**ANNEXE pour PowerShell**

PowerShell offre la possibilité de personnaliser l'affichage des colonnes.

**Get-Help about\_Calculated\_Properties**

**Format-List**

* **name/label - optionnel**
* **expression**
* **formatstring - optionnel**

**Format-Table**

* **name/label - optionnel**
* **expression**
* **formatstring - optionnel**
* **width - optionnel**
* **alignment - optionnel**

**Select-Object**

* **name/label - optionnel**
* **expression**

**Sort-Object**

* **expression**
* **ascending/descending - optionnel**

**name/label**

* Spécifie le nom de la propriété en cours de création.
* Vous pouvez utiliser **name** ou son alias **label**.

**expression**

* Un bloc de script utilisé pour calculer la valeur de la nouvelle propriété.

**alignment**

* Utilisé par les cmdlets qui produisent une sortie tabulaire pour définir comment les valeurs sont affichées dans une colonne.
* La valeur doit être **left**, **center**, ou **right**.

**formatstring**

* Spécifie une chaîne de format qui définit comment la valeur est formatée pour la sortie.
* Pour plus d’informations sur les chaînes de format, voir Types de format dans .NET.

**width**

* Spécifie la largeur maximale de la colonne d’un tableau lorsque la valeur est affichée.
* La valeur doit être supérieure à 0.

**ascending / descending**

* Vous permet de spécifier l’ordre de tri pour une ou plusieurs propriétés.
* Ce sont des valeurs booléennes.

**Exemple 1 - Exemple avec des colonnes personnalisées avec Format-List**

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

**Format-List Name,DisplayName,UserPrincipalName,DistinguishedName**

**On affiche le nom de l'unité d'organisation des utilisateurs à partir de la propriété DistinguishedName.**

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

**Format-List Name,DisplayName,UserPrincipalName,**

**@{label="Unité d'organisation"**

**expression={($PSitem.DistinguishedName -split ',',2)[1]}**

**}**

**Exemple 2 - Exemple avec des colonnes personnalisées avec Format-List**

**$NOM\_VM = "routeur"**

**Get-VMNetworkAdapter -VMName $NOM\_VM `**

**| Format-List Name,SwitchName,MacAddress,IPAddresses**

**Utilisation du paramètre -ExpandProperty avec plusieurs propriétés.**

**$NOM\_VM = "routeur"**

**Get-VMNetworkAdapter -VMName $NOM\_VM `**

**| Format-List Name,SwitchName,MacAddress,**

**@{label="IP"**

**expression={($PSItem | Select-Object -ExpandProperty IPAddresses) -Join ',' }**

**}**

**Exemple 3 - Exemple avec des colonnes personnalisées avec Format-List**

**Par défaut la capacité des barrettes de mémoire est affichée en octets.**

**Get-CIMinstance -Class Win32\_PhysicalMemory `**

**| Format-List Caption,Capacity,**

**Speed,DeviceLocator,Manufacturer,SerialNumber**

**note: pas besoin d'utiliser le paramètre FormatString pour afficher la taille des barrettes de mémoire**

**Get-CIMinstance -Class Win32\_PhysicalMemory `**

**| Format-List Caption,**

**@{label='Size(GO)'**

**expression={($PSitem.Capacity / 1GB)}**

**},**

**Speed,DeviceLocator,Manufacturer,SerialNumber**

**Exemple 4 - Exemple avec des colonnes personnalisées avec Format-Table**

**Par défaut la taille des disques est affichée en octets.**

**$partition = 'C:'**

**Get-CIMinstance -Class Win32\_LogicalDisk `**

**-Filter "DeviceId='$partition'" `**

**| Format-Table SystemName,DeviceID,VolumeName,Size,FreeSpace -Autosize**

**note 1: on modifie le titre de plusieurs colonnes du tableau avec le paramètre Label**

**note 2: on utilise le paramètre FormatString='N2' pour afficher le résultat des calculs**

**$partition = 'C:'**

**Get-CIMinstance -Class Win32\_LogicalDisk `**

**-Filter "DeviceId='$partition'" `**

**| Format-Table SystemName, `**

**@{label='Partition'**

**expression={$PSitem.DeviceId}**

**}, `**

**@{label='Description'**

**expression={$PSitem.VolumeName}**

**}, `**

**@{label='Size(GO)'**

**expression={$PSitem.Size / 1GB}**

**formatstring='N2'**

**alignment='center'**

**}, `**

**@{label='FreeSpace(GO)'**

**expression={$PSitem.FreeSpace / 1GB}**

**formatstring='N2'**

**alignment='center'**

**} -Autosize**

**Exemple 5 - Exemple avec des colonnes personnalisées avec Format-Table**

Utilisation des paramètres **width** et **alignment** avec **Format-Table**.

**$chemin = "F:\\_VIRTUEL\DISQUE"**

**$nom = @{label = 'Nom'**

**expression= {$PSItem.Name}**

**width=100**

**alignment='left'**

**}**

**$taille = @{label='KB'**

**expression={ ($PSItem.Length / 1KB) }**

**width=20**

**alignment='right'**

**}**

**Get-ChildItem -Path $chemin | Format-Table $nom,$taille**

**Exemple 6 - Exemple avec des colonnes personnalisées avec Select-Object**

**Get-NetAdapter | Select-Object Name,MacAddress**

**On modifie l'affichage de l'adresse MAC.**

**Get-NetAdapter | Select-Object Name,**

**@{label='MacAddress'**

**expression={** **$PSitem.MacAddress -replace '-',':' }**

**}**

**Exemple 7 - Exemple avec des colonnes personnalisées avec Sort-Object**

Pour utiliser **Sort-Object** avec plusieurs paramètres, il faut utiliser des "hash table" pour trier par ordre croissant, décroissant, ou une combinaison d’ordres de tri.

**Get-Service | Sort-Object -Property @{expression = "Status"**

**descending = $true**

**},**

**@{expression = "DisplayName"**

**ascending = $true**

**}**