Oauth2 Refresh token

Pre-requirements

- Azure AD App registration (Need App ID)
- Secret for App registration
- Tenant ID
- User with enough permissions for the work that needs to be done.

Create Powershell function.

```
function Get-CurrentUserProfile
{
  Param
  (
    [parameter(Mandatory=$true,
    ValueFromPipeline=$true)]
    [PSCredential]
    $credential,
    [parameter(Mandatory=$true)]
    [string]
    $scopes,
    [parameter(Mandatory=$true)]
    [string]
    $redirectUrl,
    [switch]
    $displayTokens
  )

  $clientID = $credential.Username
  $clientSecret = $credential.GetNetworkCredential().Password
  #URL encode the secret
  $clientSecret = [System.Web.HttpUtility]::UrlEncode($clientSecret)

  #v2.0 authorize URL
  $authorizeUrl = "https://login.microsoftonline.com/common/oauth2/v2.0/authorize"

  #Permission scopes
  $requestUrl = $authorizeUrl + "?scope=$scopes"

  #Code grant, will receive a code that can be redeemed for a token
  $requestUrl += "&response_type=code"

  #Add your app's Application ID
  $requestUrl += "&client_id=$clientID"

  #Add your app's redirect URL
  $requestUrl += "&redirect_uri=$redirectUrl"

  #Options for response_mode are "query" or "form_post". We want the response
  #to include the data in the querystring
  $requestUrl += "&response_mode=query"

  Write-Host
  Write-Host "Copy the following URL and paste the following into your browser:"
  Write-Host
  Write-Host $requestUrl -ForegroundColor Cyan
  Write-Host
  Write-Host "Copy the code querystring value from the browser and paste it
below."
  Write-Host
  $code = Read-Host -Prompt "Enter the code"

  $body =
"client_id=$clientID&client_secret=$clientSecret&scope=$scopes&grant_type=authoriz
ation_code&code=$code&redirect_uri=$redirectUrl"
  #v2.0 token URL
  $tokenUrl = "https://login.microsoftonline.com/common/oauth2/v2.0/token"

  $response = Invoke-RestMethod -Method Post -Uri $tokenUrl -Headers @{"Content-
Type" = "application/x-www-form-urlencoded"} -Body $body

  if($displayTokens)
  {
    $response | select * | fl
  }

  #Pass the access_token in the Authorization header to the Microsoft Graph
  $token = $response.access_token
  Invoke-RestMethod -Method Get -Uri "https://graph.microsoft.com/v1.0/me" -
Headers @{"Authorization" = "bearer $token"}
}
```

Execute Powershell

```
Add-Type -AssemblyName System.Web

#offline_access:  Allows requesting refresh tokens
#openid:  Allows your app to sign the user in and receive an app-specific
identifier for the user
#profile: Allows your app access to all other basic information such as name,
preferred username, object ID, and others
#User.Read: Allows your app to read the current's user's profile
$scopes = "offline_access+openid+profile+User.
```

Set the redirect URL to the URL set in the App registration (needs to be the same)

```
#Redirects to this URL will show a 404 in your browser, but allows you to copy the returned code from the URL bar
#Must match a redirect URL for your registered application
$redirectURL = "http://localhost"
```

Next enter the following line of code. Enter the application/Client ID as username and the secret as password

```
$credential = Get-Credential -Message "Enter the client ID and client secret"
```

Execute the function:

```
$credential = Get-Credential -Message "Enter the client ID and client secret"
Get-CurrentUserProfile $credential -scopes $scopes -redirectUrl $redirectURL -displayTokens
```

The function will output an URL, which you can copy and paste into your browser, eg.

```
https://login.microsoftonline.com/common/oauth2/v2.0/authorize?scope=offline_access+openid+profile+User.Read&response_type=code&client_id=xxxxxda1-7c0f-xxxx-8238-xxxx10ea1xxx&redirect_uri=http://localhost&response_mode=query
```

You can now login with the username and password which has the correct permissions. After the login the browser will redirect to the redirectURL provided, probably with a http 404 error.

Copy and paste this URL in notepad, notepad++, vscode or any editor you would like.

Remove the url part, and the ?code=

At the end remove the &session_state= part until the end of the string. The string that is left can be pasted into your powershell sessions, which is prompting for this

```
Copy the code querystring value from the browser and paste it below.

Enter the code:
```

After pressing enter, you receive:

- access_token
- id_token
- refresh_token

```
token_type     : Bearer
scope          : Directory.AccessAsUser.All User.Read profile openid email
expires_in     : 3599
ext_expires_in : 3599
access_token   :
eyJ0eXAiOiJKV1QiLCJub25jZSI6ImVxR00yWXQxdTBKbEhCeVVtMWxCYURQaGxHRHVvZGNUVUhnZDliUj
hWZVUiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZyIsImtpZCI6
Im5PbzNaRHJPRFhFSzFqS1doWHN...
refresh_token  :
0.AAAAfz7QNAP6KkOkjVO44ISjyKEtW3wPfAxIgjhQ2xDqHZl5AFc.AgABAAAAAAD--
DLA3VO7QrddgJg7WevrAgDs_wQA9P9bTEA8BbRmJrHzTNy6UkF9CUl-SQjSvsGbbYNMRmkhpPgxH888nE-
UESrrSx00tbWV5WFGreBiCf0TyurZr_UWnG3hTYI1Z.....
id_token       :
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZy
J9.eyJhdWQiOiI3YzViMmRhMS03YzBmLTQ4MGMtODIzOC01MGRiMTBlYTFkOTkiLCJpc3MiOiJodHRwczo
vL2xvZ2luLm1pY3Jvc29mdG9ubG....
```

Use it

This information can be used to access information.

eg.

```
$clientId = $env:ccmdataClientId
$tenantId = 'tenantid'
$clientSecret = $env:ccmdataClientSecret
$partnerAccessTokenUri =
"https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token"

$refreshToken = (refresh token)

$params = @{
    grant_type = "refresh_token";
    client_id = $clientId;
    scope = "https://api.partnercenter.microsoft.com/user_impersonation";
    refresh_token = $refreshToken;
    client_secret = $clientSecret
}

$auth = Invoke-RestMethod -Uri $partnerAccessTokenUri -Method POST -Body $params -
ContentType 'application/x-www-form-urlencoded'
#$auth.access_token
$auth

$token = $auth.access_token


$response = Invoke-RestMethod -Uri
"https://api.partnercenter.microsoft.com/v1.0/customers/customerID/orders" -
Headers @{Authorization = "Bearer " + $token}
```

```
$responseObject = $response.items | where-object {$_.status -ne "cancelled" -and
$_.lineitems.links.sku -like "*DG7GMGF0DVT9*"}
```

As long as the refresh token isn't expired, you can update it using:

```
$clientId = <applicationID>
$tenantId = <tenantid>
$clientSecret = <secret>
$partnerAccessTokenUri =
"https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token"

$refreshToken = <current token>

$params = @{
    grant_type = "refresh_token";
    client_id = $clientId;
    scope = "https://api.partnercenter.microsoft.com/user_impersonation";
    refresh_token = $refreshToken;
    client_secret = $clientSecret
}


$auth = Invoke-RestMethod -Uri $partnerAccessTokenUri -Method POST -Body $params -
ContentType 'application/x-www-form-urlencoded'
```

`$auth.refresh_token` contains your new refresh token.