

# TP Powershell CESI

---

**None**

*Baptiste DENEUVE*

*None*

# Sommaire

---

|  |    |
|--|----|
| 1. Session 1   | 3  |
| 1.1 Commandes pour débuter                                   | 3  |
| 1.2 Comprendre un script                                     | 3  |
| 1.3 Système de fichiers                                      | 3  |
| 1.4 Gestion système  | 3  |
| 2. Session 2   | 4  |
| 2.1 Les fournisseurs   | 4  |
| 2.2 Formatage de l'affichage                                 | 4  |
| 2.3 Système de fichiers                                      | 4  |
| 2.4 Gestion système  | 4  |
| 2.5 Statistiques, manipulation de texte, conversion en objet | 4  |
| 2.6 Conditions   | 5  |
| 2.7 Créer une boucle   | 5  |
| 2.8 Services WEB   | 5  |
| 3. Session 3   | 6  |
| 3.1 Gestion Active Directory avec Powershell                 | 6  |
| 4. Session 4   | 10 |
| 4.1 Utilisation de WMI/CIM                                   | 10 |
| 4.2 Envoi de message et HTML                                 | 10 |
| 5. Corrections   | 12 |
| 5.1 TP Session 1 & 2   | 12 |
| 5.2 TP Session 3   | 12 |

# 1. Session 1

---

## 1.1 Commandes pour débuter

---

- Lister toutes les commandes de type « functions »
- Lister toutes les commandes se terminant par 'path'.
- N'afficher que les exemples de l'aide pour la commande Get-Process
- Afficher de l'aide concernant For

## 1.2 Comprendre un script

---

- Que fait ce script :

```
Get-Command | Foreach {
    Get-Help $_ -detailed |Out-File -FilePath C:\temp\$_.txt -Encoding ASCII
}
```

## 1.3 Système de fichiers

---

- Lister tous les fichiers du répertoire C:\Windows\system32 et ses sous-répertoires dont la taille est supérieure ou égale à 10 Mo
- Copier seulement les sous-répertoires de l'arborescence (récuratif) du répertoire 'program files' vers un répertoire de votre choix.

## 1.4 Gestion système

---

- Récupérer tous les services arrêtés et envoyer le résultat dans un fichier.
- Lister tous les ids des mise à jour Microsoft et envoyer le résultat dans un fichier.
- Récupérer les processus utilisant au moins 100 Mo.
- Récupérer dans le journal d'événements 'Application', toutes les entrées en erreur.

⌚ December 17, 2024 23:10:26

## 2. Session 2

---

### 2.1 Les fournisseurs

#### 2.1.1 Variables d'environnement

- Récupérer dans une variable le contenu de la variable d'environnement **computername**
- Créer une variable d'environnement nommée **login** et affecter lui la valeur **paul.atreides**

#### 2.1.2 Registre

- Retrouver le chemin complet de la clé **Winlogon** dans le registre (HKEY Current User)
- À l'aide du chemin de registre suivant, extraire les versions et les services pack des Framework .net installés : **HKLM\software\microsoft\Net Framework setup\ndp**
- À l'aide du chemin de registre suivant, activer le bureau à distance de votre machine : **HKLM\SYSTEM\currentcontrolset\control\terminal server\fdenytsconnections**
- À l'aide du chemin de registre suivant, lister toutes les clés usb connectés à votre machine hôte : **HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR**

### 2.2 Formatage de l'affichage

- Afficher toutes les propriétés de la commande **Get-Process**
- Puis choisir de n'afficher que les propriétés **ID, Name et WorkingSet**

### 2.3 Système de fichiers

- Afficher les extensions de fichiers présentes dans le répertoire **c:\Windows** de manière unique et dans l'ordre alphabétique (exclure les extensions 'vides')
- Créer un fichier nommée **date.log** et reculer sa date de création d'un mois.

### 2.4 Gestion système

- Afficher le nombre de services arrêtés et démarrés en **seule ligne de commande**
- Afficher tous les processus fabricant (Microsoft, AMD, Intel, etc..) et trier en ordre décroissant leurs nombres.
- Vérifier la présence d'une erreur disque dans les journaux d'évènements (id event 7)

### 2.5 Statistiques, manipulation de texte, conversion en objet

- Mesurer le nombre de lignes, de mots et caractères d'un fichier texte
- Créer une variable nommée **chaine** avec la valeur **Paul Atreides** et effectuer les opérations suivantes :
- Convertir les majuscules en minuscule
- Remplacer l'espace par un point
- Ajouter le suffixe **@cesi.fr**
- Extraire le prénom et le nom de la chaîne

## 2.6 Conditions

---

- Proposer à l'utilisateur de choisir un chiffre en 1 et 5 et afficher lui le chiffre choisi en toutes lettres

## 2.7 Créer une boucle

---

- Créer une boucle visuelle avec la commande Write-Progress (Barre de progression)

## 2.8 Services WEB

---

- À l'aide de l'URL suivante, récupérer l'ip publique de votre connexion Internet et stocker le résultat dans une variable : <https://api.ipify.org?format=json>
- À l'aide de l'URL suivante, récupérer la liste des communes correspondant au code postal de votre choix : <https://geo.api.gouv.fr/communes?codePostal=<code postal>>

⌚ December 16, 2024 20:27:34

## 3. Session 3

---

### 3.1 Gestion Active Directory avec Powershell

#### 3.1.1 Paramétrage du serveur

Nommer la machine virtuelle à l'aide de la commande Powershell suivante :

```
Rename-Computer -NewName DC1 -force
```

Configurer la carte réseau comme indiqué ci-dessous avec Powershell :

Trouver l'id de la carte réseau local à l'aide de Powershell :

**Réponse (commande utilisée et id de la carte) :**

Pour configurer l'adresse IPv4, la passerelle et le masque de sous réseau :

```
Get-NetAdapter -InterfaceIndex <id_carte_réseau> | New-NetIPAddress -IPAddress 192.168.10.12 -DefaultGateway 192.168.10.1 -PrefixLength 24
```

Que signifie le paramètre PrefixLength dans la commande précédente ?

**Réponse :**

Vérifier la prise en compte de la configuration :

```
Get-NetAdapter -InterfaceIndex <id_carte_réseau> | Get-NetIPConfiguration
```

Tester la connexion avec le routeur :

```
Test-Connection 192.168.10.12 -count 1
```

À quelle commande historique correspond la commande ci-dessous ?

**Réponse :**

Configurer le client DNS

```
Get-NetAdapter -InterfaceIndex <id_carte_réseau> | Set-DnsClientServerAddress -ServerAddresses 192.168.10.12
```

#### 3.1.2 Installation du service d'annuaire Active Directory

Créer le domaine cesi<id>.lan à l'aide de Powershell :

Installation des outils de management et déploiement Active Directory :

```
Install-WindowsFeature AD-Domain-Services,RSAT-ADDS-Tools
```

Comment vérifier les rôles et fonctionnalités installés à l'aide de la console Powershell :

**Réponse :**

Redémarrer le serveur :

```
Restart-Computer
```

Pour installer Active Directory, lancer la commande Powershell suivante :

```
Import-Module ADDSDeployment
Install-ADDSForest -CreateDnsDelegation:$false -DatabasePath "C:\Windows\NTDS" -DomainMode "Win2012" ` 
-DirectoryName "cesi.lan" -DomainNetbiosName "CESI" -ForestMode "Win2012" -InstallDns:$true `
```

```
-LogPath "C:\Windows\NTDS" -NoRebootOnCompletion:$false -SysvolPath "C:\Windows\SYSVOL" -Force:$true ` 
-SafeModeAdministratorPassword (ConvertTo-SecureString "P@ssword" -AsPlainText -Force)
```

Redémarrer le serveur :

[Restart Computer](#)

### 3.1.3 Vérification de l'installation d'Active Directory

Rester la configuration d'Active Directory avec la commande DCDIAG.exe

Tester la résolution de nom directe et inverse du nom FQDN de votre DC :

**Réponse :**

En principe, il est conseillé de gérer son domaine AD depuis une console de type client. Dans cas les outils de gestion RSAT doivent être installés, dans le cas d'un DC, ils sont installés automatiquement. En dehors de cet exercice, il est vivement conseillé d'utiliser un poste de travail qui n'est pas un contrôleur de domaine pour administrer ActiveDirectory. Depuis Powershell v3, l'import d'un module dans la session courante est automatique : Dans ce cas la commande implicite exécutée est **Import-Module ActiveDirectory** lors du premier lancement de la commande ActiveDirectory.

- Créer une unité d'organistation (OU) nommée **Annuaire** (avec Powershell)
- Créer un utilisateur lambda avec une adresse mail et une ville à l'aide de la commande **New-Aduser**, puis lui ajouter un mot de passe. (dans l'OU Annuaire créée précédemment)
- Créer un deuxième utilisateur avec mot de passe à l'aide d'une seule ligne de commande
- Lister tous les comptes dont le mot de passe n'a pas été changé depuis un mois. (dans le cadre de ce TP, une heure suffira étant donné la date de création des comptes)
- À l'aide la commande précédente, forcer le changement de passe des comptes concernés

### 3.1.4 Crédit d'utilisateurs depuis un fichier de type CSV

- À l'aide du fichier CSV fourni par le formateur, créez un grand nombre de comptes tel que :
- Login : prenom.nom (limité à 20 caractères) sans accent
- Mot de passe : nombre aléatoire de 8 caractères
  - Bonus : utilisation d'une fonction dédiée : [Generate-Password.ps1](#)
- Email : prenom.nom@votredomaine.xxx (sans limitation de caractère)
- Forcer le changement de mot de passe à la première ouverture de session
- Créer l'utilisateur dans l'OU précédemment créée

Fichier CSV : [Lien](#)

Fonction : [Remove-Diacritics](#)

### 3.1.5 Crédit d'utilisateurs Active Directory depuis une base de données MySQL

#### Prérequis base de données MariaDB (Mysql)

- Installer MariaDB sur votre machine :
- [mariadb-10.3.7-winx64.msi](#)
- Choisir **UTF8** comme jeu de caractère
- Dans une console CMD.exe, se connecter au serveur MariaDB :

```
mysql -u root -p
```

- Créer la base de données à l'aide de la commande suivante :

```
CREATE DATABASE RH;
```

- Créer la table rh\_users :

```
USE RH;
CREATE TABLE rh_users (surname VARCHAR(100), givenname VARCHAR(100), city VARCHAR(100));
```

Charger le fichier CSV dans la table rh\_users;

```
LOAD DATA LOCAL INFILE 'C:/TP/CSV/BaseDonneesExempleCsv.csv' INTO TABLE rh_users FIELDS TERMINATED BY ';' LINES TERMINATED BY '\n' IGNORE 1 LINES (surname, givenname, city);
```

### **Création des utilisateurs depuis Mysql**

- Supprimer tous les comptes Active Directory que vous avez créés précédemment.

A l'aide des scripts fournis, créer tous les comptes présents dans la BDD MySQL dans l'AD tel que :

- Login (samaccountname) : prenom.nom (limité à 20 caractères) sans accent (fonction Remove-Diacritics fournie)
- Login (UPN) : prenom.nom@votredomaine.xxx (sans limitation de caractère)
- Mot de passe : nombre aléatoire de 8 caractères
- Bonus : utilisation d'une fonction dédiée : [Generate-Password.ps1](#)
- Email : prenom.nom@votredomaine.xxx (sans limitation de caractère)
- Forcer le changement de mot de passe à la première ouverture de session
- Créer l'utilisateur dans l'OU précédemment créée
- Créer les dossiers personnels des utilisateurs avec attribution des droits NTFS inhérents dans le dossier partagé : \\srv-data\home\<username>, voir script [SetFolderPermission.ps1](#) pour les droits NTFS

Connecteur : .Net Mysl Fonction : [Get-Mysql](#) Script : [SetFolderPermission](#)

- Remplacer le serveur de fichier hébergeant les répertoires personnels par \\srv-data2 pour tous les utilisateurs créés préalablement.
- À l'aide de la fonction [Fill-PDF](#), du fichier [Formulaire.pdf](#) et de la DLL [itextsharp.dll](#) générer un export pour chaque utilisateur de l'OU **Annuaire**.
- Simuler la transmission par mail de chaque export PDF vers l'utilisateur concerné. (Générer la commande "send-mailmessage" sous forme d'une chaîne de caractère)

### **Création des utilisateurs via CSV et GUI**

- Supprimer tous les comptes Active Directory que vous avez créés précédemment.
- Télécharger le script suivant [GUI-AD.ps1](#)
- Lancer le script suivant GUI-AD.ps1 - En principe une fenêtre vide doit s'afficher.
- Décommenter la partie "Bouton de sélection d'un fichier CSV" et relancer le script. Observez l'ajout d'un bouton concernant l'import CSV.
- Décommenter la partie "Contenu du fichier CSV" et relancer le script. Cette fois une zone de contenu vide apparaît.
- Décommenter la partie "Bouton pour créer les utilisateurs" et relancer le script. Observez l'ajout d'un bouton concernant la création des utilisateurs.
- Décommenter la partie "Action pour sélectionner un fichier" et relancer le script. Sélectionner le fichier CSV 'BaseDonneesExempleCsv.csv' et vérifier le résultat dans la zone de contenu.
- Décommenter la partie "Action pour créer les utilisateurs" et relancer le script. Effectuez les opérations suivantes :
- Adapter cette partie pour afficher le nom, le prénom et la ville dans le shell lors d'un clic sur le bouton "Créer les utilisateurs".
- Reprendre le script de la partie CSV et l'ajouter dans la partie action. Dans un premier temps, afficher seulement la commande en mode texte dans le shell : "New-Aduser -samaccountname \$login -name \$name ..."

- Si les commandes New-Aduser semblent correctes, adapter cette partie du script pour créer les comptes.
- Adapter le script pour créer une OU depuis l'interface graphique.
- Adapter le script pour créer un groupe depuis l'interface graphique.
- Adapter le script pour vérifier si les comptes existent avant création.

⌚November 16, 2025 18:17:20

## 4. Session 4

---

### 4.1 Utilisation de WMI/CIM

#### 4.1.1 Prérequis

- 2 machines Windows virtuelles nommées Client1, Client2 (Windows 10 ou 11)
- Configurer les deux VMs Client 1 & 2 sur le switch virtuel de la VM DC1
- Configurer les IPs 192.168.10.21/24 et 192.168.10.22/24 sur les VMs client1 et client2 respectivement.
- Intégrer les VMs client1 et client2 au domaine CESI.LAN
- Sur le VMs client1 et client2, lancer la commande suivante dans une console Powershell : Enable-PSRemoting

#### 4.1.2 WMI/CIM (Windows Management Instrumentation)

- Depuis le DC, lancer une console powershell
  - Tester la connectivité WSMAN avec la commande suivante :

```
Test-WsMan client1 #Puis client2
#Si la connection WSMAN fonctionne, le lignes suivantes seront affichées
wsmid      : http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd
ProtocolVersion : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
ProductVendor   : Microsoft Corporation
ProductVersion  : OS: 0.0.0 SP: 0.0 Stack: 3.0
```

- Obtenir les information de BIOS

```
$CimSession = New-CimSession -ComputerName Client1 -Credential (Get-Credential)
Get-CimInstance -CimSession $CimSession -ClassName Win32_Bios
```

- Adapter la commande précédente pour :
  - interroger les deux VMs (Client1 et 2)
  - détecter le système d'exploitation
  - afficher, le nom de la machine, le nom du système d'exploitation, sa version et son architecture (x64 ou x86).
- À l'aide de la classe WMI **win32\_LogicalDisk**, afficher la taille et l'espace libre des partitions.
- Affiner la sortie de la commande précédente pour afficher en tableau les colonnes suivantes : Name, Espace Libre (en GB) et taille (en GB). Garder ce script pour la prochaine partie.

Conseil : utiliser la syntaxe : @{label = <string> ; expression = {<script>}}

### 4.2 Envoi de message et HTML

#### 4.2.1 Send-Mailmessage

- À l'aide de la commande **Send-MailMessage**, envoyez-vous un mail de test (adaptez les paramètres de la commande en fonction de votre fournisseur de messagerie : SSL ?, Port 25 ou 587...)
- À l'aide du script de la dernière partie (concernant Win32\_LogicalDisk), envoyer la liste des partitions de vos 2 VMs à votre adresse mail.
- Répétez l'opération précédente en convertissant le retour du script en HTML.

#### 4.2.2 Module MailKit

L'usage de la commande **Send-MailMessage** est déprécié, dans les futures versions de Powershell, celle-ci ne sera plus prise en charge. En 2024, il n'y pas de commande native de remplacement. Microsoft recommande l'usage du Mailkit <https://github.com/jstedfast/MailKit>

- Installer Mailkit sur le serveur DC1 à l'aide la commande `Install-Module -Name "Send-MailKitMessage" -Scope AllUsers`
- Modifier le script [Test-MailKit.ps1](#) pour envoyer un mail

 December 19, 2024 00:35:08

## 5. Corrections

---

### 5.1 TP Session 1 & 2

---

- [Exercices Powershell Corrigés V2.4.docx](#)

### 5.2 TP Session 3

---

- [CorrectionTP\\_AD\\_CSV.ps1](#)
- [CorrectionTP\\_AD\\_Mysql.ps1](#)
- [CorrectionTP\\_AD\\_CSV\\_GUI.ps1](#)
- [CorrectionTP\\_WMI\\_CIM\\_MAIL.ps1](#)

⌚ December 20, 2024 17:46:39