

```
1 #Set-CorrectEmailAddress.ps1
2 #Written by: Jeff Brusoe
3 #Last Updated: December 15, 2020
4 #
5 #The purpose of this file is to correct any entries
6 #which have no.email@wvu.edu as an email attribute field in AD.
7
8 [CmdletBinding()]
9 param ()
10
11 #Configure environment
12 try {
13     Set-HSCEnvironment -ErrorAction Stop
14 }
15 catch {
16     Write-Warning "Unable to configure environment"
17     Invoke-HSCExitCommand -ErrorCount $Error.Count
18 }
19
20 #Generate array of AD users with no.email@wvu.edu in mail field
21 try {
22     Write-Output "Generating array of AD users"
23
24     $Properties = @(
25         "mail",
26         "extensionAttribute6",
27         "extensionAttribute15",
28         "proxyAddresses"
29     )
30
31     $GetADUserParams = @{
32         Filter = 'mail -eq "no.email@wvu.edu"'
33         Properties = $Properties
34         ErrorAction = "Stop"
35     }
36
37     $ADUsers = Get-ADUser @GetADUserParams
38 }
39 catch {
40     Write-Warning "Unable to generate list of AD Users"
41     Invoke-HSCExitCommand -ErrorCount $Error.Count
42 }
43
44 $CSVSummaryFile = "$PSScriptRoot\Logs\" +
45     (Get-Date -format yyyy-MM-dd-HH-mm) +
46     "-MailAttributeSummary.csv"
47 $ChangeCount = 0
48
49 if (($ADUsers | Measure-Object).Count -eq 0) {
50     Write-Output "`n`nNo users found.`n`n"
51 }
52 else
53 {
54     foreach ($ADUser in $ADUsers)
55     {
56         [string]$MailAttribute = $null
57         $ChangePrimarySMTP = $false
58
59         Write-Output $("Current User: " + $ADUser.SamAccountName)
60         Write-Output $("Current Mail Attribute: " + $ADUser.mail)
```

```
61 Write-Output $("extensionAttribute6: " + $ADUser.extensionAttribute6)
62 Write-Output $("Distinguished Name: " + $ADUser.DistinguishedName)
63
64 if (($ADUser.ProxyAddresses | Measure-Object).Count -gt 0)
65 {
66     Write-Output "`nProxy Addresses:"
67     foreach ($ProxyAddress in $ADUser.ProxyAddresses)
68     {
69         Write-Output "Examining Proxy Address: $ProxyAddress"
70
71         if ($ProxyAddress -clike "SMTP:*")
72         {
73             Write-Output "Found Primary SMTP Address"
74             $MailAttribute = $ProxyAddress
75             $MailAttribute = $MailAttribute -replace "SMTP:", ""
76             $MailAttribute = $MailAttribute.Trim()
77
78             if ($MailAttribute -eq "no.email@wvu.edu")
79             {
80                 #This handles the case when this was incorrectly set
81                 #as the primary SMTP address in the proxyaddresses.
82                 $ChangePrimarySMTP = $true
83                 $MailAttribute = $null
84             }
85         }
86     }
87
88     if ([string]::IsNullOrEmpty($MailAttribute)) {
89         Write-Output "Primary SMTP Address was not found"
90     }
91     elseif ($ChangePrimarySMTP)
92     {
93         #This implies primary SMTP was set to no.email@wvu.edu
94         $PrimarySMTPAddress = "SMTP:" + $ADUser.UserPrincipalName
95
96         Write-Output "Current Primary SMTP: no.email@wvu.edu"
97         Write-Output "New Primary SMTP: $PrimarySMTPAddress"
98
99         try
100         {
101             $ADUser.ProxyAddresses.Remove("SMTP:no.email@wvu.edu")
102             $ADUser.ProxyAddresses.Add($PrimarySMTPAddress)
103
104             Set-ADUser -Instance $ADUser -ErrorAction Stop
105
106             Write-Output "Successfully set new primary SMTP address"
107         }
108         catch {
109             Write-Warning "Error cleaning up primary SMTP address"
110         }
111     }
112 }
113 else {
114     Write-Output "ProxyAddress field is empty."
115 }
116
117 if ([string]::IsNullOrEmpty($MailAttribute))
118 {
119     Write-Output "`nPrimary SMTP Address not found. Examining
extensionAttribute6"
```

```

120
121     if ($ADUser.extensionAttribute6 -eq "EMPLOYEE") {
122         $MailAttribute = $ADUser.SamAccountName + "@hsc.wvu.edu"
123     }
124     elseif ($ADUser.extensionAttribute6 -eq "STUDENT") {
125         $MailAttribute = $ADUser.SamAccountName + "@mix.wvu.edu"
126     }
127     elseif ($ADUser.extensionAttribute6 -eq "HOSPITAL") {
128         $MailAttribute = $ADUser.SamAccountName + "@wvumedicine.org"
129     }
130 }
131
132 if ([string]::IsNullOrEmpty($MailAttribute)) {
133     Write-Output "`nextensionAttribute6 is null. Setting mail attribute to be
UPN"
134     $MailAttribute = $ADUser.UserPrincipalName
135 }
136
137 if (![string]::IsNullOrEmpty($MailAttribute))
138 {
139     Write-Output "`n`nSetting Mail Attribute to $MailAttribute"
140
141     try
142     {
143         $ADUser | Set-ADUser -EmailAddress $MailAttribute -ErrorAction Stop
144         Write-Output "Done setting mail attribute"
145
146         $MailAttributeInfo = New-Object -TypeName PSObject
147
148         [string]$SamAccountName = $ADUser.SamAccountName
149         [string]$UPN = $ADUser.UserPrincipalName
150
151         $AddMemberParams = @{
152             MemberType = "NoteProperty"
153             Name = "SAMAccountName"
154             Value = $SamAccountName
155             ErrorAction = "Stop"
156         }
157         $MailAttributeInfo | Add-Member @AddMemberParams
158
159         $AddMemberParams["Name"] = "UPN"
160         $AddMemberParams["Value"] = $UPN
161         $MailAttributeInfo | Add-Member @AddmemberParams
162
163         $AddMemberParams["Name"] = "MailAttribute"
164         $AddMemberParams["Value"] = $MailAttribute
165         $MailAttributeInfo | Add-Member @AddMemberParams
166
167         $MailAttributeInfo |
168             Export-Csv $CSVSummaryFile -Append -ErrorAction Stop
169
170         $ChangeCount++
171         Write-Output "Change Count: $ChangeCount"
172     }
173     catch {
174         Write-Warning "Unable to set mail attribute."
175     }
176 }
177
178 Write-Output "*****"

```

```
179     }  
180 }  
181  
182 Invoke-HSCExitCommand -ErrorCount $Error.Count
```