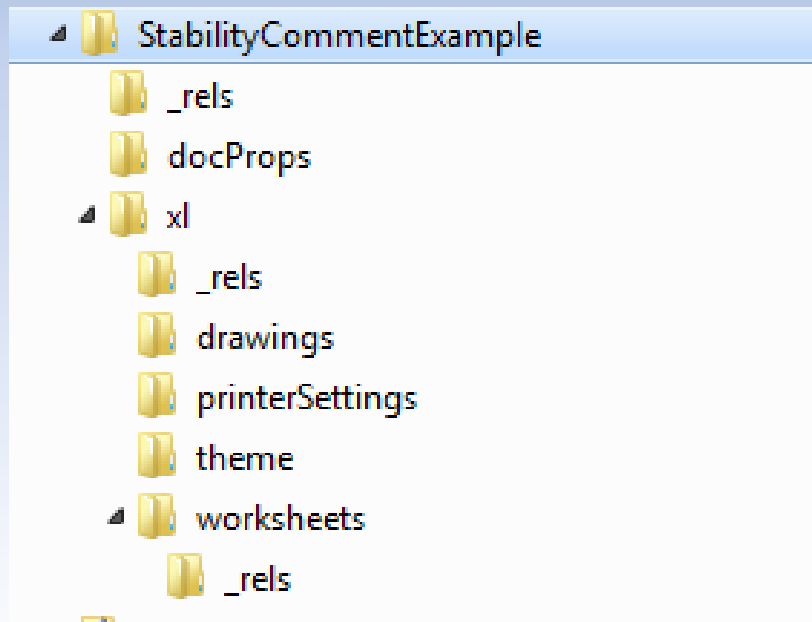
- XML standard format for office applications
- Also known as OOXML or Open XML **not** to be confused with Open Office
- History – started 1998 with the Beta version of Office 2000 released with HTML support in Word, PowerPoint, and Excel.
- 2007 - Ecma standard and submitted to ISO
- File extensions on Word, Excel, PowerPoint etc. changed and added x at the end
 - .doc => .docx

Structure of the Open XML Formats

- **Start part** - uppermost part in the hierarchy
- **XML parts** - Files or folders consisting of XML files that contain the content
- **Non-XML parts** - usually images or OLE objects
- **Relationship part** - XML files that define the relational hierarchy of the parts
- **ZIP package** - Packs the parts into a single file

Packaging Conventions

- Simplest term – a directory structure
- Excel file structure as an example



The ZIP Archive

- JMP object that allows direct access to Open XML directories
- Rename any Open XML file extension to .zip and extract the parts
- ZipArchive object operates directly on a zip file

JMP XML Gateway Commands

- Web – Opens a URL or stored file
- ZipArchive object
- Utility commands-Copy File
- XML Encode – formats text as XML fragment
- Parse XML – Parses XML and triggers events
- XPath Query (new 11)

Demonstration

- Extract comments and comment authors from [Excel file](#)
- Information found in:
 - .\filename\xl\ comments1.xml
 - .\filename\xl\ comments2.xml
 - ...
- `za = Open("C:\Temp\wrkExcelasXml.zip", zip);`
- [Link to JSL](#)

Writing to Word

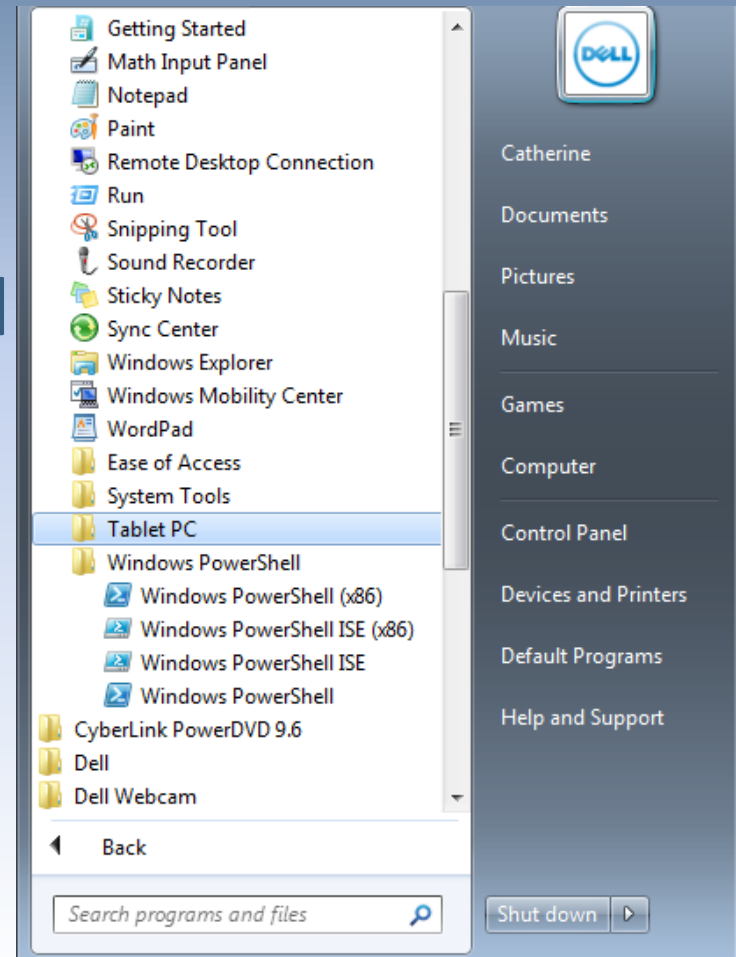
- Simple case of updating monthly report with new graph
- Requires PowerShell
- Write function of ZipZrchive does not copy over existing file and appears to turn the image into text

PowerShell

- Task-based command-line shell and scripting language
- Designed for system administration
- Built on the .NET Framework
- Scripts control and automate administration of Windows OS commands and applications
- Built-in Windows PowerShell commands are called *cmdlets*
- Accesses data stores e.g. registry and certificate store
- In this case - the file system
- Expression parser, scripting language and ISE

Accessing PowerShell

- Start menu windows 7
- >Accessories
 - > Windows PowerShell
- Downloadable earlier versions of Windows



Example Structure of ps1 file

```
Windows PowerShell ISE (x86)
File Edit View Debug Help
delPics.ps1* X zipcopy.ps1
1 $zip = "C:\Temp\wrkwordasxml.zip"
2 $shellApp = new-object -com shell.application
3 $zipFile = $shellApp.Namespace("$zip")
4
5 $files = $zipFile.Items()
6 Foreach ($file in $files)
7 {
8     $filename = $file.Name
9     write-host " File: $filename"
10    if ( $filename -eq 'word')
11    {
12        $mediafiles = $file.GetFolder.items()
13
14        Foreach ($mediafile in $mediafiles)
15        {
16            $medianame = $mediafile.Name
17            write-host " File: $medianame"
18            if ( $medianame -eq 'media')
19            {
20                $imgfiles = $mediafile.GetFolder.items()
21                Foreach ($imgfile in $imgfiles)
22                {
23                    $imgname = $imgfile.Name
24                    write-host " File: $imgname"
25                    if ( $imgname -eq 'image3.png' -or $imgname -eq 'image4.png' -or $imgname -eq 'image5.png')
26                    {
27                        $imgfile.InvokeVerb("Delete")
28                    }
29                }
30                $mediafile.CopyHere( "C:\JMP\JMP2014Discovery\Example_files\image3.png" )
31                $mediafile.CopyHere( "C:\JMP\JMP2014Discovery\Example_files\image4.png")
32                $mediafile.CopyHere( "C:\JMP\JMP2014Discovery\Example_files\image5.png" )
33            }
34        }
35    }
36 }
37 }
38 }
```

Task – Update Graph in Word

- File to be updated
- File directory
- JMP Script demo
- .bat file as glue between JMP and PowerShell
- Image files created
- Word file created by copy and paste

Creating Word Documents

- Last example edited an existing document
- Drafting a .docx file requires more
- Requires Open XML SDK for Office
- Access API via any .NET language like vb
- PowerShell cmdlets known as PowerTools

PowerTools

- Started at Microsoft circa 2008
- Version 2.2 last version to support cmdlet interface released Jan 2012
- Now a CodePlex project
- Encapsulates common document building function in to simple PowerShell command line calls
- <http://powertools.codeplex.com/>

Task – Write JMP table to Word

xml structure for a
table in a .docx file

Table start

Table Style

Columns

Row

Row

Row

End Table

```
<w:tbl>
  <w:tblPr>
    <w:tblStyle w:val="LightList-Accent1",
    <w:tblW w:w="5000" w:type="pct"/>
    <w:tblLook w:val="04A0"/>
  </w:tblPr>
  <w:tblGrid>
    <w:gridCol/>
    <w:gridCol/>
    <w:gridCol/>
  </w:tblGrid>
  <w:tr>
    <w:tc>
      <w:p>
        <w:r>
          <w:t>Title</w:t>
        </w:r>
      </w:p>
    </w:tc>
    <w:tc>
      <w:p>
        <w:r>
          <w:t>Creator</w:t>
        </w:r>
      </w:p>
    </w:tc>
    <w:tc>
      <w:p>
        <w:r>
          <w:t>Description</w:t>
        </w:r>
      </w:p>
    </w:tc>
  </w:tr>
</w:tbl>
```

Creating XML in JMP

- Excellent example of creating a file in *JSL Companion*, Utlaut, Morgan and Anderson
- Adapted to script that transposes a JMP table into a standard stability report layout
- Originally pasted the layout into Word

Task Objective

TESTNAME	LOT_NUM	STRENGTH	PACKTYPE	TIME_PT	STORCOND	STORTEMP	STOR_RH	METH_NUM	CRITERIA	RESULT
Assay	60151-L-ALU	400 mg	ALU Blister	0	25°C/60%RH	25	60	MHPRD6	95.0 % to 105.0 % w/w of label claim	99.2
Assay	60151-L-ALU	400 mg	ALU Blister	3	25°C/60%RH	25	60	MHPRD6	95.0 % to 105.0 % w/w of label claim	98.7
Assay	60151-L-ALU	400 mg	ALU Blister	6	25°C/60%RH	25	60	MHPRD6	95.0 % to 105.0 % w/w of label claim	98.8
Assay	60151-L-ALU	400 mg	ALU Blister	9	25°C/60%RH	25	60	MHPRD6	95.0 % to 105.0 % w/w of label claim	97.9
Assay	60151-L-ALU	400 mg	ALU Blister	12	25°C/60%RH	25	60	MHPRD6	95.0 % to 105.0 % w/w of label claim	97.2
Assay	60151-L-ALU	400 mg	ALU Blister	18	25°C/60%RH	25	60	MHPRD6	95.0 % to 105.0 % w/w of label claim	97.6



Stability Data for: 60151-L-ALU 25-60
Lot: 60151-L-ALU **Condition:** 25°C/60%RH **Study Start Date:** 15-Jan-12
Manufacturer: Ajor Pharmaceuticals, 58 Enterprise Drive, Livonia MI 48150 **DOM:** 30-Oct-11 **Lot Size:** 100000
Strength: 400 mg **Packaging:** ALU Blister **Package Lot:** 70158-A

Method	Test Name	Acceptance Criteria	Initial	3 mon	6 mon	9 mon	12 mon	18 mon
Appearance	Appearance	White circular tablets	Complies	Complies	Complies	Complies	Complies	Complies
Assay	Assay	95.0 % to 105.0 % w/w of label claim	99.2	98.7	98.8	97.9	97.2	97.6
Dissolution	High (30 minutes)	≥ 75 %	94	96	99	97	96	97
Dissolution	High (5 minutes)	≥ 50 %	75	80	80	76	80	84
Dissolution	Low (30 minutes)	≥ 75 %	91	92	97	94	93	93
Dissolution	Low (5 minutes)	≥ 50 %	72	77	79	74	77	82
Dissolution	Mean (30 minutes)	≥ 75 %	93	94	97	95	94	94
Dissolution	Mean (5 minutes)	≥ 50 %	74	78	80	76	79	83
Friability	Friability	≤ 0.5 %	0.2	0.2	0.2	0.2	0.2	0.1
Hardness	Average	≥ 10 kp (98 N)	16	15	15	16	15	15

Demo Writing DOCX

- JMP script
- Resulting xml file
- PowerShell Script
- Resulting word file

Conclusion

- Automating the Publishing JMP output is possible
- Has it challenges
- Worth the effort for repetitive reporting tasks