

```

1 <#
2 .SYNOPSIS
3     This file imports the hospital's unique ID numbers to extensionAttribute13 and
4     is needed for the shared user distribution list project.
5
6 .PARAMETER Domains
7     The domains that are to be searched.
8
9 .NOTES
10    Written by: Jeff Brusoe
11    Last Updated: October 27, 2020
12 #>
13
14 [CmdletBinding()]
15 param (
16     [ValidateNotNullOrEmpty()]
17     [string[]]$Domains = @("WVUH", "WVUHS"),
18
19     [switch]$Testing
20 )
21
22 Set-HSCEnvironment
23
24 #Build path to import files
25 $PathToImportFile = $PSScriptRoot
26 $PathToImportFile =
27     $PathToImportFile.substring(0, $PathToImportFile.lastIndexOf("\")+1)
28 $PathToImportFile += "Get-WVUMExt13\Logs\"
29 Write-Output "Path to Import File: $PathToImportFile"
30
31 #Change these to be more domain independent
32 $WVUHImportFile = Get-ChildItem -Path $PathToImportFile -Filter "*wvuh.wvuhs.com*"
33 |
34     Sort-Object LastWriteTime -Desc | Select-Object -first 1
35 Write-Output $("WVUH Import File: " + $WVUHImportFile.FullName)
36
37 $WVUHSImportFile = Get-ChildItem -Path $PathToImportFile -Filter "*-wvuhs.com*" |
38     Sort-Object LastWriteTime -Desc | Select-Object -first 1
39 Write-Output $("WVUHS Import File: " + $WVUHSImportFile.FullName)
40
41 #Create output file
42 $UserFoundFile = "$PSScriptRoot\Logs\" + (Get-Date -Format yyyy-MM-dd-HH-mm) + "-
43     Ext13MatchFile.csv"
44 New-Item -type file $UserFoundFile -Force
45
46 Write-Output "Importing CSV Files"
47 $WVUHUsers = Import-Csv $WVUHImportFile.FullName
48 $WVUHSUsers = Import-Csv $WVUHSImportFile.FullName
49
50 foreach ($Domain in $Domains)
51 {
52     Write-Output "Processing $Domain"
53
54     switch ($Domain)
55     {
56         "WVUH" { $DomainUsers = $WVUHUsers }
57         "WVUHS" { $DomainUsers = $WVUHSUsers }
58     }
59
60     $Error.Clear()

```

```
58
59 foreach ($DomainUser in $DomainUsers)
60 {
61     $SamAccountName = $DomainUser.SamAccountName
62     $ext13 = $DomainUser.extensionAttribute13
63
64     Write-Output "Current $Domain Account: $SamAccountName"
65     Write-Output "extensionAttribute13 From File: $ext13"
66
67     if ([string]::IsNullOrEmpty($ext13)) {
68         Write-Output "extensionAttribute13 is empty"
69     }
70     else
71     {
72         #ext13 has a value. Need to see if it should be added to the HS domain
73         $LDAPFilter = "(SamAccountName=$SamAccountName)"
74         Write-Output "LDAPFilter: $LDAPFilter"
75
76         try
77         {
78             $GetADUserParams = @{
79                 LDAPFilter = $LDAPFilter
80                 Properties = "extensionAttribute13"
81                 ErrorAction = "Stop"
82             }
83
84             $HSCADUser = Get-ADUser @GetADUserParams
85
86             if ($null -ne $HSCADUser)
87             {
88                 Write-Output "User found in HS domain" #Should go to the catch block if
user not found.
89
90                 $UserInfo = New-Object -TypeName PSObject
91
92                 $AddMemberParams = @{
93                     MemberType = "NoteProperty"
94                     Name = "SamAccountName"
95                     Value = $SamAccountName
96                     ErrorAction = "Stop"
97                 }
98
99                 $UserInfo | Add-Member @AddMemberParams
100
101                 $AddMemberParams["Name"] = "HospitalDomain"
102                 $AddMemberParams["Value"] = $Domain
103                 $UserInfo | Add-Member @AddMemberParams
104
105                 $AddMemberParams["Name"] = "ExtensionAttribute13"
106                 $AddMemberParams["Value"] = $ext13
107                 $UserInfo | Add-Member @AddMemberParams
108
109                 $AddMemberParams["Name"] = "HSDomainOU"
110                 $AddMemberParams["Value"] = $($HSCADUser.DistinguishedName)
111                 $UserInfo | Add-Member @AddMemberParams
112
113                 $UserInfo | Export-Csv $UserFoundFile -Append
114
115                 Write-Output $("Ext13 From Domain: " + $HSCADUser.extensionAttribute13)
116
```

```
117     try
118     {
119         Write-Output "Attempting to set extensionAttribute13 in the HS domain"
120
121         if([string]::IsNullOrEmpty($HSCADUser.extensionAttribute13)) {
122             $HSCADUser |
123             Set-ADUser -Add @{extensionAttribute13=$ext13} -ErrorAction Stop
124         }
125         else
126         {
127             $HSCADUser |
128             Set-ADUser -Replace @{extensionAttribute13=$ext13} -ErrorAction
Stop
129         }
130
131         Write-Output "Successfully set extensionAttribute13"
132
133         if ($Testing) {
134             Start-Sleep -s 2
135
136             if ($Error.Count -gt 0) {
137                 Invoke-HSCExitCommand -ErrorCount $Error.Count
138             }
139         }
140     }
141     catch {
142         Write-Warning "Error setting extensionAttribute13 in the HS domain"
143     }
144 }
145 else {
146     Write-Output "User not found in HS domain"
147 }
148 }
149 catch {
150     Write-Warning "Unable to find user in HS domain"
151 }
152 }
153
154 Write-Output "*****"
155 }
156 }
157
158 Invoke-HSCExitCommand -ErrorCount $Error.Count
```