

Vault Advanced Administration

Kimberley Hendrix

Manager of Data Management | @kimhendrix80

Learning Objectives

- Learn how to analyze a vault for data issues
 - Useful reports
 - Useful data pulls using PowerShell
- Learn how to set up corrective actions to clean the vault using PowerShell
 - Some helpful scripts to clean the vault
 - Scripts to a clean Job Processor
- Learn how to create some weekly tasks for reporting
 - Jobs that create reports
- Learn how to utilize the job processor to help maintain a clean healthy Vault
 - Self Cleaning Vault

Analyzing your Vault

Three ways to gather the data in your Vault



Out of Box reports
Duplicate files



Custom Queries
Using Find File and the
reporting features



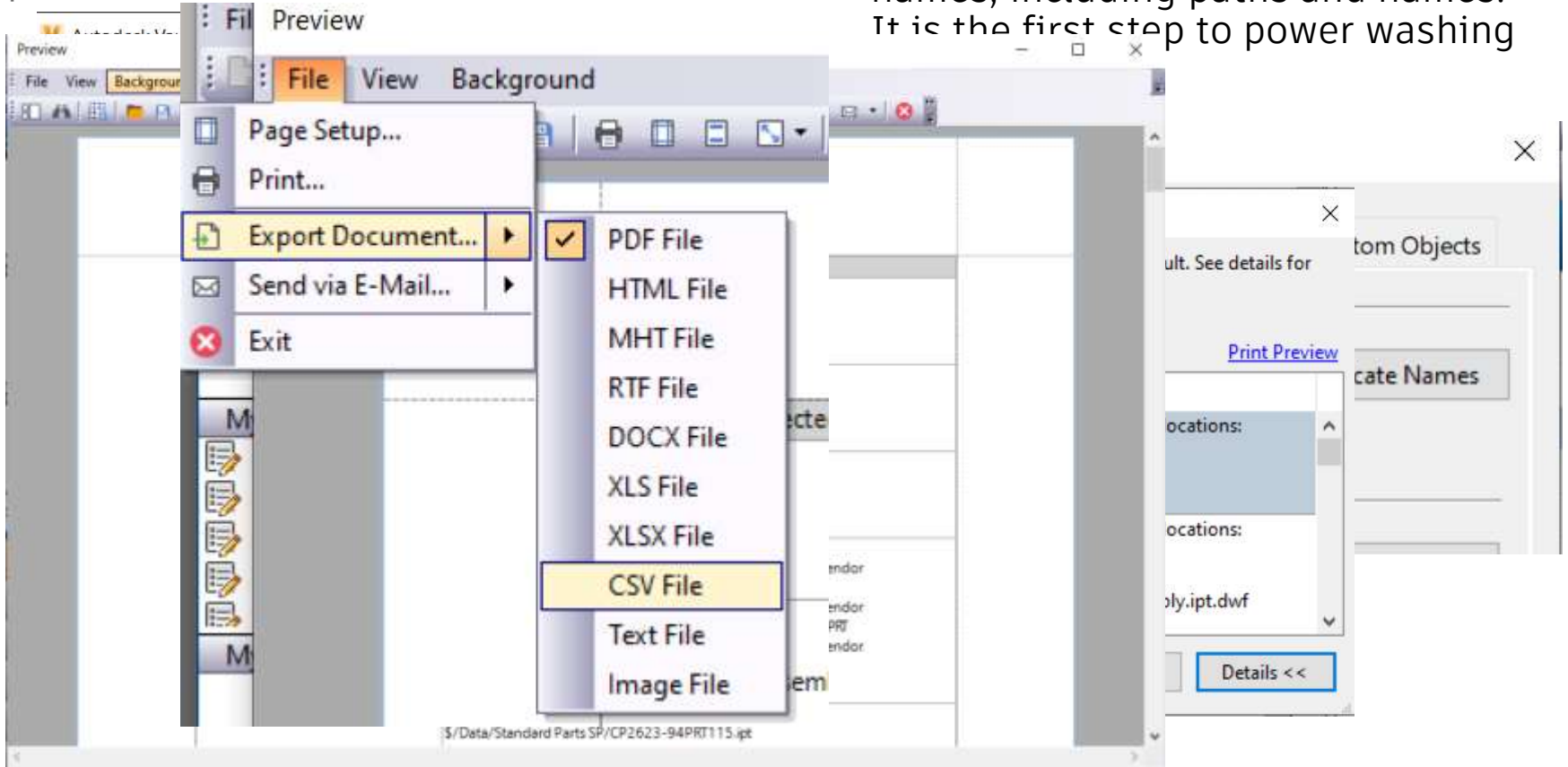
PowerShell scripts to get
more detailed
information



Out of the Box Reports

Duplicate Files

Duplicate files report will get you a list of files that have duplicate names, including paths and names. It is the first step to power washing





Custom Queries using Find File

What do we look for when cleaning a vault?

1. Files checked out
2. Change Order
3. Files with out of date
4. Orphaned Files

The screenshot shows the Autodesk Vault Professional 2022 interface. The 'Find' dialog box is open, and the 'Basic' tab is selected. The 'Look For' dropdown is set to 'Files'. The 'Look In' dropdown is set to 'Project Explorer (\$)'. The 'Search for' section shows 'Property: File Extension', 'Condition: contains', and 'Value: ipt'. The 'Find Now' button is highlighted. The results list shows files with extensions .ipt.

Annotations:

- 1: Points to the 'Find' dialog box title bar.
- 2: Points to the 'Property' dropdown menu.
- 3: Points to the 'Find Now' button.
- 4: Points to the search results list.

File Name	Revision	State	Created By
100010.ipt			Administrator
100013.ipt			Administrator
100016.ipt			Administrator



**Now that we have all
this data, what do we do
with it?**



PowerShell Scripts to manipulate data

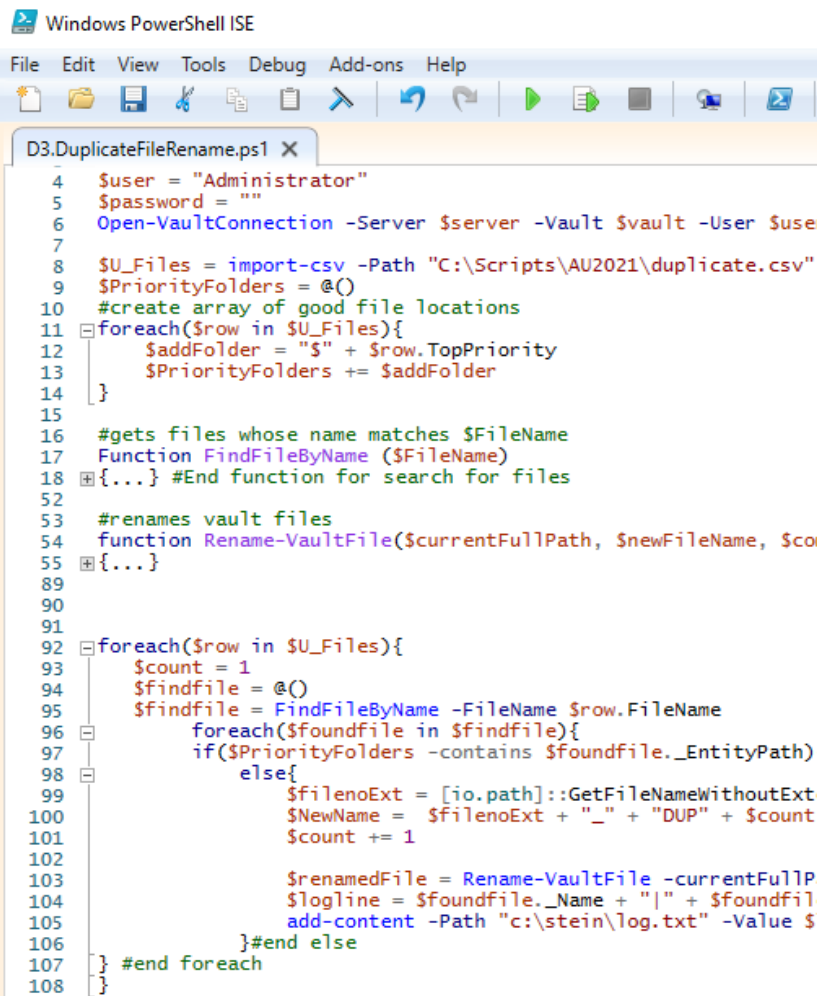
- Duplicate filenames, let's clean them all up at once.
- Moved up to Vault Workgroup or Professional?
 - Put the data in a position to take advantage of your new features.
- Loaded a lot of data, and have information from a different source?
 - Update all the properties at once reading from that source.

PowerShell

PowerShell along with coolOrange's powerVault extension provides a powerful platform to administer your Vault

The code snippet here will take the CSV file we created from the Duplicate File search in the beginning of class and execute a rename.

For each file in the CSV file, we see how many there are, then order them by date skip the newest file, and add a suffix of '_Dup#' to the end of the remaining files.



```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
D3.DuplicateFileRename.ps1 X
4 $user = "Administrator"
5 $password = ""
6 Open-VaultConnection -Server $server -Vault $vault -User $user
7
8 $U_Files = import-csv -Path "C:\Scripts\AU2021\duplicate.csv"
9 $PriorityFolders = @()
10 #create array of good file locations
11 foreach($row in $U_Files){
12     $addFolder = "$" + $row.TopPriority
13     $PriorityFolders += $addFolder
14 }
15
16 #gets files whose name matches $FileName
17 Function FindFileByName ($FileName)
18 {
19     [...] #End function for search for files
20 }
21
22 #renames vault files
23 function Rename-VaultFile($currentFullPath, $newFileName, $count)
24 {
25     [...]
26 }
27
28 foreach($row in $U_Files){
29     $count = 1
30     $findfile = @()
31     $findfile = FindFileByName -FileName $row.FileName
32     foreach($foundfile in $findfile){
33         if($PriorityFolders -contains $foundfile._EntityPath)
34         else{
35             $filenoExt = [io.path]::GetFileNameWithoutExt
36             $NewName = $filenoExt + "_" + "DUP" + $count
37             $count += 1
38
39             $renamedFile = Rename-VaultFile -currentFullPath $currentFullPath -newFileName $NewName
40             $logline = $foundfile._Name + "|" + $foundfile._EntityPath
41             add-content -Path "c:\stein\log.txt" -Value $logline
42         }
43     }
44 } #end foreach
45 }
```

Lifecycle Updates with PowerShell

You have made the move from Vault Basic to either Workgroup or Professional.

Now all your data needs assigned to appropriate Lifecycles and Categories, to take advantage of them.

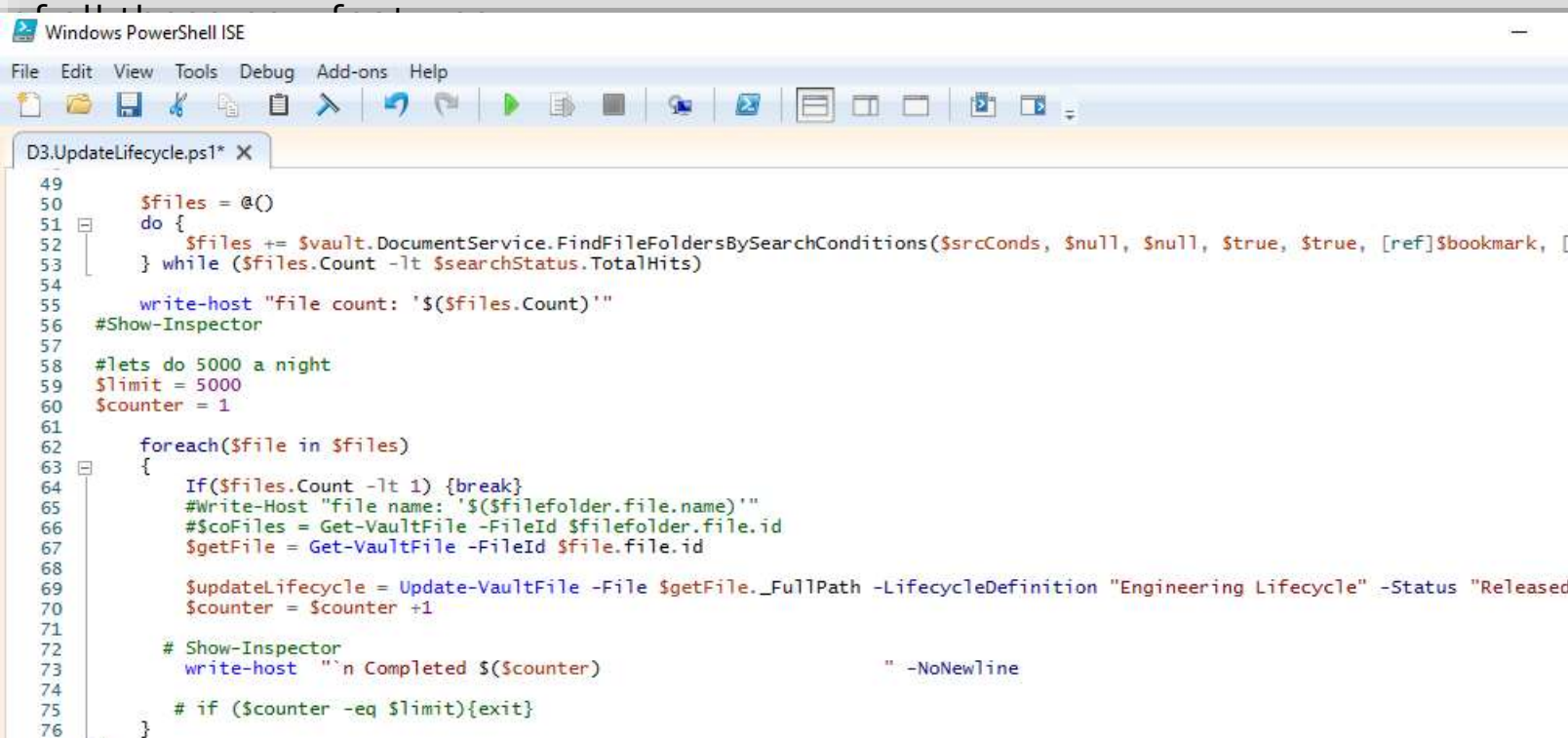
Manually

With Power

desired filt

Iterate thro

the approp



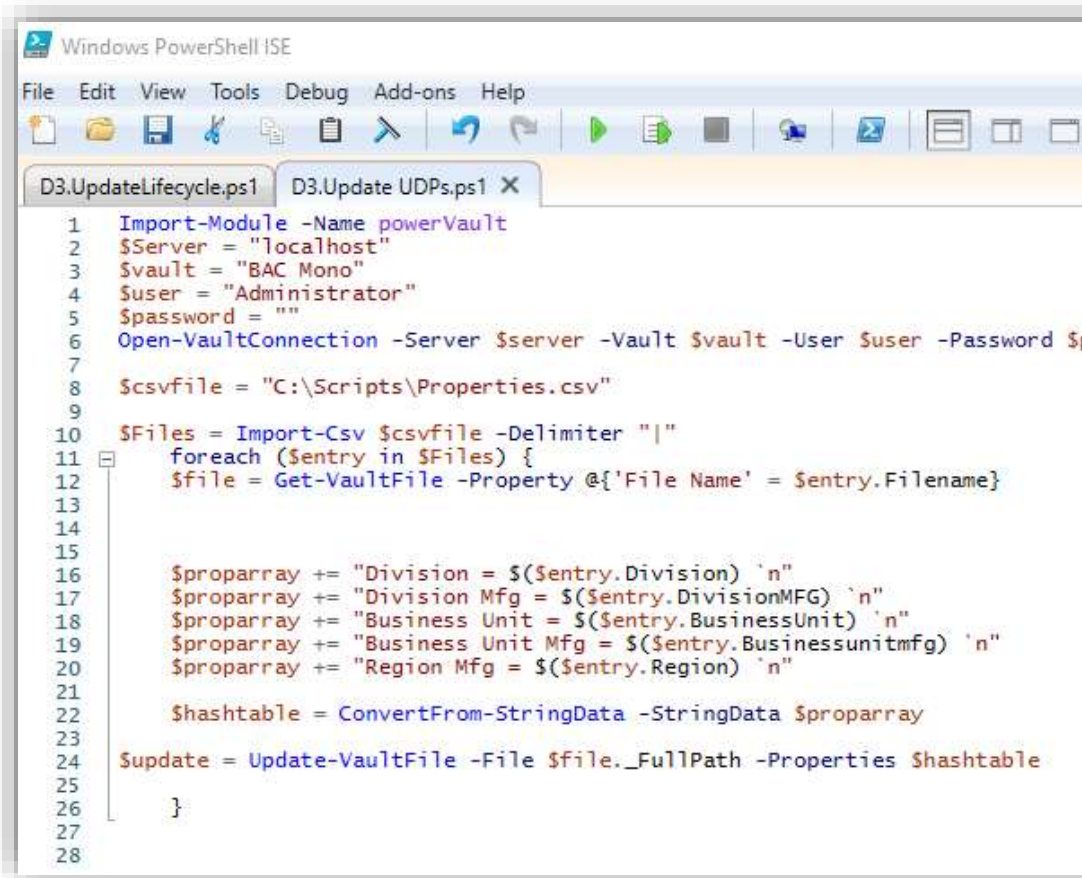
```
D3.UpdateLifecycle.ps1 X
49
50     $files = @()
51     do {
52         $files += $vault.DocumentService.FindFileFoldersBySearchConditions($srcConds, $null, $null, $true, $true, [ref]$bookmark, [ref]$searchStatus)
53     } while ($files.Count -lt $searchStatus.TotalHits)
54
55     write-host "file count: '$($files.Count)'"
56     #Show-Inspector
57
58     #lets do 5000 a night
59     $limit = 5000
60     $counter = 1
61
62     foreach($file in $files)
63     {
64         If($files.Count -lt 1) {break}
65         #Write-Host "file name: '$($filefolder.file.name)'"
66         #Get-Files = Get-VaultFile -FileId $filefolder.file.id
67         $getFile = Get-VaultFile -FileId $file.file.id
68
69         $updateLifecycle = Update-VaultFile -File $getFile._FullPath -LifecycleDefinition "Engineering Lifecycle" -Status "Released"
70         $counter = $counter +1
71
72         # Show-Inspector
73         write-host "`n Completed $($counter)" -NoNewline
74
75         # if ($counter -eq $limit){exit}
76     }
```

Data Dump from another source

Excel file or CSV file full of property values for your files.

The code snippet here will take that CSV file do a search in the vault for that file and update all the properties using the values from that source.

For each line in the CSV file, we search the vault for the file(s) and for each file found we execute an Update-VaultFile and populate a hashtable with property names and values.

A screenshot of the Windows PowerShell ISE interface. The title bar reads "Windows PowerShell ISE". The menu bar includes "File", "Edit", "View", "Tools", "Debug", "Add-ons", and "Help". The toolbar contains icons for file operations (New, Open, Save, Copy, Paste, Delete), navigation (Back, Forward), and execution (Run, Stop, Breakpoint). Two tabs are open: "D3.UpdateLifecycle.ps1" and "D3.Update UDPs.ps1". The active tab shows a PowerShell script with line numbers 1 through 28. The script imports the powerVault module, sets variables for server, vault, user, and password, opens a vault connection, and then iterates through a CSV file to update vault files with properties from the CSV.

```
1 Import-Module -Name powerVault
2 $Server = "localhost"
3 $vault = "BAC Mono"
4 $user = "Administrator"
5 $password = ""
6 Open-VaultConnection -Server $server -Vault $vault -User $user -Password $password
7
8 $csvfile = "C:\Scripts\Properties.csv"
9
10 $Files = Import-Csv $csvfile -Delimiter "|"
11 foreach ($entry in $Files) {
12     $file = Get-VaultFile -Property @{ 'File Name' = $entry.FileName }
13
14
15
16     $propparray += "Division = $($entry.Division) `n"
17     $propparray += "Division Mfg = $($entry.DivisionMfg) `n"
18     $propparray += "Business Unit = $($entry.BusinessUnit) `n"
19     $propparray += "Business Unit Mfg = $($entry.Businessunitmfg) `n"
20     $propparray += "Region Mfg = $($entry.Region) `n"
21
22     $hashtable = ConvertFrom-StringData -StringData $propparray
23
24     $update = Update-VaultFile -File $file._FullPath -Properties $hashtable
25
26 }
27
28
```



Server & Job Processor Cleaning



PowerShell Scripts to Power Wash

What else can we look for?

- Size of the SQL database?
- Drive space for filestore?
- Backups, did they run successfully?
- Self Cleaning Job Processor
 - Temp folder
 - Orphaned processes (Design Review, Inventor Server)

Cron job with the use of powerJobs*

powerJobs is a product of **coolOrange**

powerJobs allows for the use of Cron jobs, in the form of a settings file, named the same as the job, the settings file looks like this....

```
{
  "Trigger":
  {
    // This is a cron syntax expression. If you are not familiar with cron, please see: http://www.cronmaker.com/
    // Here are some common cron expressions:
    // every minute:    0 0/1 * 1/1 * ? *
    // every weekday at 8th am: 0 0 20 ? * MON,TUE,WED,THU,FRI *

    "TimeBased": "0 0 8 ? 1/1 SUN#3 *",

    // This is the name of the Vault you want to trigger the job
    "Vault": "BAC Mono",

    // And this two parameters are optional and self explaining:
    "Priority": 10,
    "Description": "This job is triggered Monthly on the 3rd Sunday at 8:00 am"
  }
}
```

Cron job with the use of powerJobs*

powerJobs is a product of **coolOrange**

The Cron job
information

- Size of t
- Size/Fre
- And Em

D3.ServerInfo.ps1 X

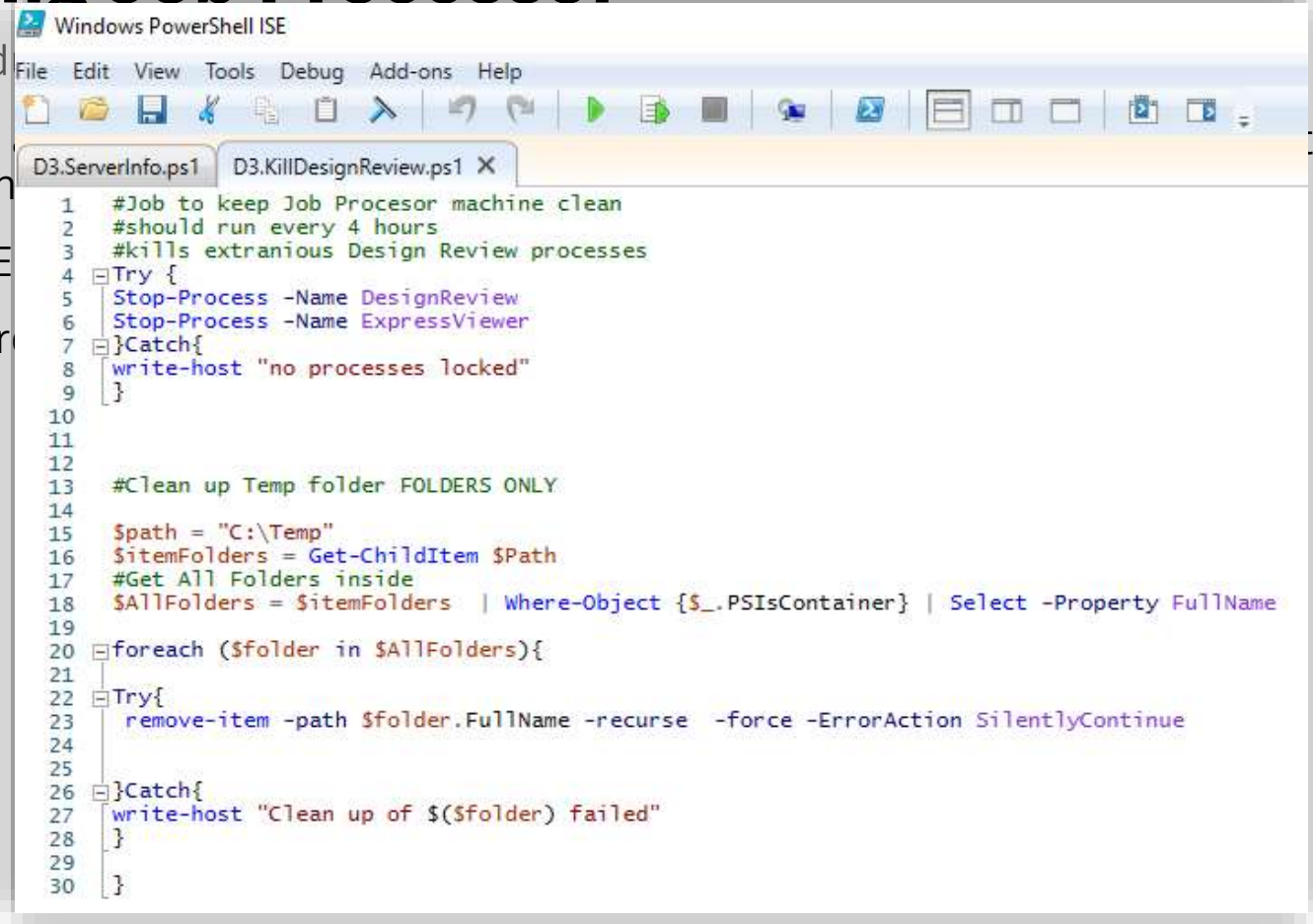
```
1 Import-Module -Name powerVault
2
3 $emailText = $null
4
5 $today = Get-Date -Format "dddd MM/dd/yyyy"
6 $CR = "`r`n"
7
8 $emailText = "Server Status for Date: " + $today + $CR + $CR
9
10 #Get the Database Size(s)
11 $dbsizes = Invoke-Sqlcmd -query "sp_databases" -Database 'BAC Mono' -serverInstance 'D3-0752\AUTODESKVAULT'
12 $bacmono = $dbsizes | Where-Object { $_.DATABASE_NAME -EQ 'BAC Mono'}
13
14 $DBSize = $bacmono.DATABASE_SIZE / 1GB
15
16 if ([int]$DBSize -lt 1){
17     $DBSize = "Less than 1GB"
18 }
19
20 $emailText = $emailText + "Database size for Vault BAC MONO is: " + $DBSize + $CR + $CR
21
22 #Get the Drive Size where the FileStore is stored
23
24 $disk = Get-WmiObject Win32_LogicalDisk -ComputerName D3-0752 -Filter "DeviceID='C:'" | Select-Object Size,FreeSpace
25
26 $dsize = $disk.Size / 1GB
27 $dfree = $disk.FreeSpace / 1GB
28 $emailText = $emailText + "Disk Size / Free Space on Server with Filestore is: " + [int]$dsize + "GB / " + [int]$dfree + "GB" + $CR
29
30 #send report to Admin
31 #BEGIN USER INPUT
32
33 $smtpServer="192.168.1.206" #SMTP Server
34 $from="kim.hendrix@teamd3.com" #Email "From"
35 $to="kim.hendrix@teamd3.com" #Email "To"
36
37 $emailSubject = "Vault Server Report for " + $today
38
39 Send-MailMessage -smtpServer $smtpServer -From $from -To $to -Subject $emailSubject -Body $emailText
40
```

Self Cleaning Job Processor

powerJobs is a prod

Using the Cron jobs
does some self main

- Cleans out the TE
- Kills orphaned pr



```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
D3.ServerInfo.ps1 D3.KillDesignReview.ps1 X
1 #Job to keep Job Processor machine clean
2 #should run every 4 hours
3 #kills extraneous Design Review processes
4 Try {
5     Stop-Process -Name DesignReview
6     Stop-Process -Name ExpressViewer
7 }Catch{
8     write-host "no processes locked"
9 }
10
11
12
13 #Clean up Temp folder FOLDERS ONLY
14
15 $path = "C:\Temp"
16 $itemFolders = Get-ChildItem $Path
17 #Get All Folders inside
18 $AllFolders = $itemFolders | Where-Object {$_.PSIsContainer} | Select -Property FullName
19
20 foreach ($folder in $AllFolders){
21
22     Try{
23         remove-item -path $folder.FullName -recurse -force -ErrorAction SilentlyContinue
24
25     }Catch{
26         write-host "Clean up of $($folder) failed"
27     }
28 }
29
30 }
```

References & Resources



References & Resources

Where can you find great ideas and examples?

- Autodesk AKN is a valuable resource of examples and tutorials
- **coolOrange** product **powerVault** is invaluable and is a great entry into the Vault API
<http://www.coolorange.com>
- Markus Koechl's GitHub is a wealth of information: <https://github.com/koechlm>
- D3 Technologies Web site and Blog: <http://www.teamd3.com>
- Code snippets from this class will be available on the AU website



Contact Information

KIMBERLEY HENDRIX
MANAGER OF DATA MANAGEMENT

D3 TECHNOLOGIES

Kim.Hendrix@TeamD3.com

LinkedIn

<https://www.linkedin.com/in/kimberleyhendrix/>

The background features several dark, metallic-looking geometric shapes, possibly cubes or prisms, arranged in a way that creates a sense of depth and perspective. These shapes are positioned around the central text, with some appearing to be in the foreground and others receding into the background. The lighting is dramatic, highlighting the edges and surfaces of the shapes against a solid black background.

AUTODESK UNIVERSITY

Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings, specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2021 Autodesk. All rights reserved.