

```
1 # Get-ADComputerInformationForMigration.ps1
2 # Written by: Jeff Brusoe
3 # Last Updated: September 28, 2021
4 #
5 # This file collects computer information from AD
6 # which is used to backup info for the AD migration project
7
8 [CmdletBinding()]
9 param(
10     [ValidateNotNullOrEmpty()]
11     [string]$OutputFilePath = (Get-HSCGitHubRepoPath) +
12         "4ADMigrationProject\"
13 )
14
15 try {
16     Set-HSCEnvironment -ErrorAction Stop
17
18     $OSInfoLog = $OutputFilePath + "ComputerOSInfo\" +
19         (Get-Date -Format yyyy-MM-dd) +
20         "-OSInfo.csv"
21
22     $ExtensionAttributeLog = $OutputFilePath + "ComputerExtensionAttributes\" +
23         (Get-Date -Format yyyy-MM-dd) +
24         "-ADComputerExtensionAttributes.csv"
25
26     $ADComputerCountLog = $OutputFilePath + "ComputerCount\" +
27         "ADComputerCount.csv"
28
29     $EnabledDisabledLog = $OutputFilePath + "ComputerEnabledDisabled\" +
30         (Get-Date -Format yyyy-MM-dd) +
31         "-ADComputerEnabledDisabled.csv"
32
33     $GroupMembershipLog = $OutputFilePath + "ComputerGroupMembership\" +
34         (Get-Date -Format yyyy-MM-dd) +
35         "-ADComputerGroupMembership.csv"
36
37     $PasswordLog = $OutputFilePath + "ComputerPassword\" +
38         (Get-Date -Format yyyy-MM-dd) +
39         "-ADComputerPassword.csv"
40
41     $LastLogonLog = $OutputFilePath + "ComputerLastLogon\" +
42         (Get-Date -Format yyyy-MM-dd) +
43         "-ADComputerLastLogon.csv"
44
45     $SPNLog = $OutputFilePath + "ComputerSPN\" +
46         (Get-Date -Format yyyy-MM-dd) +
47         "-ADComputerSPN.csv"
48
49     $AllComputerInfoLog = $OutputFilePath + "ComputerAllInfo\" +
50         (Get-Date -Format yyyy-MM-dd) +
51         "-AllComputerInfo.csv"
52
53     $AllComputerInfoLogUnfiltered = $OutputFilePath + "ComputerAllInfo\" +
54         (Get-Date -Format yyyy-MM-dd) +
55         "-AllComputerInfoUnfiltered.csv"
56
57     New-Item -ItemType File -Force -Path $OSInfoLog -ErrorAction Stop
58     New-Item -ItemType File -Force -Path $ExtensionAttributeLog -ErrorAction Stop
59     New-Item -ItemType File -Force -Path $EnabledDisabledLog -ErrorAction Stop
60     New-Item -ItemType File -Force -Path $GroupMembershipLog -ErrorAction Stop
```

```

61     New-Item -ItemType File -Force -Path $PasswordLog -ErrorAction Stop
62     New-Item -ItemType File -Force -Path $LastLogonLog -ErrorAction Stop
63     New-Item -ItemType File -Force -Path $SPNLog -ErrorAction Stop
64     New-Item -ItemType File -Force -Path $AllComputerInfoLog -ErrorAction Stop
65     New-Item -ItemType File -Force -Path $AllComputerInfoLogUnfiltered -ErrorAction
    Stop
66 }
67 catch {
68     Write-Warning "Unable to configure environment"
69     Invoke-HSCExitCommand -ErrorCount $Error.Count
70 }
71
72 try {
73     Write-Output "Getting list of AD Computers"
74
75     $ADComputers = Get-ADComputer -Filter * -Properties *
76 }
77 catch {
78     Write-Warning "Unable to get list of AD computers"
79     Invoke-HSCExitCommand -ErrorCount $Error.Count
80 }
81
82 $ComputerCount = 1
83 foreach ($ADComputer in $ADComputers)
84 {
85     Write-Output $("Current Computer: " + $ADComputer.Name)
86     Write-Output "Computer Count: $ComputerCount"
87     $ComputerCount++
88
89     Write-Output "`nLogging Computer OS Info"
90     $OSInfoObject = [PSCustomObject]@{
91         SamAccountName = $ADComputer.SamAccountName
92         ComputerName = $ADComputer.Name
93         ComputerCreationDate = $ADComputer.whenCreated
94         ComputerChanged = $ADComputer.whenChanged
95         OperatingSystem = $ADComputer.OperatingSystem
96         OperatingSystemVersion = $ADComputer.OperatingSystemVersion
97         OperatingSystemServicePack = $ADComputer.OperatingSystemServicePack
98         OperatingSystemHotfix = $ADComputer.OperatingSystemHotfix
99         DistinguishedName = $ADComputer.DistinguishedName
100     }
101
102     $OSInfoObject
103
104     try {
105         $OSInfoObject |
106         Export-Csv -Path $OSInfoLog -ErrorAction Stop -NoTypeInformation -
    Append
107     }
108     catch {
109         Write-Warning "Unable to log computer OS ifno"
110     }
111
112     Write-Output "`n`nLogging Computer Extension Attributes"
113     $ExtensionAttributeObject = [PSCustomObject]@{
114         SamAccountName = $ADComputer.SamAccountName
115         ComputerName = $ADComputer.Name
116         ComputerCreationDate = $ADComputer.whenCreated
117         ComputerChanged = $ADComputer.whenChanged
118         extensionAttribute1 = $ADComputer.extensionAttribute1

```

```

119     extensionAttribute2 = $ADComputer.extensionAttribute2
120     extensionAttribute3 = $ADComputer.extensionAttribute3
121     extensionAttribute4 = $ADComputer.extensionAttribute4
122     extensionAttribute5 = $ADComputer.extensionAttribute5
123     extensionAttribute6 = $ADComputer.extensionAttribute6
124     extensionAttribute7 = $ADComputer.extensionAttribute7
125     extensionAttribute8 = $ADComputer.extensionAttribute8
126     extensionAttribute9 = $ADComputer.extensionAttribute9
127     extensionAttribute10 = $ADComputer.extensionAttribute10
128     extensionAttribute11 = $ADComputer.extensionAttribute11
129     extensionAttribute12 = $ADComputer.extensionAttribute12
130     extensionAttribute13 = $ADComputer.extensionAttribute13
131     extensionAttribute14 = $ADComputer.extensionAttribute14
132     extensionAttribute15 = $ADComputer.extensionAttribute15
133     DistinguishedName = $ADComputer.DistinguishedName
134 }
135
136 $ExtensionAttributeObject
137
138 try {
139     $ExtensionAttributeObject |
140     Export-Csv -Path $ExtensionAttributeLog -ErrorAction Stop -
NoTypeInfoInformation -Append
141 }
142 catch {
143     Write-Warning "Unable to get computer extension attributes"
144 }
145
146 Write-Output "`n`nLogging Computer Enabled/Disabled Status"
147 $EnabledDisabledObject = [PSCustomObject]@{
148     SamAccountName = $ADComputer.SamAccountName
149     ComputerName = $ADComputer.Name
150     ComputerCreationDate = $ADComputer.whenCreated
151     ComputerChanged = $ADComputer.whenChanged
152     Enabled = $ADComputer.Enabled
153     DistinguishedName = $ADComputer.DistinguishedName
154 }
155
156 $EnabledDisabledObject
157 try {
158     $EnabledDisabledObject |
159     Export-Csv -Path $EnabledDisabledLog -ErrorAction Stop -
NoTypeInfoInformation -Append
160 }
161 catch {
162     Write-Warning "Unable to get enabled/disabled status"
163 }
164
165 Write-Output "`n`nLogging Computer Group Membership"
166 $GroupMembershipObject = [PSCustomObject]@{
167     SamAccountName = $ADComputer.SamAccountName
168     ComputerName = $ADComputer.Name
169     ComputerCreationDate = $ADComputer.whenCreated
170     ComputerChanged = $ADComputer.whenChanged
171     GroupMembership = $ADComputer.MemberOf -join ";"
172     DistinguishedName = $ADComputer.DistinguishedName
173 }
174
175 try {
176     $GroupMembershipObject |

```

```

177         Export-Csv -Path $GroupMembershipLog -ErrorAction Stop -
NoTypeInfoInformation -Append
178     }
179     catch {
180         Write-Warning "Unable to get group membership information"
181     }
182
183     Write-Output "`n`nLogging Computer Password Info"
184     $PasswordObject = [PSCustomObject]@{
185         SamAccountName = $ADComputer.SamAccountName
186         ComputerName = $ADComputer.Name
187         ComputerCreationDate = $ADComputer.whenCreated
188         ComputerChanged = $ADComputer.whenChanged
189         LastBadPasswordAttempt = $ADComputer.LastBadPasswordAttempt
190         PasswordExpired = $ADComputer.PasswordExpired
191         PasswordLastSet = $ADComputer.PasswordLastSet
192         PasswordNeverExpires = $ADComputer.PasswordNeverExpires
193         PasswordNotRequired = $ADComputer.PasswordNotRequired
194         DistinguishedName = $ADComputer.DistinguishedName
195     }
196
197     $PasswordObject
198     try {
199         $PasswordObject |
200         Export-Csv -Path $PasswordLog -ErrorAction Stop -NoTypeInfoInformation -
Append
201     }
202     catch {
203         Write-Warning "Unable to pull computer password info"
204     }
205
206     $LastLogonInfoObject = [PSCustomObject]@{
207         SamAccountName = $ADComputer.SamAccountName
208         ComputerName = $ADComputer.Name
209         ComputerCreationDate = $ADComputer.whenCreated
210         ComputerChanged = $ADComputer.whenChanged
211         LastLogon = $ADComputer.LastLogonDate
212         LastLogonTimeStamp =
[DateTime]::FromFileTime($ADComputer.lastLogonTimeStamp)
213         DistinguishedName = $ADComputer.DistinguishedName
214     }
215
216     $LastLogonInfoObject
217     $LastLogonInfoObject |
218     Export-Csv -Path $LastLogonLog -ErrorAction Stop -NoTypeInfoInformation -Append
219
220     $ComputerSPNObject = [PSCustomObject]@{
221         SamAccountName = $ADComputer.SamAccountName
222         ComputerName = $ADComputer.Name
223         ComputerCreationDate = $ADComputer.whenCreated
224         ComputerChanged = $ADComputer.whenChanged
225         SPN = $ADComputer.servicePrincipalName -join ";"
226         DistinguishedName = $ADComputer.DistinguishedName
227     }
228
229     $ComputerSPNObject
230     $ComputerSPNObject |
231     Export-Csv -Path $SPNLog -ErrorAction Stop -NoTypeInfoInformation -Append
232
233     $AllComputerInfoObject = [PSCustomObject]@{

```

```

234     SamAccountName = $ADComputer.SamAccountName
235     ComputerName = $ADComputer.Name
236     ComputerCreationDate = $ADComputer.whenCreated
237     ComputerChanged = $ADComputer.whenChanged
238     Enabled = $ADComputer.Enabled
239     OperatingSystem = $ADComputer.OperatingSystem
240     OperatingSystemVersion = $ADComputer.OperatingSystemVersion
241     OperatingSystemServicePack = $ADComputer.OperatingSystemServicePack
242     OperatingSystemHotfix = $ADComputer.OperatingSystemHotfix
243     extensionAttribute1 = $ADComputer.extensionAttribute1
244     extensionAttribute2 = $ADComputer.extensionAttribute2
245     extensionAttribute3 = $ADComputer.extensionAttribute3
246     extensionAttribute4 = $ADComputer.extensionAttribute4
247     extensionAttribute5 = $ADComputer.extensionAttribute5
248     extensionAttribute6 = $ADComputer.extensionAttribute6
249     extensionAttribute7 = $ADComputer.extensionAttribute7
250     extensionAttribute8 = $ADComputer.extensionAttribute8
251     extensionAttribute9 = $ADComputer.extensionAttribute9
252     extensionAttribute10 = $ADComputer.extensionAttribute10
253     extensionAttribute11 = $ADComputer.extensionAttribute11
254     extensionAttribute12 = $ADComputer.extensionAttribute12
255     extensionAttribute13 = $ADComputer.extensionAttribute13
256     extensionAttribute14 = $ADComputer.extensionAttribute14
257     extensionAttribute15 = $ADComputer.extensionAttribute15
258     GroupMembership = $ADComputer.MemberOf -join ";"
259     LastBadPasswordAttempt = $ADComputer.LastBadPasswordAttempt
260     PasswordExpired = $ADComputer.PasswordExpired
261     PasswordLastSet = $ADComputer.PasswordLastSet
262     PasswordNeverExpires = $ADComputer.PasswordNeverExpires
263     PasswordNotRequired = $ADComputer.PasswordNotRequired
264     LastLogon = $ADComputer.LastLogonDate
265     LastLogonTimeStamp =
[DateTime]::FromFileTime($ADComputer.lastLogonTimeStamp)
266     SPN = $ADComputer.servicePrincipalName -join ";"
267     DistinguishedName = $ADComputer.DistinguishedName
268 }
269
270 $AllComputerInfoObject
271 $AllComputerInfoObject |
272     Export-Csv -Path $AllComputerInfoLog -ErrorAction Stop -NoTypeInformation -
Append
273
274     Write-Output "*****"
275 }
276
277 Write-Output "Writing All Computer Info (Unfiltered)"
278 $ADComputers |
279     Export-Csv -Path $AllComputerInfoLogUnfiltered
280
281 Write-Output "Logging Count of AD Computer"
282 [PSCustomObject]@{
283     Date = Get-Date -Format yyyy-MM-dd-HH-mm
284     ADComputerCount = $ADComputers.Count
285 } | Export-Csv -Path $ADComputerCountLog -NoTypeInformation -Append
286
287 Invoke-HSCEExitCommand -ErrorCount $Error.Count

```