

```

1  #Set-ADMailAttribute.ps1
2  #
3  #Written by: Jeff Brusoe
4  #Last Modified: May 12, 2020
5  #Version: 1.0
6  #
7  #The purpose of this file is to copy the primary SMTP address over to the mail
  attribute field in AD.
8
9  [CmdletBinding()]
10 param (
11     #Common HSC PowerShell Parameters
12     [switch]$NoSessionTranscript,
13     [switch]$StopOnError = $true, # $true is used for testing purposes,
14     [string]$LogFilePath = "$PSScriptRoot\Logs", #Running on sysscript2
15     [int]$DaysToKeepLogFiles = 5, #This value is used to clean up old log files.
16
17     #File Specific Parameters
18     [switch]$Testing,
19     [string]$User=$null, #Used to set for one specific user
20     [string]$ChangeLog = "$PSScriptRoot\Logs\" + (Get-Date -format yyyy-MM-dd-hh-mm-ss)
    + "-ChangeLog.txt"
21 )
22
23 #####
24 # Configure Environment Block #
25 #####
26
27 $Error.Clear()
28
29 #Build path to HSC PowerShell Modules
30 $PathToHSCPowerShellModules = $PSScriptRoot
31 $PathToHSCPowerShellModules =
32 $PathToHSCPowerShellModules.substring(0,$PathToHSCPowerShellModules.lastIndexOf("\")+1)
33 $PathToHSCPowerShellModules += "1HSC-PowerShell-Modules"
34 Write-Output $PathToHSCPowerShellModules
35
36 #Attempt to load common code module
37 $CommonCodeModule = $PathToHSCPowerShellModules + "\HSC-CommonCodeModule.psm1"
38 Write-Output "Path to common code module: $CommonCodeModule"
39 Import-Module $CommonCodeModule -Force -ArgumentList
40 $NoSessionTranscript,$LogFilePath,$true,$DaysToKeepLogFiles
41
42 Write-Output "PSScriptRoot: $PSScriptRoot"
43 Set-Location $PSScriptRoot
44
45 Set-Environment
46 Set-WindowTitle
47
48 New-Item -type file -path $ChangeLog -Force
49
50 if ($Error.Count -gt 0)
51 {
52     #Indicates an error configuring environment
53     Write-Output "An error has occurred. Program is exiting."
54     Exit-Commands
55 }
56
57 #####
58 # End of Environment Congifuration Block #
59 #####
60
61 if ([string]::IsNullOrEmpty($User))
62 {
63     #This is the case where the $user parameter was is $null. An array of all users
64     will be generated.
65     Write-Output "Generating list of all users."

```

```

63
64     try
65     {
66         $ADUsers = Get-ADUser -Filter * -Properties proxyaddresses,mail -ErrorAction Stop
67         Write-Output "Finished generating all user list.`n`n"
68     }
69     catch
70     {
71         Write-Warning "Unable to generate AD user list. Program is exiting."
72         Exit-Command
73     }
74 }
75 else
76 {
77     Write-Output "Searching for user: $User"
78
79     try
80     {
81         $ADUsers = Get-ADUser $User -ErrorAction Stop
82         Write-Output "User has been found"
83     }
84     Catch
85     {
86         Write-Warning "Unable to find user. Program is exiting."
87         Exit-Command
88     }
89 }
90
91 Write-Output "Beginning to loop through user array."
92
93 foreach ($ADUser in $ADUsers)
94 {
95     Write-Output $("Current User: " + $ADUser.SamAccountName)
96
97     $PrimarySMTPAddress = $ADUser.ProxyAddresses | where {$_ -clike "SMTP:*"}
98     $PrimarySMTPAddress = $PrimarySMTPAddress -replace "SMTP:", ""
99     Write-Output "Primary SMTP Address: $PrimarySMTPAddress"
100    Write-Output $("Current Mail Attribute: " + $ADUser.mail)
101
102    if ($Error.Count -gt 0)
103    {
104        return
105    }
106
107    if (![string]::IsNullOrEmpty($PrimarySMTPAddress))
108    {
109        if ($PrimarySMTPAddress.IndexOf("@hsc.wvu.edu") -gt 0 -OR
110            $PrimarySMTPAddress.IndexOf("@wvumedicine.org") -gt 0)
111        {
112            if ((![string]::IsNullOrEmpty($ADUser.mail) -AND
113                ($PrimarySMTPAddress.ToLower().Trim()) -eq $ADUser.mail.ToLower().Trim()))
114            {
115                Write-Output "Primary SMTP address matches mail attribute."
116            }
117            else
118            {
119                Write-Output "Primary SMTP address does not match mail attribute."
120
121                if ($Testing)
122                {
123                    Read-Host "Press enter key to continue"
124                }
125
126                try
127                {

```

```

128         $ADUser | Set-ADUser -EmailAddress $PrimarySMTPAddress -ErrorAction
129         Stop
130
131         Write-Output "Successfully changed mail attribute"
132
133         Add-Content -Value "Original Mail Attribute:
134         $OriginalMailAttribute" -Path $ChangeLog
135         Add-Content -Value "New Mail Attribute: $PrimarySMTPAddress" -Path
136         $ChangeLog
137         Add-Content -Value "Successfully changed attribute" -Path $ChangeLog
138         Add-Content -Value "*****" -Path $ChangeLog
139     }
140     catch
141     {
142         Write-Warning "Error setting primary SMTP address"
143
144         if ($StopOnError)
145         {
146             Write-Warning "Program is exiting."
147             #Exit-Command
148             Read-Host "Error: "
149         }
150     }
151 }
152 else
153 {
154     Write-Output "Proxy address is null and is being skipped."
155 }
156 Write-Output "*****"
157 }
158 if (!$NoSessionTranscript)
159 {
160     Stop-Transcript
161 }

```