

```

1  #Get-ExchangeForwardingRule.ps1
2  #
3  #Written by: Jeff Brusoe
4  #Last Modified: May 12, 2020
5  #Version: 1.0
6  #
7  #Purpose: Get exchange forwarding rule reports
8
9  [CmdletBinding()]
10 param (
11     #Common HSC PowerShell Parameters
12     [switch]$NoSessionTranscript,
13     [switch]$StopOnError, #true is used for testing purposes,
14     [string]$LogFilePath = "$PSScriptRoot\Logs",
15     [int]$DaysToKeepLogFiles = 5 #This value is used to clean up old log files
16 )
17
18 #Reset environment
19 Clear-Host
20 $Error.Clear()
21 Set-Location $PSScriptRoot
22
23 #####
24 #Import HSC PowerShell Modules#
25 #####
26
27 #Build path to HSC PowerShell Modules
28 $PathToHSCPowerShellModules = $PSScriptRoot
29 $PathToHSCPowerShellModules =
30 $PathToHSCPowerShellModules.substring(0,$PathToHSCPowerShellModules.lastIndexOf("\")+1)
31 $PathToHSCPowerShellModules += "1HSC-PowerShell-Modules"
32 Write-Output $PathToHSCPowerShellModules
33
34 #Attempt to load common code module
35 $CommonCodeModule = $PathToHSCPowerShellModules + "\HSC-CommonCodeModule.psml"
36 Write-Output "Path to common code module: $CommonCodeModule"
37 Import-Module $CommonCodeModule -Force -ArgumentList
38 $NoSessionTranscript,$LogFilePath,$true,$DaysToKeepLogFiles
39
40 #Attempt to load HSC Office 365 Module
41 $Office365Module = $PathToHSCPowerShellModules + "\HSC-Office365Module.psml"
42 Write-Output "Path to HSC Office 365 module: $Office365Module"
43 Import-Module $Office365Module -Force
44
45 #####
46 #Configure environment
47 #####
48
49 Write-Output "Getting Parameter Information"
50 Get-Parameter -ParameterList $PSBoundParameters
51
52 Set-Environment
53 Set-WindowTitle
54
55 #See this page to understand what is going on here.
56 #https://www.thecloudjournal.net/2016/07/create-your-own-powershell-module-for-exchange-online/
57
58 ConnectTo-Office365 #from Office 365 module
59 Import-Module ExchangeOnline -Force #comes from HSC-Office365Module.psml
60
61 $InboxRuleFile = "$LogFilePath\" + (Get-Date -format yyyy-MM-dd-HH-mm-ss) +
62 "-InboxRules.csv"
63
64 #####
65 #End of configure environment block
66 #####
67

```

```

64 try
65 {
66     Write-Output "Generating user list"
67     #Users = Get-AzureADUser -All $true -Filter "accountEnabled eq true" -ErrorAction
        Stop | where {($_.UserPrincipalName -notlike "*#EXT#") -AND ($_.UserPrincipalName
        -notlike "*wvuh.wvuhs.com") -AND ($_.UserPrincipalName -notlike
        "*wvumedicine.com") } | select ObjectID, UserPrincipalName, FirstName, LastName,
        StrongAuthenticationRequirements, StsRefreshTokensValidFrom,
        StrongPasswordRequired, LastPasswordChangeTimestamp
68     $Mailboxes = Get-Mailbox -ResultSize Unlimited | where {$_ .PrimarySMTPAddress
        -notlike "*rni.*" -AND $_.PrimarySMTPAddress -notlike "*wvurni*"}
        Write-Output "Successfully generated user list."
69 }
70 catch
71 {
72     Write-Warning "An error has occurred getting mailbox list."
73     Exit-Commands
74 }
75
76 $Error.Clear()
77 $MailboxCount = 0
78
79 foreach ($Mailbox in $Mailboxes)
80 {
81     Write-Output $("Current user: " + $Mailbox.PrimarySMTPAddress)
82
83     Get-InboxRule -Mailbox $Mailbox.PrimarySMTPAddress | where {($_.ForwardTo -ne
84     $null) -or ($_.ForwardAsAttachmentTo -ne $null) -or ($_.RedirectTo -ne $null)} |
        Select Name, Description, Enabled, Priority, ForwardTo, ForwardAsAttachmentTo,
        RedirectTo, DeleteMessage | Export-Csv $InboxRuleFile -NoTypeInformation -Append
85
86     if ($Error.Count -gt 0 -AND $StopOnError)
87     {
88         Write-Warning "An error has occurred. Program is exiting"
89         Exit-Commands
90     }
91
92     $MailboxCount++
93     Write-Output "Mailbox Count: $MailboxCount"
94
95     Write-Output "*****"
96 }
97
98 if (!$NoSessionTranscript)
99 {
100     Stop-Transcript
101 }
102
103 Exit
104

```