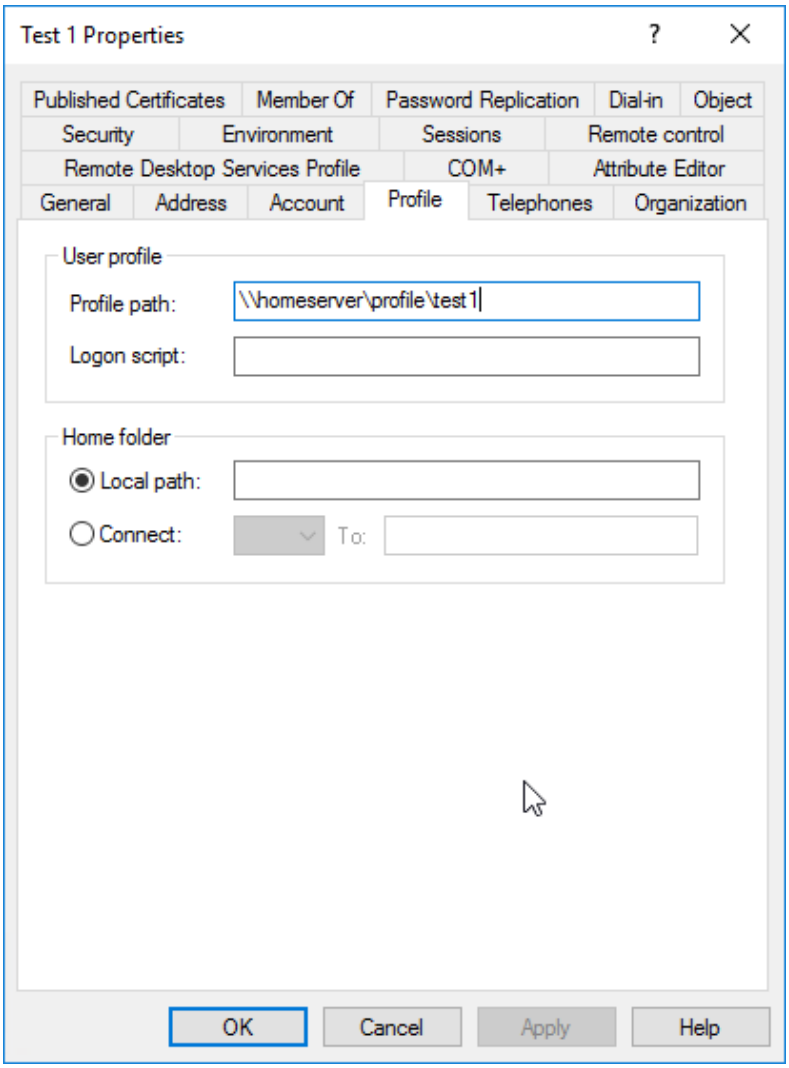
Modify the users' profile path in Active Directory



Profile path in AD.

The customer I am working for asked me to write a script that removes the profile path from the users' profile in Active Directory. One of the requirements was to do it on a batch-by-batch basis. Thus not a big bang.

I decided to write a PowerShell script and use a .csv file as the input. The results should be written to a log file.

I created the following script:

```
<#
.NOTES
================================================================
================================
Created with:     Windows PowerShell ISE
Created on:       03-August-2018
Created by:       Willem-Jan Vroom
Organization:
Filename:         RemoveProfilePathFromUser (v02).ps1
================================================================
================================
.DESCRIPTION:

This script removes the profile path from the users' profile
in Active Directory.

.USAGE:

Create a CSV file with the following layout:

Userid
test1
test2

And mention the CSV file as a parameter -
FileWithUseridsInCSVFormat.

.VERSION HISTORY:
 v0.1:
   * Initial version.

 v.0.2:
   * Option -FileWithUseridsInCSVFormat has been added.
   * Added help text by the options.

 v.0.3:
   * Minor cosmetic changes.

#>

param
```

```
(
[Parameter(HelpMessage="CSV Filename that contains all the
userids that should be modified. If not mentioned than the
script name is used.")]
[String] $FileWithUseridsInCSVFormat=""
)

#
================================================================
================================
# Function block
#
================================================================
================================

Function Write-EntryToResultsFile
{

<#
.NOTES
================================================================
================================
Created with:     Windows PowerShell ISE
Created on:       03-August-2018
Created by:       Willem-Jan Vroom
Organization:
Functionname:     Write-EntryToResultsFile
================================================================
================================
.DESCRIPTION:

This function adds the success or failure information to the
array that contains the log
information.

#>
param
 (
  $strUserid,
  $ErrorMessage = ""
 )
```

```
 $Record            = [ordered] @{"Username" = "";"Error"= ""}
 $Record."Username" = $strUserid
 $Record."Error"    = $ErrorMessage
 $objRecord         = New-Object PSObject -Property $Record
 $Global:arrTable   += $objRecord
}

Function Process-User

<#
.NOTES
================================================================
================================
Created with:     Windows PowerShell ISE
Created on:       03-August-2018
Created by:       Willem-Jan Vroom
Organization:
Functionname:     Process-User
================================================================
================================
.DESCRIPTION:

Determine if the user can be modified.

#>
{
param
(
 $strUserid
)
    Try
       {
        $UN = Get-ADUser -Identity $strUserid
          Remove-ProfilePathFromUserProfileInAD -strUserid
$strUserid
       }
        Catch
       {
         Write-EntryToResultsFile -strUserid $strUserid -
ErrorMessage $_.Exception.Message
        Continue
```

```
       }

}


Function Remove-ProfilePathFromUserProfileInAD
{

<#
.NOTES
===========================================================================
================================
Created with:     Windows PowerShell ISE
Created on:       03-August-2018
Created by:       Willem-Jan Vroom
Organization:
Functionname:     Write-EntryToResultsFile
===========================================================================
================================
.DESCRIPTION:

This function adds the success or failure information to the
array that contains the log
information.

#>

  param
   (
    $strUserid
   )
   Try
      {
       Set-ADUser -Identity $strUserid -Clear profilePath
       Write-EntryToResultsFile -strUserid $strUserid
      }
       Catch
      {
         Write-EntryToResultsFile -strUserid $strUserid -
ErrorMessage $_.Exception.Message
       Continue
```

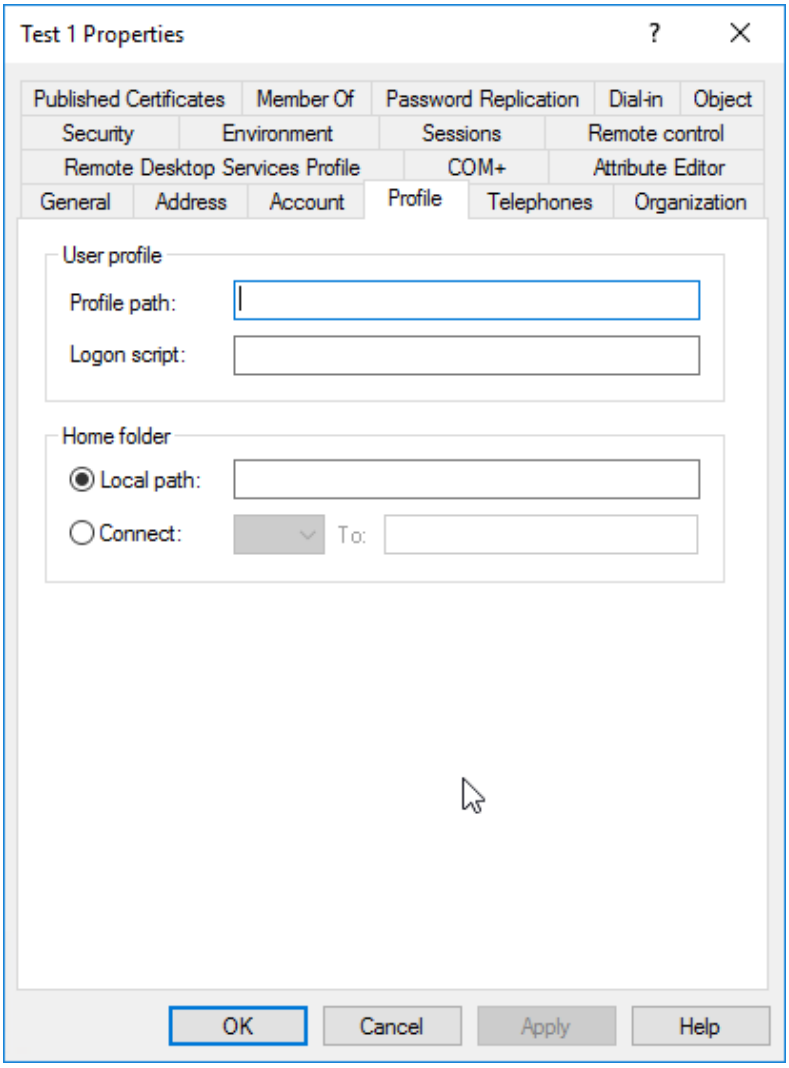
```
      }
}

#
================================================================
================================
# End function block
#
================================================================
================================

#
================================================================
================================
# Define the CSV Import File.
#
================================================================
================================

  $currentPath                        = Split-Path -parent
$MyInvocation.MyCommand.Definition
    $strCurrentFile                                       =
$MyInvocation.MyCommand.Name
  if($FileWithUseridsInCSVFormat.Length -eq 0)
   {
    $strCSVFileName                 = $strCurrentFile -Replace
".ps1",".csv"
   }
    else
   {
    if($FileWithUseridsInCSVFormat.ToLower().IndexOf(".csv") -
eq -1)
    {
     $FileWithUseridsInCSVFormat =+ ".csv"
    }
    $strCSVFileName = $FileWithUseridsInCSVFormat
   }

#
================================================================
================================
```

```
# Check if the string $strCSVFileName is a path. In that case,
nothing has to be done.
# In case it is not a path, then the current location should
be added.
#
================================================================
================================


  if (-not(Split-Path($strCSVFileName)))
  {
   $strCSVFileName = $currentPath + "\" + $strCSVFileName
  }

#
================================================================
================================
# Declares the variables.
#
================================================================
================================

  $valCounter                     = 1
  $Global:arrTable                = @()
  $Record                          = [ordered] @{"Username" =
"";"Error"= ""}

#
================================================================
================================
# Define the log file. This log file contains all the results.
#
================================================================
================================

    $strCSVLogFileSucces  =  $strCSVFileName  -Replace
".csv","_(Results).csv"
  If(Test-Path $strCSVLogFileSucces)
  {
  Remove-Item $strCSVLogFileSucces
  }
```

```
#
=================================================================
==================================
# Read the CSV file.
#
=================================================================
==================================

  if(Test-Path -LiteralPath $strCSVFileName)
  {
   $arrUserids = Import-Csv $strCSVFileName
  }
   Else
  {
    Write-Host "The import file $strCSVFileName does not
exists."
   Exit 1
  }

#
=================================================================
==================================
# Modify the users' profile path.
# Write the success or failure to the array with the results.
#
=================================================================
==================================

  Import-Module ActiveDirectory
  Clear-Host
  Write-Host (Get-Date).ToString('T')  "Started."
  ForEach ($objUserid in $arrUserids)
   {
     $strUserid = $objUserid.Userid
     if ($strUserid.Length -gt 0)
     {
        Write-Progress -Activity "Removing profile path from
user in Active Directory" -Status "Processing user $strUserid"
-PercentComplete ($valCounter / $arrUserids.Count * 100)
       Process-User -strUserid $strUserid
       Start-Sleep -s 1
```

```
       $valCounter ++
     }
       else
     {
        Write-Host "The input file $strCSVFileName has an
invalid layout. The column header should be named 'Userid'."
       Exit 1
     }
   }
  Start-Sleep -s 1

#
==================================================================
==================================
# Write the results to the csv file.
#
==================================================================
==================================

  If($Global:arrTable.Count -gt 0)
  {
     $Global:arrTable | Export-Csv $strCSVLogFileSucces -
NoTypeInformation
  }
   Else
  {
    Write-Host "Something went wrong while writing the logfile
$strCSVLogFileSucces. Maybe nothing to report..."
  }
Write-Host (Get-Date).ToString('T')  "Finished."
```

After running the script with the correct permissions, the profile path looks like:



Profile path after running the script.

The scripts can be downloaded in ZIP format:

1. Link to RemoveProfilePathFromUser (v0.1)
2. Link to RemoveProfilePathFromUser (v0.2)
3. Link to RemoveProfilePathFromUser (v0.3)