

```

1  #Import-WVUMExt13.ps1
2  #Written by: Jeff Brusoe
3  #Last Updated: May 6, 2020
4
5  [CmdletBinding()]
6  param (
7      #Common HSC PowerShell Parameters
8      [switch]$NoSessionTranscript,
9      [string]$LogFilePath = "$PSScriptRoot\Logs",
10     [switch]$StopOnError, #true is used for testing purposes
11     [int]$DaysToKeepLogFiles = 5, #this value used to clean old log files
12
13     #File specific parameters
14     [string[]]$Domains = @("WVUH","WVUHS")
15 )
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 += "1HSC-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 #####
35 #End of Import HSC PowerShell Modules#
36 #####
37
38 #####
39 #Configure environment block#
40 #####
41 Set-Environment
42 Set-WindowTitle
43
44 Write-Output "Getting Parameter Information"
45 Get-Parameter -ParameterList $PSBoundParameters
46
47 Import-Module ActiveDirectory
48 #####
49 #End of environment configuration block#
50 #####
51
52 #Build path to import files
53 $PathToImportFile = $PSScriptRoot
54 $PathToImportFile = $PathToImportFile.substring(0,$PathToImportFile.lastIndexOf("\")+1)
55 $PathToImportFile += "Get-WVUMExt13\Logs\"
56 Write-Output "Path to Import File: $PathToImportFile"
57
58 #Change these to be more domain independent
59 $WVUHImportFile = Get-ChildItem -Path $PathToImportFile -Filter "*wvuh.wvuhs.com*" |
60     Sort LastWriteTime -Desc | Select -first 1
61 Write-Output $("WVUH Import File: " + $WVUHImportFile.FullName)
62
63 $WVUHSImportFile = Get-ChildItem -Path $PathToImportFile -Filter "*-wvuhs.com*" |
64     Sort LastWriteTime -Desc | Select -first 1
65 Write-Output $("WVUHS Import File: " + $WVUHSImportFile.FullName)
66
67 #Create output file

```

```

66 $UserFoundFile = "$PSScriptRoot\Logs\" + (Get-Date -Format yyyy-MM-dd-HH-mm) +
    "-Ext13MatchFile.csv"
67 New-Item -type file $UserFoundFile -Force
68
69 Write-Output "Importing CSV Files"
70 $WVUHUsers = Import-Csv $WVUHImportFile.FullName
71 $WVUHSUsers = Import-Csv $WVUHSImportFile.FullName
72
73 foreach ($Domain in $Domains)
74 {
75     Write-Output "Processing $Domain"
76
77     #This needs to be changed so it can accept any domain name.
78     switch ($Domain)
79     {
80         "WVUH" { $DomainUsers = $WVUHUsers }
81         "WVUHS" { $DomainUsers = $WVUHSUsers }
82     }
83
84     foreach ($DomainUser in $DomainUsers)
85     {
86         $SamAccountName = $DomainUser.SamAccountName
87         $ext13 = $DomainUser.extensionAttribute13
88
89         Write-Output "Current $Domain Account: $SamAccountName"
90
91         if ([string]::IsNullOrEmpty($ext13))
92         {
93             Write-Output "extensionAttribute13 is empty"
94         }
95         else
96         {
97             #ext13 has a value. Need to see if it should be added to the HS domain
98             $LDAPFilter = "(SamAccountName=$SamAccountName)"
99             Write-Output "LDAPFilter: $LDAPFilter"
100
101             try
102             {
103                 $user = Get-ADUser -LDAPFilter $LDAPFilter -ErrorAction Stop
104
105                 if ($user -ne $null)
106                 {
107                     Write-Output "User found in HS domain" #Should go to the catch
108                     block if user not found.
109
110                     $UserInfo = New-Object -TypeName PSObject
111
112                     $UserInfo | Add-Member -MemberType NoteProperty -Name
113                     SamAccountName -Value $SamAccountName
114                     $UserInfo | Add-Member -MemberType NoteProperty -Name
115                     HospitalDomain -Value $Domain
116                     $UserInfo | Add-Member -MemberType NoteProperty -Name
117                     ExtensionAttribute13 -Value $ext13
118                     $UserInfo | Add-Member -MemberType NoteProperty -Name HSDomainOU
119                     -Value $user.DistinguishedName
120
121                     $UserInfo | Export-Csv $UserFoundFile -Append
122
123                     try
124                     {
125                         Write-Output "Attempting to set extensionAttribute13 in the HS

```

```

126             ##else
127             ##{
128             ##  $user | Set-ADUser -Replace @{extensionAttribute13=$ext13}
129             ##}
130         }
131     catch
132     {
133         Write-Warning "Error setting extensionAttribute13 in the HS
134         domain"
135     }
136     else
137     {
138         Write-Output "User not found in HS domain"
139     }
140 }
141 catch
142 {
143     Write-Warning "Unable to find user in HS domain"
144 }
145 }
146
147 Write-Output "*****"
148 }
149 }
150
151 Exit-Command

```