

```

1  #-----
2  #6Get-MainCampusMailboxes.ps1
3  #
4  #Written by: Jeff Brusoe
5  #
6  #Last Modified: April 30, 2020
7  #
8  #Version: 2.0
9  #
10 #Purpose
11 #1. Export main campus email addresses which will be imported (via a separate script)
    the the HSC tenant.
12 #2. Create file to import SIP addresses into the HSC tenant. (Does this have to be done
    separately?)
13 #
14 #Mailboxes that are selected must meet the following requirements
15 #1. Primary SMTP address of @mail.wvu.edu or @wvu.edu
16 #2. Mailbox is not hidden in main campus GAL
17 #3. First AND last names must be present
18 #3. Gordon Gee - For some reason, his display name is correct, but he has no first/last
    name in the address book. By rule 3, his account would normally not be exported
19 #4. Does not contain the word "test" in the name
20 #
21 #This file assumes a connection to the main campus Office 365 tenant has been
    established. If it isn't, then it will
22 #look for the Connect-ToOffice365-MC.ps1 file to attempt a connection.
23 #
24 #To Be Done
25 #1. Determine if separate SIP file is needed.
26 #-----
27
28 <#
29 .SYNOPSIS
30     The purpose of this program is to export mailbox information from the WVU main
    campus Office 365 tenant.
31
32 .DESCRIPTION
33     This file is used to export mailbox information from the WestVirginiaUniversity
    Office 365 tenant.
34     Mailboxes that are selected for export must meet the following requirements.
35     1. The mailbox must be a user mailbox.
36     2. Primary SMTP Address is either @mail.wvu.edu or @wvu.edu.
37     3. Not hidden in address book
38     4. Contain a first and last name
39
40     There are some other minor clean up to account for discrepancies in the main campus
    address book as well.
41
42 .PARAMETER
43     In addition to the common parameters, the file specific parameters are:
44
45 .PARAMETER FileLocation
46     This is the directory in the file system that the ps1 file is located in.
47
48 .PARAMETER WindowTitle
49
50 .NOTES
51     Author: Jeff Brusoe
52     Last Updated: February 19, 2018
53 #>
54
55 [CmdletBinding()]
56 param (
57     #Common HSC PowerShell Parameters
58     [switch]$NoSessionTranscript,

```

```

59     [string]$LogFilePath = "$PSScriptRoot\Logs",
60     [switch]$StopOnError, #true is used for testing purposes
61     [int]$DaysToKeepLogFiles = 5, #this value used to clean old log files
62
63     #File specific parameters
64     [string]$MainCampusMailboxFile = "$LogFilePath\" + (Get-Date -Format
yyyy-MM-dd-HH-mm) + "-MainCampusExchangeExport.csv",
65     [string]$SIPFile = "$LogFilePath\" + (Get-Date -Format yyyy-MM-dd-HH-mm) +
"-SIP-MC.csv",
66     [string]$ErrorFilePath = "$LogFilePath\" + (Get-Date -Format yyyy-MM-dd-HH-mm) +
"-MainCampusExport-ErrorFile.txt"
67 )
68
69 Clear-Host
70
71 #####
72 #Import HSC PowerShell Modules#
73 #####
74
75 #Build path to HSC PowerShell Modules
76 $PathToHSCPowerShellModules = $PSScriptRoot
77 $PathToHSCPowerShellModules =
78 $PathToHSCPowerShellModules.substring(0,$PathToHSCPowerShellModules.lastIndexOf("\")+1)
79 $PathToHSCPowerShellModules += "1HSC-PowerShell-Modules"
80 Write-Output $PathToHSCPowerShellModules
81
82 #Attempt to load common code module
83 $CommonCodeModule = $PathToHSCPowerShellModules + "\HSC-CommonCodeModule.psm1"
84 Write-Output "Path to common code module: $CommonCodeModule"
85 Import-Module $CommonCodeModule -Force -ArgumentList
$NoSessionTranscript,$LogFilePath,$true,$DaysToKeepLogFiles
86
87 #Attempt to load HSC Office 365 Module
88 $Office365Module = $PathToHSCPowerShellModules + "\HSC-Office365Module.psm1"
89 Write-Output "Path to HSC Office 365 module: $Office365Module"
90 Import-Module $Office365Module -Force
91
92 if ($Error.Count -gt 0)
93 {
94     #Any errors at this point are from loading modules. Program must stop.
95     Write-Warning "There was an error configuring the environment. Program is exiting."
96     Exit-Commands
97 }
98
99 #####
100 #End of Import HSC PowerShell Modules#
101 #####
102
103 #####
104 #Configure environment block#
105 #####
106 Write-Output "Getting Parameter Information"
107 Get-Parameter -ParameterList $PSBoundParameters
108
109 Set-Environment
110 Set-WindowTitle
111
112 #See this page to understand what is going on here.
113 #https://www.thecloudjournal.net/2016/07/create-your-own-powershell-module-for-exchange-o
nline/
114 #ConnectTo-Office365 #from Office 365 module
115 #Import-Module ExchangeOnline -Force #comes from HSC-Office365Module.psm1
116
117 &
"C:\Users\microsoft\Documents\GitHub\HSC-PowerShell-Repository\1HSC-PowerShell-Modules\Co
nnect-ToOffice365-MC.ps1"

```

```

118 #Remove old CSV log files
119 Remove-OldLogFile -CSV -Path $LogFilePath -Days $DaysToKeepLogFiles -Verbose -Delete
120
121 #####
122 #End of environment configuration block#
123 #####
124
125 #Initialization of output files
126 New-Item -path $SIPFile -type "file" -force
127 New-Item -path $MainCampusMailboxFile -type "file" -force
128
129 #Get main campus mailboxes
130 Write-Output "Generating list of main campus mailboxes"
131
132 try
133 {
134     $Mailboxes = Get-Recipient -identity "*" -Resultsize Unlimited -ErrorAction Stop |
where {(($_ .PrimarySMTPAddress -like "*@mail.wvu.edu") -OR ($_ .PrimarySMTPAddress
-like "*@wvu.edu")) -AND ($_ .HiddenFromAddressListsEnabled -eq $false) -AND
($_ .Title -ne "Retiree") -AND ($_ .Alias -ne "cwassick")}
135 }
136 catch
137 {
138     Write-Warning "Unable to query main campus mailboxes. Program is exiting."
139     Exit-Command
140 }
141
142 Write-Output "`nBeginning to process main campus mailboxes"
143
144 foreach ($Mailbox in $Mailboxes)
145 {
146     if ($Error.Count -gt 0 -AND $StopOnError)
147     {
148         Write-Warning "Stopping due to error. Program is exiting."
149         $Error | FL
150         Exit-Command
151     }
152
153     Write-Output $("`n`nCurrent Mailbox: " + $Mailbox.PrimarySMTPAddress)
154
155     if ($Mailbox.PrimarySMTPAddress -eq "Gordon.gee@mail.wvu.edu")
156     {
157         #Make sure to include this address
158         #Currently (April 25, 2016) there is no first or last name entry for him, but
his display name is correct.
159         Write-Output "Adding Gordon Gee to export file."
160         $Mailbox | select
DisplayName,FirstName,LastName,UserPrincipalName,Title,Department,Phone,PrimarySM
TPAddress,Alias,
WhenChanged,legacyExchangeDN,customattribute8,hiddenfromaddresslistsenabled |
export-csv $MainCampusMailboxFile -Append
161
162         $MailboxCount++
163     }
164     elseif ([string]::IsNullOrEmpty($Mailbox.LastName)) -OR
([string]::IsNullOrEmpty($Mailbox.FirstName))
165     {
166         if ([string]::IsNullOrEmpty($Mailbox.LastName))
167         {
168             Write-Warning "Skipping mailbox because of missing last name"
169         }
170         else
171         {
172             Write-Warning "Skipping mailbox because of missing first name"
173         }
174     }
175     elseif ($Mailbox.DisplayName.ToLower().IndexOf("test") -ge 0)

```

```

176 {
177     Write-Warning "Test account - Skipping"
178 }
179 else
180 {
181     Write-Output "Exporting user information"
182     $Mailbox | select
        DisplayName,FirstName,LastName,UserPrincipalName,Title,Department,Phone,PrimarySM
        TPAddress,Alias,
        WhenChanged,legacyExchangeDN,customattribute8,hiddenfromaddresslistsenabled |
        export-csv $MainCampusMailboxFile -Append

183
184     $MailboxCount++
185 }
186
187 #Examine email addresses to see if any are SIP ones.
188 Write-Host -ForegroundColor Green "Looking for SIP Addresses"
189
190 $EmailAddresses = $Mailbox.EmailAddresses
191
192 foreach ($EmailAddress in $EmailAddresses)
193 {
194     "EmailAddress: " + $EmailAddress
195
196     if ($EmailAddress.indexof("SIP") -ge 0)
197     {
198         Write-Host -ForegroundColor Green $("Writing " + $EmailAddress + " to SIP
        file")
199
200         $Mailbox | select primarySMTPAddress,Alias,@{Label
        ='SIPAddress';Expression={$EmailAddress}} | export-csv $SIPFile -Append
201     }
202 }
203
204
205 if (!$NoSessionTranscript)
206 {
207     Stop-Transcript
208 }
209
210 #Exit - Uncomment when done with testing.

```