

The Excel spreadsheet used iterative calculations and lots of VBA routines to accomplish its task. However since CalculationXL supported all this there was no problem and the resulting compiled workbook could be accessed using Microsoft PowerShell Scripting technology as we describe below:

POWERSHELL SCRIPTING

All our technologies export their capability to Microsoft's PowerShell Scripting Environment.

Microsoft has created PowerShell as their overall administrative scripting tool.

PowerShell natively supports .NET components and is extensible.

Many third party vendors are exposing their functionality through PowerShell which increases the reach of any software that uses PowerShell.

In particular, we were able to integrate our ability to apply workbooks to datasets in PowerShell, with our ability to extract spatial data from various sources and process it using PowerShell.

ILLUSTRATING THE SCRIPT

```

1 Add-Pssnapin SpatialVUSnapin
2 Add-PsSnapin CalculationServerSnapin
3
4 $InputFilePath = 'mtot200812.dm'
5 $MaxRecords = 50000
6 $XLCPATH = 'GBE.xlc'
7 $InputRangeName = 'InputCells'
8 $OutputRangeName = 'OutputCells'
9 $OutputFilePath = 'GeoDMOutput.dm'
10 $TMPDMFILE = 'tmp.dm'
11 $sw = [System.Diagnostics.Stopwatch]::StartNew()

```

This section of script adds in the SpatialVU PowerShell snap-in and the CalculationServer PowerShell snap-in, making available all the features of these products to PowerShell.

It further defines some variables depicting the input data.

```

13 Get-DMFileData $InputFilePath -MaxRecords $MaxRecords |
14 where-Object {$_.ZONE -eq 1 -or $_.ZONE -eq 1.5 -or $_.ZONE -eq 2} |
15 Aggregate-Object IJK, "sum$$_.XINC*$__.YINC*$__.ZINC*$__.DENSITY;"TONS"" |
16 DeAggregate-Object -Summaries 1 |
17 Select-Columns IJK, "[MATH]::Truncate($_.IJK/2472)*30+78950+15;"XC"", "[MATH]::Truncate($_.IJK/24)%(30*30-48110+15);"YC"",
18 "$_.IJK%24*30+300+15;"ZC"", "30;"XINC"", "30;"YINC"", "$_.VOLUME/900;"ZINC"", "$_.TOTM/$_.VOLUME;"DENSITY"", TONS |
19 Select-Columns IJK, XC, YC, ZC, XINC, YINC, ZINC, "$_.DENSITY;"SG"", DENSITY, "975-$_.ZC;"DEPTH"", TONS |
20 Select-Columns IJK, XC, YC, ZC, XINC, YINC, ZINC, "0;"NI"", SG, "9;"RES_CAT"", "9;"PCAF"", "(0.023*$_.DEPTH+17.81+2.49)/20,3;"MCAF"", TONS, DENSITY |
21 New-DMFile $TMPDMFILE -ExtendedPrecision 1

```

This section of script filters only the null blocks in the zones of interest and aggregates them before saving them to an extended precision Datamine binary file.

```

23 Get-DMFileData $InputFilePath -MaxRecords $MaxRecords |
24 where-Object {$_.ZONE -eq 0} |
25 where-Object {$_.RES_CAT -eq 0 -or $_.RES_CAT -eq 1 -or $_.RES_CAT -eq 2} |
26 Aggregate-Object IJK, RES_CAT, "sum$$_.NI*$__.XINC*$__.YINC*$__.ZINC*$__.DENSITY;"TNI"", "sum$$_.XINC*$__.YINC*$__.ZINC*$__.DENSITY;"TOTM"" |
27 "$_.XINC*$__.YINC*$__.ZINC;"VOLUME"" |
28 DeAggregate-Object -Summaries 1 |
29 Select-Columns RES_CAT, "[MATH]::Truncate($_.IJK/2472)*30+78950+15;"XC"", "[MATH]::Truncate($_.IJK/24)%(30*30-48110+15);"YC"",
30 "$_.IJK%24*30+300+15;"ZC"", "30;"XINC"", "30;"YINC"", "$_.VOLUME/900;"ZINC"", IJK, "$_.TOTM/$_.VOLUME;"DENSITY"", "$_.TNI/$_.TOTM;"NI"" |
31 Select-Columns RES_CAT, XC, YC, ZC, "975-$_.ZC;"DEPTH"", XINC, YINC, ZINC, IJK, DENSITY, NI |
32 Apply-WorkBook $XLCPATH $InputRangeName $OutputRangeName
33 Select-Columns 0:RES_CAT, 1:XC, 2:YC, 3:ZC, 4:DEPTH, 5:XINC, 6:YINC, 7:ZINC, 8:IJK, 9:DENSITY, 10:INPUTNI, 11:CU, 12:TONS,
34 13:FEEDMASS, 14:FERONI, 15:FEEDCU, 16:PRODMASS, 17:PROONI, 18:PRODCU, 19:PD, 20:PT, 21:AG, 22:MCAF, 23:PCAF, 24:SALENIMASS, 25:NI, 26:MP, 27:SG |
35 Select-Columns IJK, XC, YC, ZC, XINC, YINC, ZINC, NI, SG, RES_CAT, PCAF, MCAF, TONS, DENSITY |
36 Add-DMFile $TMPDMFILE

```

