

```

1  #Set-MailContactCompanyField.ps1
2  #Written by: Jeff Brusoe
3  #Last Updated: May 13, 2020
4  #
5  #The purpose of this file is to set the mail contact company field for MSOL contacts.
6  #It is not designed to run as a scheduled task because
7  #these are all being migrated over to AzureAD contacts.
8
9  [CmdletBinding()]
10 param (
11     #Common HSC PowerShell Parameters
12     [switch]$NoSessionTranscript,
13     [string]$LogFilePath = "$PSScriptRoot\Logs",
14     [switch]$StopOnError, #true is used for testing purposes
15     [int]$DaysToKeepLogFiles = 5 #this value used to clean old log files
16 )
17 #####
18 #Import HSC PowerShell Modules#
19 #####
20
21 #Build path to HSC PowerShell Modules
22 $PathToHSCPowerShellModules = $PSScriptRoot
23 $PathToHSCPowerShellModules =
24 $PathToHSCPowerShellModules.substring(0,$PathToHSCPowerShellModules.lastIndexOf("\")+1)
25 $PathToHSCPowerShellModules += "HSC-PowerShell-Modules"
26 Write-Output $PathToHSCPowerShellModules
27
28 #Attempt to load common code module
29 $CommonCodeModule = $PathToHSCPowerShellModules + "\HSC-CommonCodeModule.psm1"
30 Write-Output "Path to common code module: $CommonCodeModule"
31 Import-Module $CommonCodeModule -Force -ArgumentList
32 $NoSessionTranscript,$LogFilePath,$true,$DaysToKeepLogFiles
33
34 if ($Error.Count -gt 0)
35 {
36     #Any errors at this point are from loading modules. Program must stop.
37     Write-Warning "There was an error configuring the environment. Program is exiting."
38     Exit-Commands
39 }
40 #####
41 #End of Import HSC PowerShell Modules#
42 #####
43
44 #####
45 #Configure environment block#
46 #####
47 Write-Output "Getting Parameter Information"
48 Get-Parameter -ParameterList $PSBoundParameters
49
50 Set-Environment
51 Set-WindowTitle
52
53 #See this page to understand what is going on here.
54 #https://www.thecloudjournal.net/2016/07/create-your-own-powershell-module-for-exchange-online/
55 ##ConnectTo-Office365 #from Office 365 module
56 ##Import-Module ExchangeOnline -Force #comes from HSC-Office365Module.psm1
57 &
58 "C:\Users\microsoft\Documents\GitHub\HSC-PowerShell-Repository\6OldConnectionFiles\Connect-ToOffice365-MS4.ps1"
59
60 $InvalidEmailFile = "$PSScriptRoot\Logs\" + (Get-Date -format yyyy-MM-dd-HH-mm) +
61 "-InvalidEmailFile.txt"
62 New-Item $InvalidEmailFile -type file -Force

```

```

61 #####
62 #End of environment configuration block#
63 #####
64
65 $CurrentUserCount = 1
66 $CompanyAddCount = 0
67
68 Write-Output "Getting Msol Contacts"
69 try
70 {
71     $MSOLContacts = Get-MsolContact -All -ErrorAction Stop
72     Write-Output "Successfully generated list of Msol contacts"
73 }
74 catch
75 {
76     Write-Warning "Unable to get list of Msol users. Program is exiting."
77     Exit-Command
78 }
79
80 Write-Output $("Total Number of MsolContacts: " + ($MSOLContacts | Measure).Count)
81
82 foreach ($MSOLContact in $MSOLContacts)
83 {
84     $EmailAddress = $MSOLContact.EmailAddress
85     Write-Output "Current User: $EmailAddress"
86
87     $CompanyName = $MSOLContact.CompanyName
88     Write-Output "Company Name: $CompanyName"
89
90     if ([string]::IsNullOrEmpty($CompanyName) -OR ($CompanyName -ne "WVU")) -AND
91     ($EmailAddress -like "*@mail.wvu.edu*" -OR $EmailAddress -like "*@wvu.edu*"))
92     {
93         Set-Contact -Identity $EmailAddress -Company "WVU"
94         $CompanyAddCount++
95
96         Start-Sleep -s 1
97     }
98     elseif ([string]::IsNullOrEmpty($CompanyName) -OR $CompanyName -ne "WVUF") -AND
99     ($EmailAddress -like "*wvuf.org*"))
100     {
101         Set-Contact -Identity $EmailAddress -Company "WVUF"
102
103         Start-Sleep -s 1
104     }
105     elseif ($EmailAddress -notlike "*mail.wvu.edu*" -AND $EmailAddress -notlike
106     "*wvuf.org*" -AND $EmailAddress -notlike "*@wvu.edu*")
107     {
108         #This is here to ensure that no @gmail.com etc. type of contacts.
109         Add-Content $InvalidEmailFile -Value $EmailAddress
110     }
111
112     Write-Output "Current User Count: $CurrentUserCount"
113     $CurrentUserCount++
114
115     Write-Output "Company Add Count: $CompanyAddCount"
116
117     Write-Output "*****"
118 }
119
120 if (!$NoSessionTranscript)
121 {
122     Stop-Transcript
123 }
124
125 Exit

```