PAST THE BELOW SCRIPT INTO POWERSHELL

#-----------------------------------------------------------------------------------

#--This script will remove existing / expired ClientSecret tokens for

#--Template Manager and install new ones with an expiry date of 6 years from time of running

#--You will need to add your SharePoint administrator credentials when prompted

#-----------------------------------------------------------------------------------

Clear-Host

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted

#Checking if MSOLService is installed, if not install it

if (-Not (Get-Module -ListAvailable -Name MSOnline)) {

 Write-Host "MSOnline module not installed. Installing MSOnline"

 Install-Module MSOnline

 }

#Checking if SPOService is installed, if not install it

if (-Not (Get-Module -ListAvailable -Name Microsoft.Online.SharePoint.Powershell)) {

 Write-Host "SPOnline module not installed. Installing SPOnline"

 Install-Module Microsoft.Online.SharePoint.Powershell

 }

Import-Module Microsoft.Online.SharePoint.Powershell -DisableNameChecking

#the client id of Template Manager application

$clientId = '894367b3-6a82-40f4-9d9b-b9b85ed5f96a'

#the new client secret to be installed

$newClientSecret1 = 'i4BaLQ2jRj9+VjYVE5EWDPS9oh4HvcVgSNRIVsVvJY4='

$cred = Get-Credential

Write('Connecting to SharePoint tenancy...')

Connect-MsolService -Credential $cred

Write('Preparing to retrieve expiring access keys to One Practice Template Manager and remove them...')

$keys = Get-MsolServicePrincipalCredential -AppPrincipalId $clientId -ReturnKeyValues $true | Where-Object { ($\_.Type -ne "Other") -and ($\_.Type -ne "Asymmetric") }

foreach($kk in $keys){

 Write ($kk)

 Remove-MsolServicePrincipalCredential -AppPrincipalId $clientId -KeyIds @($kk.KeyId)

}

Write('Checking all keys removed...')

$keysCheck = Get-MsolServicePrincipalCredential -AppPrincipalId $clientId -ReturnKeyValues $true | Where-Object { ($\_.Type -ne "Other") -and ($\_.Type -ne "Asymmetric") }

Write('Installing new keys with expiry of 6 years...')

$dtStart = [System.DateTime]::Now

$dtEnd = $dtStart.AddYears(6)

New-MsolServicePrincipalCredential -AppPrincipalId $clientId -Type Symmetric -Usage Sign -Value $newClientSecret1 -StartDate $dtStart –EndDate $dtEnd

New-MsolServicePrincipalCredential -AppPrincipalId $clientId -Type Symmetric -Usage Verify -Value $newClientSecret1 -StartDate $dtStart –EndDate $dtEnd

New-MsolServicePrincipalCredential -AppPrincipalId $clientId -Type Password -Usage Verify -Value $newClientSecret1 -StartDate $dtStart –EndDate $dtEnd

Write('Getting keys again after adding new...')

$keys2 = Get-MsolServicePrincipalCredential -AppPrincipalId $clientId -TenantId $theTenantId -ReturnKeyValues $true | Where-Object { ($\_.Type -ne "Other") -and ($\_.Type -ne "Asymmetric") }

foreach($kk in $keys2){

 Write ($kk)

}

Write('REMEMBER IN SOME CASES YOU MAY NEED TO WAIT UP TO 12 HOURS FOR CHANGES TO PROPAGATE!!')