

TP6 : Introduction à PowerShell – Module Active Directory

Sommaire

2 – Prise en main :	2
3 – Les commandes de base :	5
4 – Navigation dans les répertoires et fichiers :	10
6 – Module Active Directory :	16
7 – TP : Création des comptes utilisateurs par lots :	18

2 – Prise en main :

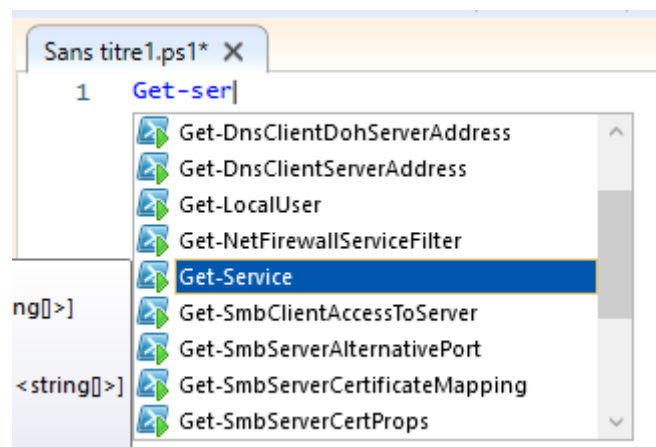
- Pour savoir quelle version est installée le poste de travail ou serveur, je tape la commande \$PSVersionTable dans la console :

```
PS C:\Users\Administrateur> $PSVersionTable

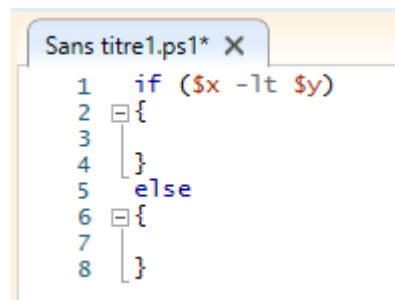
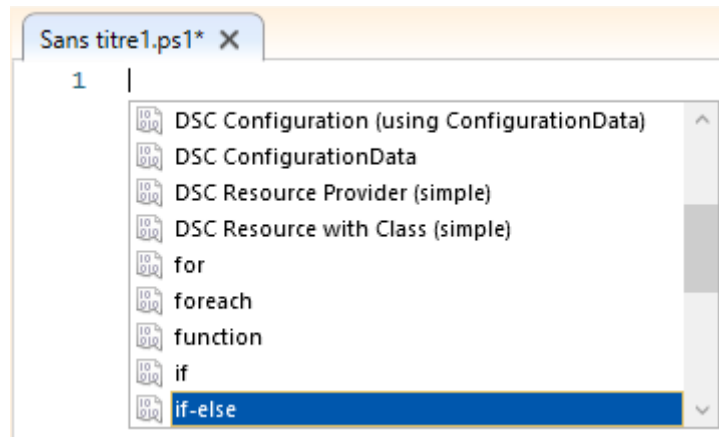
Name                           Value
----                           -
PSVersion                       5.1.26100.3624
PSEdition                       Desktop
PSCompatibleVersions            {1.0, 2.0, 3.0, 4.0...}
BuildVersion                    10.0.26100.3624
CLRVersion                      4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1

PS C:\Users\Administrateur>
```

- Avec la fonctionnalité « IntelliSense », il suffit à présent de commencer à frapper au clavier le début d'une commande pour qu'une proposition soit faite tout en indiquant la syntaxe :



- Enfin pour terminer, il existe une autre fonctionnalité des plus appréciables : les Snippets. Ces dernières permettent de coller du code ou des structures de code PowerShell directement dans l'éditeur. Pour les utiliser, saisissez la combinaison [Ctrl] J et une liste déroulante apparaît pour choisir un élément de code :



- Je saisi par exemple la commande dir :

```

PS C:\Users\Administrateur> dir

Répertoire : C:\Users\Administrateur

Mode                LastWriteTime         Length Name
----                -
d-----          16/05/2025   14:37      .VirtualBox
d-----          02/04/2025   18:01      Cisco Packet Tracer 8.2.2
d-r---          13/03/2025   09:47      Contacts
d-r---          02/04/2025   18:07      Desktop
d-r---          27/03/2025   15:51      Documents
d-r---          24/04/2025   10:15      Downloads
d-r---          13/03/2025   09:47      Favorites
d-r---          13/03/2025   09:47      Links
d-r---          13/03/2025   09:47      Music
d-r---          13/01/2025   18:07      OneDrive
d-r---          13/03/2025   09:47      Pictures
d-r---          13/03/2025   09:47      Saved Games
d-r---          13/03/2025   09:47      Searches
d-r---          13/03/2025   09:47      Videos
d-----          09/05/2025   16:18      VirtualBox VMs
-a-----          02/04/2025   16:51      194 .packettracer

PS C:\Users\Administrateur>

```

Et maintenant la même commande dans l'invite de commande CMD :

```
Invite de commandes
Microsoft Windows [version 10.0.26100.3775]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Administrateur>cd c:\

c:\>dir
Le volume dans le lecteur C s'appelle Windows
Le numéro de série du volume est E4C7-CCEA

Répertoire de c:\

17/03/2025  11:46    <DIR>          DGLogs
28/04/2025  10:04    <DIR>          inetpub
01/04/2024  09:26    <DIR>          PerfLogs
28/04/2025  14:48    <DIR>          Program Files
21/03/2025  13:51    <DIR>          Program Files (x86)
07/03/2025  17:56    <DIR>          Users
12/05/2025  11:15    <DIR>          Windows
                0 fichier(s)                0 octets
                7 Rép(s)  23 457 554 432 octets libres

c:\>
```

- Pour connaître la liste complète des alias disponibles, je tape la commande suivante : Get-Alias. Et pour les fonctions, je tape : Get-Command -CommandType function.

```
PS C:\Users\Administrateur> Get-Alias

CommandType      Name                                     Version      Source
-----
Alias            % -> ForEach-Object
Alias            ? -> Where-Object
Alias            ac -> Add-Content
Alias            asnp -> Add-PSSnapin
Alias            cat -> Get-Content
Alias            cd -> Set-Location
Alias            CFS -> ConvertFrom-String              3.1.0.0     Microsoft...
Alias            chdir -> Set-Location
Alias            clc -> Clear-Content
Alias            clear -> Clear-Host
Alias            clhy -> Clear-History
Alias            cli -> Clear-Item
Alias            clp -> Clear-ItemProperty
Alias            cls -> Clear-Host
Alias            clv -> Clear-Variable
Alias            cnsn -> Connect-PSSession
Alias            compare -> Compare-Object
Alias            copy -> Copy-Item
Alias            cp -> Copy-Item
Alias            cpi -> Copy-Item
```

```
PS C:\Users\Administrateur> Get-Command -CommandType function
```

CommandType	Name	Version	Source
Function	A:		
Function	Add-BCDataCacheExtension	1.0.0.0	BranchCache
Function	Add-BitLockerKeyProtector	1.0.0.0	BitLocker
Function	Add-DnsClientDohServerAddress	1.0.0.0	DnsClient
Function	Add-DnsClientNrptRule	1.0.0.0	DnsClient
Function	Add-DtcClusterTMMapping	1.0.0.0	MsDtc
Function	Add-EtwTraceProvider	1.0.0.0	EventTraci...
Function	Add-InitiatorIdToMaskingSet	2.0.0.0	Storage
Function	Add-MpPreference	1.0	ConfigDefe...
Function	Add-MpPreference	1.0	Defender
Function	Add-NetEventNetworkAdapter	1.0.0.0	NetEventPa...
Function	Add-NetEventPacketCaptureProvider	1.0.0.0	NetEventPa...
Function	Add-NetEventProvider	1.0.0.0	NetEventPa...
Function	Add-NetEventVFPPProvider	1.0.0.0	NetEventPa...
Function	Add-NetEventVmNetworkAdapter	1.0.0.0	NetEventPa...
Function	Add-NetEventVmSwitch	1.0.0.0	NetEventPa...
Function	Add-NetEventVmSwitchProvider	1.0.0.0	NetEventPa...
Function	Add-NetEventWFPCaptureProvider	1.0.0.0	NetEventPa...
Function	Add-NetIPHttpsCertBinding	1.0.0.0	NetworkTra...
Function	Add-NetLbfoTeamMember	2.0.0.0	NetLbfo
Function	Add-NetLbfoTeamNic	2.0.0.0	NetLbfo
Function	Add-NetNatExternalAddress	1.0.0.0	NetNat
Function	Add-NetNatStaticMapping	1.0.0.0	NetNat
Function	Add-NetSwitchTeamMember	1.0.0.0	NetSwitchTeam
Function	Add-OdbcDsn	1.0.0.0	Wdac

3 – Les commandes de base :

- Je saisis la commande Get-Verb permettant de retrouver la liste complète des verbes officiels et leurs groupes d'applications.

```
PS C:\Users\Administrateur> Get-Verb
```

Verb	Group
Add	Common
Clear	Common
Close	Common
Copy	Common
Enter	Common
Exit	Common
Find	Common
Format	Common
Get	Common
Hide	Common
Join	Common
Lock	Common
Move	Common
New	Common
Open	Common
Optimize	Common
Pop	Common
Push	Common
Redo	Common
Remove	Common
Rename	Common
Reset	Common
Resize	Common
Search	Common
Select	Common
Set	Common
Show	Common

- Pour l'instant, je m'intéresse uniquement aux commandelettes, et pour cela, j'ajoute le paramètre -CommandType suivi du type de commandes choisi, à savoir cmdlet :

```
PS C:\Users\Administrateur> Get-Command -CommandType cmdlet
```

CommandType	Name	Version	Source
Cmdlet	Add-AppProvisionedSharedPackageContainer	3.0	DisM
Cmdlet	Add-AppSharedPackageContainer	2.0.1.0	Appx
Cmdlet	Add-AppvClientConnectionGroup	1.0.0.0	AppvClient
Cmdlet	Add-AppvClientPackage	1.0.0.0	AppvClient
Cmdlet	Add-AppvPublishingServer	1.0.0.0	AppvClient
Cmdlet	Add-AppxPackage	2.0.1.0	Appx
Cmdlet	Add-AppxProvisionedPackage	3.0	DisM
Cmdlet	Add-AppxVolume	2.0.1.0	Appx
Cmdlet	Add-BitsFile	2.0.0.0	BitsTransfer
Cmdlet	Add-CertificateEnrollmentPolicyServer	1.0.0.0	PKI
Cmdlet	Add-Computer	3.1.0.0	Microsoft...
Cmdlet	Add-Content	3.1.0.0	Microsoft...
Cmdlet	Add-History	3.0.0.0	Microsoft...
Cmdlet	Add-JobTrigger	1.1.0.0	PSSchedule...
Cmdlet	Add-KdsRootKey	1.0.0.0	Kds
Cmdlet	Add-LocalGroupMember	1.0.0.0	Microsoft...
Cmdlet	Add-Member	3.1.0.0	Microsoft...
Cmdlet	Add-PSSnapin	3.0.0.0	Microsoft...
Cmdlet	Add-SignerRule	1.0	ConfigCI
Cmdlet	Add-Type	3.1.0.0	Microsoft...
Cmdlet	Add-WindowsCapability	3.0	DisM
Cmdlet	Add-WindowsDriver	3.0	DisM
Cmdlet	Add-WindowsImage	3.0	DisM
Cmdlet	Add-WindowsPackage	3.0	DisM
Cmdlet	Checkpoint-Computer	3.1.0.0	Microsoft...
Cmdlet	Clear-Content	3.1.0.0	Microsoft...
Cmdlet	Clear-EventLog	3.1.0.0	Microsoft...
Cmdlet	Clear-History	3.0.0.0	Microsoft...

- Dans la première version de PowerShell, les commandes de base étaient au nombre de 129. Dans la version 5.1, leur nombre beaucoup plus important varie en fonction du type de système (client ou serveur), du nombre de rôles. Pour les compter, je saisis la commande suivante :

```
PS C:\Users\Administrateur> (Get-Command -CommandType cmdlet).count
706
PS C:\Users\Administrateur>
```

- Get-Command possède le paramètre -verb et ce dernier permet de retourner toutes les commandes commençant par un verbe donné. J'obtiens par exemple toutes les commandes dont le verbe commence par Write :

```
PS C:\Users\Administrateur> Get-Command -Verb write
```

CommandType	Name	Version	Source
Alias	Write-FileSystemCache	2.0.0.0	Storage
Alias	Write-FileSystemCache	1.0.0.0	VMDirectSt...
Function	Write-DtcTransactionsTraceSession	1.0.0.0	MsDtc
Function	Write-PrinterNfcTag	1.1	PrintManag...
Function	Write-VolumeCache	2.0.0.0	Storage
Cmdlet	Write-Debug	3.1.0.0	Microsoft...
Cmdlet	Write-Error	3.1.0.0	Microsoft...
Cmdlet	Write-EventLog	3.1.0.0	Microsoft...
Cmdlet	Write-Host	3.1.0.0	Microsoft...
Cmdlet	Write-Information	3.1.0.0	Microsoft...
Cmdlet	Write-Output	3.1.0.0	Microsoft...
Cmdlet	Write-Progress	3.1.0.0	Microsoft...
Cmdlet	Write-Verbose	3.1.0.0	Microsoft...
Cmdlet	Write-Warning	3.1.0.0	Microsoft...

```
PS C:\Users\Administrateur> |
```

- De manière similaire, avec cette fois-ci le paramètre -noun, j'affiche les commandes qui s'appliquent, par exemple, aux objets, c'est-à-dire celles dont la partie nom est objet :

```
PS C:\Users\Administrateur> Get-Command -Noun object
```

CommandType	Name	Version	Source
Cmdlet	Compare-Object	3.1.0.0	Microsoft...
Cmdlet	ForEach-Object	3.0.0.0	Microsoft...
Cmdlet	Group-Object	3.1.0.0	Microsoft...
Cmdlet	Measure-Object	3.1.0.0	Microsoft...
Cmdlet	New-Object	3.1.0.0	Microsoft...
Cmdlet	Select-Object	3.1.0.0	Microsoft...
Cmdlet	Sort-Object	3.1.0.0	Microsoft...
Cmdlet	Tee-Object	3.1.0.0	Microsoft...
Cmdlet	Where-Object	3.0.0.0	Microsoft...

```
PS C:\Users\Administrateur> |
```

- J'affiche les commandes de type alias :

```
PS C:\Users\Administrateur> Get-Command -Commandtype alias
```

CommandType	Name	Version	Source
Alias	% -> ForEach-Object		
Alias	? -> Where-Object		
Alias	ac -> Add-Content		
Alias	Add-AppPackage	2.0.1.0	Appx
Alias	Add-AppPackageVolume	2.0.1.0	Appx
Alias	Add-AppProvisionedPackage	3.0	Dism
Alias	Add-MsixPackage	2.0.1.0	Appx
Alias	Add-MsixPackageVolume	2.0.1.0	Appx
Alias	Add-MsixVolume	2.0.1.0	Appx
Alias	Add-ProvisionedAppPackage	3.0	Dism
Alias	Add-ProvisionedAppSharedPackageContainer	3.0	Dism
Alias	Add-ProvisionedAppxPackage	3.0	Dism
Alias	Add-ProvisioningPackage	3.0	Provisioning
Alias	Add-TrustedProvisioningCertificate	3.0	Provisioning
Alias	algm ->	1.0.0.0	Microsoft...
Alias	Apply-WindowsUnattend	3.0	Dism
Alias	asnp -> Add-PSSnapin		
Alias	blsmba ->	2.0.0.0	SmbShare
Alias	blsmbclas ->	2.0.0.0	SmbShare
Alias	cat -> Get-Content		
Alias	cd -> Set-Location		
Alias	CFS -> ConvertFrom-String	3.1.0.0	Microsoft...
Alias	chdir -> Set-Location		
Alias	clc -> Clear-Content		
Alias	clear -> Clear-Host		
Alias	clhy -> Clear-History		
Alias	cli -> Clear-Item		

- Je recherche une commande dont on ignore le nom mais sachant qu'elle commence par le verbe Get :

```
PS C:\Users\Administrateur> Get-Command Get-*
```

CommandType	Name	Version	Source
Alias	Get-AppPackage	2.0.1.0	Appx
Alias	Get-AppPackageAutoUpdateSettings	2.0.1.0	Appx
Alias	Get-AppPackageDefaultVolume	2.0.1.0	Appx
Alias	Get-AppPackageLastError	2.0.1.0	Appx
Alias	Get-AppPackageLog	2.0.1.0	Appx
Alias	Get-AppPackageManifest	2.0.1.0	Appx
Alias	Get-AppPackageVolume	2.0.1.0	Appx
Alias	Get-AppProvisionedPackage	3.0	Dism
Alias	Get-DiskSNV	2.0.0.0	Storage
Alias	Get-DiskSNV	1.0.0.0	VMDirectSt...
Alias	Get-Language	1.0	LanguagePa...
Alias	Get-MsixDefaultVolume	2.0.1.0	Appx
Alias	Get-MsixLastError	2.0.1.0	Appx
Alias	Get-MsixLog	2.0.1.0	Appx
Alias	Get-MsixPackage	2.0.1.0	Appx
Alias	Get-MsixPackageAutoUpdateSettings	2.0.1.0	Appx
Alias	Get-MsixPackageDefaultVolume	2.0.1.0	Appx
Alias	Get-MsixPackageLastError	2.0.1.0	Appx
Alias	Get-MsixPackageLog	2.0.1.0	Appx
Alias	Get-MsixPackageManifest	2.0.1.0	Appx
Alias	Get-MsixPackageVolume	2.0.1.0	Appx
Alias	Get-MsixVolume	2.0.1.0	Appx
Alias	Get-PhysicalDiskSNV	2.0.0.0	Storage
Alias	Get-PhysicalDiskSNV	1.0.0.0	VMDirectSt...
Alias	Get-PreferredLanguage	1.0	LanguagePa...
Alias	Get-ProvisionedAppPackage	3.0	Dism
Alias	Get-ProvisionedAppSharedPackageContainer	3.0	Dism

- Je crée la variable \$maVariable et je lui affecte une valeur de type chaîne (String).

```
PS C:\Users\Administrateur> $maVariable = 'Bonjour tout le monde !'  
PS C:\Users\Administrateur>
```

- Je tape la commande \$maVariable | get-member :

```
PS C:\Users\Administrateur> $maVariable | gm

    TypeName : System.String

Name      MemberType      Definition
----      -
Clone     Method          System.Object Clone(), System.Object ICloneable.C...
CompareTo Method          int CompareTo(System.Object value), int CompareTo...
Contains  Method          bool Contains(string value)
CopyTo    Method          void CopyTo(int sourceIndex, char[] destination, ...
EndsWith  Method          bool EndsWith(string value), bool EndsWith(string...
Equals    Method          bool Equals(System.Object obj), bool Equals(strin...
GetEnumera Method         System.CharEnumerator GetEnumerator(), System.Col...
tor        Method          int GetHashCode()
GetType   Method          type GetType()
GetTypeCode Method         System.TypeCode GetTypeCode(), System.TypeCode IC...
IndexOf   Method          int IndexOf(char value), int IndexOf(char value, ...
IndexAny  Method          int IndexOfAny(char[] anyOf), int IndexOfAny(char...
Insert    Method          string Insert(int startIndex, string value)
IsNormali Method         bool IsNormalized(), bool IsNormalized(System.Tex...
zed        Method          int LastIndexOf(char value), int LastIndexOf(char...
LastIndex Method         int LastIndexOfAny(char[] anyOf), int LastIndexO...
OfAny     Method          string Normalize(), string Normalize(System.Text...
Normaliz  Method          string PadLeft(int totalWidth), string PadLeft(in...
e         Method          string PadRight(int totalWidth), string PadRight(...
PadLeft   Method          string Remove(int startIndex, int count), string ...
PadRight  Method          string Replace(char oldChar, char newChar), strin...
Remove    Method          string[] Split(Params char[] separator), string[...
Replac    Method          bool StartsWith(string value), bool StartsWith(st...
e         Method          string Substring(int startIndex), string Substrin...
Split     Method          bool IConvertible.ToBoolean(System.IFormatProvide...
StartsWi Method         byte IConvertible.ToByte(System.IFormatProvider p...
th        Method          char IConvertible.ToChar(System.IFormatProvider p...
Substrin Method         char[] ToCharArray(), char[] ToCharArray(int star...
g         Method          datetime IConvertible.ToDateTime(System.IFormatPr...
g         Method          decimal IConvertible.ToDecimal(System.IFormatProv...
g         Method          double IConvertible.ToDouble(System.IFormatProvid...
g         Method          int16 IConvertible.ToInt16(System.IFormatProvider...
g         Method          int IConvertible.ToInt32(System.IFormatProvider p...
g         Method          long IConvertible.ToInt64(System.IFormatProvider ...
g         Method          string ToLower(), string ToLower(cultureinfo cult...
g         Method          string ToLowerInvariant()
g         Method          sbyte IConvertible.ToSByte(System.IFormatProvider...
g         Method          float IConvertible.ToSingle(System.IFormatProvide...
g         Method          string ToString(), string ToString(System.IFormat...
g         Method          System.Object IConvertible.ToType(type conversion...
g         Method          uint16 IConvertible.ToUInt16(System.IFormatProvid...
g         Method          uint32 IConvertible.ToUInt32(System.IFormatProvid...
g         Method          uint64 IConvertible.ToUInt64(System.IFormatProvid...
g         Method          string ToUpper(), string ToUpper(cultureinfo cult...
g         Method          string ToUpperInvariant()
g         Method          string Trim(Params char[] trimChars), string Trim()
g         Method          string TrimEnd(Params char[] trimChars)
g         Method          string TrimStart(Params char[] trimChars)
Chars     ParameterizedProperty char Chars(int index) {get;}
Length    Property        int Length {get;}

```

- J'utilise la méthode ToUpper afin de retourner la chaîne contenue dans \$maVariable en majuscules.

```
PS C:\Users\Administrateur> $maVariable.ToUpper()
BONJOUR TOUT LE MONDE !
PS C:\Users\Administrateur>
```

- De la même façon, j'utilise la propriété Length afin d'obtenir le nombre de caractères contenus dans notre chaîne.

```
PS C:\Users\Administrateur> $maVariable.Length
23
PS C:\Users\Administrateur>
```

4 – Navigation dans les répertoires et fichiers :

- Pour le vérifier, je tape la commande suivante :

```
PS C:\Users\Administrateur> Get-Alias dir

CommandType      Name                                     Version          Source
-----
Alias             dir -> Get-ChildItem
```

- Par exemple, j'observe le résultat de la commande suivante :

```
PS C:\Users\Administrateur> gci c:\

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----          17/03/2025   10:46         DGLogs
d-----          28/04/2025   10:04         inetpub
d-----          01/04/2024   09:26         PerfLogs
d-r---          28/04/2025   14:48         Program Files
d-r---          21/03/2025   12:51         Program Files (x86)
d-r---          07/03/2025   16:56         Users
d-----          12/05/2025   11:15         Windows

PS C:\Users\Administrateur>
```

- Pour afficher les fichiers cachés, j'ajoute à la commande Get-Childitem le paramètre -Force :

```
PS C:\Users\Administrateur> gci c:\ -Force

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d--hs-             27/02/2025    10:33           $Recycle.Bin
d-----            17/03/2025    10:46           DGLogs
d--hs|             13/01/2025    18:02           Documents and Settings
d-----            28/04/2025    10:04           inetpub
d-----            01/04/2024    09:26           PerfLogs
d-r---            28/04/2025    14:48           Program Files
d-r---            21/03/2025    12:51           Program Files (x86)
d--h--            21/03/2025    12:52           ProgramData
d--hs-             14/03/2025    07:50           Recovery
d--hs-             15/05/2025    17:35           System Volume Information
d-r---            07/03/2025    16:56           Users
d-----            12/05/2025    11:15           Windows
-a-hs-             16/05/2025    14:37           12288 DumpStack.log.tmp
-a-hs-             16/05/2025    14:37           27451232256 hiberfil.sys
-a-hs-             16/05/2025    14:37           4294967296 pagefile.sys
-a-hs-             16/05/2025    14:37           16777216 swapfile.sys

PS C:\Users\Administrateur>
```

- J'affiche (récursivement) tous les fichiers ayant l'extension .log contenus à l'intérieur d'une arborescence :

```
PS C:\Users\Administrateur> Get-ChildItem C:\Windows\Temp -Include *.log -Recurse
Get-ChildItem : L'accès au chemin d'accès 'C:\Windows\Temp\vmware-Système' est refusé.
Au caractère Ligne:1 : 1
+ Get-ChildItem C:\Windows\Temp -Include *.log -Recurse
+ ~~~~~
+ CategoryInfo          : PermissionDenied: (C:\Windows\Temp\vmware-Système:String) [G
  et-ChildItem], UnauthorizedAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.Get
  ChildItemCommand

Répertoire : C:\Windows\Temp

Mode                LastWriteTime         Length Name
----                -
-a-----            16/05/2025    14:52           139038 MpSigStub.log
-a-----            09/05/2025    16:17           268518 vminst.log

PS C:\Users\Administrateur>
```

- J'affiche le nom des fichiers dont la taille est supérieure à 32 Ko :

```
PS C:\Users\Administrateur> Get-ChildItem | Where-Object {$_.Length -gt 32KB}
```

- J'affiche les fichiers dont la date de dernier enregistrement est postérieure au 31/01/2018 :

```
PS C:\Users\Administrateur> Get-ChildItem | Where-Object {$_.LastWriteTime -gt '01/31/2018'}
```

Répertoire : C:\Users\Administrateur

Mode	LastWriteTime	Length	Name
d-----	16/05/2025 15:04		.VirtualBox
d-----	02/04/2025 18:01		Cisco Packet Tracer 8.2.2
d-r---	13/03/2025 09:47		Contacts
d-r---	02/04/2025 18:07		Desktop
d-r---	27/03/2025 15:51		Documents
d-r---	24/04/2025 10:15		Downloads
d-r---	13/03/2025 09:47		Favorites
d-r---	13/03/2025 09:47		Links
d-r---	13/03/2025 09:47		Music
d-r---	13/01/2025 18:07		OneDrive
d-r---	13/03/2025 09:47		Pictures
d-r---	13/03/2025 09:47		Saved Games
d-r---	13/03/2025 09:47		Searches
d-r---	13/03/2025 09:47		Videos
d-----	09/05/2025 16:18		VirtualBox VMs
-a----	02/04/2025 16:51	194	.packettracer

- J'affiche tous les fichiers ou répertoires cachés à la racine de la partition système :

```
PS C:\Users\Administrateur> Get-ChildItem c:\ -Attributes Hidden
```

Répertoire : C:\

Mode	LastWriteTime	Length	Name
d--hs-	27/02/2025 10:33		\$Recycle.Bin
d--hs	13/01/2025 18:02		Documents and Settings
d--h--	21/03/2025 12:52		ProgramData
d--hs-	14/03/2025 07:50		Recovery
d--hs-	15/05/2025 17:35		System Volume Information
-a-hs-	16/05/2025 14:37	12288	DumpStack.log.tmp
-a-hs-	16/05/2025 14:37	27451232256	hiberfil.sys
-a-hs-	16/05/2025 14:37	4294967296	pagefile.sys
-a-hs-	16/05/2025 14:37	16777216	swapfile.sys

```
PS C:\Users\Administrateur>
```

- Par exemple, filtre uniquement les fichiers cachés et non les répertoires en combinant les attributs « caché » et « n'est pas un répertoire ».

```
PS C:\Users\Administrateur> Get-ChildItem c:\ -Attributes Hidden+!Directory
```

Répertoire : C:\

Mode	LastWriteTime	Length	Name
-a-hs-	16/05/2025 14:37	12288	DumpStack.log.tmp
-a-hs-	16/05/2025 14:37	27451232256	hiberfil.sys
-a-hs-	16/05/2025 14:37	4294967296	pagefile.sys
-a-hs-	16/05/2025 14:37	16777216	swapfile.sys

```
PS C:\Users\Administrateur> |
```

- Set-Location, Elle me permet de nous déplacer dans une arborescence de dossiers.

```
PS C:\Users\Administrateur> Set-Location D:\  
PS D:\> |
```

- Get-Location, Cette commande retourne l'emplacement actuel à l'intérieur d'une arborescence.

```
PS C:\users> Get-Location  
Path  
----  
C:\users  
  
PS C:\users>
```

- J'affiche les propriétés et méthodes :

```
PS C:\users> Get-Location | Get-Member  
  
TypeName : System.Management.Automation.PathInfo  
  
Name      MemberType Definition  
----      -  
Equals    Method     bool Equals(System.Object obj)  
GetHashCode Method     int GetHashCode()  
GetType   Method     type GetType()  
ToString  Method     string ToString()  
Drive     Property   System.Management.Automation.PSDriveInfo Drive {get;}  
Path      Property   string Path {get;}  
Provider  Property   System.Management.Automation.ProviderInfo Provider {get;}  
ProviderPath Property   string ProviderPath {get;}  
  
PS C:\users> |
```

- Je récupère dans une variable le chemin (path) de l'emplacement courant en une seule ligne de commande.

```
PS C:\users> $CheminCourant = (Get-Location).Path  
PS C:\users>
```

- Je vien de stocker la valeur du chemin courant, en l'occurrence C:\users, dans la variable \$CheminCourant. Affichez-le dans la console :

```
PS C:\users> $CheminCourant  
C:\users  
PS C:\users>
```

- Je crée le répertoire Temp :

```
PS C:\WINDOWS\system32> New-Item -ItemType directory -Name Temp

Répertoire : C:\WINDOWS\system32

Mode                LastWriteTime         Length Name
----                -
d-----           16/05/2025   15:22             Temp

PS C:\WINDOWS\system32>
```

- Si le dossier avait contenu un espace, vous auriez dû le mettre entre guillemets. Par exemple :

```
PS C:\WINDOWS\system32> New-Item -Name 'Dossier Test' -ItemType directory

Répertoire : C:\WINDOWS\system32

Mode                LastWriteTime         Length Name
----                -
d-----           16/05/2025   15:22             Dossier Test

PS C:\WINDOWS\system32> |
```

- Je crée un fichier nommé monFichier.txt qui contient la phrase « Madame Pasquier for ever ! » :

```
PS C:\WINDOWS\system32> ni -Name monFichier.txt -ItemType file -Value 'Madame Pasquier for ever'

Répertoire : C:\WINDOWS\system32

Mode                LastWriteTime         Length Name
----                -
-a-----           16/05/2025   15:23             25 monFichier.txt

PS C:\WINDOWS\system32> |
```

- Je supprime tous les fichiers .log contenus dans le répertoire C:\Temp :

```
PS C:\WINDOWS\system32> Remove-Item C:\Windows\System32\Temp\*.log
PS C:\WINDOWS\system32>
```

- Je supprime sélectivement tous les fichiers contenus dans une arborescence de dossiers dont l'extension est .txt :

```
PS C:\WINDOWS\system32> Get-ChildItem C:\Windows\System32\Temp\* -Include *.txt -Recurse | Remove-Item
PS C:\WINDOWS\system32>
```

- Pour supprimer un fichier système, masqué ou en lecture seule, il suffit tout simplement d'utiliser le paramètre -Force, comme dans l'exemple ci-dessous :

```
PS C:\Users\Administrateur\Documents> ri fichierASupprimer.txt -Force
PS C:\Users\Administrateur\Documents> |
```

- Déplacement de fichiers, exemple : déplacement des fichiers *.jpg du répertoire courant vers le dossier « mes photos ».

```
PS C:\Users\Administrateur\Documents> Move-Item -Path *.jpg -destination 'Mes photos'
PS C:\Users\Administrateur\Documents>
```

- Le déplacement d'un répertoire est similaire au déplacement de fichiers.

```
PS C:\Users\Administrateur\Documents> Move-Item 'Rep1' 'Rep2'
PS C:\Users\Administrateur\Documents> |
```

- Je vien de déplacer l'intégralité du répertoire Rep1 dans le répertoire Rep2.

```
PS C:\Users\Administrateur\Documents> Move-Item 'Rep1' 'Rep3'
PS C:\Users\Administrateur\Documents>
```

- Exemple : renommage du fichier monFichierDeLog.txt en ficlog.txt.

```
PS C:\Users\Administrateur\Documents> Rename-Item -Path C:\Users\Administrateur\Documents\monFichierDeLog.txt -Newname ficlog.
PS C:\Users\Administrateur\Documents>
```

- Exemple : renommage du répertoire Dossier1 en Dossier2

```
PS C:\Users\Administrateur\Documents> Rename-Item -Path C:\Users\Administrateur\Documents\Dossier1 -Newname Dossier2
PS C:\Users\Administrateur\Documents>
```

- Copie un fichier d'un répertoire source vers un répertoire destination.

```
PS C:\Users\Administrateur\Documents> Copy-Item -Path C:\Users\Administrateur\Documents\ficLog.txt -destination C:\Users\Admi
PS C:\Users\Administrateur\Documents>
```

6 – Module Active Directory :

- J'importe le module Active Directory :

```
PS C:\Users\Administrateur> Import-Module ActiveDirectory

PS C:\Users\Administrateur> Get-Command -Module ActiveDirectory
```

CommandType	Name	Version	Source
Cmdlet	Add-ADCentralAccessPolicyMember	1.0.1.0	ActiveDirectory
Cmdlet	Add-ADComputerServiceAccount	1.0.1.0	ActiveDirectory
Cmdlet	Add-ADDomainControllerPasswordReplicationPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Add-ADFineGrainedPasswordPolicySubject	1.0.1.0	ActiveDirectory
Cmdlet	Add-ADGroupMember	1.0.1.0	ActiveDirectory
Cmdlet	Add-ADPrincipalGroupMembership	1.0.1.0	ActiveDirectory
Cmdlet	Add-ADResourcePropertyListMember	1.0.1.0	ActiveDirectory
Cmdlet	Clear-ADAccountExpiration	1.0.1.0	ActiveDirectory
Cmdlet	Clear-ADClaimTransformLink	1.0.1.0	ActiveDirectory
Cmdlet	Disable-ADAccount	1.0.1.0	ActiveDirectory
Cmdlet	Disable-ADOptionalFeature	1.0.1.0	ActiveDirectory
Cmdlet	Enable-ADAccount	1.0.1.0	ActiveDirectory
Cmdlet	Enable-ADOptionalFeature	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADAccountAuthorizationGroup	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADAccountResultantPasswordReplicationPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADAuthenticationPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADAuthenticationPolicySilo	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADCentralAccessPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADCentralAccessRule	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADClaimTransformPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADClaimType	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADComputer	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADComputerServiceAccount	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADDCCloningExcludedApplicationList	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADDefaultDomainPasswordPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADDomain	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADDomainController	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADDomainControllerPasswordReplicationPolicy	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADDomainControllerPasswordReplicationPolicy...	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADFineGrainedPasswordPolicy	1.0.1.0	ActiveDirectory

- Vous pouvez obtenir l'ensemble des commandes apportées par le module Active Directory en tapant ceci :

```
PS C:\Users\Administrateur> Get-Command -Module ActiveDirectory -Name *user*
```

CommandType	Name	Version	Source
Cmdlet	Get-ADUser	1.0.1.0	ActiveDirectory
Cmdlet	Get-ADUserResultantPasswordPolicy	1.0.1.0	ActiveDirectory
Cmdlet	New-ADUser	1.0.1.0	ActiveDirectory
Cmdlet	Remove-ADUser	1.0.1.0	ActiveDirectory
Cmdlet	Set-ADUser	1.0.1.0	ActiveDirectory

```
PS C:\Users\Administrateur> |
```

- Pour créer un compte utilisateur, il faut au minimum préciser la propriété Name

```
PS C:\Users\Administrateur> New-ADUser -Name Winnie

PS C:\Users\Administrateur>
```

- Comme la valeur attendue pour le paramètre -AccountPassword est de type chaîne sécurisée (SecureString), je dois effectuer quelques petites manipulations supplémentaires :

```
PS C:\Users\Administrateur> $passwd = '123AZEqsd!'
PS C:\Users\Administrateur> $passwd = ConvertTo-SecureString $passwd -AsPlainText -Force
```

```
PS C:\Users\Administrateur> New-ADUser -SamAccountName Johnny -Name 'Jean-Philippe Smet'
```

- Au lieu de stocker le mot de passe en clair, ce qui n'est pas idéal du point de vue de la sécurité, nous aurions pu utiliser la commande Read-Host associée au paramètre -AsSecureString :

```
PS C:\Users\Administrateur> $secure_passwd = read-host "Entrer un mot de passe :" -assecurestring
PS C:\Users\Administrateur>
```

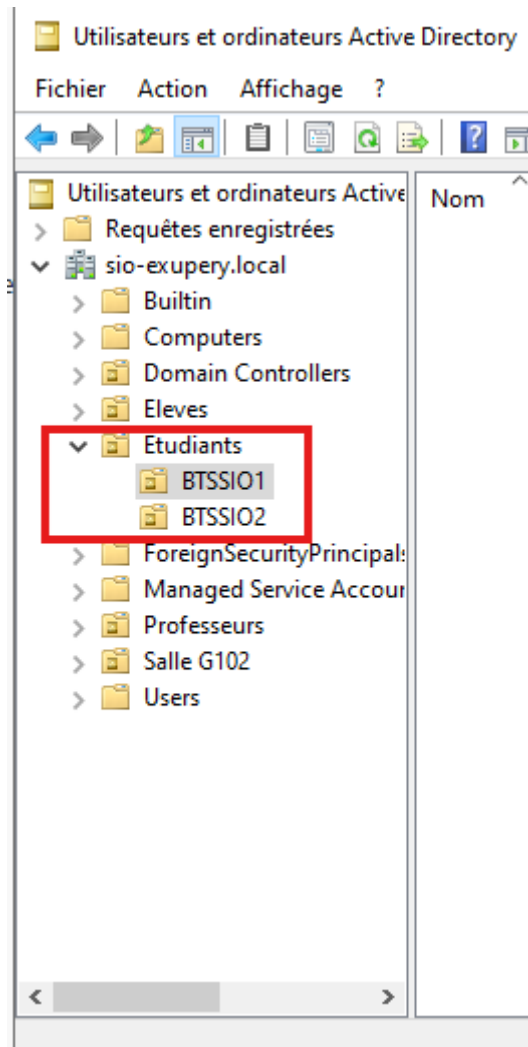
- Pour activer un compte à la création, il est nécessaire de lui affecter un mot de passe et d'utiliser le paramètre Enabled.

```
PS C:\Users\Administrateur> $secure_passwd = read-host "Entrer un mot de passe :" -assecurestring
PS C:\Users\Administrateur> $passwd = '123AZEqsd!'
PS C:\Users\Administrateur> $passwd = ConvertTo-SecureString $passwd -AsPlainText -Force
```

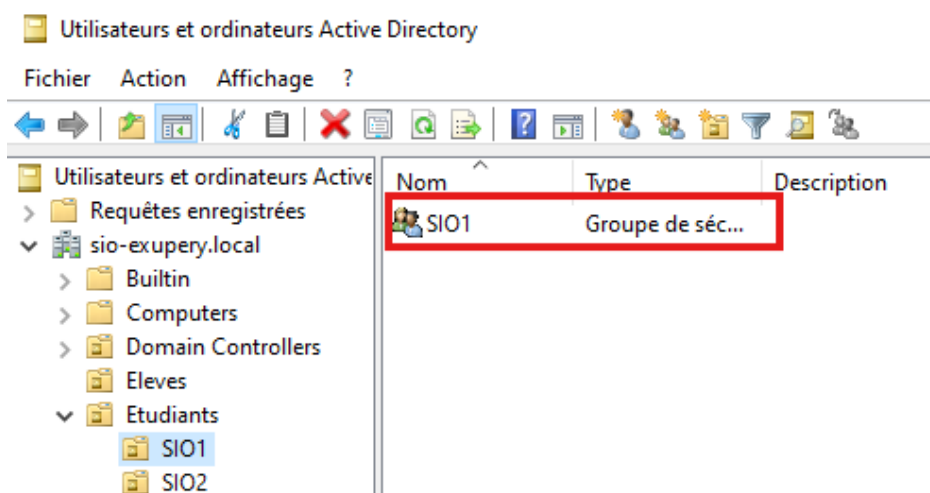
```
PS C:\Users\Administrateur> New-ADGroup -Name BTSSIO1 -GroupScope Global -GroupCategory Security
New-ADGroup : Le groupe spécifié existe déjà
Au caractère Ligne:1 : 1
+ New-ADGroup -Name BTSSIO1 -GroupScope Global -GroupCategory Security
+ ~~~~~
+ CategoryInfo          : NotSpecified: (CN=BTSSIO1,CN=U...xupery,DC=local:String) [New-ADGroup], ADException
+ FullyQualifiedErrorId : ActiveDirectoryServer:1318,Microsoft.ActiveDirectory.Management.Commands.NewADGroup
PS C:\Users\Administrateur> |
```

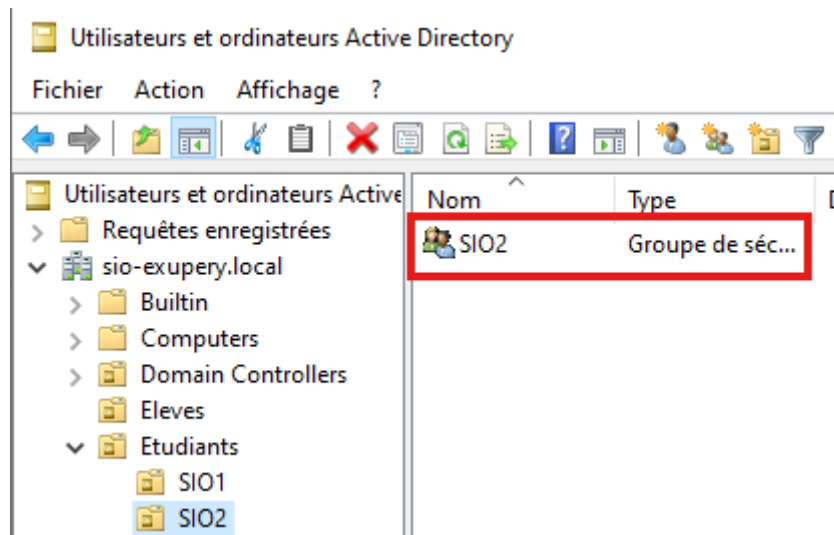
7 – TP : Création des comptes utilisateurs par lots :

- Je crée l'OU Etudiants puis, à l'intérieur de celle-ci, les OU SIO1 et SIO2



- Je crée les groupes de sécurité SIO1 et SIO2 respectivement dans les OU SIO1 et SIO2.





- Je crée, sur le serveur Windows, un fichier .csv (avec le Bloc-notes ou Excel) où chaque ligne contiendra la description d'un compte utilisateur. Pour ce faire, il sera nécessaire de commencer votre fichier par une ligne d'en-têtes. Ces derniers seront des propriétés des objets manipulés dans le script.

```

comptes.csv - Bloc-notes
Fichier Edition Format Affichage Aide
name,givenname,classe
cotta,gabriel,SI02
vanpraet,erwan,SI02
cavin,joakim,SI02
gauthier,etienne,SI02
kaczmarek,cleophee,SI02
bersia,baptiste,SI02
lantrua,lucas,SI01
witz,tanguy,SI01
ruggeri,jamy,SI01
bellanger,baptiste,SI01
barkallah,adam,SI01
perillat,arnaud,SI01

```

- J'affiche les cmdlets contenant NTFS :

```
PS C:\Users\Administrateur> get-command *NTFS*
```

CommandType	Name	Version	Source
Cmdlet	Add-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Add-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Clear-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Clear-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Disable-NTFSAccessInheritance	4.2.6	NTFSSecurity
Cmdlet	Disable-NTFSAuditInheritance	4.2.6	NTFSSecurity
Cmdlet	Enable-NTFSAccessInheritance	4.2.6	NTFSSecurity
Cmdlet	Enable-NTFSAuditInheritance	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSEffectiveAccess	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSHardLink	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSInheritance	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSOrphanedAccess	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSOrphanedAudit	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSOwner	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSSecurityDescriptor	4.2.6	NTFSSecurity
Cmdlet	Get-NTFSSimpleAccess	4.2.6	NTFSSecurity
Cmdlet	New-NTFSHardLink	4.2.6	NTFSSecurity
Cmdlet	New-NTFSSymbolicLink	4.2.6	NTFSSecurity
Cmdlet	Remove-NTFSAccess	4.2.6	NTFSSecurity
Cmdlet	Remove-NTFSAudit	4.2.6	NTFSSecurity
Cmdlet	Set-NTFSInheritance	4.2.6	NTFSSecurity

- J'écris le script dans PowerShell ISE :

```
Add-Users.ps1 X
4 foreach ($user in $users)
5 {
6     $name = $user.name
7     $givenname = $user.givenname
8     $samname = $givenname.substring(0,1) + $name
9     $surname = $user.surname
10    $section = $user.classe
11    $domain = "sio-exupery"
12    $ext = "local"
13    $drive = "H:"
14    $hddirectory = "\\AD\Home\$"
15    $dossier = "$hddirectory\$samname"
16    $profil = "\\AD\Profiles\$samname"
17
18    New-ADUser -Name "$name $givenname" -UserPrincipalName "$name@$domain.$ext" -SamAccountName $samname -GivenName $givenname -Surname $surname -Path "$ouSection,$ouEtudiants,$ouSio-exupery,$ouLocal" -HomeDrive $drive -HomeDirectory $dossier -ProfilePath $profil
19    Set-Location $hddirectory
20    New-Item -Name $samname -ItemType directory
21    Get-Item "$dossier" | Disable-NTFSAccessInheritance
22    Set-NTFSOwner -Path "$dossier" -Account "sio-exupery.local\$samname"
23    Add-NTFSAccess -Path "$dossier" -Account "sio-exupery.local\$samname" -AccessRights FullControl
24    Get-NTFSAccess -Path "$dossier" -Account "utilisateurs" -ExcludeInherited | Remove-NTFSAccess
25    Add-ADGroupMember -Identity "section" -Members $samname
26 }
```

- J'exécute le script :

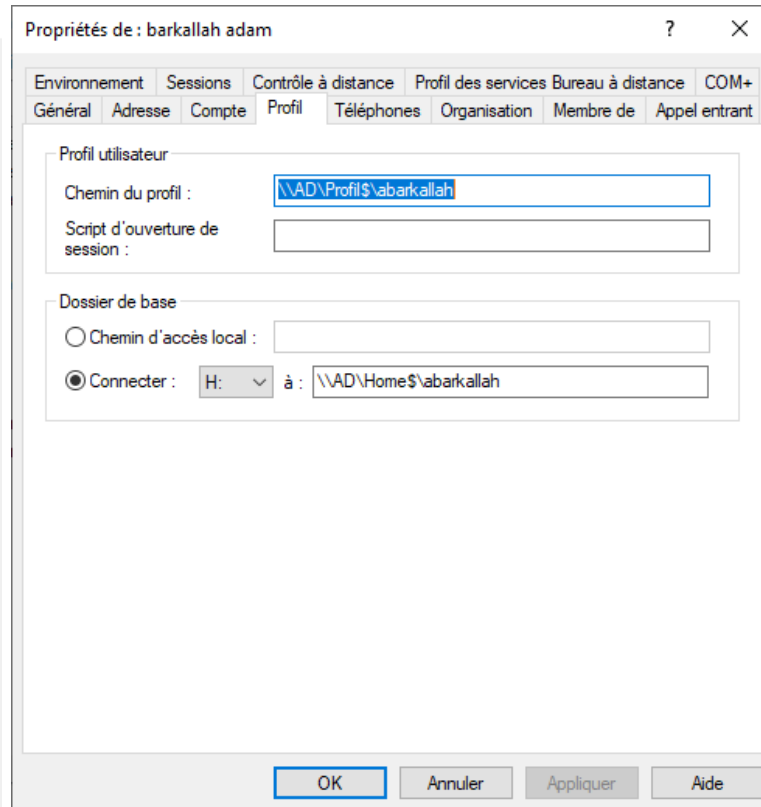
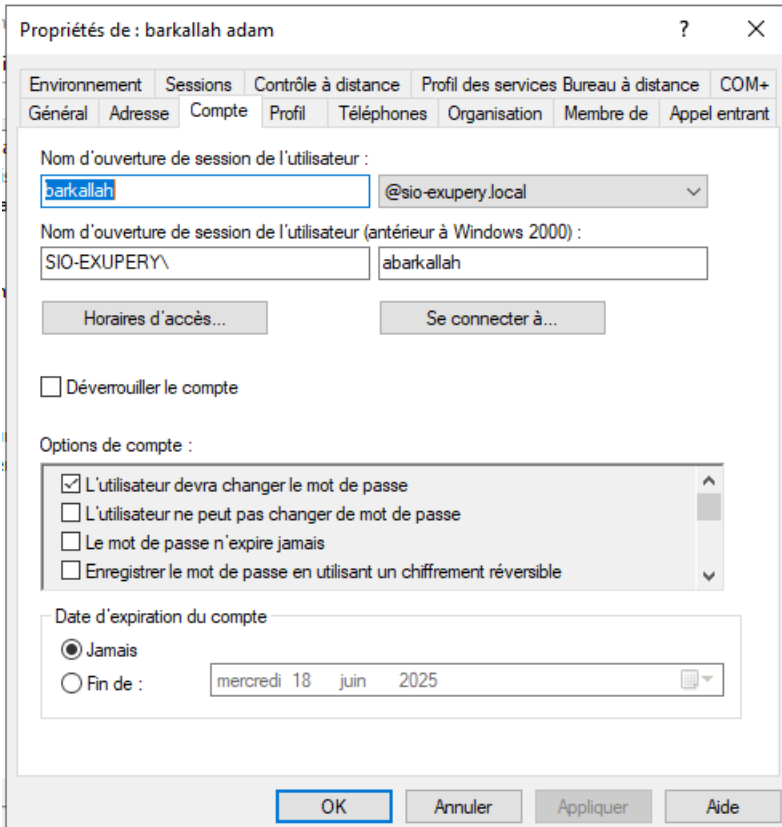
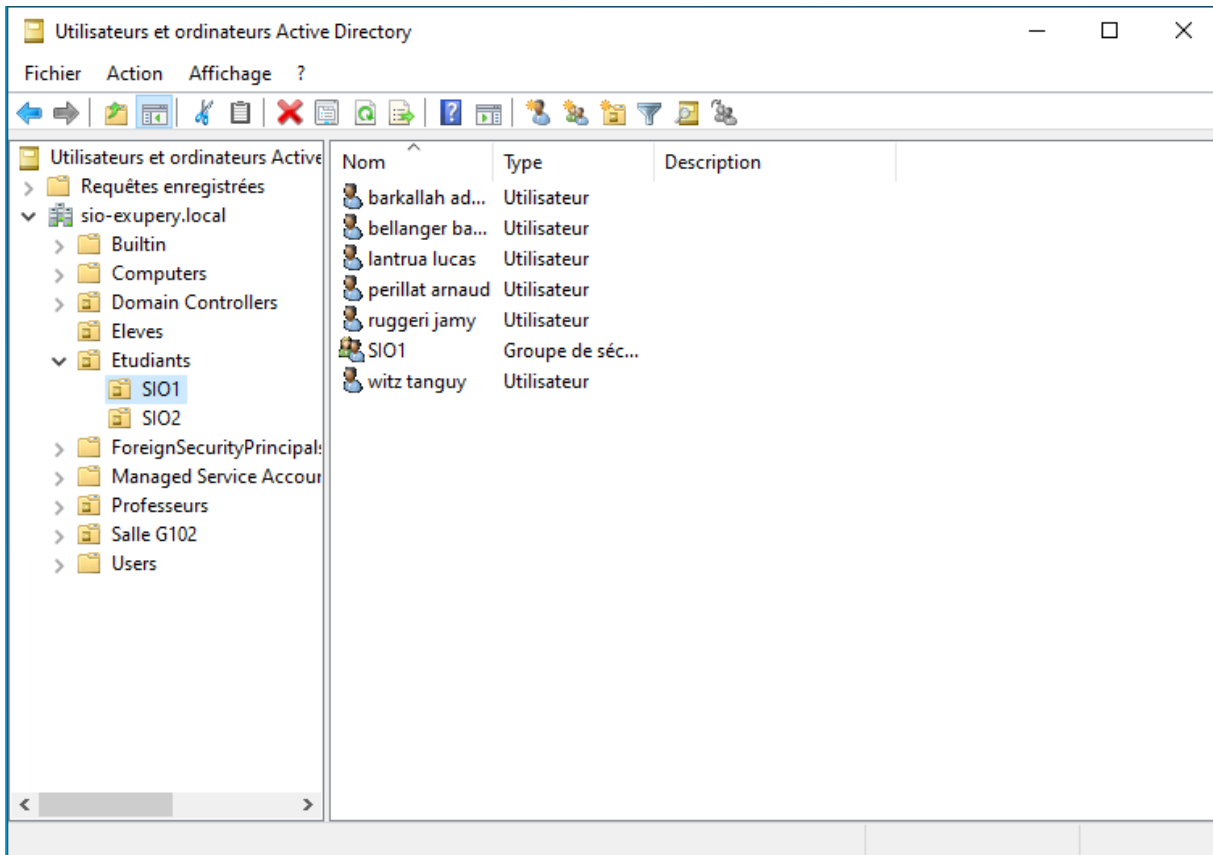
```
PS Microsoft.PowerShell.Core\FileSystem::\\AD\Home$> C:\Users\Administrateur\Documents\Add-Users.ps1
```

```
Répertoire : \\AD\Home$
```

Mode	LastWriteTime	Length	Name
d----	19/05/2025 13:16		gcotta
d----	19/05/2025 13:16		evanpraet
d----	19/05/2025 13:16		jcavin
d----	19/05/2025 13:16		egauthier
d----	19/05/2025 13:16		ckaczmarek
d----	19/05/2025 13:16		bbersia
d----	19/05/2025 13:16		llantrua
d----	19/05/2025 13:16		twitz
d----	19/05/2025 13:16		jruggeri
d----	19/05/2025 13:16		bbellanger
d----	19/05/2025 13:16		abarkallah
d----	19/05/2025 13:16		aperillat

```
PS Microsoft.PowerShell.Core\FileSystem::\\AD\Home$>
```

- Je constate la création des comptes au sein de chaque UO et l'affectation aux groupes :



- Je vérifie les autorisations NTFS définies sur les répertoires personnels :

