

Systemes clients Microsoft

Module 11 – Plus loin avec PowerShell



1

1

Plus loin avec PowerShell

Objectifs

- Savoir rediriger les flux des cmdlets
- Utiliser le pipeline
- Filtrer puis formater le résultat



2

2

Plus loin avec PowerShell

Plus loin avec PowerShell

- Redirection de flux
- Le pipeline
- Filtrer l'affichage d'un résultat



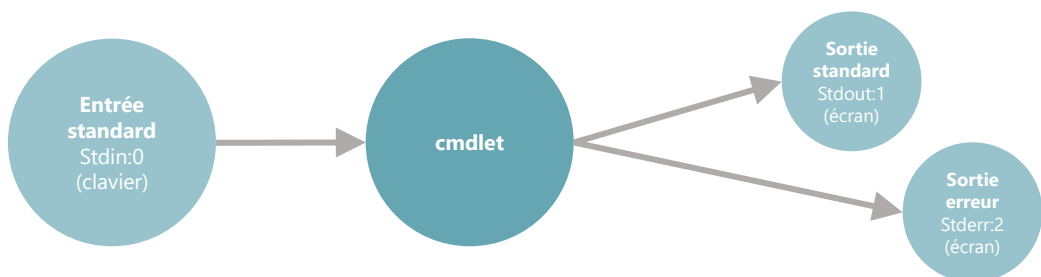
3

3

Plus loin avec PowerShell

Redirection

- Une cmdlet :
 - Reçoit des paramètres par l'entrée standard (stdin:0) par défaut le clavier
 - Fournit des résultats sur la sortie standard (stdout:1) par défaut l'écran
 - Affiche éventuellement les erreurs sur la sortie standard (stderr:2) par défaut l'écran



4

4

Redirection

Redirection du flux de sortie standard	
Cmdlet > fichier Cmdlet 1>fichier	Redirige le flux de sortie standard vers 'fichier'
Cmdlet >> fichier Cmdlet 1>> fichier	Redirige le flux de sortie standard à la fin de 'fichier'
Redirection du flux de sortie d'erreur	
Cmdlet 2> fichier.err	Redirige le flux de sortie d'erreur vers fichier.err
Cmdlet 2>> fichier.err	Redirige le flux de sortie d'erreur à la fin de fichier.err
Redirection des flux	
Cmdlet > fichier 2>&1	Redirige le flux de sortie standard et le flux de sortie d'erreur dans 'fichier'
Cmdlet 2>\$null	Redirige le flux de sortie d'erreur "dans le vide"

Redirection

- Exemple de redirection dans un fichier
 - `Get-LocalUser > c:\temp\LocalUserList.txt`

- Lecture du fichier
 - `Get-Content c:\temp\LocalUserList.txt`

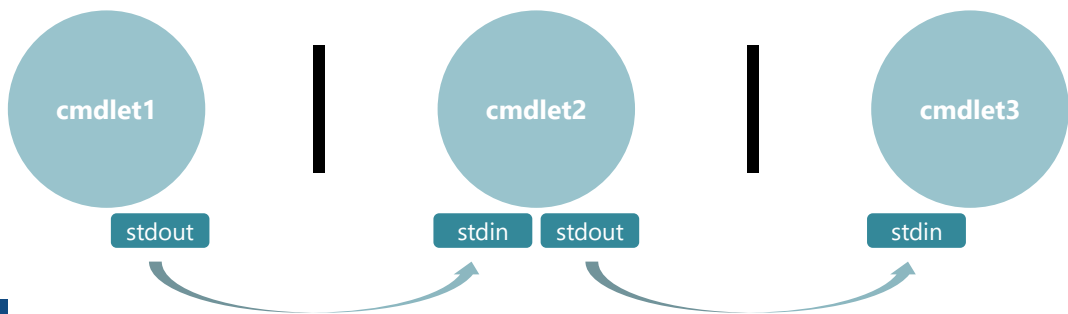
Name	Enabled	Description
-----	-----	-----
Administrateur	False	Compte d'utilisateur d'administration
Alfred	True	
DefaultAccount	False	Compte utilisateur géré par le système.
Georges	True	
Invité	False	Compte d'utilisateur invité
Martin	True	
romain	True	
Ron	True	
Stanley	True	
WDAGUtilityAccount	False	Compte d'utilisateur géré et utilisé par

Démonstration



Pipeline

- Déjà utilisé avec **Select-Object**, le *pipeline* transmet le résultat de la sortie standard d'une cmdlet dans l'entrée standard d'une autre cmdlet
- Enchaînement de cmdlet possible



Plus loin avec PowerShell

Pipeline

- Utilisé pour fournir un jeu restreint de propriétés
 - exemple avec **Get-NetAdapter** (sans paramètre ni argument) qui affiche certaines propriétés

Name	InterfaceDescription	ifIndex	Status	MacAddress	LinkSpeed
-----	-----	-----	-----	-----	-----
VMware Network Adapte...	VMware Virtual Ethernet Adapter for ...	23	Up	00-50-56-C0-00-01	100 Mbps
Wi-Fi	Intel(R) Dual Band Wireless-AC 7260	21	Disconnected	80-19-34-B1-CB-05	0 bps
VMware Network Adapte...	VMware Virtual Ethernet Adapter for ...	15	Up	00-50-56-C0-00-07	100 Mbps
Ethernet 2	Fortinet Virtual Ethernet Adapter (N...	14	Disconnected	00-09-0F-FE-00-01	100 Mbps
VMware Network Adapte...	VMware Virtual Ethernet Adapter for ...	10	Up	00-50-56-C0-00-08	100 Mbps
Ethernet 3	Remote NDIS based Internet Sharing D...	9	Up	02-06-63-35-30-64	426.0 Mbps
Connexion au réseau local	PPPoE WAN Adapter	5	Disconnected	00-00-00-00-00-00	0 bps
Ethernet	Intel(R) Ethernet Connection I218-LM	3	Disconnected	34-E6-D7-09-FE-03	0 bps

- **Get-NetAdapter** | **Select Name,Status,LinkSpeed**

Name	Status	LinkSpeed
-----	-----	-----
VMware Network Adapter VMnet1	Up	100 Mbps
Wi-Fi	Disconnected	0 bps
VMware Network Adapter VMnet7	Up	100 Mbps
Ethernet 2	Disconnected	100 Mbps
VMware Network Adapter VMnet8	Up	100 Mbps
Ethernet 3	Up	426.0 Mbps
Connexion au réseau local	Disconnected	0 bps
Ethernet	Disconnected	0 bps



9

9

Plus loin avec PowerShell

Démonstration



10

10

Filtrage

- Les objets fournis par le pipeline sont parfois nombreux
- Mettre en place le **filtrage** pour ne garder que les propriétés nécessaires
- Prérequis : connaître les **opérateurs de comparaisons**
 - Par défaut insensible à la casse
 - Les préfixer de "c" pour les rendre sensibles...
 - ... **Get-Help About_Comparison_Operator** ...



Filtrage

Comparaison	Insensibilité à la casse	Sensibilité à la casse
Egalité	-eq	-ceq
Inégalité	-ne	-cne
Supérieur à	-gt	-cgt
Supérieur ou égal à	-ge	-cge
Inférieur à	-lt	-clt
Inférieur ou égal à	-le	-cle
Comparaison d'égalité d'expression	-like	-clike
Comparaison d'inégalité d'expression	-notlike	-cnotlike



Plus loin avec PowerShell

Filtrage basique

- Ne peut filtrer qu'une seule propriété
- Nécessite la cmdlet **Where-Object**
- Exemple **Get-NetAdapter** | **select Name,Status,LinkSpeed**

Name	Status	LinkSpeed
-----	-----	-----
VMware Network Adapter VMnet1	Up	100 Mbps
Wi-Fi	Disconnected	0 bps
VMware Network Adapter VMnet7	Up	100 Mbps
Ethernet 2	Disconnected	100 Mbps
VMware Network Adapter VMnet8	Up	100 Mbps
Ethernet 3	Up	426.0 Mbps
Connexion au réseau local	Disconnected	0 bps
Ethernet	Disconnected	0 bps

- **Get-Netadapter** | **Select Name,Status,LinkSpeed** | **Where Status -like Up**

Name	Status	LinkSpeed
-----	-----	-----
VMware Network Adapter VMnet1	Up	100 Mbps
VMware Network Adapter VMnet7	Up	100 Mbps
VMware Network Adapter VMnet8	Up	100 Mbps
Ethernet 3	Up	426.0 Mbps



13

13

Plus loin avec PowerShell

Démonstration



14

14

Filtrage

Le filtrage avancé

- Pour filtrer plusieurs propriétés
- Nécessite aussi la cmdlet **Where-Object**
- Nécessite un script de filtrage **-FilterScript {}** dans la cmdlet **Where-Object**
- Possibilité d'utiliser la variable **\$PSITEM** (ou **\$_**) pour être plus productif
- **\$_** contient tous les objets transmis à **Where-Object**



Filtrage

- Exemple **Get-NetAdapter** | **select Name,Status,LinkSpeed**

Name	Status	LinkSpeed
-----	-----	-----
VMware Network Adapter VMnet1	Up	100 Mbps
Wi-Fi	Disconnected	0 bps
VMware Network Adapter VMnet7	Up	100 Mbps
Ethernet 2	Disconnected	100 Mbps
VMware Network Adapter VMnet8	Up	100 Mbps
Ethernet 3	Up	426.0 Mbps
Connexion au réseau local	Disconnected	0 bps
Ethernet	Disconnected	0 bps

- **Get-NetAdapter** | **select Name,Status,LinkSpeed** | **Where -FilterScript {\$_ .Status -like "Up" -and \$_.LinkSpeed -gt "100 Mbps"}**

Name	Status	LinkSpeed
----	-----	-----
Ethernet 3	Up	426.0 Mbps



Démonstration



17

17

Formatage

- Par défaut, PowerShell "formate" les résultats des cmdlets avant de les afficher à l'écran
- Le format des résultats peut être modifié à l'aide des cmdlets
 - **Format-List**
 - **Format-Table**
 - **Format-Wide**
- Les cmdlets **Format-*** sont à réaliser en dernier lieu, "après avoir terminé le travail"



18

18

Formatage

- Affichage sous forme de liste
 - `Get-Service` | `format-List`
 - `Get-Service` | `format-List -Property *`
 - `Get-Service` | `FL -Property Name,Status,DisplayName`
 - Équivalent à `Get-Service` | `select Name,Status,DisplayName` | `FL`
- Affichage sous forme de tableau
 - `Get-TimeZone` | `Format-Table`
 - `Get-TimeZone` | `Format-Table -Property *`
 - `Get-TimeZone` | `FT -Property StandardName,BaseUtcOffset -autosize`



Formatage

- Affichage sous forme de liste élargie
 - Afficher une liste sur une ou plusieurs colonnes
 - Une seule propriété de la collection d'objets est sélectionnée
- `Get-Process` | `Format-Wide`
- `Get-Process` | `Format-Wide -Property ID`
- `Get-Process` | `FW -Property ID -column 5`
- `Get-Process` | `FW -Property ID -autosize`



Plus loin avec PowerShell

Démonstration



21

21

Plus loin avec PowerShell

TP



22

22

Plus loin avec PowerShell

Conclusion

- Maîtriser PowerShell, l'essence même d'un technicien moderne
- Aller plus loin, être curieux
- S'éveiller au scripting PowerShell

