

```
1 #Get-NonADGroupO365LicenseUser.ps1
2 #Written By: Jeff Brusoe
3 #Last Updated: July 6, 2021
4
5 [CmdletBinding()]
6 param(
7     [ValidateNotNullOrEmpty()]
8     [string]$UsersToBeChangedFile = "$PSScriptRoot\Logs\" +
9                                     (Get-Date -Format yyyy-MM-dd) +
10                                     "-UsersToAddToGroup.csv"
11 )
12
13 try {
14     Write-Output "Configuring Environment"
15     Set-HSCEnvironment -ErrorAction Stop
16
17     New-Item $UsersToBeChangedFile -ErrorAction Stop -Force
18 }
19 catch {
20     Write-Warning "Unable to configure environment"
21     Invoke-HSCExitCommand -ErrorCount $Error.Count
22 }
23
24 try {
25     Write-Output "Generating List of AD Users"
26
27     $GetADUserParams = @{
28         Filter = "*"
29         SearchBase = "OU=HSC,DC=HS,DC=WVU-AD,DC=WVU,DC=EDU"
30         Properties = @(
31             "extensionAttribute7",
32             "memberOf"
33         )
34         ErrorAction = "Stop"
35     }
36
37     $ADUsers = Get-ADUser @GetADUserParams |
38         Where-Object {$_.extensionAttribute7 -eq "Yes365" -AND
39                     $_.Enabled}
40 }
41 catch {
42     Write-Warning "Unable to generate list of AD users"
43     Invoke-HSCExitCommand -ErrorCount $Error.Count
44 }
45
46 $UserCount = 1
47 foreach ($ADUser in $ADUsers)
48 {
49     Write-Output $("Current AD User: " + $ADUser.SamAccountName)
50
51     Write-Output "User Count: $UserCount"
52     $UserCount++
53
54     $ADUserGroups = $ADUser.memberOf
55     $ADUserGroups
56
57     $InO365LicenseGroup = $false
58     foreach ($ADUserGroup in $ADUserGroups) {
59         if ($ADUserGroup -like "*Office 365 Base Licensing*") {
60             Write-Output "User is in Base Licensing Group"
```

```
61         $InO365LicenseGroup = $true
62     }
63 }
64
65 if (!$InO365LicenseGroup) {
66     Write-Output "User is not in base licensing group"
67
68     $ADUser |
69     Select-Object
70     SamAccountName,Enabled,extensionAttribute7,DistinguishedName |
71     Export-Csv $UsersToBeChangedFile -NoTypeInformation -Append
72 }
73 Write-Output "*****"
74 }
75
76 Write-Output "Finished Search"
77 Invoke-HSCExitCommand -ErrorCount $Error.Count
```