

```

1  #1Disable-POPIMAP.ps1
2  #
3  #Written by: Jeff Brusoe
4  #Last Modified: September 26, 2019
5  #
6  #This file ensures that POP and IMAP are disabled on all Exchange mailboxes.
7  #
8  #Version History
9  #September 26, 2019
10 # - Migrated to GitHub directory
11 # - Added HSC PowerShell modules
12 # - Misc cleanup of code
13 # - Split POP & IMAP disable due to cosmetic error related to licensing
14 #
15 #April 29,2020
16 # - Added error handling code for certain hospital accounts
17 # - Added code to handle random errors setting accounts that have been correctly found
18
19 [CmdletBinding()]
20 param (
21     #Common HSC PowerShell Parameters
22     [switch]$NoSessionTranscript,
23     [string]$LogFilePath = "$PSScriptRoot\Logs",
24     [switch]$StopOnError, #true is used for testing purposes
25     [int]$DaysToKeepLogFiles = 7 #this value used to clean old log files
26 )
27
28 $Error.Clear()
29 Set-Location $PSScriptRoot
30
31 #Build path to HSC PowerShell Modules
32 $PathToHSCPowerShellModules = $PSScriptRoot
33 $PathToHSCPowerShellModules =
34 $PathToHSCPowerShellModules.substring(0,$PathToHSCPowerShellModules.lastIndexOf("\")+1)
35 $PathToHSCPowerShellModules += "1HSC-PowerShell-Modules"
36 Write-Output $PathToHSCPowerShellModules
37
38 #Attempt to load common code module
39 $CommonCodeModule = $PathToHSCPowerShellModules + "\HSC-CommonCodeModule.psm1"
40 Write-Output "Path to common code module: $CommonCodeModule"
41 Import-Module $CommonCodeModule -Force -ArgumentList
42 $NoSessionTranscript,$LogFilePath,$true,$DaysToKeepLogFiles
43
44 #Attempt to load HSC Office 365 Module
45 $Office365Module = $PathToHSCPowerShellModules + "\HSC-Office365Module.psm1"
46 Write-Output "Path to HSC Office 365 module: $Office365Module"
47 Import-Module $Office365Module -Force
48
49 if ($Error.Count -gt 0)
50 {
51     #Any errors at this point are from loading modules. Program must stop.
52     Write-Warning "Error loading HSC PowerShell modules. Program is exiting."
53     return
54 }
55
56 #####
57 #Configure environment block
58 #####
59 Write-Verbose "Getting Parameter Information"
60 Get-Parameter -ParameterList $PSBoundParameters
61
62 Set-Environment
63 Set-WindowTitle
64
65 #See this page to understand what is going on here.
66 #https://www.thecloudjournal.net/2016/07/create-your-own-powershell-module-for-exchange-online/

```

```

65 ConnectTo-Office365 #from Office 365 module
66 Import-Module ExchangeOnline -Force #comes from HSC-Office365Module.psm1
67
68 #Remove old log files
69 Remove-OldLogFile -TXT -Path $LogFilePath -Days $DaysToKeepLogFiles -Verbose -Delete
70
71 [int]$UserNumber = 0
72
73 if ($Error.Count -gt 0)
74 {
75     #Any errors at this point are from configuring the environment. Program must stop.
76     Write-Warning "Error configuring environment. Program is exiting."
77     Exit-Commands
78 }
79
80 #####
81 #End of environment configuration block
82 #####
83
84 #####
85 #Begin Main Program#
86 #####
87
88 Write-Output "`nGenerating list of mailboxes..."
89
90 $Mailboxes = Get-CasMailbox -ResultSize Unlimited | where {($_.IMAPEnabled -eq $true)
91 -OR ($_.POPEnabled -eq $true)}
92
93 if (($Mailboxes | Measure-Object).Count -eq 0)
94 {
95     Write-Output "POP and IMAP are disabled on all mailboxes.`n"
96 }
97 else
98 {
99     foreach ($Mailbox in $Mailboxes)
100     {
101         $UserNumber++
102         Write-Output "Current user number: $UserNumber"
103         Write-Output $("PrimarySMTPAddress: " + $Mailbox.PrimarySMTPAddress)
104         Write-Output $("Mailbox DistinguishedName: " + $Mailbox.DistinguishedName)
105         Write-Output $("IMAPEnabled: " + (Get-CasMailbox
106 $Mailbox.PrimarySMTPAddress).IMAPEnabled)
107         Write-Output $("POPEnabled: " + (Get-CasMailbox
108 $Mailbox.PrimarySMTPAddress).POPEnabled)
109
110         try
111         {
112             If ($($mailbox.PrimarySMTPAddress) -like "FPAdmin*")
113             {
114                 #This is an exclusion to allow IMAP on these mailboxes only
115                 Write-Output $($mailbox.PrimarySMTPAddress + " skipped - only disabling
116 POP")
117                 Set-CasMailbox -Identity $Mailbox.PrimarySMTPAddress -POPEnabled $false
118                 -ErrorAction Stop
119                 Write-Output "POP Disabled"
120             }
121             else
122             {
123                 #POP and IMAP disables have been split up due to issue with proxy
124                 address licenses.
125
126                 if ($Mailbox.IMAPEnabled)
127                 {
128                     Write-Output "Disabling IMAP"
129
130                     try

```

```

126         {
127             #Mailbox | Set-CasMailbox -IMAPEnabled $false -ErrorAction Stop
128             Set-CasMailbox -Identity $Mailbox.DistinguishedName
129                 -IMAPEnabled $false -ErrorAction Stop
130         }
131     catch
132     {
133         #Error | fl
134         Write-Warning "Error disabling IMAP. Trying again."
135
136         try
137         {
138             Set-CasMailbox -Identity $Mailbox.PrimarySMTPAddress
139                 -IMAPEnabled $false -ErrorAction Stop
140         }
141         catch
142         {
143             #Error | FL
144             Write-Warning "Unable to disable IMAP"
145         }
146     }
147
148     Write-Output "IMAP has been disabled."
149 }
150 else
151 {
152     Write-Output "IMAP Is already disabled."
153 }
154
155 if ($Mailbox.POPEnabled)
156 {
157     Write-Output "Disabling POP"
158     try
159     {
160         Set-CasMailbox -Identity $Mailbox.DistinguishedName -POPEnabled
161             $false -ErrorAction Stop
162     }
163     catch
164     {
165         Write-Warning "Error disabling POP. Trying again."
166
167         try
168         {
169             Set-CasMailbox -Identity $Mailbox.PrimarySMTPAddress
170                 -POPEnabled $false -ErrorAction Stop
171         }
172         catch
173         {
174             #Error | FL
175             Write-Warning "Unable to disable POP"
176         }
177     }
178
179     Write-Output "POP has been disabled."
180 }
181 else
182 {
183     Write-Output "POP is already disabled"
184 }
185 }
186 }
187 catch
188 {
189     Write-Warning $("Error Setting " + $Mailbox.PrimarySMTPAddress)
190     Write-Warning "Trying again"
191 }

```

```
189         try
190         {
191             Set-CasMailbox -Identity $Mailbox.DistinguishedName -POPEnabled $false
                -IMAPEnabled $false -ErrorAction Stop
192         }
193         catch
194         {
195             Write-Error "Unable to disable POP & IMAP"
196
197             if ($StopOnError)
198             {
199                 Write-Warning "Program is exiting"
200                 Exit-Command
201             }
202         }
203     }
204     Finally
205     {
206         $Error.Clear()
207     }
208
209     Write-Output "*****"
210 }
211
212
213 if (!$NoSessionTranscript)
214 {
215     Stop-Transcript
216 }
217
218 Exit
```