**Programmation avec le GET**

Ce laboratoire doit être fait individuellement sur l'ordinateur virtuel 2

**Objectif**

* Utiliser les cmdlets de PowerShell afin de se familiariser avec les objets d’un domaine
* Maîtriser l’utilisation de plusieurs cmdlets GET du module ActiveDirectory

**Mise en place**

Pour conserver dans un fichier texte le contenu de l'aide sur le paramètre "-Filter" des cmdlets de l'Active Directory en exécutant la commande suivante:

**Accès à la structure du domaine**

Dans l’invite PowerShell exécuter les commandes suivantes et répondre aux questions.

**Get-ADForest**

Trouvez la valeur des propriétés suivantes:

|  |  |
| --- | --- |
| **Nom de la propriété** | **Valeur de la propriété** |
| ForestMode | **Windows2016Forest** |
| DomainNamingMaster | **SERVEUR1.FORMATION.LOCAL** |
| GlobalCatalogs | **{SERVEUR1.FORMATION.LOCAL}** |
| SchemaMaster | **SERVEUR1.FORMATION.LOCAL** |

**Get-ADDomain**

Trouvez la valeur des propriétés suivantes:

|  |  |
| --- | --- |
| **Nom de la propriété** | **Valeur de la propriété** |
| DistinguishedName | **DC=FORMATION,DC=LOCAL** |
| DNSRoot | **FORMATION.LOCAL** |
| DomainMode | **Windows2016Domain** |
| Name | **FORMATION** |
| Forest | **FORMATION.LOCAL** |

La commande qui retourne SEULEMENT la valeur de la propriété DistinguishedName de votre domaine:

**(Get-ADDomain).DistinguishedName**

La commande qui retourne SEULEMENT la valeur de la propriété DNSRoot de votre domaine:

**(Get-ADDomain).DNSRoot**

**Accès aux ordinateurs d’un domaine avec Get-ADComputer**

La commande qui affiche les principales propriétés de l'ordinateur SEVEUR2.

**Get-ADComputer -Identity "SERVEUR 2"**

La commande qui affiche toutes les propriétés de l'ordinateur SERVEUR2.

**Get-ADComputer -Identity "SERVEUR2" -Properties \***

La commande qui affiche les principales propriétés des ordinateurs dont le nom commence par "SERVEUR"

**Get-ADcomputer -Filter {Name -like "SERVEUR\*"}**

La commande qui affiche SEULEMENT le nom de tous les ordinateurs de votre domaine

**(Get-ADComputer -Filter \*).Name**

La commande qui affiche les propriétés "Name,DNSHostName,IPv4Address" de tous les ordinateurs de votre domaine

**Get-ADComputer -Filter \* -Properties IPv4Address | Format-Table Name,DNSHostName,IPv4Address -Autosize**

Voici une commande qui affiche plusieurs propriétés de tous les ordinateurs de votre domaine en triant deux paramètres en ordre croissant et en triant un paramètre en ordre décroissant.

**Get-ADComputer -Filter \* `**

**-Properties \* | `**

**Sort-Object -Property @{Expression = "Description"; Ascending = $true},**

**@{Expression = "OperatingSystem"; Ascending = $true},**

**@{Expression = "OperatingSystemVersion"; Descending = $true} | `**

**Format-Table Description,Name,OperatingSystem,OperatingSystemVersion,WhenChanged -AutoSize**

Voici le code qui permet d'afficher le nom de l'utilisateur qui a joint un ordinateur à l'Active Directory.

[**Si l’utilisateur a une délégation pour créer des objets dans l'Active Directory ou est un membre du groupe "Admins du domaine"**](https://www.reverso.net/translationresults.aspx?lang=FR&sourcetext=If%20the%20user%20has%20delegated%20rights%20to%20create%20computer%20objects,%20or%20is%20a%20member%20of%20domain%20admins%20then,%20ms-DS-CreatorSID%20is%20left%20empty&action_form=translate&direction_translation=eng-fra-7) **alors mS-DS-CreatorSID est vide.**

**$nom = Get-ADComputer -Identity SERVEUR2 `**

**-Properties mS-DS-CreatorSID | `**

**Select-Object -Expandproperty mS-DS-CreatorSID | `**

**Select-Object -ExpandProperty Value | `**

**Foreach-Object {Get-ADUser -Filter {SID -eq $PSItem}}**

**if ([string]::IsNullOrEmpty($nom))**

**{**

**Write-Host "mS-DS-CreatorSID est vide" -ForegroundColor Yellow**

**}**

**else**

**{**

**Write-Host "mS-DS-CreatorSID correspond à $nom" -ForegroundColor Cyan**

**}**

**Accès aux groupes d’un domaine avec Get-ADGroup**

La commande qui affiche les principales propriétés du groupe Administrateurs

**Get-ADGroup -Identity "Administrateurs"**

La commande qui affiche toutes les propriétés du groupe Administrateurs

**Get-ADGroup -Identity "Administrateurs" -Properties \***

La commande qui affiche les principales propriétés des groupes dont le nom commence par "Adm"

**Get-ADGroup -Filter {Name -like "Adm\*"}**

La commande qui affiche SEULEMENT le nom des groupes dont le nom commence par "Adm".

**(Get-ADGroup -Filter {Name -like "Adm\*"}).Name**

**Accès aux utilisateurs d’un domaine avec Get-ADUser**

La commande qui affiche les principales propriétés de l’utilisateur dont le nom d’ouverture de session est EMP01

**Get-ADuser -Identity "EMP01"**

La commande qui affiche toutes les propriétés de l’utilisateur dont le nom d’ouverture de session est EMP01

**Get-ADuser -Identity "EMP01" -Properties \***

La commande qui affiche les principales propriétés des utilisateurs dont le nom de famille est Coutu*.*

**Get-ADuser -Filter 'Surname -eq "Coutu"'**

La commande qui affiche les principales propriétés des utilisateurs dont le nom d’ouverture de session débute par "EMP".

**Get-ADUser -Filter {SamAccountName -like "EMP\*"}**

La commande qui affiche les principales propriétés des utilisateurs qui sont dans la OU "FORMATION" qui est directement sous le domaine "FORMATION.LOCAL".

**$emplacement= "OU=formation,DC=formation,DC=local"**

**Get-ADUser -Filter \* -SearchBase $emplacement -SearchScope Subtree**

Voici le code qui permet d’afficher les utilisateurs dont le nom débute par 0,1,2,3,4,5,6,7,8,9.

**for ($i = 0; $i -le 9; $i++)**

**{**

**$nom = -join ($i,"\*")**

**Get-ADUser -Filter {Name -like $nom}**

**}**

**Trouver la liste des utilisateurs dont le DistinguishedName se termine par ou=formation,dc=formation,dc=local**

**Essai 1**: Le paramètre **-Filter** ne fonctionne pas parce que l'attribut "**DistinguishedName**" est un attribut étendu.

**~~Get-ADuser -Filter {DistinguishedName -like "\*ou=formation,dc=formation,dc=local"}~~**

**Essai 2**: On doit utiliser **Where-Object** qui s'exécute sur le résultat de la commande qui est à gauche.

**Get-ADUser -Filter \* | `**

**Where-Object {$PSItem.DistinguishedName -like "\*ou=formation,dc=formation,dc=local"}**

**Trouver les utilisateurs dans l'Active Directory qui ont des propriétés qui ne sont pas configurées.**

La liste des utilisateurs qui n'ont pas d'adresse de messagerie en utilisant **-Filter**

**$sb = (Get-ADDomain).DistinguishedName**

**Get-ADUser -SearchBase $sb -Filter {mail -notlike "\*"} -Properties \***

La liste des utilisateurs qui n'ont pas de "Gestionnaire" en utilisant **-Filter**

L'attribut "**manager**" contient le "**DistinguishedName**" du gestionnaire.

**Essai 1**: Le paramètre **-Filter** ne fonctionne pas, parce que l'attibut "**manager**" est un attribut étendu.

**~~Get-ADUser -SearchBase $sb -Filter {manager -notlike "\*"} -Properties \*~~**

**Essai 2**: On doit utiliser **Where-Object** qui s'exécute sur le résultat de la commande qui est à gauche.

**Get-ADUser -SearchBase $sb -Filter \* -Properties \* | Where-Object {$PSitem.manager -eq $null}**

**Utilisation du paramètre -LDAPFilter avec Get-ADUser**

Il est possible d'effectuer des requêtes en utilisant la syntaxe LDAP (Lightweight Directory Access Protocol).

* LDAP est un protocole ouvert et multiplateforme utilisé pour l’authentification des services d’annuaire.
* LDAP est un moyen de communiquer avec Active Directory.

**Get-Aduser** permet d'utiliser le paramètre **-LDAPFilter**.

Les opérateurs sont différents selon l'utilisation du paramètre -Filter ou l'utilisation du paramètre -LDAPFilter.

|  |  |
| --- | --- |
| **Opérateurs utilisés avec -Filter** | **Opérateurs utilisés avec -LDAPFilter** |
| **-eq** | **=** |
| **-ne** | **!x=y** |
| **-le** | **<=** |
| **-lt** | **!x >= y** |
| **-ge** | **>=** |
| **-gt** | **!x <= y** |
| **-like** | **=** |
| **-notlike** | **!x = y** |
| **-and** | **&** |
| **-or** | **|** |
| **-not** | **!** |

**Voici plusieurs exemples qui utilisent le paramètre -LDAPFilter avec Get-ADUser**

# le nom débute par R **ET** l'adresse de messagerie se termine par @cvm.qc.ca

**$q1 = "(&(cn=R\*)(mail=\*@cvm.qc.ca))"**

**Get-ADUser -SearchBase (Get-ADDomain).DistinguishedName -LDAPFilter $q1 -Properties \***

# le nom = Richard **OU** le nom = Michelle **OU** le nom = Patrick

**$q2 = "(|(cn=Richard)(cn=Michelle)(cn=Patrick))"**

**Get-ADUser -SearchBase (Get-ADDomain).DistinguishedName -LDAPFilter $q2 -Properties \***

# le prénom = Richard **ET** (la ville = Laval **OU** la ville = Verdun)

**$q3 = "(&(givenName=Richard)(|(l=Laval)(l=Verdun)))"**

**Get-ADUser -SearchBase (Get-ADDomain).DistinguishedName -LDAPFilter $q3 -Properties \***

# La liste des utilisateurs qui n'ont pas de "Gestionnaire" en utilisant **-LDAPFilter**

**$q4 = "(!(manager=\*))"**

**Get-ADUser -SearchBase $sb -LDAPFilter $q4 -Properties \***

**Accès aux unités d'organisation d’un domaine avec Get-ADOrganizationalUnit**

La commande qui affiche les principales propriétés de l’unité d'organisation FORMATION

**Get-ADOrganizationalUnit -Identity "OU=formation,DC=formation,DC=local"**

La commande qui affiche toutes les propriétés de l’unité d'organisation FORMATION

**Get-ADOrganizationalUnit -Identity "OU=formation,DC=formation,DC=local" -Properties \***

La commande qui affiche les principales propriétés des unités d'organisation dont le nom contient la suite de caractères "gestion"

**Get-ADOrganizationalUnit -Filter {Name -like "\*gestion\*"}**

La commande qui affiche les principales propriétés des unités d'organisation dont le nom commence par "inf"

**Get-ADOrganizationalUnit -Filter {Name -like "inf\*"}**

La commande qui affiche les principales propriétés de toutes les unités d'organisation de votre domaine

**Get-ADOrganizationalUnit -Filter \***

La commande qui affiche les propriétés "Name" et "DistinguishedName" de toutes les unités d'organisation de votre domaine

**Get-ADOrganizationalUnit -Filter \* | Format-Table Name,DistinguishedName -Autosize**

La commande qui affiche les propriétés "Name" et "Created" de toutes les unités de votre domaine

**Get-ADOrganizationalUnit -Filter \* -Properties created | Format-Table Name,Created -Autosize**

La commande qui affiche toutes les propriétés de toutes les unités d'organisation

**Get-ADOrganizationalUnit -Filter \* -Properties \***

La commande qui affiche le nom des unités d'organisation en utilisant la propriété CanonicalName

**Get-ADOrganizationalUnit -Filter \* `**

**-Properties CanonicalName `**

**| Select-Object -Property CanonicalName `**

**| Sort-Object CanonicalName**

**Exemples supplémentaires avec Get-ADOrganizationalUnit**

On peut enregistrer le résultat des cmdlets dans un fichier et spécifier le type d'encodage.

**# Malheuresement, l'utilisation du paramètre "-Encoding utf8" crée**

**# un fichier dont l'encodage sera "UTF8 avec BOM".**

**# Il n'y a pas de paramètre pour utiliser l'encodage "UTF8".**

**$NomFichier\_v1 = "C:\\_TEMP\info\_OU\_v1.txt"**

**Get-ADOrganizationalUnit -Filter \* -Properties \* `**

**| Out-File -FilePath $Nomfichier\_v1 -Encoding utf8**

**# Version améliorée du code pour forcer l'utilisation de l'encodage UTF8.**

**$Utf8NoBomEncoding = New-Object System.Text.UTF8Encoding $False**

**$NomFichier\_v2 = "C:\\_TEMP\info\_OU\_v2.TXT"**

**$contenu\_v2 = Get-ADOrganizationalUnit -Filter \* -Properties \***

**[System.IO.File]::WriteAllLines($NomFichier\_v2,$contenu\_v2,$Utf8NoBomEncoding)**

**$NomFichier\_v3 = "C:\\_TEMP\info\_OU\_v3.TXT"**

**$contenu\_v3 = Get-ADOrganizationalUnit -Filter \* `**

**-Properties CanonicalName `**

**| Select-Object -Property CanonicalName `**

**| Sort-Object CanonicalName | Out-String**

**[System.IO.File]::WriteAllLines($NomFichier\_v3,$contenu\_v3,$Utf8NoBomEncoding)**

**Accès aux objets d'un domaine avec Get-ADObject**

Requête qui permet d'afficher les utilisateurs de l'Active Directory

**Get-ADObject -Filter {(objectCategory -eq "person") -and (objectClass -eq "user")}**

**Get-ADObject -LDAPFilter "(&(objectCategory=person)(objectClass=user))"**

Requête qui permet d'afficher les groupes de l'Active Directory

**Get-ADObject -Filter {(objectCategory -eq "group")}**

**Get-ADObject -LDAPFilter "(&(objectCategory=group))"**

Requête qui permet d'afficher les ordinateurs de l'Active Directory

**Get-ADObject -Filter {(objectCategory -eq "computer")}**

**Get-ADObject -LDAPFilter "(&(objectCategory=computer))"**

Requête qui permet d'afficher les unités d'organisation de l'Active Directory

**(Get-ADObject -Filter {ObjectClass -eq "organizationalunit"})**

**Get-ADObject -LDAPFilter "(&(objectClass=organizationalunit))"**

Requête qui permet de trouver les objets de l'Active Directory dont le nom débute par "Adm"

**$sb = "DC=formation,DC=local"**

**(Get-ADObject -SearchBase $sb -Filter {Name -like "Adm\*"}) | `**

**Sort-Object ObjectClass | `**

**Format-Table ObjectClass,DistinguishedName -AutoSize**

Requête qui permet d'afficher le contenu de l'attribut ms-DS-MachineAccountQuota du domaine

**Get-ADObject -Identity ((Get-ADDomain).DistinguishedName) `**

**-Properties ms-DS-MachineAccountQuota**

**Modification des objets d'un domaine avec Get-ADObject**

Requête qui permet de protéger l'utilisateur TECH d'une suppression accidentelle.

**Get-ADUser -Identity TECH | Set-ADObject -ProtectedFromAccidentalDeletion:$true**