

Effiziente Administration Ihrer Netzwerkumgebung

Inbetriebnahme Azure Modul

UNICAT
unified computing and
technology gmbh

Inhaltsverzeichnis

1. Voraussetzungen	3
2. Umgebung in HDB discovern	4
3. Zugriff über einen http(s) Proxy	4
4. Powershell Modulfunktion und Elementaroperation	5
5. Session Handling	5
6. Zugriff auf Exchange Online	6
6.1 SYNCHRONISIEREN UND DISCOVERN VON POSTFÄCHERN	7
7. Alternative Anmeldung	7
8. Connection Parameter	7
9. Connect Beispiel AzureAD	7
10. Anhang1: Erzwingung TLS 1.2 für Cloud Kommunikation	8

1. Voraussetzungen

Installieren sie den Microsoft Manangement Framework 5.1:

<https://www.microsoft.com/en-us/download>

Installieren Sie die erforderlichen Commandlets für Azure Cloud:

Um auf die Microsoft cloud verbinden zu können und Azure- und MSO-Commandlets nutzen zu können, müssen die folgenden Pakete auf jedem Operations Manager Server installiert werden:

Powershell:

Install-Module -Name AzureAD

Install-Module MSOnline

Ggfs. Register-PSRepository -Default -Verbose

Ggfs. Set-PSRepository -Name "PSGallery" -InstallationPolicy Trusted

ExchangeOnline 2.0:

Set-ItemProperty -Path

'HKLM:\SOFTWARE\Wow6432Node\Microsoft\.NetFramework\v4.0.30319' -Name 'SchUseStrongCrypto' -Value '1' -Type DWord

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\.NetFramework\v4.0.30319' -Name 'SchUseStrongCrypto' -Value '1' -Type DWord

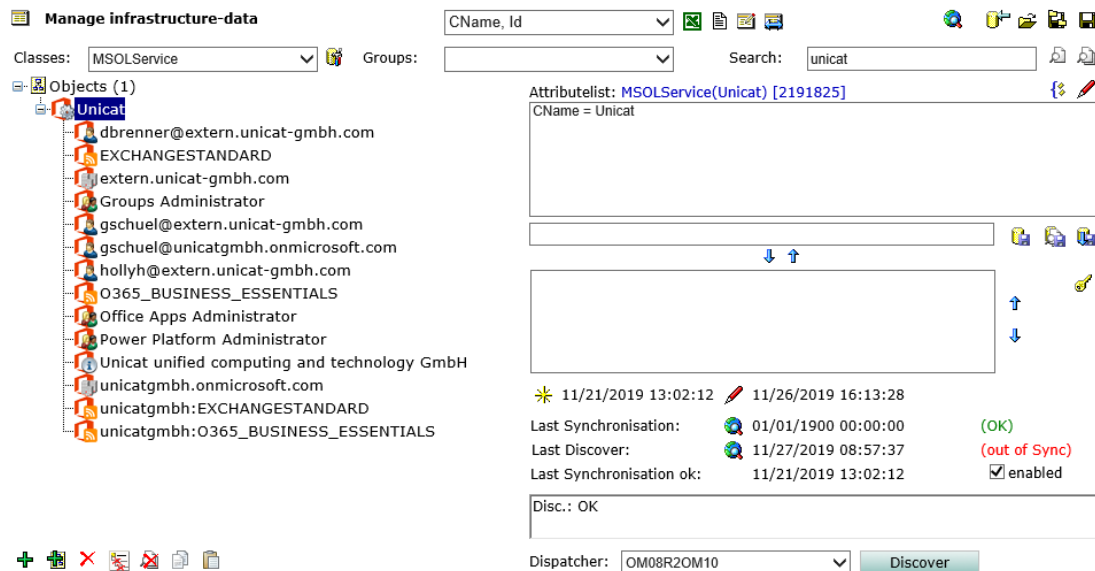
Check protocols: [Net.ServicePointManager]::SecurityProtocol

Install-Module PowershellGet -Force

Install-Module -Name ExchangeOnlineManagement -Force

2. Umgebung in HDB discovern

Um die Microsoft Service Objekte in die HDB zu laden muss ein Objekt der Klasse ‚MSOLService‘ erstellt werden. Der Name ist zunächst ohne Bedeutung – wählen Sie am besten die Bezeichnung Ihrer Organisation:



Hinterlegen Sie nun Anmeldeinformationen für das HDB-Objekt. Hier sollten Sie ein Microsoft Online Konto verwenden, welches Administrationsberechtigungen für den Service besitzt. In aller Regel ist das etwa Admin@{Organization}.onmicrosoft.com.

Nun kann über den üblichen Weg mit Discover die Objektstruktur eingelesen werden.

3. Zugriff über einen http(s) Proxy

Da der Dispatcher und damit die OMExecute(32/64) im Benutzerkontext des ‚LocalSystem‘ Benutzers ausgeführt werden, muss der Proxy für diesen Benutzer konfiguriert werden.

```
C:\windows\System32\bitsadmin.exe /Util /SetIEProxy LocalSystem Manual_proxy
http://<proxyserver>:<proxy port> "<Any bypasses to be added>"
```

Wenn die Option ‚Bypass proxy server for local addresses‘ aktiviert werden soll, lautet der Befehl

```
C:\windows\System32\bitsadmin.exe /Util /SetIEProxy LocalSystem Manual_proxy
http://<proxyserver>:<proxy port> "local"
```

Anschließend muss der Server zwingend neu gestartet werden!

Der MSOL-Connect nutzt die Proxy-Einstellungen des IE.

Für Exchange-Online muss ggfs. an der Klasse MSOLService das Attribut ‚ExchangeOnlineProxyAccessType‘ auf den Wert IEConfig gesetzt werden.

Generell kann für .net Framework der Proxy in der machine.config hinterlegt werden:

C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\machine.config:
und
C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\machine.config:

```
<system.net>
  <defaultProxy>
    <proxy
      usesystemdefault="true"
      proxyaddress="http://<PROXYADDRESS>:<PROXYPORT>"
      bypassonlocal="true"
    />
    <bypasslist>
      <add address="[a-z]+\.\contoso\.com$" />
    </bypasslist>
  </defaultProxy>
</system.net>
```

Erklärung ist zu finden unter:

[Azure AD Connect: Voraussetzungen und Hardware | Microsoft Docs](#)

Unter Umständen ist eine Manipulation wegen Seiteneffekten auf andere .net Anwendungen nicht immer möglich. Daher kann grundsätzlich für Azure-Connect und Exchange-Connect der Proxy im IE für das Dienstkonto was sich unter dem ausführenden Host in der HDB befindet mit den Proxy-Einstellungen konfiguriert werden. Vor Ausführung der MSOL-Eos wird jeweils mit dem Dienstkonto unter dem Host lokal impersoniert (Kontextwechsel).

4. Powershell Modulfunktion und Elementaroperation

Die folgenden Powershell-Befehle werden in den Powershell Funktionen für MSOL vorangestellt:

```
import-module -name "MSOnline"
$password = ConvertTo-SecureString "{Pwd}" -AsPlainText -Force
$Credential = New-Object PSCredential ("{User}", $password)
Connect-MSOLService -Credential $Credential
```

5. Session Handling

Die gleichzeitige Anzahl von Sitzungen wird durch die Verwendung von Semaphoren gesteuert und begrenzt. Die Anzahl der Sitzungen kann an der Klasse ‚MSOLService‘ für die einzelnen Subsysteme festgelegt werden:

MSOLService.AzureADMaxSessionCount
MSOLService.ExchangeOnlineMaxSessionCount
MSOLService.MicrosoftTeamsMaxSessionCount
MSOLService.MSOLServiceMaxSessionCount

Der Wert wird als Ganzzahl angegeben, der Standardwert ist 1.

6. Zugriff auf Exchange Online

Um die Elementaroperationen für den Zugriff auf Exchange Online verwenden zu können, müssen vorab auf jedem Dispatcher die benötigten Komponenten installiert werden:

```
Install-Module PowerShellGet -Force  
Install-Module -Name ExchangeOnlineManagement
```

Hinweis: Auf dem Server muss TLS 1.2 aktiv sein, ansonsten kann die Installation nicht vorgenommen werden!

TLS 1.2 aktivieren

Um keinen größeren Eingriff auf dem Server vorzunehmen, kann über die Powershell TLS 1.2 temporär aktiviert werden:

```
[Net.ServicePointManager]::SecurityProtocol =  
[Net.SecurityProtocolType]::Tls12
```

Die eigentliche Verbindung erfolgt dann mit folgendem Script im Hintergrund:

```
import-module -name "ExchangeOnlineManagement"  
$password = ConvertTo-SecureString "{Pwd}" -AsPlainText -Force  
$Credential = New-Object PSCredential ("{User}", $password)  
Connect-ExchangeOnline -Credential $Credential
```

Wenn Exchange Online über einen Proxy erreicht wird und an der Klasse MSOLService das Attribut ‚ExchangeOnlineProxyAccessType‘ auf einen der Werte ‚IEConfig‘, ‚WinHttpConfig‘ oder ‚Auto‘ gesetzt wurde, erfolgt die eigentliche Verbindung im Hintergrund mit folgendem Script:

```
import-module -name "ExchangeOnlineManagement"  
$proxysettings = New-PSSessionOption -ProxyAccessType {MSOLService.  
ExchangeOnlineProxyAccessType}  
$password = ConvertTo-SecureString "{Pwd}" -AsPlainText -Force  
$Credential = New-Object PSCredential ("{User}", $password)  
Connect-ExchangeOnline -Credential $Credential -PSSessionOption $  
proxysettings
```

6.1 Synchronisieren und Discovern von Postfächern

In der Klasse MSOLMailbox werden die Online Postfachdaten gespeichert.

Die Objekte liegen unterhalb der MSOLDomain und können über ‚discover_MSOLDomain‘ (EO) ergänzt werden. Einzelne Mailboxen können mit ‚Sync_MSOLMailbox‘ synchronisiert werden. Es ist jedoch wegen dem Overhead des Connect empfehlenswert, die Synchronisation über ‚Sync_MSOLDomain‘ durch zu führen. Hierbei werden alle Postfächer, die sich seit der letzten Synchronisation (LastMailboxSync) geändert haben, aktualisiert. Synchronisationsattribute können wie üblich an der Klasse MSOLMailbox hinterlegt werden.

7. Alternative Anmeldung

Alternativ zu den üblichen Anmeldeinformationen Benutzername mit Kennwort kann auch die Anmeldung mit einem CertificateThumbprint oder anderen Faktoren erfolgen. Für diesen Fall sind u.U. die normalen Anmeldedaten zu unterdrücken. Dies kann über die folgenden Parameter am MSOLService erreicht werden:

ExchangeOnlineUseCredentials = 0
MicrosoftTeamsUseCredentials = 0
AzureADUseCredentials = 0

8. Connection Parameter

Ergänzend können die MSOL-Connect Commandlets mit weiteren Parametern ergänzt werden. Diese können im Attribut [Umgebung]ConnectionParameter des MSOLService Objekts gesetzt werden, z.B.

-TenantId *TenantId* -ApplicationId *ApplicationId* -CertificateThumbprint
CertificateThumbprint

Die Parameter werden dabei als Text unverändert hinter den Connection-Befehl angefügt. Sie werden verschlüsselt in der HDB hinterlegt.

9. Connect Beispiel AzureAD

```
$password = ConvertTo-SecureString "{pwd}" -AsPlainText -Force
$Credential = New-Object PSCredential ("{User}", $password)
Connect-AzureAD -Credential $Credential -TenantId "{TenantId}"
get-AzureADUser
```

10. Anhang1: Erzwingung TLS 1.2 für Cloud Kommunikation

```
If (-Not (Test-Path 'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319'))
{
    New-Item 'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319' -Force | Out-Null
}
New-ItemProperty -Path 'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319' -Name
'SystemDefaultTlsVersions' -Value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -Path 'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319' -Name
'SchUseStrongCrypto' -Value '1' -PropertyType 'DWord' -Force | Out-Null

If (-Not (Test-Path 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319'))
{
    New-Item 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319' -Force | Out-Null
}
New-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319' -Name
'SystemDefaultTlsVersions' -Value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319' -Name
'SchUseStrongCrypto' -Value '1' -PropertyType 'DWord' -Force | Out-Null

If (-Not (Test-Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server'))
{
    New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Server' -Force | Out-Null
}
New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server' -
Name 'Enabled' -Value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server' -
Name 'DisabledByDefault' -Value '0' -PropertyType 'DWord' -Force | Out-Null

If (-Not (Test-Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client'))
{
    New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Client' -Force | Out-Null
}
New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client' -
Name 'Enabled' -Value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client' -
Name 'DisabledByDefault' -Value '0' -PropertyType 'DWord' -Force | Out-Null

Write-Host 'TLS 1.2 has been enabled. You must restart the Windows Server for the changes to
take affect.' -ForegroundColor Cyan
```