



PowerShell

Operációs rendszerek I

2012

Windisch Gergely

PowerShell

- Automatizmust elősegítő parancsok gyűjteménye
- Parancssorhoz hasonló, de annál hatékonyabb
- Alkalmazások, Windows komponensek vezérlése parancssorból, kívülről
- Minden objektum

Powershell verziók

- PowerShell v1.0: 2006
 - 129 parancs
 - Windows 2008 kiegészítés
 - Letölthető Windows 2003 és XP verzió
- PowerShell v2.0: 2009
 - Windows 7 és 2008 R2 alapból tartalmazza
 - Letölthető Windows 2003 és XP verzió
 - <http://support.microsoft.com/kb/968930>
- PowerShell v3.0: 2012

Használata

- Start / kellékek / WinPowerShell
- Parancssorral kompatibilis - látszólag
 - Ez is átszinezhető
 - cd, dir, mkdir, copy - minden parancs elérhető
 - de csak hasonló parancs!
 - mkdir proba
 - echo "hello" > proba\mintafajl.txt
 - del proba - könyvtár törlésére is - suspend új
- Nem case sensitive - mkdir == MKdir
 - get-command == gET-COMMand

Miért más, mint mások?

- Objektumokon dolgozik, szöveg helyett
 - A parancsok eredménye egymás után fűzhető programozás nélkül
 - Hagyományosan (pl: linux - bash)
 - `kill `ps ax | grep java | cut -f1``
 - A `ps ax` lekérdezi a futó folyamatokat. A `grep` kikeresi a `java` nevűt, a `cut` feldarabolja a kimenetet tabonként, ahol az első érték a process ID, ami kell a kilövéshez.
 - A ``-en belüli rész fut le először, visszaad egy azonosító számot, és lefut a `kill <PID>` parancs (ahol PID helyére bekerül a java folyamat azonosítója)
 - Powershellben ugyanez
 - `get-process java | stop-process`
 - `ps java | kill`

ps kimenete (linux)

- PID TTY STAT TIME COMMAND
- 1 ? Ss 0:00 /sbin/init
- 1611 ? Ss 0:00 /usr/sbin/sshd
- 1930 ? S 0:10 metacity
- 1932 ? S 0:26 gnome-power-manager
- 1933 ? S 2:21 /usr/libexec/gdm-simple-greeter
- 17867 ? SN 0:00 /usr/bin/python /usr/bin/denyhosts.py --daemon --config=/etc/denyhosts.conf
- 18093 ? Ss 0:00 sshd: winger [priv]
- 18096 ? S 0:00 sshd: winger@pts/0
- 18097 pts/0 Ss 0:00 -bash
- 18143 pts/0 S+ 0:00 /usr/bin/mc -P /tmp/mc-winger/mc.pwd.18097
- 18145 pts/1 Ss 0:00 bash -rcfile .bashrc
- **18171 pts/1 R+ 0:00 java**
- 24960 ? Ss 0:13 /usr/sbin/httpd

PowerShell parancs típusok

- cmdlet (command-let)
 - Powershell parancs
 - pl: Get-ChildItem - könyvtár listázása
- alias
 - cmdlet-ek ismerős nevei
 - pl: Get-ChildItem cmdlet-re mutató alias: ls, dir, gci
- natív parancs
 - "normál" windows parancsok, programok
 - pl: notepad, ping, ipconfig
- script
 - Powershell scriptek

cmdlet

- Ige-Főnév formátum (verb-noun ; csináld-mit)

Verb	Noun
Add	Object
Get	QADUser
Import	VM
Export	Alias
New	Content
Remove	QADGroup


Windows PowerShell (Cmdlet)	Windows PowerShell (Alias)	cmd.exe / COMMAND.COM (MS-DOS, Windows, OS/2, etc.)	Bash (BSD, Linux, X etc.)	Description
Get-ChildItem	gci, dir, ls	dir	ls	List all files / directories in the (current) directory
Get-Content	gc, type, cat	type	cat	Get the content of a file
Get-Command	gcm	help	which	List available commands
Get-Help	help, man	help	man	Help on commands
Clear-Host	cls, clear	cls	clear	Clear the screen ^[Note 1]
Copy-Item	cpi, copy, cp	copy	cp	Copy one or several files / a whole directory tree
Move-Item	mi, move, mv	move	mv	Move a file / a directory to a new location
Remove-Item	ri, del, erase, rmdir, rd, rm	del , erase , rmdir , rd	rm , rmdir	Delete a file / a directory
Rename-Item	mi, ren, mv	ren , rename	mv	Rename a file / a directory
Get-Location	gl, pwd	cd	pwd	Display the current directory/present working directory.
Pop-Location	popd	popd	popd	Change the current directory to the directory most recently pushed onto the stack
Push-Location	pushd	pushd	pushd	Push the current directory onto the stack
Set-Location	sl, cd, chdir	cd , chdir	cd	Change the current directory
Tee-Object	tee	n/a	tee	Pipe input to a file or variable, then pass the input along the pipeline
Write-Output	echo, write	echo	echo	Print strings, variables etc. to standard output
Get-Process	gps, ps	tlist , ^[Note 2] tasklist ^[Note 3]	ps	List all currently running processes
Stop-Process	spps, kill	kill , ^[Note 2] taskkill ^[Note 3]	kill	Stop a running process
Select-String		find , findstr	grep	Print lines matching a pattern
Set-Variable	sv, set	set	set	Set the value of a variable / create a variable

Példa parancsok I

- `Dir C:\windows`
- `Alias Dir`
- `$a = dir C:\windows`
- `$a.count`
- `$a | Select Name`
- `$a | Select Name, Extension`
- `$a | group extension`
- `Get-ChildItem PS:\powershell`
- `$DateToCompare = (Get-date).AddDays(-2)`
- `Get-Childitem PS:\windows-recurse | where-object {$_.lastwritetime -gt $DateToCompare}`

Powershell

- Get-Command
- Get-Help *
- Get-Help about_*
- get-help get-command -detailed
- get-help get-command -example
- get-command -commandType function
- get-command write-*
- get-command *-object
- get-command more
- get-command more.com | get-member
- get-command more.com | foreach {\$_.FileVersionInfo}
- get-qaduser -logonname JDoe | get-member
- get-qaduser -logonname JDoe | get-member -MemberType Property
- get-qaduser -logonname JDoe | select DisplayName, PhoneNumber
- Get-PSDrive
- Get-ChildItem HKLM:
- Get-ChildItem Function:
- Get-ChildItem cert:

- 
- `get-command`
 - `get-help`
 - `get-member`
 - `get-psdrive`


Objektumok

- Get-Service -name fax
- Get-Service | Get-Member
- Get-Service | Get-Member -MemberType Property
- Get-Service | Get-Member -MemberType Method

```
Windows PowerShell
PS C:\MyScripts> Get-Service | Get-Member

TypeName: System.ServiceProcess.ServiceController

Name                MemberType Definition
-----
Name                AliasProperty Name = ServiceName
add_Disposed        Method      System.Void add_Disposed(EventH...
Close               Method      System.Void Close()
Continue            Method      System.Void Continue()
CreateObjRef         Method      System.Runtime.Remoting.ObjRef ...
Dispose             Method      System.Void Dispose()
Equals              Method      System.Boolean Equals(Object obj)
ExecuteCommand       Method      System.Void ExecuteCommand(Int32...
GetHashCode          Method      System.Int32 GetHashCode()
GetLifetimeService   Method      System.Object GetLifetimeService()
GetType             Method      System.Type GetType()
get_CanPauseAndContinue Method      System.Boolean get_CanPauseAndContinue()
get_CanShutdown      Method      System.Boolean get_CanShutdown()
get_CanStop          Method      System.Boolean get_CanStop()
get_Container         Method      System.ComponentModel.IContainer...
get_DependentServices Method      System.ServiceProcess.ServiceCo...
get_DisplayName       Method      System.String get_DisplayName()
get_MachineName       Method      System.String get_MachineName()
get_ServiceHandle     Method      System.Runtime.InteropServices....
get_ServiceName       Method      System.String get_ServiceName()
get_ServicesDependedOn Method      System.ServiceProcess.ServiceCo...
get_ServiceType       Method      System.ServiceProcess.ServiceTy...
get_Site              Method      System.ComponentModel.ISite get...
get_Status           Method      System.ServiceProcess.ServiceCo...
InitializeLifetimeService Method      System.Object InitializeLifetime...
Pause               Method      System.Void Pause()
Refresh             Method      System.Void Refresh()
remove_Disposed      Method      System.Void remove_Disposed(Eve...
set_DisplayName       Method      System.Void set_DisplayName(Str...
set_MachineName       Method      System.Void set_MachineName(Str...
set_ServiceName       Method      System.Void set_ServiceName(Str...
set_Site             Method      System.Void set_Site(ISite value)
Start               Method      System.Void Start(), System.Voi...
Stop               Method      System.Void Stop()
ToString            Method      System.String ToString()
WaitForStatus       Method      System.Void WaitForStatus(Servi...
CanPauseAndContinue Property     System.Boolean CanPauseAndConti...
CanShutdown         Property     System.Boolean CanShutdown {get;}
CanStop             Property     System.Boolean CanStop {get;}
Container            Property     System.ComponentModel.IContainer...
DependentServices   Property     System.ServiceProcess.ServiceCo...
DisplayName          Property     System.String DisplayName {get;}
MachineName         Property     System.String MachineName {get;}
ServiceHandle        Property     System.Runtime.InteropServices....
ServiceName         Property     System.String ServiceName {get;}
ServicesDependedOn   Property     System.ServiceProcess.ServiceCo...
ServiceType         Property     System.ServiceProcess.ServiceTy...
Site                Property     System.ComponentModel.ISite Sif...
```

- 
- Start-service -name fax
 - Get-ChildItem | Get-Member
 - Get-ChildItem -Path C:\ -Recurse |
Where-Object {\$_.LastWriteTime -gt
"08/25/2007"}<enter>

Kimenet formázása

- Get-Command Format-*
- Get-ChildItem C:\Windows | Format-Table
- Get-ChildItem C:\Windows | Format-Table -AutoSize
- Get-ChildItem C:\Windows | Format-List
- Get-ChildItem C:\Windows -Recurse | Format-List -Property
FullName,CreationTime,LastWriteTime
- Get-ChildItem C: | Format-Wide -Column 3

Kimenet átirányítás

- Get-Process | ConvertTo-html
- out-file cmdlet - hasonló a > parancshoz
 - Get-Process | ConvertTo-html | out-file
“Processes.html”
- Megnyitás:
 - Invoke-Item Processes.html
- Get-Process | Export-CSV Processes.csv

Változók

\$ a változók neve előtt

- \$a = Get-Content c:\file.txt
- \$users = Get-QADUser -SizeLimit 0
- \$VMs = Get-VM

\$_ az aktuális objektumot jelenti

```
get-process | where-object { $_.WS -gt  
1000MB } | stop-process
```

Elágazás

- `$x = 2` #creates a variable x and assigns 2 as the value
- `if ($x -eq 5) {Write-Host "Hello my name is Bob"}`
- `elseif ($x -eq 4) {Write-Host "Hello, my name is Sue"}`
- `elseif ($x -eq 2) {Write-Host "Hello, my name is Troy"}`
- `elseif ($x -gt 1) {Write-Host "Hello, my name is Mary"}`
- `else {"I have no idea what my name is?"}`

Iterációk

- do while

```
do
{
    Write-Host $i
    $i++
} while ($i -le 5)
```

- while

```
while ($i -le 5)
{
    Write-Host $i
    $i++
}
```

- foreach

```
$ints = @(1, 2, 3, 4, 5)
```

```
foreach ($i in $ints)
{Write-Host $i}
```

for

```
for ($i=1; $i -le 5; $i++)
{
    Write-Host $i
}
```

until

```
do
{
    Write-Host $i
    $i++
} until ($i -gt 5)
```

foreach

- Nincs feltételvizsgálat, gyűjtemény elemeit járja be - pl: processzorok listázása

```
$strComputer = "."
```

```
$collItems = get-wmiobject -class "Win32_Processor" -namespace  
"root\CIMV2" -computername $strComputer
```

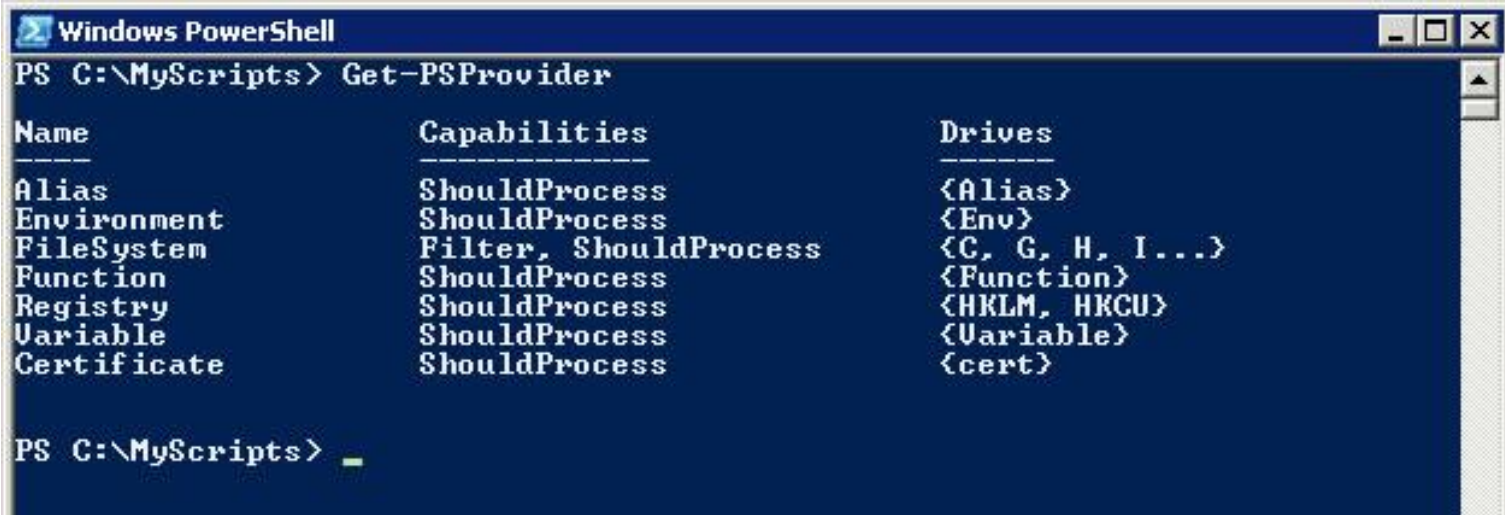
```
foreach ($objItem in $collItems) {  
write-host "Caption: " $objItem.Caption  
write-host "CPU Status: " $objItem.CpuStatus  
write-host "Current Clock Speed: " $objItem.CurrentClockSpeed  
write-host "Device ID: " $objItem.DeviceID  
write-host "L2 Cache Size: " $objItem.L2CacheSize  
write-host "Name: " $objItem.Name  
}
```

Külső programok is vezérelhetőek

- Microsoft alkalmazások mindegyike fog PS támogatással rendelkezni
- pl: MS SQL szerver kezelése távolról - új adatbázis létrehozása
- Set-Location
SQLSERVER:\SQL\localhost\DEFAULT\Databases
- \$MyDBVar = New-Object
Microsoft.SqlServer.Management.SMO.Database
- \$MyDBVar.Parent = (Get-Item ..)
- \$MyDBVar.Name = "NewDB"
- \$MyDBVar.Create()
- \$MyDBVar.State

Providerek

- A rendszer elemeihez való hozzáférés providereken keresztül történik.
- `get-psprovider` - elérhető providerek



```
Windows PowerShell
PS C:\MyScripts> Get-PSProvider

Name                Capabilities                Drives
----                -
Alias                ShouldProcess                {Alias}
Environment          ShouldProcess                {Env}
FileSystem           Filter, ShouldProcess        {C, G, H, I...}
Function             ShouldProcess                {Function}
Registry             ShouldProcess                {HKLM, HKCU}
Variable             ShouldProcess                {Variable}
Certificate          ShouldProcess                {cert}
```

PS C:\MyScripts> _

Provider használata

- Provider használatához csatlakozni kell az általa kínált erőforráshoz
 - Get-PSDrive

```
Windows PowerShell
PS C:\MyScripts> Get-PSDrive
```

Name	Provider	Root	CurrentLocation
A	FileSystem	A:\	
Alias	Alias		
C	FileSystem	C:\	MyScripts
cert	Certificate	\	
Env	Environment		
Function	Function		
G	FileSystem	G:\	
H	FileSystem	H:\	
HKCU	Registry	HKEY_CURRENT_USER	
HKLM	Registry	HKEY_LOCAL_MACHINE	
I	FileSystem	I:\	
Variable	Variable		
Z	FileSystem	Z:\	

Csatlakozás providerhez

- Set-Location Alias:

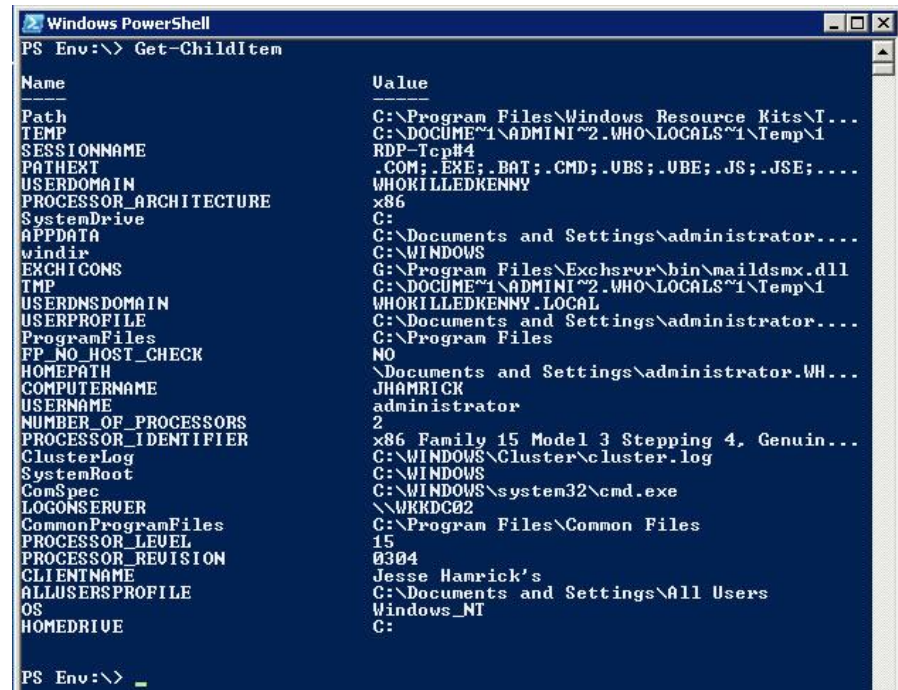
- get-childitem

- Set-Location Env:

- get-childitem

- Set-Location c:

- get-childitem



```
Windows PowerShell
PS Env:\> Get-ChildItem

Name                                     Value
----
Path                                     C:\Program Files\Windows Resource Kits\T...
TEMP                                    C:\DOCUME~1\ADMINI~2\WHO\LOCALS~1\Temp\1
SESSIONNAME                            RDP-Tcp#4
PATHEXT                                 .COM;.EXE;.BAT;.CMD;.VBS;.UBE;.JS;.JSE;...
USERDOMAIN                              WHOKILLEDKENNY
PROCESSOR_ARCHITECTURE                  x86
SystemDrive                             C:
APPDATA                                 C:\Documents and Settings\administrator...
windir                                  C:\WINDOWS
EXCHICONS                               G:\Program Files\Exchsrvr\bin\maildsmx.dll
TMP                                     C:\DOCUME~1\ADMINI~2\WHO\LOCALS~1\Temp\1
USERDNSDOMAIN                           WHOKILLEDKENNY.LOCAL
USERPROFILE                             C:\Documents and Settings\administrator...
ProgramFiles                           C:\Program Files
FP_NO_HOST_CHECK                        NO
HOMEPATH                                \Documents and Settings\administrator.WH...
COMPUTERNAME                            JHAMRICK
USERNAME                                administrator
NUMBER_OF_PROCESSORS                    2
PROCESSOR_IDENTIFIER                    x86 Family 15 Model 3 Stepping 4, Genuin...
ClusterLog                             C:\WINDOWS\Cluster\cluster.log
SystemRoot                              C:\WINDOWS
ComSpec                                 C:\WINDOWS\system32\cmd.exe
LOGONSERVER                             \\WKKDC02
CommonProgramFiles                     C:\Program Files\Common Files
PROCESSOR_LEVEL                          15
PROCESSOR_REVISION                      0304
CLIENTNAME                              Jesse Hamrick's
ALLUSERSPROFILE                         C:\Documents and Settings\All Users
OS                                       Windows_NT
HOMEDRIVE                               C:
```

PS Env:\> _

- A Get-ChildItem és társai parancsok nem lemez kezelők, hanem a provideren belül kezelik az elemeket
- pl:
 - fájl átnevezése
 - Rename-Item -Path c:\valami -newname másik
 - környezeti változó átnevezése
 - set-location env:
 - Rename-Item -Path env:var1 -NewName var2

PowerShell scriptek

- Scriptek készíthetők
- .ps I kiterjesztéssel
- Execution policy
 - megadja, hogy milyen scriptek futtathatóak
 - Restricted – scriptek letiltva
 - RemoteSigned – aláírt távoli, vagy helyi scriptek
 - AllSigned – aláírt scriptek
 - Unrestricted – bármi fut
- Get-ExecutionPolicy
- Set-ExecutionPolicy <policy name>
- #: comment

PowerShell script - paraméterek

- Bejövő paraméterek
 - args tömb
- echo \$args
- echo \$args[0]
- echo \$args[1]
- foreach(\$arg in \$args){
 echo \$arg
}

PowerShell script - paraméterek

- Formális paraméterlistát is generálhatunk
 - első sorba kell, hogy kerüljön
- `param([string]$foo = "x", [string]$bar = "y")`
 - `[string]`: paraméter típusa
 - `$foo`: paraméter neve
 - `„x”`: default érték
 - `fussal.ps | alma korte`
 - `fussal.ps | alma`
 - `fussal.ps | alma -foo korte`

PowerShell mintascript

- `param([string]$foo = "x", [string]$bar = "y")`
- `Write-Host "Arg: $foo"`
- `Write-Host "Arg: $bar"`
- `foreach ($svc in Get-Service){`
- `if($svc.displayName.StartsWith($args[0]))`
- `{`
- `echo $svc`
- `}`
- `}`

Feladat

- Készítsünk egy olyan scriptet, ami első paraméterként megadott nevű (azzal kezdődő, azt tartalmazó stb) szolgáltatást megkeres, és ha fut, akkor leállítja.

Feladat

- Az előző scriptet bővítsük ki. Két paramétert fogadjon:
 - -name és -muvelet
 - name: a szolgáltatás neve, amit berhelni akarunk
 - muvelet: lehet indit vagy leallit
 - Értelem szerűen működjön (és persze ellenőrizze, hogy az adott feladat értelmes –e) (ha fut, akkor ne indítsa el megint)