

```
1 [CmdletBinding()]
2 param ()
3
4 function Get-HSCConnectionString
5 {
6     [CmdletBinding()]
7     Param (
8
9         [ValidateNotNullOrEmpty()]
10        [string]$DataSource="sql01.hsc.wvu.edu",
11        [ValidateNotNullOrEmpty()]
12        [string]$Database = "BannerData",
13        [ValidateNotNullOrEmpty()]
14        [string]$Username = "itsnetworking",
15
16        [Parameter(Mandatory=$true)]
17        [Alias("Password")]
18        [string]$SQLPassword
19    )
20
21    $ConnectionString = "Data Source=$DataSource;Initial Catalog=$Database;User
22    Id=$Username;Password=$SQLPassword;"
23
24    $CleanedConnectionString =
25    $ConnectionString.substring(0,$ConnectionString.lastIndexOf("=")+1) + "
26    <PasswordRemoved>"
27    Write-Output "Connection string: $CleanedConnectionString" | Out-Host
28
29    return $ConnectionString
30 }
31
32 function Get-HSCSQLConnectionString0365
33 {
34     [CmdletBinding()]
35     Param (
36
37         [ValidateNotNullOrEmpty()]
38        [string]$DataSource="hscpowershell.database.windows.net",
39        [ValidateNotNullOrEmpty()]
40        [string]$Database = "HSCPowerShell",
41        [ValidateNotNullOrEmpty()]
42        [string]$Username = "HSCPowershell",
43
44        [Parameter(Mandatory=$true)]
45        [Alias("Password")]
46        [string]$SQLPassword
47    )
48
49    $ConnectionString = "Data Source=$DataSource;Initial Catalog=$Database;User
50    Id=$Username;Password=$SQLPassword;"
51
52    $CleanedConnectionString =
53    $ConnectionString.substring(0,$ConnectionString.lastIndexOf("=")+1) + "
54    <PasswordRemoved>"
55    Write-Output "Connection string: $CleanedConnectionString" | Out-Host
56
57    return $ConnectionString
58 }
59
60 function Get-HSCSQLPassword
```

```

55 {
56     <#
57     .SYNOPSIS
58         This function decrypts the SQL connection file.
59
60     .NOTES
61         Written by: Jeff Brusoe
62         Last Updated: September 9, 2020
63     #>
64
65     [CmdletBinding()]
66     param (
67         [ValidateNotNull()]
68         [string]$SQLConnectionFile = (Get-HSCSQLSecureFile),
69
70         [switch]$SOLEDB
71     )
72
73     begin
74     {
75         if ($SOLEDB)
76         {
77             $SQLConnectionFile = Get-HSCSQLSecureFile -SOLEDB
78         }
79
80         Write-Verbose "Preparing to decrypt SQL Connection File"
81         Write-Verbose "SQL Connection File: $SQLConnectionFile"
82
83         [string]$SQLPassword = $null
84     }
85
86     process
87     {
88         if (($null -ne $SQLConnectionFile) -AND (Test-Path $SQLConnectionFile))
89         {
90             Write-Verbose "Connection file exists and is being decrypted"
91
92             $PwdString = Convertto-SecureString -String (Get-Content $SQLConnectionFile)
93             $bstr =
94             [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($PwdString)
95             $SQLPassword =
96             [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($bstr)
97         }
98         else {
99             Write-Warning "Invalid SQL Connection File"
100             Invoke-HSCExitCommand -ErrorCount $Error.Count
101         }
102     }
103
104     end
105     {
106         return $SQLPassword
107     }
108 }
109
110 function Get-HSCSQLSecureFile
111 {
112     <#
113     .SYNOPSIS

```

```
112     This function gets the path to the SQL Server connection file for that
particular server.
113     By default, it assumes the connection should go to the HSC 0365 SQL
instance.
114
115     .PARAMETER SOLEDB
116     Indicates that the SQL connection file for SOLE should be used instead of the
0365 instance
117
118     .NOTES
119     Written by: Jeff Brusoe
120     Last Updated: September 9, 2020
121     #>
122
123     [CmdletBinding()]
124     param (
125         [switch]$SOLEDB
126     )
127
128     begin
129     {
130         $ServerName = Get-HSCServerName
131         Write-Verbose "ServerName: $ServerName"
132
133         $ConnectionFile = Get-HSCEncryptedDirectoryPath
134         Write-Verbose "Encrypted Directory Path: $ConnectionFile"
135     }
136
137     process
138     {
139         if ($ServerName -eq "HSVDIWIN10JB" -AND $SOLEDB)
140         {
141             $ConnectionFile += "SOLESQL-HSVDIWIN10JB.txt"
142         }
143         elseif ($ServerName -eq "HSVDIWIN10JB")
144         {
145             $ConnectionFile = $ConnectionFile + "0365SQLInstance-HSVDIWIN10JB.txt"
146         }
147         elseif ($ServerName -eq "DESKTOP-1MQ9DJ0")
148         {
149             $ConnectionFile = $ConnectionFile + "0365SQLInstance-JeffSurface.txt"
150         }
151         else {
152             $ServerNumber = $ServerName.substring($ServerName.Length-1)
153             Write-Verbose "Server Number: $ServerNumber"
154
155             if ($SOLEDB) {
156                 $ConnectionFile += "SOLESQL$ServerNumber.txt"
157             }
158             else {
159                 $ConnectionFile += "0365SqlInstance$ServerNumber.txt"
160             }
161
162             Write-Verbose "Connection File: $ConnectionFile"
163         }
164     }
165
166     end
167     {
168         return $ConnectionFile
169     }
```

```
169 }  
170 }
```