Stream Stream

The Best Streaming Under One Brook

Hello Mr. Waterfield,

Our team identified the following file shares at these locations:

- /shares/config and /shares/Sharedir @ columbia 192.168.220.253
- /share @ Virgin 192.168.220.222
- /Public @ American 192.168.220.78
- /root/m @ Pattagansett 192.168.220.44

We have consolidated the files on these shares to a file share called “stream-share” located at Colorado 192.168.220.11. This file share is available for use by all Stream Domain Users. We chose to host the file share using the familiar SMB format that our team is familiar with, using their domain credentials for access. We chose Colorado as the location to host the file share because as it functions as a Domain Controller, it is rarely powered off during working hours.

We used the following script to check for duplicate files. The script requires Windows Powershell to execute and produces a report in csv format which contains the path of the removed files, the path to the saved file, and the MD5 hash of the file(s). We have left the original file shares in place as we will need time to brief the team about the new location. Please reach out if you run into any issues with the script or have any further questions.

Thank you,
Cameron Cisneros

Deduplication.ps1 (available on the stream-share file share):

```
# function to return md5 hash of each file, accepts path as a string
function GetFileHash {
    param (
        [Parameter(Position=0)]
        [string]$Path
    )
    (Get-FileHash -Algorithm MD5 -Path $Path).Hash

}
# get current working directory as string
$shareLocation = (Get-Location).Path

# choose to continue or not

Write-Host "Depuplication of files in $shareLocation, continue?(y/n)?"
$choice = Read-Host
if ($choice.ToLower() -ne 'y'){
    Write-Host "exiting..."
```

```powershell
    exit
}

# create hashtable of all files in format key=md5hash value=file path
$fileListTable =@{}
Get-ChildItem -Path $shareLocation -Recurse -File | ForEach-Object{
    $md5Hash = GetFileHash -Path $_.FullName

    if ($fileListTable.ContainsKey($md5Hash)){
        $fileListTable[$md5Hash] += $_.FullName
    } else {
        $fileListTable[$md5Hash] =@($_.FullName)
    }
}


# create file report as  PSCustomObject
$fileReport = @()
foreach ($hash in $fileListTable.Keys){
    $files = $fileListTable[$hash]
    if ($files.Count -gt 1 ){
        for ($i=1; $i -lt $files.Count; $i++){
            $savedFile = $files[0]
            $removedFile =$files[$i]
            Remove-item -Force $files[$i]
            $fileReport += [PSCustomObject]@{
                "Removed File" = $removedFile
                "Saved File" = $savedFile
                "File Hash" = $hash
            }
        }
    }
}

# export fileReport to csv with custom properties
$fileReport | Select-Object "Removed File","Saved File", "File Hash" | Export-Csv
-Path "output.csv" -NoTypeInformation
if ($fileReport.Length -eq 0){
    Write-Host "no duplicates found"
    Remove-Item -Force "output.csv"
}else {
    Write-Host "Report saved as output.csv"
}
```