



PowerShell

NOTES DE COURS

Filtrer des objets

Automne 2019

Table des matières

1	Filtrer des objets	3
1.1	Opérateurs de comparaison	3
1.2	Opérateurs binaires	13
1.3	Paramètres.....	14
1.4	Syntaxe de base.....	15
1.5	Syntaxe avancée.....	16

NOTE IMPORTANTE

Les exemples de ce module proviennent de la documentation de PowerShell.

1 Filtrer des objets

La commande **Where-Object** permet de filtrer des objets selon les valeurs des propriétés.

1.1 Opérateurs de comparaison

PowerShell supporte les opérateurs de comparaison suivants :

Comparaison	Non sensible à la casse		Sensible à la casse
Égal	-eq	-ieq	-ceq
Non égal	-ne	-ine	-cne
Plus grand	-gt	-igt	-cgt
Plus petit	-lt	-ilt	-clt
Plus grand ou égal	-ge	-ige	-cge
Plus petit ou égal	-le	-ile	-cle
Correspondance avec le caractère générique *	-Like	-iLike	-cLike
	-NotLike	-iNotLike	-cNotLike
Correspondance avec une expression régulière	-Match	-iMatch	-cMatch
	-NotMatch	-iNotMatch	-cNotMatch
Collection contenant un élément	-Contains	-iContains	-cContains
	-NotContains	-iNotContains	-cNotContains
Élément fait partie d'une collection	-In	-iIn	-cIn
	-NotIn	-iNotIn	-cNotIn
Remplacement	-Replace	-iReplace	-cReplace

Par défaut, tous les opérateurs ne sont pas sensibles à la casse. Pour changer ce défaut il suffit de préfixer l'opérateur par le caractère *c*.

Il est possible aussi de préfixer l'opérateur par le caractère *i* pour le rendre insensible à la casse mais ceci est le comportement défaut.

Lorsque la valeur à gauche de l'opérateur est une valeur scalaire, l'opérateur de comparaison retourne une booléenne *True* ou *False*.

Lorsque la valeur à gauche de l'opérateur est une collection de valeurs, l'opérateur de comparaison retourne les valeurs qui vérifient l'opérateur. Si aucune valeur de la collection ne vérifie l'opérateur alors l'opérateur ne retourne rien.

Les exceptions sont les opérateurs de contenance (-Contains, -NotContains), les opérateurs In (-In, -NotIn), et les opérateurs de type (-Is, -IsNot), qui retournent toujours une valeur booléenne *True* ou *False*.

Opérateur eq (Equal)

```
PS C:\> "abc" -eq "abc"  
True
```

```
PS C:\> "abc" -eq "abc", "def"  
False
```

```
PS C:\> "abc", "def" -eq "abc"  
abc
```

Opérateur ne (Not equal)

```
PS C:\> "abc" -ne "def"  
True
```

```
PS C:\> "abc" -ne "abc"  
False
```

```
PS C:\> "abc" -ne "abc", "def"  
True
```

```
PS C:\> "abc", "def" -ne "abc"  
Def
```

Opérateur gt (Greater-than)

```
PS C:\> 8 -gt 6  
True
```

```
PS C:\> 7, 8, 9 -gt 8  
9
```

Opérateur ge (Greater-than or equal)

```
PS C:\> 8 -ge 8  
True
```

```
PS C:\> 7, 8, 9 -ge 8  
8  
9
```

Opérateur lt (Less-than)

```
PS C:\> 8 -lt 6  
False
```

```
PS C:\> 7, 8, 9 -lt 8  
7
```

Opérateur le (Less-than or equal to)

```
PS C:\> 6 -le 8  
True
```

```
PS C:\> 7, 8, 9 -le 8  
7  
8
```

Opérateur Like

```
PS C:\> "PowerShell" -like "*shell"  
True
```

```
PS C:\> "windows PowerShell", "Server" -like "*shell"  
windows PowerShell
```

Opérateur NotLike

```
PS C:\> "windows PowerShell" -NotLike "*shell"  
False
```

```
PS C:\> "windows PowerShell", "Server" -NotLike  
"*shell"  
Server
```

Opérateur Match

Lorsque la valeur à gauche de l'opérateur est un scalaire, le résultat sera contenu dans la variable **\$Matches**.

```
PS C:\> "Sunday" -Match "sun"
True
```

```
PS C:\> $matches
Name Value
----
0     Sun
```

```
PS C:\> "Sunday", "Monday" -Match "sun"
Sunday
```

```
PS C:\> "windows", "PowerShell" -Match ".shell"
PowerShell
```

```
PS C:\> (Get-Command Get-Member -Syntax) -Match "-view"
True
```

```
PS C:\> (Get-Command Get-Member -Syntax)
-NotMatch "-path"
True
```

```
PS C:\> (Get-Content Servers.txt) -Match "^Server\d\d"
Server01
Server02
PS C:\> "Sunday", "Monday", "Tuesday" -Match "sun"
Sunday
```

```
PS C:\> $matches
PS C:\>
```

```
PS C:\> "Sunday" -Match "sun"
True
```

```
PS C:\> $matches
Name value
----
0     Sun
```

Opérateur NotMatch

```
PS C:\> "Sunday" -NotMatch "sun"  
False
```

```
PS C:\> $matches  
Name Value  
----  
0     sun
```

```
PS C:\> "Sunday", "Monday" -NotMatch "sun"  
Monday
```

```
PS C:\> "Sunday" -NotMatch "rain"  
True
```

```
PS C:\> $matches  
PS C:\>  
PS C:\> "Sunday" -NotMatch "day"  
False
```

```
PS C:\> $matches  
PS C:\>  
Name Value  
----  
0     day
```

Opérateur Contains

Déterminer si une collection de valeurs inclut une valeur spécifique.

```
PS C:\> "abc", "def" -Contains "def"
True
```

```
PS C:\> "windows", "PowerShell" -Contains "Shell"
False
#Not an exact match
```

```
# Does the list of computers in $domainServers
# include $thisComputer?
# -----
PS C:\> $domainServers -Contains $thisComputer
True
```

```
PS C:\> "abc", "def", "ghi" -Contains "abc", "def"
False
```

```
PS C:\> $a = "abc", "def"
PS C:\> "abc", "def", "ghi" -Contains $a
False
```

```
PS C:\> $a, "ghi" -Contains $a
True
```

```
C:PS> 1,2,3 -contains 2
True
```

```
C:PS> "PowerShell" -contains "Shell"
False
```

```
C:PS> "Windows", "PowerShell" -contains "Shell"
False
```

```
PS C:\> "abc", "def", "123" -contains "def"
True
```

```
PS C:\> "true", "blue", "six" -contains "true"
True
```

L'opérateur *Contains* est similaire à l'opérateur d'égalité. Cependant, l'opérateur *Contains* retourne toujours une valeur booléenne même avec les collections.

Une autre différence est que l'opérateur *Contains* retourne une valeur à la première correspondance contrairement à l'opérateur d'égalité qui retourne toutes les correspondances.

```
PS C:\> 1,2,3,4,5,4,3,2,1 -eq 2  
2  
2
```

```
PS C:\> 1,2,3,4,5,4,3,2,1 -contains 2  
True
```

-NotContains

```
PS C:\> "windows", "PowerShell" -NotContains "Shell"  
True
```

Opérateur In

Déterminer si une valeur apparaît dans une collection de valeur.

```
PS C:\> "def" -in "abc", "def"
True
```

```
PS C:\> "shell" -in "windows", "PowerShell"
False #Not an exact match
```

```
PS C:\> "windows" -in "windows", "PowerShell"
True
#An exact match
```

```
PS C:\> "windows", "PowerShell" -in
"windows", "PowerShell", "ServerManager"
False
```

```
#Using reference equality
PS C:\> $a = "windows", "PowerShell"
PS C:\> $a -in $a, "ServerManager"
True
```

```
#Using reference equality
# Does the list of computers in $domainServers
# include $thisComputer?
# -----
```

```
PS C:\> $thisComputer -in $domainServers
True
```

Opérateur NotIn

```
PS C:\> "def" -NotIn "abc", "def"  
False
```

```
PS C:\> "ghi" -NotIn "abc", "def"  
True
```

```
PS C:\> "Shell" -NotIn "Windows", "PowerShell"  
True
```

```
#Not an exact match  
PS C:\> "Windows" -NotIn "Windows", "PowerShell"  
False
```

```
#An exact match  
# Find unapproved verbs in the functions in my module  
# -----  
PS C:\> $ApprovedVerbs = Get-Verb | foreach {$_.verb}
```

```
PS C:\> $myVerbs = Get-Command -Module MyModule |  
foreach {$_.verb}
```

```
PS C:\> $myVerbs | where {$_ -NotIn $ApprovedVerbs}  
ForEach  
Sort  
Tee  
Where
```

Opérateur Replace

Cet opérateur permet de modifier les valeurs des éléments.

```
PS C:\> "Get-Process" -Replace "Get", "Stop"  
Stop-Process
```

```
# Change all .GIF file name extension to .JPG  
PS C:\> dir *.gif | foreach {$_ -Replace ".gif",  
".jpg"}
```

```
Get-ChildItem |  
Rename-Item -NewName { $_ -Replace '.gif$', '.jpg$' }  
PS C:\> "book" -Replace "B", "C"  
Cook
```

```
PS C:\> "book" -iReplace "B", "C"  
Cook
```

```
PS C:\> "book" -cReplace "B", "C"  
book
```

```
PS C:\> '<command:parameter required="false"  
variableLength="true" globbing="false"'`  
| foreach {$_ -replace 'globbing="false"',  
'globbing="true"'}`  
<command:parameter required="false"  
variableLength="true" globbing="true"
```

1.2 Opérateurs binaires

Windows PowerShell support les opérateurs binaires suivants:

Opérateur	Description	Exemple
-bAnd	AND	PS C:\> 10 -band 3 2
-bOr	OR (inclusive)	PS C:\> 10 -bor 3 11
-bXor	OR (exclusive)	PS C:\> 10 -bxor 3 9
-bNot	NOT	PS C:\> -bNot 10 11
-shl	Shift-left	PS C:\> 100 -shl 2 400
-shr	Shift-right	PS C:\> 100 -shr 1 50

1010	(10)	
0011	(3)	
-----		bAND
0010	(2)	

1010	(10)	
0011	(3)	
-----		bOR (inclusive)
1011	(11)	

1010	(10)	
0011	(3)	
-----		bXOR (exclusive)
1001	(9)	

1.3 Paramètres

Paramètre **Property**

Ce paramètre permet de spécifier la propriété sur laquelle le filtre d'applique.

Paramètre **Value**

Ce paramètre permet de spécifier la valeur de la propriété.

```
Get-Process | where-Object -Property PriorityClass -eq  
-Value "Normal"
```

Cette commande est équivalente à la commande suivante :

```
Get-Process | where-Object PriorityClass -eq "Normal"
```

Paramètre **Contains**

Ce paramètre permet de spécifier l'opérateur **Contains**. Avec ce paramètre, la correspondance (match) doit être exacte.

```
Get-Process | where ProcessName -contains "svchost"
```

Paramètre **Like**

```
Get-Process | where ProcessName -like "*host"
```

1.4 Syntaxe de base

- Supporte une seule condition.
- On doit référencer directement une propriété d'un objet.

```
Get-Service | where Name.Length -gt 8
```

```
PS C:\> Get-Service|where-Object status -eq stopped
```

Status	Name	DisplayName
-----	----	-----
Stopped	AeLookupSvc	Expérience d'application
Stopped	ALG	Service de la passerelle
Stopped	AppIDSvc	Identité de l'application

1.5 Syntaxe avancée

- Supporte plusieurs conditions.
- Pas de restriction sur les expressions utilisées.

Nécessite l'utilisation d'un script de filtre qui contient les critères du filtre. Le script de filtre retourne une valeur booléenne True ou False.

A l'intérieur d'un script de filtre, on utilise **\$PSItem** ou **\$_** pour référencer l'objet courant du pipe.

```
PS C:\> Get-Service|where-Object
{$_ .name -Like "r*" -and $_.status -eq "stopped" }
```

Status	Name	DisplayName
Stopped	RasAuto	Gestionnaire des connexions
Stopped	RasMan	Gestionnaire des connexions
Stopped	RemoteAccess	Routage et accès distant
Stopped	RemoteRegistry	Registre à distance
Stopped	RpcLocator	Localisateur d'appels de

```
Get-Service |
where-Object -Filter {$PSItem.Status -eq 'Running' }
Get-Service | where { $_.Status -eq 'Running' }
Get-Service | ? { $PSItem.Status -eq 'Running' }
```

Critères multiples

```
Get-Volume | where-Object -Filter {
    $PSItem.HealthStatus -ne 'Healthy'
    -or
    $PSItem.SizeRemaining -lt 100MB
}
```