

Scripts : Introduction à PowerShell – Module Active Directory

Table des matières

Scripts : Introduction à PowerShell – Module Active Directory	1
2. Prise en main.....	2
2.1. Découverte de la console ligne de commandes	2
2.2. L’environnement d’écriture de scripts intégré (interface ISE).....	2
2.3. Transition avec le passé.....	3
3. Les commandes de base.....	4
3.1. Constitution des commandes	4
3.2. Get-Command.....	5
3.3. Get-Help.....	7
3.4. Get-Member et le concept des objets.....	7
4. Navigation dans les répertoires et les fichiers.....	9
4.1. Les nouvelles commandes	9
4.2. Get-ChildItem (Alias : gci, ls, dir)	9
4.3. Set-Location (Alias : sl, cd, chdir)	11
4.4. Get-Location (Alias : gl, pwd)	11
4.5. New-Item (Alias : ni, md).....	12
4.6. Remove-Item (Alias : ri, rm, rmdir, rd, erase, del)	13
4.7. Move-Item (Alias : mi, move, mv)	13
4.8. Rename-Item (Alias : ren, rni)	14
4.9. Copy-Item (Alias : cpi, cp, copy).....	14
6. Module Active Directory.....	14
6.1. Mise en route du module.....	14
6.2. Gestion des utilisateurs	14
6.3. Gestion des groupes	Erreur ! Signet non défini.
7. TP : Création des comptes utilisateurs par lots.	15

2. Prise en main.

2.1. Découverte de la console ligne de commandes

Nous effectuons un `$PSVersionTable` pour savoir quelle version est installée sur notre poste de travail.

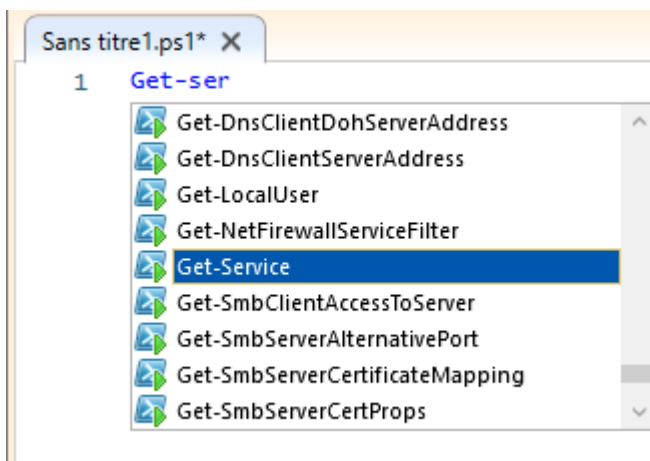
```
PS C:\Users\nnicolau> $PSVersionTable

Name                           Value
-----                           -
PSVersion                       5.1.26100.4061
PSEdition                       Desktop
PSCompatibleVersions             {1.0, 2.0, 3.0, 4.0...}
BuildVersion                    10.0.26100.4061
CLRVersion                      4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1

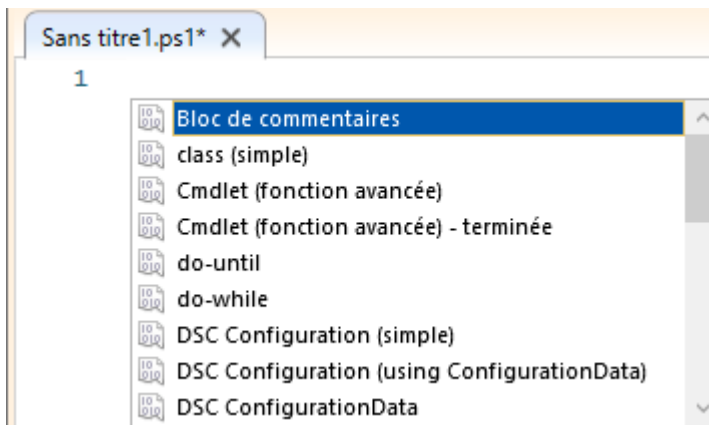
PS C:\Users\nnicolau>
```

2.2. L'environnement d'écriture de scripts intégré (interface ISE)

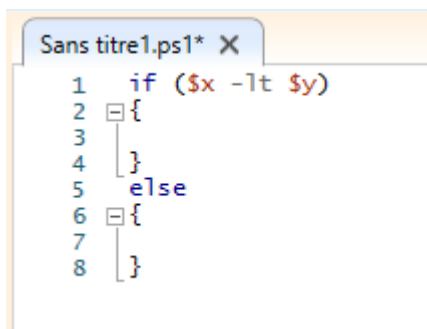
Avec la fonctionnalité « IntelliSense », il nous suffit à présent de commencer à frapper au clavier le début d'une commande pour qu'une proposition soit faite, tout en indiquant la syntaxe :



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 :

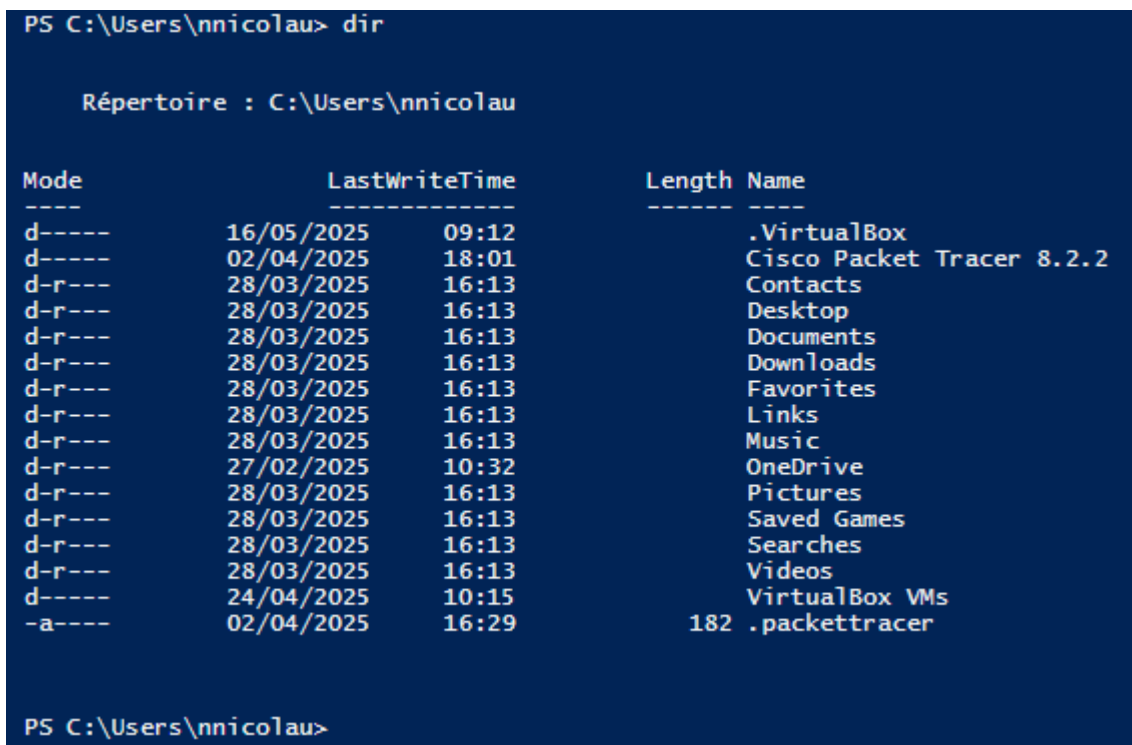


Une fois sélectionné, le code PowerShell s'inscrit directement dans la page de script :



2.3. Transition avec le passé

Nous saisissons par exemple la commande dir :



Et nous retapons la même commande dans l'invite de commande CMD :

```

PS C:\Users\nnicolau> cd c:\
PS C:\> dir

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----           25/04/2025         12:56    inetpub
d-----           01/04/2024         09:26    PerfLogs
d-r---           29/04/2025         08:29    Program Files
d-r---           01/04/2025         15:12    Program Files (x86)
d-r---           12/05/2025         13:48    Users
d-----           15/05/2025         17:29    Windows

```

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

```

PS C:\> 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.PowerShell...
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
Alias             cpp -> Copy-ItemProperty
Alias             curl -> Invoke-WebRequest
Alias             cvpa -> Convert-Path
Alias             dbp -> Disable-PSBreakpoint
Alias             del -> Remove-Item
Alias             diff -> Compare-Object
Alias             dir -> Get-ChildItem
Alias             dnsn -> Disconnect-PSSession
Alias             ebp -> Enable-PSBreakpoint
Alias             echo -> Write-Output
Alias             epal -> Export-Alias

```

3. Les commandes de base.

3.1. Constitution des commandes

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

```
PS C:\> 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
```

3.2. Get-Command

Nous ajoutons le paramètre -CommandType suivi du type de commandes choisi, à savoir cmdlet :

```
PS C:\> 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-AzureAccount                       5.1.2       Azure
Cmdlet           Add-AzureAnalysisServicesAccount        0.5.0       Azure.AnalysisServices
Cmdlet           Add-AzureApplicationGatewaySslCertificate 5.1.2       Azure
Cmdlet           Add-AzureCertificate                   5.1.2       Azure
Cmdlet           Add-AzureDataDisk                       5.1.2       Azure
```

Pour compter le nombre qu'il y a, nous saisissons la commande suivante :

```
PS C:\> (Get-Command -CommandType cmdlet).count
3214
PS C:\>
```

NICOLAU Noah

Get-Command permet d'obtenir par exemple toutes les commandes dont le verbe commence par Write :

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

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

```
PS C:\>
```

De manière similaire, avec cette fois-ci le paramètre -noun, nous affichons les commandes qui s'appliquent, par exemple, aux objets.

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

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

```
PS C:\>
```

Nous affichons les commandes de type alias :

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

CommandType	Name	Version	Source
Alias	% -> ForEach-Object		
Alias	? -> Where-Object		
Alias	ac -> Add-Content		
Alias	Add-AdlAnalyticsDataSource	4.2.3	AzureRM.DataLakeAnal...
Alias	Add-AdlAnalyticsFirewallRule	4.2.3	AzureRM.DataLakeAnal...
Alias	Add-AdlStoreFirewallRule	5.2.0	AzureRM.DataLakeStore
Alias	Add-AdlStoreItemContent	5.2.0	AzureRM.DataLakeStore
Alias	Add-AdlStoreTrustedIdProvider	5.2.0	AzureRM.DataLakeStore
Alias	Add-AppPackage	2.0.1.0	Appx
Alias	Add-AppPackageVolume	2.0.1.0	Appx
Alias	Add-AppProvisionedPackage	3.0	Dism
Alias	Add-AzureHDInsightConfigValues	5.1.2	Azure
Alias	Add-AzureHDInsightMetastore	5.1.2	Azure
Alias	Add-AzureHDInsightStorage	5.1.2	Azure
Alias	Add-AzureRmAccount	4.6.0	AzureRM.Profile
Alias	Add-AzureRmIotHubEHCG	3.1.2	AzureRM.IotHub

NICOLAU Noah

Nous recherchons une commande dont on ignore le nom, mais sachant qu'elle commence par le verbe get :

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

CommandType	Name	Version	Source
Alias	Get-AdlAnalyticsAccount	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlAnalyticsComputePolicy	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlAnalyticsDataSource	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlAnalyticsFirewallRule	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlCatalogItem	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlJob	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlJobPipeline	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlJobRecurrence	4.2.3	AzureRM.DataLakeAnal...
Alias	Get-AdlStore	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreChildItem	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreFirewallRule	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreItem	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreItemAclEntry	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreItemContent	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreItemOwner	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreItemPermission	5.2.0	AzureRM.DataLakeStore
Alias	Get-AdlStoreTrustedIdProvider	5.2.0	AzureRM.DataLakeStore
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

3.3. Get-Help

3.4. Get-Member et le concept des objets

Nous créons la variable \$maVariable et affectons une valeur de type chaîne.

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

Nous tapons la commande \$maVariable | get-member :

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

    TypeName : System.String

Name      MemberType      Definition
-----
Clone     Method          System.Object Clone(), System.Object ICloneable.Clone()
CompareTo Method          int CompareTo(System.Object value), int CompareTo(string st...
Contains  Method          bool Contains(string value)
CopyTo    Method          void CopyTo(int sourceIndex, char[] destination, int destin...
EndsWith  Method          bool EndsWith(string value), bool EndsWith(string value, Sy...
Equals    Method          bool Equals(System.Object obj), bool Equals(string value), ...
GetEnumerator Method        System.CharEnumerator GetEnumerator(), System.Collections.I...
GetHashCode Method        int GetHashCode()
GetType   Method          type GetType()
GetTypeCode Method        System.TypeCode GetTypeCode(), System.TypeCode IConvertible...
IndexOf   Method          int IndexOf(char value), int IndexOf(char value, int startI...
IndexOfAny Method        int IndexOfAny(char[] anyOf), int IndexOfAny(char[] anyOf, ...
Insert    Method          string Insert(int startIndex, string value)
IsNormalized Method        bool IsNormalized(), bool IsNormalized(System.Text.Normaliz...
LastIndexOf Method        int LastIndexOf(char value), int LastIndexOf(char value, in...
LastIndexOfAny Method        int LastIndexOfAny(char[] anyOf), int LastIndexOfAny(char[]...
Normalize Method          string Normalize(), string Normalize(System.Text.Normalizat...
PadLeft   Method          string PadLeft(int totalWidth), string PadLeft(int totalWid...
PadRight  Method          string PadRight(int totalWidth), string PadRight(int totalW...
Remove    Method          string Remove(int startIndex, int count), string Remove(int...
Replace   Method          string Replace(char oldChar, char newChar), string Replace(...
Split     Method          string[] Split(Params char[] separator), string[] Split(cha...
StartsWith Method        bool StartsWith(string value), bool StartsWith(string value...
Substring Method        string Substring(int startIndex), string Substring(int star...
ToBoolean Method        bool IConvertible.ToBoolean(System.IFormatProvider provider)
ToByte    Method        byte IConvertible.ToByte(System.IFormatProvider provider)
ToChar    Method        char IConvertible.ToChar(System.IFormatProvider provider)
ToCharArray Method        char[] ToCharArray(), char[] ToCharArray(int startIndex, in...
ToDateTime Method        datetime IConvertible.ToDateTime(System.IFormatProvider pro...
ToType    Method        System.Object IConvertible.ToType(type conversionType, Syst...
ToUInt16  Method        uint16 IConvertible.ToUInt16(System.IFormatProvider provider)
ToUInt32  Method        uint32 IConvertible.ToUInt32(System.IFormatProvider provider)
ToUInt64  Method        uint64 IConvertible.ToUInt64(System.IFormatProvider provider)
ToUpper   Method        string ToUpper(), string ToUpper(cultureinfo culture)
ToUpperInvariant Method    string ToUpperInvariant()
Trim      Method        string Trim(Params char[] trimChars), string Trim()
TrimEnd   Method        string TrimEnd(Params char[] trimChars)
TrimStart Method        string TrimStart(Params char[] trimChars)
Chars     ParameterizedProperty char Chars(int index) {get;}
Length    Property        int Length {get;}

PS C:\>
```

Nous utilisons la méthode ToUpper afin de retourner la chaîne contenue dans \$maVariable en majuscules.

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

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

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

4. Navigation dans les répertoires et les fichiers.

4.1. Les nouvelles commandes

Nous tapons la commande Get-Alias dir pour voir L'alias qui est exécuté derrière.

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

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

4.2. Get-ChildItem (Alias : gci, ls, dir)

Nous observons le résultat de la commande suivante.

```
PS C:\WINDOWS\system32> gci c:\

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----           25/04/2025   12:56      inetpub
d-----           01/04/2024   09:26      PerfLogs
d-r---           29/04/2025   08:29      Program Files
d-r---           01/04/2025   15:12      Program Files (x86)
d-r---           12/05/2025   13:48      Users
d-----           15/05/2025   17:29      Windows
```

Nous observons la colonne Mode.

```
PS C:\WINDOWS\system32> gci c:\

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----           25/04/2025   12:56      inetpub
d-----           01/04/2024   09:26      PerfLogs
d-r---           29/04/2025   08:29      Program Files
d-r---           01/04/2025   15:12      Program Files (x86)
d-r---           12/05/2025   13:48      Users
d-----           15/05/2025   17:29      Windows
```

NICOLAU Noah

Nous affichons les fichiers cachés et ajoutons à la commande Get-ChildItem le paramètre -Force :

```
PS C:\WINDOWS\system32> gci c:\ -Force

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d--hs-             12/05/2025         13:48      $Recycle.Bin
d--hs-             15/05/2025         16:30      Config.Msi
d--hsl            13/01/2025         16:17      Documents and Settings
d-----           25/04/2025         12:56      inetpub
d-----           01/04/2024          09:26      PerfLogs
d-r---            29/04/2025          08:29      Program Files
d-r---            01/04/2025         15:12      Program Files (x86)
d--h--            01/04/2025         15:13      ProgramData
d--hs-            22/04/2025         16:21      Recovery
d--hs-            16/05/2025          09:06      System Volume Information
d-r---            12/05/2025         13:48      Users
d-----           15/05/2025         17:29      Windows
-a-hs-            16/05/2025         14:38           12288 DumpStack.log.tmp
-a-hs-            16/05/2025         14:38      27451232256 hiberfil.sys
-a-hs-            16/05/2025         14:38      4294967296 pagefile.sys
-a-hs-            16/05/2025         14:38           16777216 swapfile.sys

PS C:\WINDOWS\system32>
```

Nous affichons tous les fichiers ayant l'extension .log contenus à l'intérieur d'une arborescence.

```
PS C:\> Get-ChildItem C:\ -Include *.log -Recurse

Répertoire : C:\Program Files\LibreOffice\program\logs

Mode                LastWriteTime         Length Name
----                -
-a----            22/04/2025         11:54           5108 maintenanceservice-1.log
-a----            27/02/2025         10:36           5141 maintenanceservice-2.log
-a----            27/02/2025         10:34           5141 maintenanceservice-3.log
-a----            23/01/2025         11:29            268 maintenanceservice-install.log
-a----            22/04/2025         11:54           5111 maintenanceservice.log
```

Nous affichons tous les fichiers ou répertoires cachés à la racine de la partition système.

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

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d--hs-             12/05/2025   13:48          $Recycle.Bin
d--hs-             15/05/2025   16:30          Config.Msi
d--hsl             13/01/2025   16:17      Documents and Settings
d--h--             01/04/2025   15:13          ProgramData
d--hs-             22/04/2025   16:21          Recovery
d--hs-             16/05/2025   09:06      System Volume Information
-a-hs-             16/05/2025   14:38      12288 DumpStack.log.tmp
-a-hs-             16/05/2025   14:38  27451232256 hiberfil.sys
-a-hs-             16/05/2025   14:38  4294967296 pagefile.sys
-a-hs-             16/05/2025   14:38   16777216 swapfile.sys
```

Nous filtrons uniquement les fichiers cachés et non les répertoires en combinant les attributs « caché » et « n'est pas un répertoire ».

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

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
-a-hs-             16/05/2025   14:38      12288 DumpStack.log.tmp
-a-hs-             16/05/2025   14:38  27451232256 hiberfil.sys
-a-hs-             16/05/2025   14:38  4294967296 pagefile.sys
-a-hs-             16/05/2025   14:38   16777216 swapfile.sys

PS C:\>
```

4.3. Set-Location (Alias : sl, cd, chdir)

Nous saisissons, par exemple.

```
PS C:\> Set-Location D:\
PS D:\>
```

4.4. Get-Location (Alias : gl, pwd)

Nous saisissons Get-Location.

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

Path
----
C:\users

PS C:\users>
```

NICOLAU Noah

Nous affichons 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>
```

Nous récupérons dans une variable le chemin de l'emplacement courant en une seule ligne de commande.

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

Nous stockons la valeur du chemin courant, en l'occurrence C:\users, dans la variable \$CheminCourant et nous l'affichons dans la console :

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

4.5. New-Item (Alias : ni, md)

4.5.1. Création d'un répertoire

Nous créons le répertoire Temp.

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

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----           16/05/2025   15:06         Temp

PS C:\>
```

Si le dossier avait contenu un espace, nous aurions dû mettre entre guillemets le nom. Par exemple :

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

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----            16/05/2025   15:07             Dossier Test

PS C:\>
```

4.5.2. Création d'un fichier

Nous créons un fichier nommé monFichier.txt qui contient la phrase « Madame Pasquier for ever ! » :

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

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
-a-----            16/05/2025   15:08             25 monFichier.txt

PS C:\>
```

4.6. Remove-Item (Alias : ri, rm, rmdir, rd, erase, del)

Nous supprimons tous les fichiers. logs contenus dans le répertoire C:\Temp :

```
PS C:\> Remove-Item C:\Temp\*.log
PS C:\>
```

Nous supprimons tous les fichiers contenus dans une arborescence de dossiers dont l'extension est .txt :

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

4.7. Move-Item (Alias : mi, move, mv)

4.7.1. Déplacement de fichiers

Nous déplaçons les fichiers *.jpg du répertoire courant vers le dossier « mes photos ».

```
PS C:\> Move-Item *.jpg Mes photos
PS C:\>
```

4.7.2. Déplacement d'un répertoire

NICOLAU Noah

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

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

4.8. Rename-Item (Alias : ren, rni)

4.8.1. Renommer un fichier

Nous renommons le fichier monFichierDeLog.txt en ficlog.txt.

```
PS C:\Users\nnicolau\Documents> Rename-Item C:\Users\nnicolau\Documents\MonFichierDeLog.txt ficlog.txt  
PS C:\Users\nnicolau\Documents>
```

4.8.2. Renommer un dossier

Nous renommons le répertoire Dossier1 en Dossier2.

```
PS C:\Users\nnicolau\Documents> Rename-Item C:\Users\nnicolau\Documents\Dossier1 Dossier2  
PS C:\Users\nnicolau\Documents>
```

4.9. Copy-Item (Alias : cpi, cp, copy)

Nous pouvons copier des fichiers ou des répertoires, voire les deux à la fois.

```
PS C:\Users\nnicolau\Documents> Copy-Item -Path C:\Users\nnicolau\Documents\ficLog.txt -destination D:\Logs  
PS C:\Users\nnicolau\Documents>
```

6. Module Active Directory.

6.1. Mise en route du module

Nous pouvons obtenir l'ensemble des commandes apportées par le module Active Directory en tapant ceci.

```
PS C:\Users\Administrateur> Get-Command -Module ActiveDirectory  
  
CommandType      Name  
-----  
Cmdlet           Add-ADCentralAccessPolicyMember  
Cmdlet           Add-ADComputerServiceAccount  
Cmdlet           Add-ADDomainControllerPasswordReplication...  
Cmdlet           Add-ADFineGrainedPasswordPolicySubject  
Cmdlet           Add-ADGroupMember  
Cmdlet           Add-ADPrincipalGroupMembership  
Cmdlet           Add-ADResourcePropertyListMember  
Cmdlet           Clear-ADAccountExpiration  
Cmdlet           Clear-ADClaimTransformLink  
Cmdlet           Disable-ADAccount  
Cmdlet           Disable-ADOptionalFeature  
Cmdlet           Enable-ADAccount
```

6.2. Gestion des utilisateurs

Pour la gestion des utilisateurs, nous disposons d'un jeu de quelques commandes que nous pouvons obtenir de la manière suivante :

```
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>
```

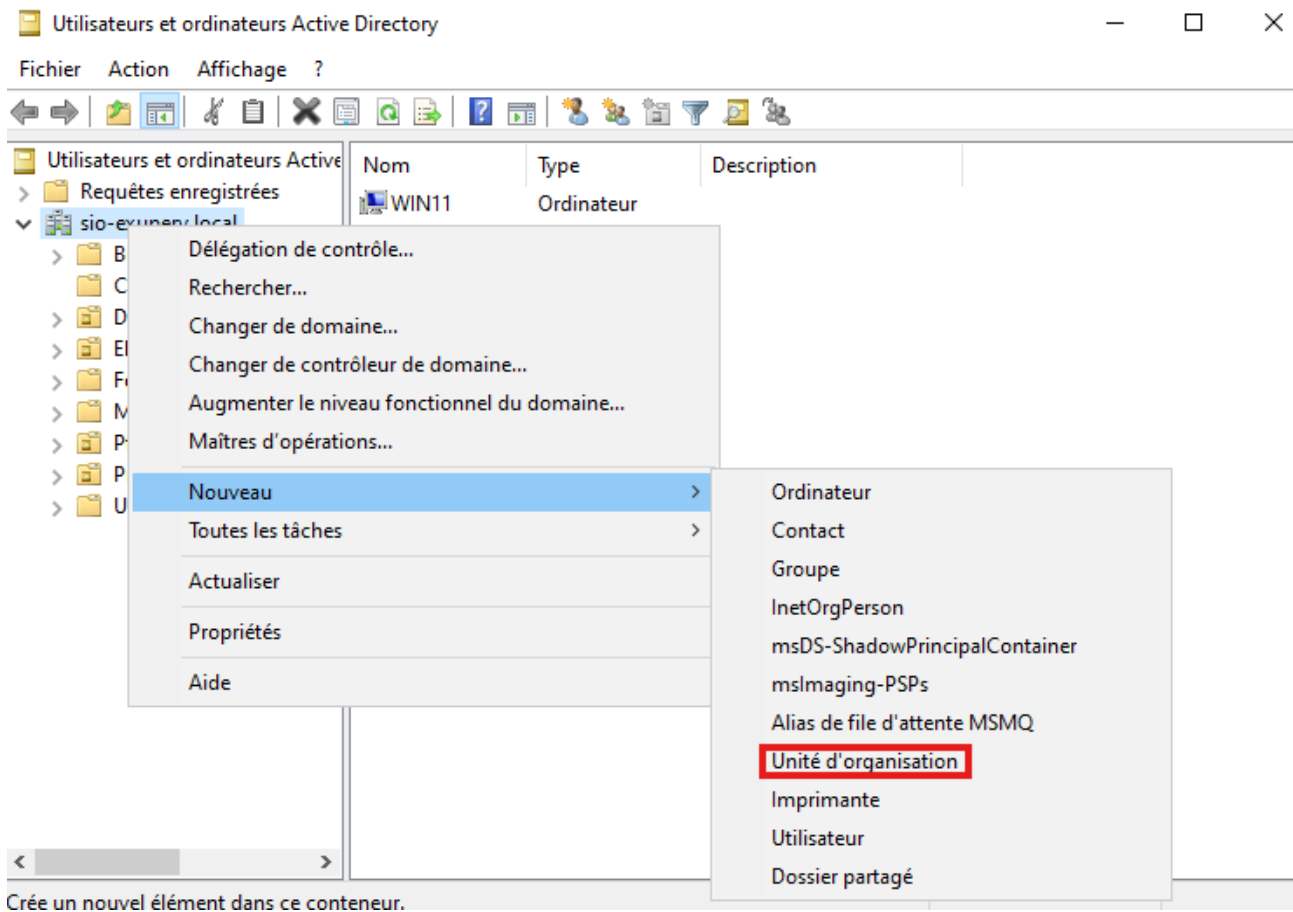
6.2.1. Créer des utilisateurs

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>
```

7. TP : Création des comptes utilisateurs par lots.

Nous créons l'UO Etudiants puis, à l'intérieur de celle-ci, les UO BTSSIO1 et BTSSIO2.





Créer dans : sio-exupery.local/

Nom :

Protéger le conteneur contre une suppression accidentelle

OK Annuler Aide

Utilisateurs et ordinateurs Active Directory

Fichier Action Affichage ?

Utilisateurs et ordinateurs Active Directory

- Requêtes enregistrées
- sio-exupery.local
 - Builtin
 - Computers
 - Domain Controllers
 - Elevés
 - ForeignSecurityPrincipal...
 - Managed Service Account...
 - PfSense
 - Professeurs
 - Users
 - Etudiants**

Aucun élément à afficher dans cet aperçu.


Nom	Type	Description
Aucun élément à afficher dans cet aperçu.		

Délégation de contrôle...
Déplacer...
Rechercher...
Nouveau >
Toutes les tâches >
Affichage >
Couper
Supprimer
Renommer
Actualiser

- Ordinateur
- Contact
- Groupe
- InetOrgPerson
- msDS-ShadowPrincipalContainer
- msImaging-PSPs
- Alias de file d'attente MSMQ
- Unité d'organisation**

Nouvel objet - Unité d'organisation




 Créer dans : sio-exupery.local/Etudiants

Nom :

Protéger le conteneur contre une suppression accidentelle

Nouvel objet - Unité d'organisation

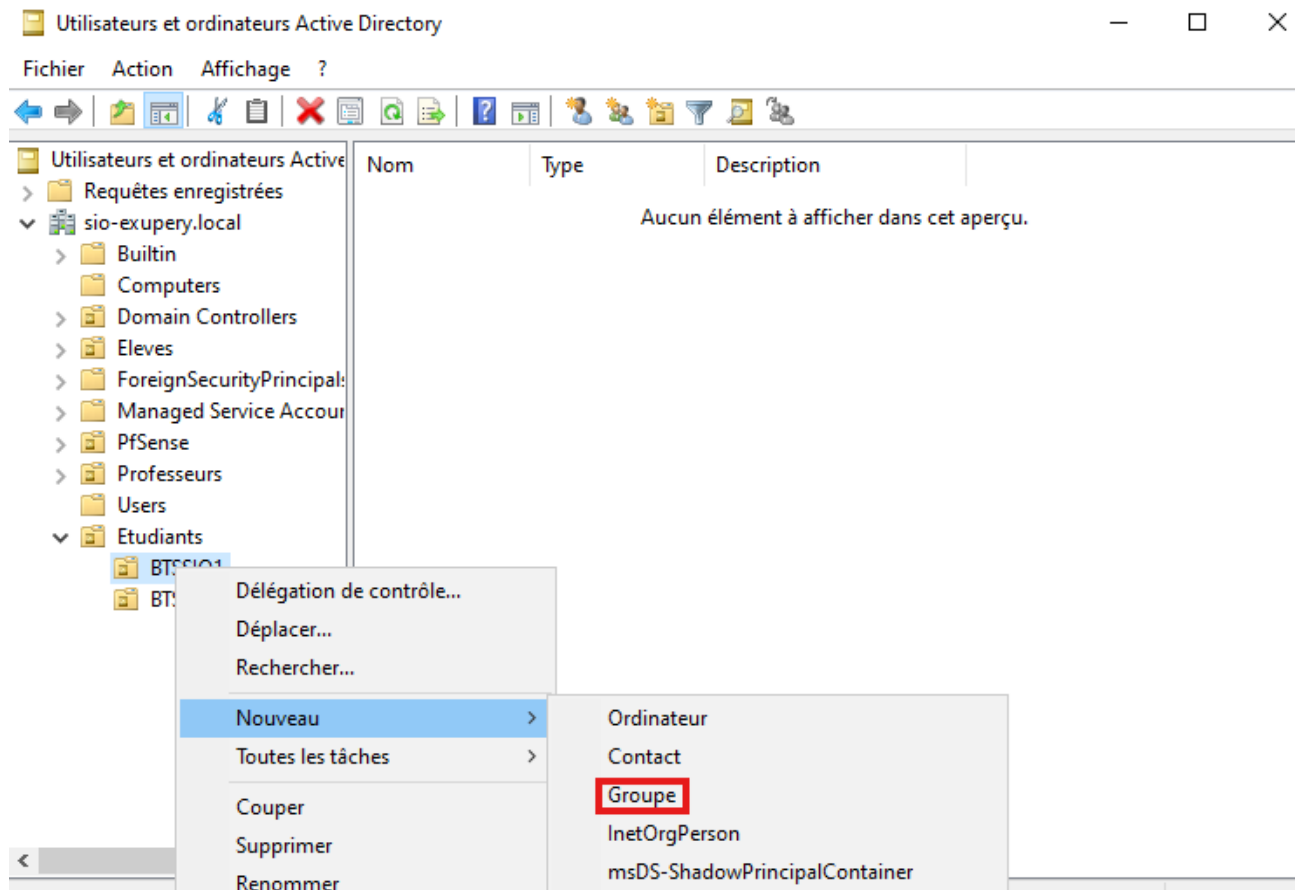


 Créer dans : sio-exupery.local/Etudiants

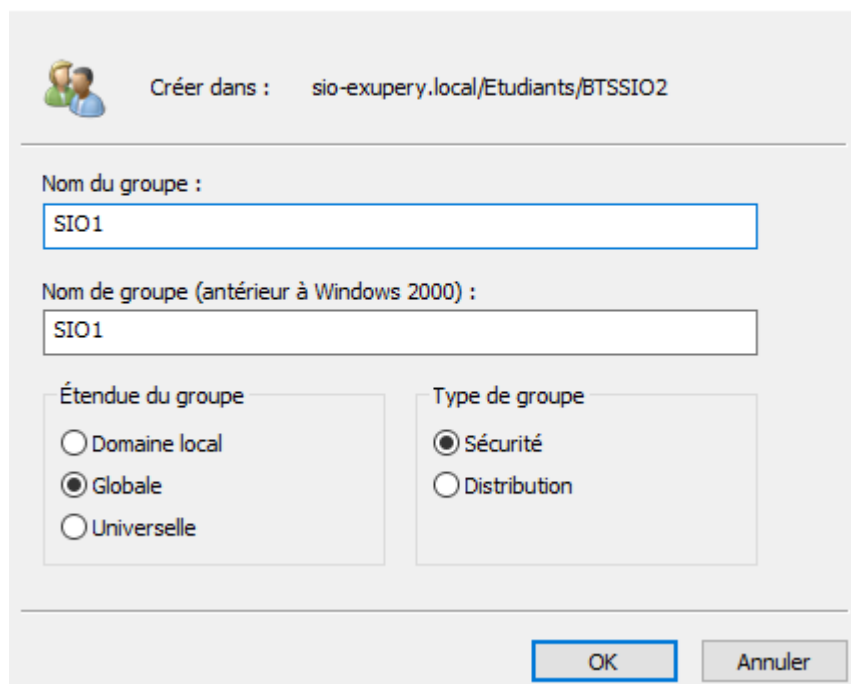
Nom :

Protéger le conteneur contre une suppression accidentelle


Nous créons les groupes de sécurité SIO1 et SIO2 respectivement dans les UO BTSSIO1 et BTSSIO2.



Nouvel objet - Groupe






 Créer dans : sio-exupery.local/Etudiants/BTSSIO2

Nom du groupe :

Nom de groupe (antérieur à Windows 2000) :

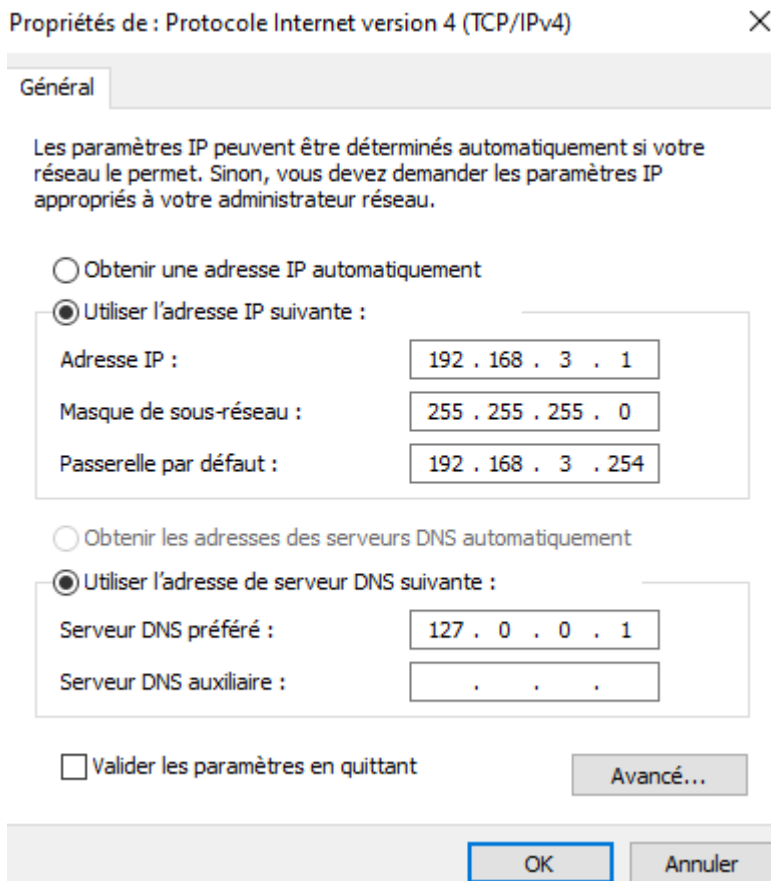
Étendue du groupe	Type de groupe
<input type="radio"/> Domaine local	<input checked="" type="radio"/> Sécurité
<input checked="" type="radio"/> Globale	<input type="radio"/> Distribution
<input type="radio"/> Universelle	

Nous créons un fichier .csv qui sera utilisé dans un script pour créer des users.

 *Sans titre - Bloc-notes

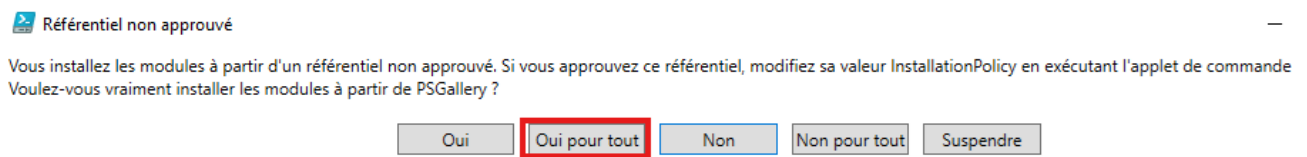
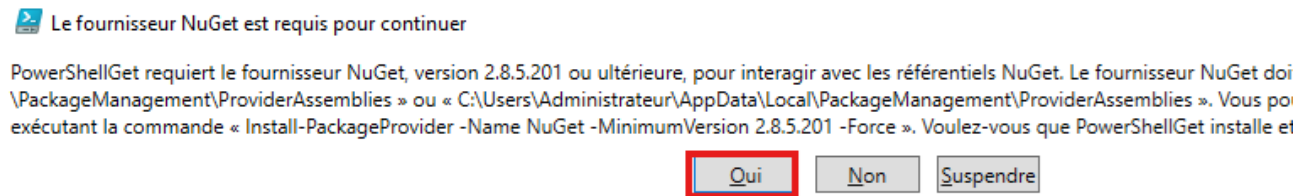
Fichier Edition Format Affichage Aide

```
name,givenname,classe
cotta,gabriel,SI02
vanpraett,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|
```



Pour afficher le NTFS, nous l'installons avant :

```
PS C:\Users\Administrateur> Install-Module -name NTFSSecurity
```



Nous affichons 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
Cmdlet	Set-NTFSOwner	4.2.6	NTFSSecurity
Cmdlet	Set-NTFSSecurityDescriptor	4.2.6	NTFSSecurity
Cmdlet	Show-NTFSSimpleAccess	4.2.6	NTFSSecurity
Application	chkntfs.exe	10.0.20...	C:\Windows\system32\chkntfs.exe

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

Nous écrivons le script dans PowerShell ISE :

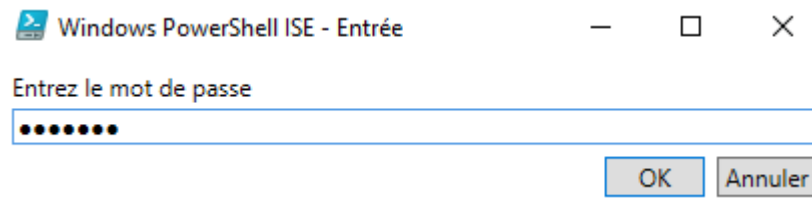
```

1 $inpfile = "C:\comptes.csv"
2 $password = Read-Host "Entrez le mot de passe" -AsSecureString
3 $users = Import-Csv $inpfile -Delimiter ','
4 Foreach ($user in $users)
5 {
6     $name = $user.name
7     $givenname = $user.givenname
8     $samname = $givenname.substring(0,1) + $name
9     $surname = $user.name
10    $section = $user.classe
11    $domain = "sio-exupery"
12    $ext = "local"
13    $drive = "H:"
14    $hdirectory = "\\AD\Homes"
15    $dossier = "$hdirectory\$samname"
16    $profil = "\\AD\Profils\$samname"
17
18    New-ADUser -Name "$name $givenname" -UserPrincipalName "$name@$domain.$ext" -
19    SamAccountName $samname -GivenName $givenname -Surname $surname -Path "ou=Section,ou=Etudiants,dc=sio-exupery,dc=local" -HomeDrive $drive -HomeDirectory $dossier -ProfilePath $profil -Enabled $true -ChangePasswordAtLogon $true - AccountPassword $password
20    Set-Location $hdirectory
21    New-Item -Name $samname -ItemType directory
22    Get-Item "$dossier" | Disable-NTFSAccessInheritance
23    Set-NTFSOwner -Path "$dossier" -Account "sio-exupery.local\$samname"
24    Add-NTFSAccess -Path "$dossier" -Account "sio-exupery.local\$samname" -AccessRights
25    FullControl
26    Get-NTFSAccess -Path "$dossier" -Account "Utilisateurs" -ExcludeInherited | Remove-NTFSAccess
27    Add-ADGroupMember -Identity "$section" -Members $samname
28 }
    
```

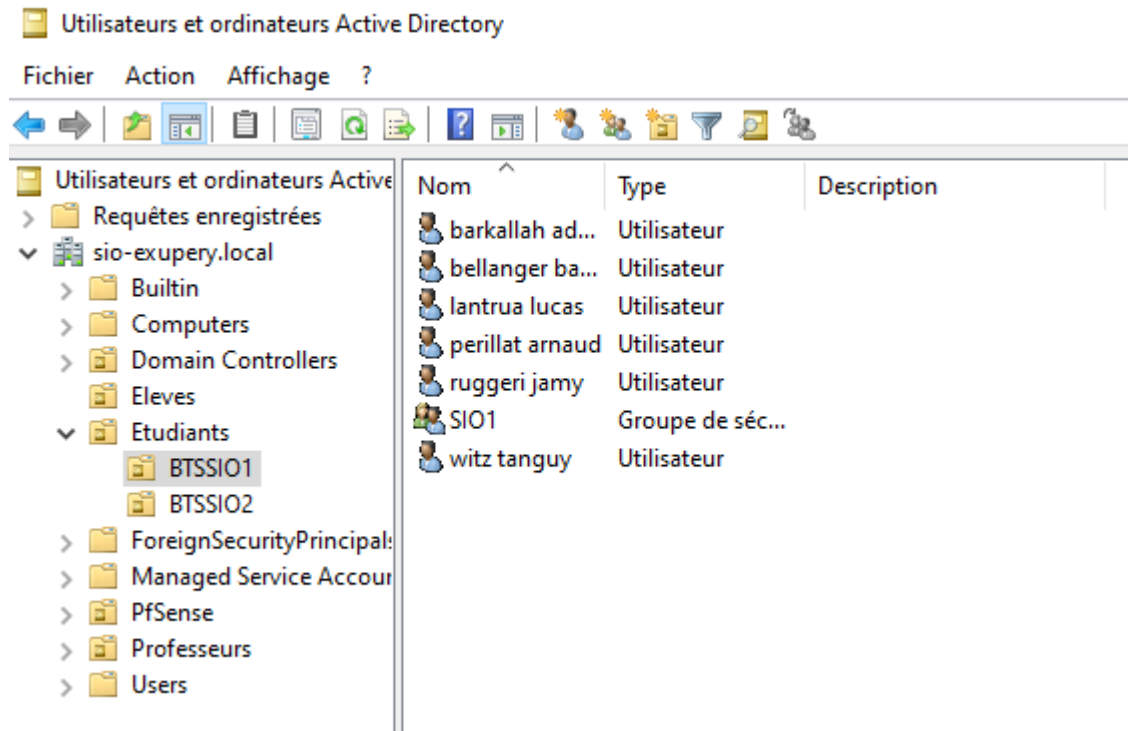
Nous exécutons le script :

```

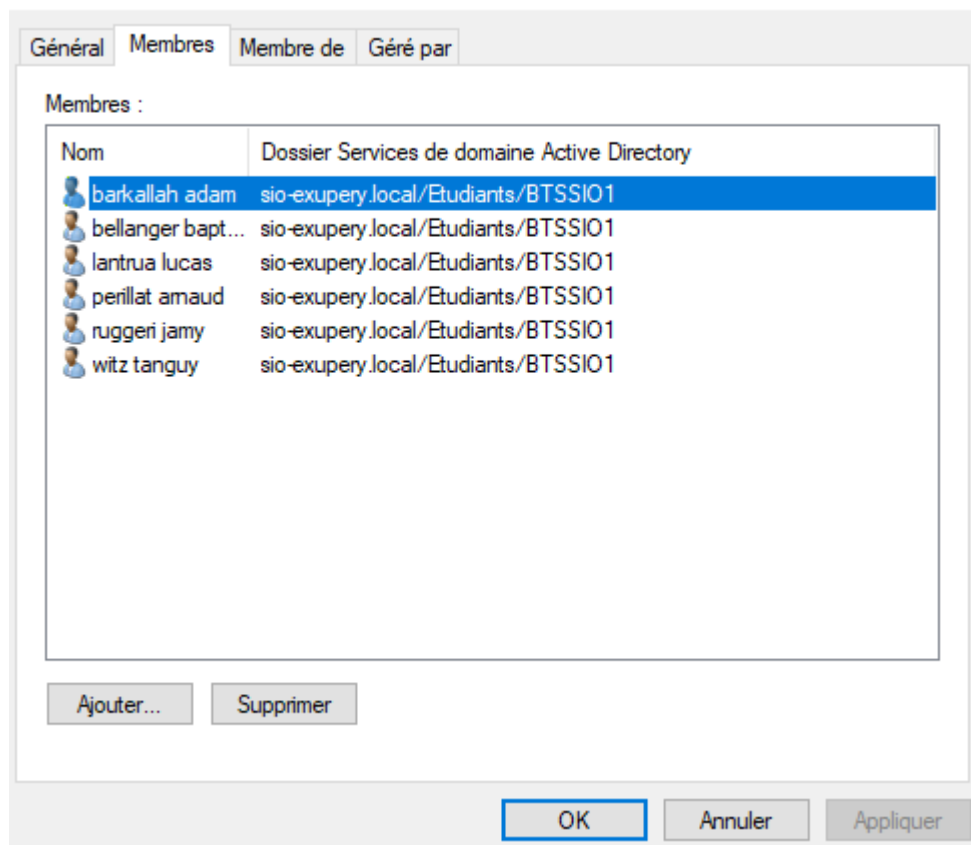
1 $inpfile = "C:\comptes.csv"
2 $password = Read-Host "Entrez le mot de passe" -AsSecureString
3 $users = Import-Csv $inpfile -Delimiter ','
4 Foreach ($user in $users)
5 {
6     $name = $user.name
7     $givenname = $user.givenname
8     $samname = $givenname.substring(0,1) + $name
9     $surname = $user.name
10    $section = $user.classe
11    $domain = "sio-exupery"
12    $ext = "local"
13    $drive = "H:"
14    $hdirectory = "\\AD\Homes"
15    $dossier = "$hdirectory\$samname"
16    $profil = "\\AD\Profils\$samname"
17
18    New-ADUser -Name "$name $givenname" -UserPrincipalName "$name@$domain.$ext" -
19    SamAccountName $samname -GivenName $givenname -Surname $surname -Path "ou=Section,ou=Etudiants,dc=sio-exupery,dc=local" -HomeDrive $drive -HomeDirectory $dossier -ProfilePath $profil -Enabled $true -ChangePasswordAtLogon $true - AccountPassword $password
20    Set-Location $hdirectory
21    New-Item -Name $samname -ItemType directory
22    Get-Item "$dossier" | Disable-NTFSAccessInheritance
23    Set-NTFSOwner -Path "$dossier" -Account "sio-exupery.local\$samname"
24    Add-NTFSAccess -Path "$dossier" -Account "sio-exupery.local\$samname" -AccessRights
25    FullControl
26    Get-NTFSAccess -Path "$dossier" -Account "Utilisateurs" -ExcludeInherited | Remove-NTFSAccess
27    Add-ADGroupMember -Identity "$section" -Members $samname
28 }
    
```

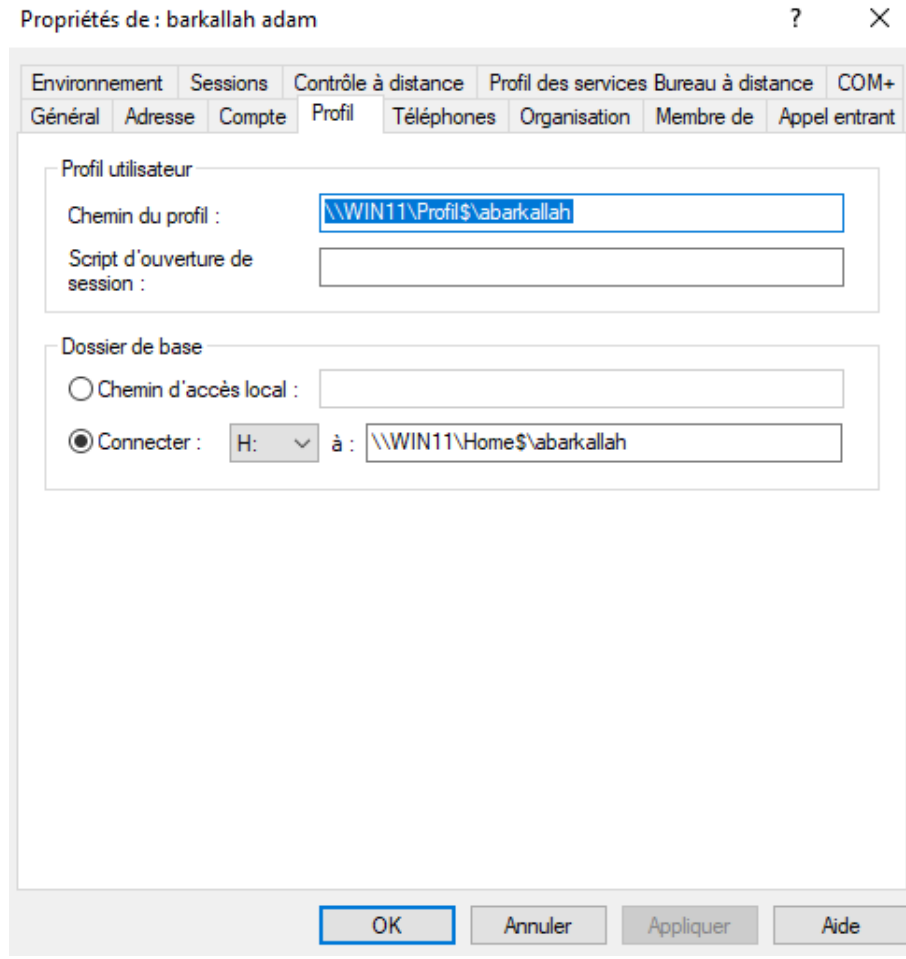


Après le lancement du script, nous constatons la création des comptes au sein de chaque UO et l'affectation aux groupes :



Propriétés de : SIO1





Propriétés de : barkallah adam

? X

Environnement Sessions Contrôle à distance Profil des services Bureau à distance COM+

Général Adresse Compte Profil Téléphones Organisation Membre de Appel entrant

Nom d'ouverture de session de l'utilisateur :
barkallah @sio-exupery.local

Nom d'ouverture de session de l'utilisateur (antérieur à Windows 2000) :
SIO-EXUPERY\ abarkallah

Horaires d'accès... Se connecter à...

Déverrouiller le compte

Options de compte :

- L'utilisateur devra changer le mot de passe
- L'utilisateur ne peut pas changer de mot de passe
- Le mot de passe n'expire jamais
- Enregistrer le mot de passe en utilisant un chiffrement réversible

Date d'expiration du compte

Jamais


Fin de : lundi 16 juin 2025

OK Annuler Appliquer Aide

Propriétés de : SIO2

? X

Général Membres Membre de Géré par

 SIO2

Nom de groupe (antérieur à Windows 2000) : SIO2

Description :

Adresse de messagerie :

Étendue du groupe

- Domaine local
- Globale
- Universelle

Type de groupe

- Sécurité
- Distribution

Remarques :

OK Annuler Appliquer

NICOLAU Noah

Nous vérifions les autorisations NTFS définies sur les répertoires personnels :

Principal : abarkallah (barkallah@sio-exupery.local) [Sélectionnez un principal](#)

Type :

S'applique à :

Autorisations de base :

- Contrôle total
- Modification
- Lecture et exécution
- Affichage du contenu du dossier
- Lecture
- Écriture
- Autorisations spéciales

Appliquer ces autorisations uniquement aux objets et/ou aux conteneurs faisant partie de ce conteneur