

Automation Workflow for:
One Drive Prep Script.ps1

.NOTES
Script Name : Get-DeletedTeamsSites.ps1
Author : christopher.brennan@kirkland.com
Date Created : 2025-12-12
Last Modified : 2025-12-12
This Version : 1.1.0

Run the Script:

Run the script with appropriate parameters & permissions to modify users OneDrive folders.



```
### Run the Script with Parameters:  
### -----  
### Example:  
.\\OneDrivePrepScript.ps1 -TenantUrl "https://tenantID.sharepoint.us"  
-LogPath "c:\\tmp\\logs"  
-DataFile "c:\\tmp\\UserData.csv"
```

Script Parameters:

You could manually edit the script parameters to match your environment.



```
### Script Parameters:  
### -----  
param (  
    ### Project Name/Client:  
    $Project = "Test Dev Script"  
,  
    ### Tenant URL:  
    $TenantUrl = "https://$tenantID-my.sharepoint.us"  
,  
    ### Path to the .csv file containing error log file:  
    $LogPath = ".\\Projects\\OneDrive Prep Script"  
,  
    ### Path to the .csv file containing user data:  
    $DataFile = "UserData.csv")
```

Import User Data:

Import User .csv file containing user UPN.



```
### Import User Data:  
### -----  
try {  
    $users = Import-Csv -Path $DataFile  
}  
catch {  
    Write-RtxLog -Message $_.Exception.Message -Level Error  
    throw }
```

Script Logic:

```
Foreach ($user in $users){  
    1. Connect User Site  
    2. Create Folders  
    3. Get Items  
    4. Move Files  
    5. Log & Report  
    6. Disconnect User  
}
```

Out of scope options:
- Email User Report
- Code Sign Script

Loop through All User



```
### Connect to User's site:  
### -----  
### Replace '@' and '.' with '-'  
$userUpn = $($user.Upn).Replace("@", "-").Replace(".", "-")  
### User's OneDrive Relative URL  
$userRelativeUrl = "$tenantUrl/personal/$userUpn"  
$pnpSiteConnection = Connect-PnPOnline -Url $userRelativeUrl  
-ReturnConnection -UseWebLogin
```



"DO NOT MIGRATE",
"APPROVED TO MIGRATE"

```
### Create Folders:  
### -----  
@("DO NOT MIGRATE", "APPROVED TO MIGRATE", "TestFolder") | ForEach-Object {  
    Resolve-PnPFolder -siteRelativePath "Documents/$_" | Out-Null }
```



```
### Get OneDrive Items:  
### -----  
$allItems = Get-PnPListItem -List 'documents' | Where-Object {  
    $($_.FieldValues.Title) -ne "DO NOT MIGRATE"  
    -and $($_.FieldValues.Title) -ne "APPROVED TO MIGRATE"  
    -and $($_.FieldValues.Title) -ne "TestFolder"}
```

Move Items:

Move Item:



```
### Move Items:  
### -----  
$movePnPFileParams = @{  
    ServerRelativeUrl = $item.FileRef  
    TargetServerRelativeLibrary = $item.FileRef  
    AllowItemNameMismatch = $false  
    IgnoreVersionHistory = $false  
    Connection = $pnpSiteConnection  
    Force = $true  
    ErrorAction = 'Stop'
```

Log & Report:



Loop through All User Items



Summary Log/Report/Errors

Email User Report

Out of Scope

Disconnect User



Summary Report

Transcript

Out of Scope

Email Summary Reports



```
.NOTES  
Script Name : Get-DeletedTeamsSites.ps1  
Author : christopher.brennan@kirkland.com  
Date Created : 2025-12-12  
Last Modified : 2025-12-12  
This Version : 1.1.0
```