SPMT Home Drive to OneDrive Import

Microsoft have a tool called SharePoint Migration Tool (SPMT) that can be used to migrate SharePoint sites and file shares to SharePoint online and OneDrive.

https://docs.microsoft.com/en-us/sharepointmigration/introducing-the-sharepoint-migration-tool

You can use it to migrate home drives to OneDrive, but the MS documentation is somewhat lacking on this bit of its functionality. You need to create a CSV file mapping each users home drive folder to a OneDrive URL, but they don't give any examples of the format you require for OneDrive. Use the following procedure to generate the CSV file.

Export Users and Groups

If the tenant is not synchronised with AD and you need to export groups and memberships to Azure, use this script:

https://github.com/ScottKnights/Azure/blob/main/Export-ADGroupsToAzureAD.ps1

Install required PowerShell modules

- Azure AD: **https://docs.microsoft.com/en-us/powershell/module/azuread/?view=azureadps-2.0**
- SharePoint Online: **https://docs.microsoft.com/en-us/powershell/module/sharepoint-online/?view=sharepoint-ps**
- Active Directory: **https://docs.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2019-ps**

Pre-Provision OneDrive

Users need to have their OneDrive provisioned before you attempt to migrate the files.

Connect to Azure AD and generate a text file containing UPNs:

```
$UPNFile=".\UPNFile.csv"
Connect-AzureAD # Supply required credentials
(Get-AzureADuser|ft userprincipalname -HideTableHeaders|out-string).trim()|out-file -FilePath $UPNFile -Force -Encoding ascii
Set-Content -Path $UPNFile -Value (get-content -Path $UPNFile | Select-String -Pattern 'onmicrosoft.com' -NotMatch)
# Get tenant name for next steps
$tenant=((Get-AzureADTenantDetail).verifieddomains |Where-Object {$_.name -like "*onmicrosoft.com"}).name.replace(".onmicrosoft.com","")
```

Review the file and delete any accounts you don't need.

Connect to the SharePoint Online Management Shell and provision OneDrive for each user and export a list of the provisioned OneDrive paths:

```
$tenanturl="https://$tenant-admin.sharepoint.com"
connect-sposervice -url $tenanturl # Supply required credentials
$users = Get-Content -path $UPNFile

# Request-SPOPersonalSite can only process up to 200 logins at a time
while ($users.count -gt 200) {
    Request-SPOPersonalSite -UserEmails ($users | select -first 200)
    $users=$users|select-object -skip 200
}
Request-SPOPersonalSite -UserEmails $users

# Wait a bit for OneDrive to be provisioned. As with many things in Azure, it runs on its own time frame.
# The CMDLet documentation states:
# "If you're pre-provisioning OneDrive for a large number of users, it might take multiple days for the OneDrive locations to be created."

# Export list of generated OneDrive Paths
$ODPaths=".\ODPaths.txt"
Get-SPOSite -IncludePersonalSite $true -Limit all -Filter "Url -like '-my.sharepoint.com/personal/'" | select Url| out-file -filepath
$ODPaths
```

Generate the SPMT Mapping file

Use the following script to generate the SPMT mapping file. Specify the tenant name, OneDrive suffix and path to the folder containing the home folders. Supply the path to the ODPaths file generated in the last step. As the paths are generated, they will be compared to the paths exported from SharePoint to ensure that they exist. Investigate any that don't exist. It will often meant that the AD username doesn't match the Azure username but there could be other reasons. Fixup or delete paths in the csv file as required.

```
# Generate an SPMT mapping file for home drives

# ================================================================================
#region Constants
# ================================================================================
```

```
# Tenant name as in the onmicrosoft.com domain (@TENANTNAME.onmicrosoft.com)
$tenant="TENANTNAME"

# OneDrive domain suffix. This should be their UPN domain name with a leading _ and with all . characters replace with _
$odsuffix="_TENANT_COM"

# Path to home folders
$homepath="\\SERVER\PATH"

# Path to the file containing the list of exported OneDrive paths
$ODPathFile=".\ODPaths.txt"

# Output file path
$outfile=".\homedrives.csv"

# Log file path and encoding
$logfile=".\spmthome.log"
$encoding="ASCII"

#endregion Constants

# ============================================================================
#region Functions
# ============================================================================
function out-log {
    param ( [Parameter(ValueFromPipeline = $true)]
        [PSCustomObject[]]
        $iobjects
    )

    process {
        foreach ($iobject in $iobjects)
            {
            $iobject|add-content $logfile -passthru -Encoding $encoding |write-output
            }
    }
}
#endregion Functions
```

```
# ================================================================================
#region Execute
# ================================================================================
# Check that we can create the log file
$start=get-date
try {
    set-Content -Path $logfile -encoding $encoding -Value "Start run at $start" -erroraction stop
} catch {
    write-output "Cannot create log file $logfile. Check path and permissions. Exiting."
    Return
}

# Delete the output file if it already exists
if (test-path -path $outfile) {
    remove-item $outfile -force
}

# Get the domain netbiosname. Exit on failure.
try {
    $domain=(get-addomain).netbiosname
} catch {
    "Error. AD powershell module is probably missing. Exiting."
    return
}

# Add trailing \ to the homepath if it is missing
if (-not $homepath.endswith("\")) {
    $homepath="$homepath\"
}

# Exit if home path is invalid
if (-not (test-path -path $homepath -erroraction silentlycontinue)) {
    "Homepath is invalid. Exiting."
    return
}

# Exit if $ODPathFile is invalid
if (-not (test-path -path $ODPathFile -erroraction silentlycontinue)) {
    "OneDrive path file is invalid. Exiting."
```

```
    return
}

# Get list of OneDrive paths from the OneDrive path file
$ODPaths=get-content -path $ODPathFile

# Get home folders from the home share
$homefolders=get-childitem $homepath -directory

foreach ($homefolder in $homefolders) {
    $hftemp=$homefolder.name
    # Strip the domain name from the end of the folder name if it exists
    if ($hftemp.endswith(".$domain")) {
        $hfname=$hftemp -replace ".$domain",""
    } else {
        $hfname=$hftemp
    }
    # Test if the user exists, write their mapping to the output file if they do
    try {
        if (get-aduser $hfname) {
            $modname=$hfname.replace(".","_")
            $ODTarget="https://$tenant-my.sharepoint.com/personal/$modname$odsuffix"
            # Check if the generated OneDrive path exists in the list exported from SharePoint
            if ($ODPaths -match $ODTarget) {
                "$homepath$hftemp,,,$ODTarget,Documents,"|out-file -filepath $outfile -append -encoding utf8
            } else {
                "$ODTarget not found in list of ODPaths"|out-log
            }
        }
    } catch {
        "$hfname does not exist in AD"|out-log
    }
}
#endregion Execute
```

The resulting CSV file should look like this. Note the multiple commas after the first field and the trailing comma. There must not be any field headers.

```
\\server\home$\user1.ad,,,https://tenant-my.sharepoint.com/personal/user1_tenant_com,Documents,
\\server\home$\user2.ad,,,https://tenant-my.sharepoint.com/personal/user2_tenant_com,Documents,
\\server\home$\user3.ad,,,https://tenant-my.sharepoint.com/personal/user3_tenant_com,Documents,
\\server\home$\user4.ad,,,https://tenant-my.sharepoint.com/personal/user4_tenant_com,Documents,
```