

PS1 files used in demo

Copy last word doc and replace image files

```
DATA { "Copyright (C) 2014 (Catherine Hosage Norman, Module 3 Solutions, LLC)"  
      "This code is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or"  
      "FITNESS FOR A PARTICULAR PURPOSE." }
```

```
$zip = "C:\Temp\WrkWordasXml.zip"  
$shellApp = new-object -com shell.application  
$zipFile = $shellApp.NameSpace("$zip")
```

```
$Files = $zipFile.Items()  
Foreach ($File in $Files)  
{  
  $Filename = $File.Name  
  if ( $Filename -eq 'word')  
  {  
    $Mediafiles = $File.GetFolder.items()  
  
    Foreach ($Mediafile in $Mediafiles)  
    {  
      $Medianame = $Mediafile.Name  
      if ( $Medianame -eq 'media')  
      {  
        $fileRoot = $Mediafile.GetFolder  
        $Imgfiles = $Mediafile.GetFolder.items()
```

```
Foreach ($Imgfile in $Imgfiles)
{
$Imgname = $Imgfile.Name

if ( $Imgname -eq 'image3.png' )
{
$Imgfile.InvokeVerb("Delete")
$fileRoot.CopyHere( "C:\JMP\JMP2014Discovery\Example_files\image3.png", 8 -and 64 -and 256 )

        Start-sleep -seconds 1

}

if ( $Imgname -eq 'image4.png' )
{
$Imgfile.InvokeVerb("Delete")
$fileRoot.CopyHere( "C:\JMP\JMP2014Discovery\Example_files\image4.png", 8 -and 64 -and 256 )

        Start-sleep -seconds 1

}

if ( $Imgname -eq 'image5.png' )
{
$Imgfile.InvokeVerb("Delete")
$fileRoot.CopyHere( "C:\JMP\JMP2014Discovery\Example_files\image5.png", 8 -and 64 -and 256 )

        Start-sleep -seconds 1

}

}
```

}

}

}

}

Create New Word Doc from XML file created by JMP script

```
DATA { "Copyright (C) 2014 (Catherine Hosage Norman, Module 3 Solutions, LLC)"
      "This code is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or"
      "FITNESS FOR A PARTICULAR PURPOSE." }
```

```
function AddParagraph
```

```
{
  $s1 = "<w:p>"
  $s2 = ""
  if ($args[2] -ne $NULL)
  {
    $s2 = "<w:pPr><w:pStyle w:val='' + $args[2] + ''/></w:pPr>"
  }
  $s3 = "<w:r><w:t>" + $args[1] + "</w:t></w:r></w:p>"
  $stringToAdd = $s1 + $s2 + $s3

  return Add-OpenXmlContent -Document $args[0] -PartPath '/word/document.xml' `
    -InsertionPoint '/w:document/w:body/w:*[last()]|/w:styles/w:style[1]' `
    -Content $stringToAdd
}
```

```
function AddHeader
```

```
{
  $s1 = "<w:p>"
  $s4 = "<w:pPr> <w:pStyle w:val='Heading2'/><w:rPr> <w:rFonts w:ascii='Times New Roman' w:hAnsi='Times New Roman' w:cs='Times New Roman'/> </w:rPr> </w:pPr>"

  $s3 = "<w:r><w:t>" + $args[1] + "</w:t></w:r></w:p>"
```

```
$stringToAdd = $s1 + $s4 + $s3
```

```
return Add-OpenXmlContent -Document $args[0] -PartPath '/word/document.xml' `
  -InsertionPoint '/w:document/w:body/w:*[last()]|/w:styles/w:style[1]' `
  -Content $stringToAdd
}
```

```
function AddLandscape
```

```
{
  $content = @"
    <w:sectPr >
<w:pgSz w:w="15840" w:h="12240" w:orient="landscape"/>
<w:pgMar w:top="1440" w:right="1440" w:bottom="1440" w:left="1440" w:header="720" w:footer="720" w:gutter="0"/>
<w:cols w:space="720"/>
<w:docGrid w:linePitch="360"/>
</w:sectPr>
"@
```

```
return Add-OpenXmlContent -Document $args[0] -PartPath '/word/document.xml' `
  -InsertionPoint '/w:document/w:body/w:*[last()]|/w:styles/w:style[1]' `
  -Content $content
}
```

```
function CreateWordProcessingDocument
```

```
{
  $d = Export-OpenXmlWordprocessing -Text $args[1] -OutputPath $args[0]

  $fullPath = "C:\JMP\JMP2014Discovery\Example_files" + "\Templates\Template3.docx"
```

```
$d = Get-OpenXmlStyle -Document $fullPath | Set-OpenXmlStyle -Document $args[0]
```

```
return $d  
}
```

```
function AddRow($markup, $valueArray, $rightJustifiedArray, $bold)
```

```
{  
    $result = $markup  
    $result += "<w:tr>"  
    $formatting = ""  
    if ($bold)  
    {  
        $formatting = "<w:rPr><w:b/></w:rPr>"  
    }  
    $index = 0
```

```
foreach ($c in $valueArray)
```

```
{  
    $rightJustifiedFormatting = ""  
    if ($rightJustifiedArray[$index])  
    {  
        $rightJustifiedFormatting = "<w:pPr>  
            <w:jc w:val='left' />  
        </w:pPr>"  
    }  
    else  
    {  
        $rightJustifiedFormatting = "<w:pPr>
```

```

                <w:jc w:val='center'/>
            </w:pPr>"
    }

    $result += "<w:tc>
        <w:tcPr>
            <w:tcW w:w='0'
                w:type='auto'/>
            </w:tcPr>
        <w:p>"
    $result += $rightJustifiedFormatting
    $result += "<w:r>"
    $result += $formatting
    $result += "<w:t>" + $c + "</w:t>
        </w:r>
        </w:p>
    </w:tc>"
    $index++
}
$result += "</w:tr>"
$result
}

```

```

function AddColHeads($markup, $valueArray, $bold)
{
    $result = $markup
    $result += "<w:tr>"
    $formatting = ""
    if ($bold)
    {

```

```
$formatting = "<w:rPr><w:b/></w:rPr>"
}
$index = 0

foreach ($c in $valueArray)
{

$result += "<w:tc>
    <w:tcPr>
        <w:tcW w:w='0'
            w:type='auto'/'>
        <w:noWrap/'>
    </w:tcPr>
    <w:p>"
$result += "<w:pPr>
    <w:jc w:val='center'/'>
    </w:pPr>"
$result += "<w:r>"
$result += $formatting
$result += "<w:t>" + $c + "</w:t>
    </w:r>
    </w:p>
    </w:tc>"
    $index++
}
$result += "</w:tr>"
$result
}
```



```

# 0 = document name
# 1 = dynamic object from xml
function CreateDocForData
{
#get lot info attributes and put in borderless table

$s = $args[1].name
$s2 = "Stability Data for: " + $args[1].name
$doc = AddHeader $args[0] $s2
$isAccelerated = $s.IndexOf("40-75")
$tabcontent = "<w:tbl>
  <w:tblPr>
<w:tblStyle w:val='TableGrid'/> <w:tblW w:w='0' w:type='auto'/>
<w:tblBorders> <w:top w:val='none' w:sz='0' w:space='0' w:color='auto'/> <w:left w:val='none' w:sz='0' w:space='0' w:color='auto'/> <w:right w:val='none' w:sz='0' w:space='0' w:color='auto'/> <w:bottom w:val='none' w:sz='0' w:space='0' w:color='auto'/>
<w:insideH w:val='none' w:sz='0' w:space='0' w:color='auto'/> <w:insideV w:val='none' w:sz='0' w:space='0' w:color='auto'/>
</w:tblBorders>
<w:tblLook w:val='04A0' w:firstRow='1' w:lastRow='0' w:firstColumn='1' w:lastColumn='0' w:noHBand='0' w:noVBand='1'/>
</w:tblPr>
<w:tblGrid>
<w:gridCol w:w='4068'/>
<w:gridCol w:w='4950'/>
<w:gridCol w:w='2070'/>
<w:gridCol w:w='2088'/>
</w:tblGrid>
<w:tr>
  <w:tc><w:tcPr><w:tcW w:w='4068' w:type='dxa'/></w:tcPr>
  <w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr> <w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>

```

```
<w:t xml:space='preserve'>Lot: </w:t>
</w:r>
    <w:r><w:rPr> <w:sz w:val='24' /> <w:szCs w:val='24' /></w:rPr>
    <w:t>" + $args[1].lot + "</w:t>
</w:r>
</w:p>
</w:tc>
<w:tc><w:tcPr><w:tcW w:w='4950' w:type='dxa' /></w:tcPr>
<w:p><w:rPr> <w:sz w:val='24' /> <w:szCs w:val='24' /> </w:rPr>
    <w:r><w:rPr><w:rStyle w:val='Strong' /> <w:sz w:val='24' /> <w:szCs w:val='24' /></w:rPr>
    <w:t xml:space='preserve'>Condition: </w:t>
</w:r>
    <w:r><w:rPr> <w:sz w:val='24' /> <w:szCs w:val='24' /></w:rPr>
    <w:t>" + $args[1].condition + "</w:t>
</w:r>
</w:p>
</w:tc>
<w:tc><w:tcPr><w:tcW w:w='4158' w:type='dxa' /></w:tcPr>
<w:p><w:rPr> <w:sz w:val='24' /> <w:szCs w:val='24' /> </w:rPr>
    <w:r><w:rPr><w:rStyle w:val='Strong' /> <w:sz w:val='24' /> <w:szCs w:val='24' /></w:rPr>
    <w:t xml:space='preserve'>Study Start Date: </w:t>
</w:r>
    <w:r><w:rPr> <w:sz w:val='24' /> <w:szCs w:val='24' /></w:rPr>
    <w:t>" + $args[1].start + "</w:t>
</w:r>
</w:p>
</w:tc>
</w:tr>
<w:tr>
    <w:tc><w:tcPr><w:tcW w:w='9018' w:type='dxa' /> <w:gridSpan w:val='2' /></w:tcPr>
```

```
<w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr><w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t xml:space='preserve'>Manufacturer: </w:t>
</w:r>
  <w:r><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t>" + $args[1].manufacturer + "</w:t>
</w:r>
</w:p>
</w:tc>
<w:tc><w:tcPr><w:tcW w:w='2070' w:type='dxa'/></w:tcPr>
  <w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr><w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t xml:space='preserve'>DOM: </w:t>
</w:r>
  <w:r><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t>" + $args[1].dom + "</w:t>
</w:r>
</w:p>
</w:tc>
<w:tc><w:tcPr><w:tcW w:w='2088' w:type='dxa'/></w:tcPr>
  <w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr><w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t xml:space='preserve'>Lot Size:</w:t>
</w:r>
  <w:r><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t>" + $args[1].lot_size + "</w:t>
</w:r>
</w:p>
</w:tc>
</w:tr>
```

```
<w:tr>
  <w:tc><w:tcPr><w:tcW w:w='4068' w:type='dxa'/></w:tcPr>
  <w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr><w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t xml:space='preserve'>Strength: </w:t>
</w:r>
  <w:r><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t>" + $args[1].strength + "</w:t>
</w:r>
</w:p>
</w:tc>
<w:tc><w:tcPr><w:tcW w:w='4950' w:type='dxa'/></w:tcPr>
  <w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr><w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t xml:space='preserve'>Packaging: </w:t>
</w:r>
  <w:r><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t>" + $args[1].packaging + "</w:t>
</w:r>
</w:p>
</w:tc>
<w:tc><w:tcPr><w:tcW w:w='4158' w:type='dxa'/><w:gridSpan w:val='2'/></w:tcPr>
  <w:p><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/> </w:rPr>
  <w:r><w:rPr><w:rStyle w:val='Strong'/> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t xml:space='preserve'>Package Lot: </w:t>
</w:r>
  <w:r><w:rPr> <w:sz w:val='24'/> <w:szCs w:val='24'/></w:rPr>
  <w:t>" + $args[1].packaginglot + "</w:t>
</w:r>
</w:p>
```

```
</w:tc>
</w:tr>
</w:tbl>"
```

```
$doc = Add-OpenXmlContent -Document $args[0] -PartPath '/word/document.xml' `
-InsertionPoint '/w:document/w:body/w:p[last()]/w:styles/w:style[1]' `
-Content $tabcontent
```

```
$doc = AddParagraph $args[0] ""
```

```
$markup = `
"<w:tbl>
  <w:tblPr>
    <w:tblStyle w:val='TableGrid' />
    <w:tblW w:w='0'
      w:type='auto' />
    <w:tblLook w:val='04A0' />
  </w:tblPr>
"
```

```
if($isAccelerated -eq -1)
{
  $rja = @($True, $True, $True, $False, $False, $False, $False, $False, $False)
  $markup = AddColHeads $markup @("Method", "Test Name", "Acceptance Criteria", "Initial", "3 mon", "6 mon", "9 mon", "12 mon", "18
mon") $true
}
else
{
  $rja = @($True, $True, $True, $False, $False, $False)
  $markup = AddColHeads $markup @("Method", "Test Name", "Acceptance Criteria", "Initial", "3 month", "6 month") $true
}
```

```
}

foreach ($o in $args[1].analysis)
{
  if ($o.ACCEPTANCE_CRITERIA -ne "" -and $o.ACCEPTANCE_CRITERIA -ne $null)
  {
    $Crit = $o.ACCEPTANCE_CRITERIA
  }
  else
  {
    $Crit = "NA"
  }
  if ($o.month_0 -ne "" -and $o.month_0 -ne $null)
  {
    $zero = $o.month_0
  }
  else
  {
    $zero = "."
  }
  if ($o.month_3 -ne "" -and $o.month_3 -ne $null)
  {
    $three = $o.month_3
  }
  else
  {
    $three = "."
  }
  if ($o.month_6 -ne "" -and $o.month_6 -ne $null)
```

```
    {
    $SIX = $o.month_6
    }
    else
    {
    $SIX = "."
    }
if ($o.month_9 -ne "" -and $o.month_9 -ne $null)
    {
    $nine = $o.month_9
    }
    else
    {
    $nine = "."
    }
if ($o.month_12 -ne "" -and $o.month_12 -ne $null)
    {
    $twelve = $o.month_12
    }
    else
    {
    $twelve = "."
    }
if ($o.month_18 -ne "" -and $o.month_18 -ne $null)
    {
    $eighteen = $o.month_18
    }
    else
    {
    $eighteen = "."
    }
```

```

    }
    if($isAccelerated -eq -1)
    {
        $markup = AddRow $markup @($o.METHOD , $o.TESTNAME, $Crit, $zero, $three, $SIX, $nine, $twelve, $eighteen) $rja $false
    }
    else
    {
        $markup = AddRow $markup @($o.METHOD , $o.TESTNAME, $Crit, $zero, $three, $SIX) $rja $false
    }
}

$markup += "</w:tbl>"
$markup += "<w:p ><w:r> <w:br w:type='page'/> </w:r> </w:p><w:p > <w:bookmarkStart w:id='0' w:name='_GoBack'/> <w:bookmarkEnd
w:id='0'/> </w:p>"
$doc = Add-OpenXmlContent -Document $args[0] -PartPath '/word/document.xml' `
-InsertionPoint '/w:document/w:body/w:p[last()]/w:styles/w:style[1]' `
-Content $markup

}

Write-Host "Generate Graphs Start"
Import-Module OpenXmlPowerTools

$filename = "C:\JMP\jmp2014Discovery\Example_files\" + "stabilitiygraphs" + ".docx"
$doc = CreateWordProcessingDocument $filename " "
AddLandscape $filename

$xml]$d = get-content "C:\JMP\jmp2014Discovery\Example_files\DrugData.xml"
foreach ($c in $d.SelectNodes("/studies/study"))
{

```



```
CreateDocForData $filename $c
```

```
}
```

```
Write-Host "Genereate Graphs End"
```