**Office 365 - Migrating users onto OneDrive for Business**

**Introduction**:

In this part of migrating to Office 365 we’ll have to help users move their data. In a perfect world users would follow instructions and move it themselves but this world isn’t perfect so here’s how we can automatically move their data.

**Preparation and Prerequisites:**

Before we can move the files we need to prepare a some things. There are some GPO settings we can configure, review the OneDrive admin settings, we need to inform our users and we have to deploy the script

**Limitations**

Another potential issue are invalid file names or to longs paths. Just like SharePoint has OneDrive for Business some file limitations:

* Invalid file or folder names
* Invalid characters ( ” \* : < > ? / \ |)
* Max 15Gb file size
* 400 character path limit

Read the full list of limitations here.

<https://support.office.com/en-us/article/invalid-file-names-and-file-types-in-onedrive-onedrive-for-business-and-sharepoint-64883a5d-228e-48f5-b3d2-eb39e07630fa?ui=en-US&rs=en-US&ad=US>

**Checking OneDrive for Business Admin Center**

It’s always good to check the Office 365 Admin Center before you are running out this kind of migrations. New settings are added regularly and you might want to limit for example the sharing options for OneDrive for Business.

Make sure the settings are inline with your business security policies.

**OneDrive for Business Known Folder Move**

Moving the known folders (Desktop, Documents and Pictures) to OneDrive is something you really should enable. It only requires two GPO settings to set it up and it will roll out by itself. Your users can work and store the files as they are used to do, but they are safely backupped in the cloud.

**Get your Office 365 Tenant ID**

Before we can configure the GPO’s we need to get our tenant ID. With PowerShell we can get it with only one command after logging into Azure Run the following:

* Connect - azuread
* get-azureadtenantdetail

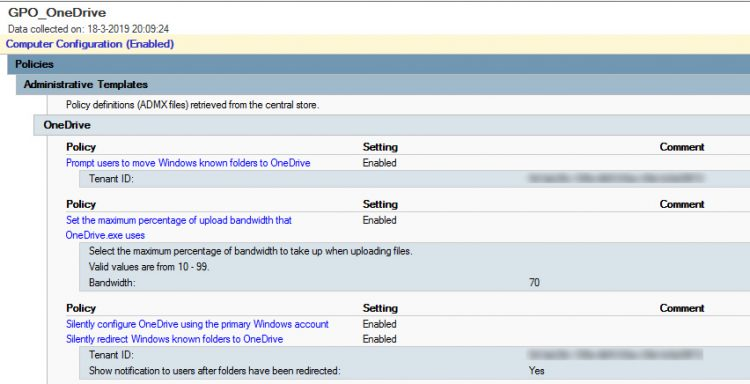
**Migration Steps**

**Step 1. Configuring the GPO’s**

There are a couple of GPO’s we need to configure. Keep in mind if you enable this for the entire organization there will be a heavy spike in network traffic because of all the OneDrive syncronization. You should enable this to a group at a time. You can use a temporary security group to target the GPO

Create a new GPO and navigate to the following polices:

Computer configuration\Policies\Administrative Templates\OneDrive



1. Enable the policy Silently configure OneDrive using the primary Windows account. This will save the user an extra login when they setup OneDrive the first-time.
2. Limit the upload bandwidth, this way one user can’t slow down the whole internet connection. Set the policy Set the maximum percentage of upload bandwidth that OneDrive.exe uses to something like 70.
3. To move the known folder, enable the policy Silently redirect Windows Known folder to OneDrive and enter the tenant ID.
4. (Optional) Enable user notification
5. Enable the Prompt user to move Windows Known folder to OneDrivepolicy to support any legacy OneDrive clients.

**Step 2 Moving Users Home Folder**

In this step we’’ll be using PowerShell Script to check and move data to OneDrive. Here are the steps:

1. Download the OneDriveLib.dll from Github. Put this file somewhere like the netlogon folder where everyone can get it. This project is Free and OpenSource:

https://github.com/rodneyviana/ODSyncService

1. Below is the breakdown to code of the script. You can also download this on my own github. <https://github.com/mrallentam/docs/moveonedrive.ps1>
2. First thing we need to do is import the module

Import-Module \\contoso.local\netlogon\OnedriveLib.dll

1. Next, we need to get the path to the users OneDrive folder. The default location for this folder is the users profile folder.

$path = $env:OneDriveCommercial

# Alternative:

# $path = $env:USERPROFILE + '\OneDrive

1. To check if we have the right folder and that OneDrive is connected we can get the OneDrive status:

#Get OneDrive status

$ODStatus = Get-ODStatus -ByPath $path

# or in short:

$ODStatus = Get-ODStatus -ByPath $env:OneDriveCommercial

1. If it’s connected it will return either ‘UpToDate’ or ‘Syncing’. So we check the result and use robocopy to copy the files.

if ($ODStatus = 'UpToDate' -or 'Syncing') {

#OneDrive is connected

write-host 'OneDrive connected and found'

robocopy $env:HOMESHARE $env:OneDriveCommercial /E /SEC

new-item $env:HOMESHARE -name '\_FILES COPIED TO ONEDRIVE.txt' -ItemType

'file' -Value 'Files Copied}

Make sure you double check this into your PowerShell scripting environment (ISE or Visual Studio Code) You have to run this script as the user, so you should put it in as a login script or push it out through SCCM, or which ever method your organization uses.