



PowerShell

17 mai 2010

Windows PowerShell 2.0

- ▶ Construit folosind .NET Common Language Runtime (CLR) si .NET Framework
- ▶ Windows PowerShell 2.0
 - ❑ Preinstalat pe Windows 7, Windows Server 2008 R2
 - ❑ Microsoft .NET Framework 2.0
- ▶ Windows PowerShell Integrated Scripting Environment (ISE)
 - ❑ Preinstalat pe Windows 7
 - ❑ Microsoft .NET Framework 3.5



Windows PowerShell

- ▶ Command line
- ▶ Scripting Language
- ▶ Automation Engine
- ▶ Inlocuitor pentru cmd.exe
- ▶ Orientat pe obiecte
- ▶ Comenzi verb-noun, demunite cmdlets
- ▶ Include multe alias-uri din Linux/Unix

Windows PowerShell

- ▶ Command line
- ▶ Scripting Language
- ▶ Automation Engine
- ▶ Orice set de comenzi poate devini un script
- ▶ Scripturile nu ruleaza implicit
- ▶ PowerShell Integrated Scripting Environment

Windows PowerShell

- ▶ Command line
- ▶ Scripting Language
- ▶ Automation Engine
- ▶ Poate fi integrat cu aplicatiile .NET
- ▶ Folosit pentru administrarea remote a calculatoarelor

Cmdlets

- ▶ Denumirea comenzilor din PowerShell
- ▶ Scrise in C#.NET
- ▶ Parametri prefixati de semnul -
- ▶ Oferă suport pentru obiecte
 - COM (Component Object Model)
 - ▶ comunicare între diverse componente software
 - WMI (Windows Management Instrumentation)
 - ▶ interfata cu sistemul de operare
 - .NET
 - ▶ obiecte din .NET Framework
 - ▶ obiecte create de utilizatori
- ▶ ~240 de comenzi in Windows 7
 - PS C:\> Get-Command | where {\$_.CommandType -eq "Cmdlet"}
| measure

Comenzi de bază

- ▶ `get-command`
- ▶ `get-help`
- ▶ `get-member`
- ▶ `get-psdrive`
- ▶ informații de bază despre un cmdlet
- ▶ datele extrase direct din codul sursă
- ▶ exemplu
 - ❑ PS C:\> `get-command ls`

CommandType	Name	Definition
-----	----	-----
Alias	ls	Get-ChildItem

Comenzi de bază

- ▶ `get-command`
- ▶ `get-help`
- ▶ `get-member`
- ▶ `get-psdrive`
- ▶ afiseaza informatii despre o comanda
- ▶ `alias-uri help si man`
- ▶ `help - full`
 - ❑ afiseaza informatii detaliate, inclusiv exemple
- ▶ `help – examples`
 - ❑ afiseaza doar exemple
- ▶ `help –online`
 - ❑ deschide resursele online disponibile

Comenzi de bază

▶ `get-command`

▶ `get-help`

▶ `get-member`

▶ `get-psdrive`

▶ afiseaza proprietatile si metodele unui obiect

▶ Exemplu

❑ `C:\PS>get-service | get-member`

Name	MemberType	Definition
----	-----	-----
Close	Method	System.Void Close()
CanStop	Property	System.Boolean CanStop {get;}

Comenzi de bază

- ▶ `get-command`
- ▶ `get-help`
- ▶ `get-member`
- ▶ `get-psdrive`
- ▶ afiseaza resursele disponibile pentru sesiunea curenta
- ▶ dispozitivele logice, inclusiv cele mapate din retea
- ▶ accesarea registrilor
- ▶ dispozitivele create prin comanda `new-psdrive`

Operatii asupra listelor de obiecte

Comanda	Actiune
Compare	Compara doua obiecte
Group	Separarea unui set de obiecte in grupuri
Measure	Masoara o anumita proprietate a unui obiect
Select	Selectia unui grup de proprietati
Sort	Ordonarea obiectelor
Tee	Duplicarea rezultatului
Where	Selectarea unui subset de obiecte

Alias-uri

- ▶ Exista alias-uri create pentru utilizatori Linux/CMD
 - *PS C:|Users|Mihai> alias | where {\$_.Definition -eq "Get-ChildItem"}*
- | <i>CommandType</i> | <i>Name</i> | <i>Definition</i> |
|--------------------|-------------|----------------------|
| <i>-----</i> | <i>----</i> | <i>-----</i> |
| <i>Alias</i> | <i>dir</i> | <i>Get-ChildItem</i> |
| <i>Alias</i> | <i>gci</i> | <i>Get-ChildItem</i> |
| <i>Alias</i> | <i>ls</i> | <i>Get-ChildItem</i> |
- ▶ Se pot crea noi alias-uri folosind comanda new-alias
 - *C:|PS>new-alias list get-childitem*
- ▶ Functii pentru aliasuri comenzi cu parametri
 - *function as {help *about_alias*}*
- ▶ Posibilitatea de export/import intre calculatoare

Pipeline si variabile

- ▶ Definirea se face folosind \$
 - ❑ `$current_process = Get-Process`
- ▶ Afisarea unei variabile se poate face folosind comanda `echo`

- ▶ Trimiterea de obiecte intre doua comenzi
- ▶ Variabila `$_` reprezinta obiectul curent
- ▶ `Format-list`
 - ❑ folosit pentru afisarea proprietatilor

If, elseif si else

▶ Sintaxa

- ❑ *if (condition) {command(s) to execute}*
- ❑ *elseif (condition2){command(s) to execute}*
- ❑ *else{command(s) to execute}*

▶ Parametrii

Operator	Descriere
-eq	Equals
-match	Match regular expression -> \$matches
-ne	Not Equal
-notmatch	Does not match
-gt -ge	Greater than/ Greater than or equal to
-lt -le	Less than/ Less than or equal to

▶ Exemplu

- ❑ *\$current_process = Get-Process
if (\$current_process.count -lt 50)
{ Write-Host "Low server load"; }*

Folosirea unui loop

► Sintaxa

- ❑ *while (condition) {command(s) to execute}*
- ❑ *do {command(s) to execute} while (condition)*
- ❑ *do {command(s) to execute} until (condition)*
- ❑ *for (initializer; exit condtion; step action) {comand(s) to execute}*
- ❑ *foreach (\$item in \$collection) {command(s) to execute}*

► Exemple

- ❑ *\$var = 1; while (\$var -le 50) {\$var; \$var += 1}*
- ❑ *\$var = 2; for (\$var = 1; \$var -le 50; \$i++) {\$var}*

► Accesarea manualului se poate face folosind help

- ❑ *help about_do, help about_for, etc.*

WMI support

Windows Management Instrumentation

- ▶ baza de date cu informatii pentru sistemele windows
 - ▶ serviciu wmi folosit pentru interogarea bazei de date
 - ▶ namespace-ul folosit este "root\CIMV2"
 - *Get-WmiObject -List -Namespace "root\CIMV2"*
 - ▶ pentru listarea proprietatilor unei clase folosim
 - *Get-WmiObject -Class "win32_Processor" | Get-Member*
-
- ▶ CIMV2 – Common Information Model Version 2

Folosirea directa a resurselor .NET

- ▶ spatiul .NET este ierarhizat folosind namespace
- ▶ accesarea se face folosind []
- ▶ Exemplu
 - ❑ *[System.Console]::Clear()*
 - ❑ *[System.Math]::Log10(100)*
 - ❑ *\$p = [System.Diagnostics.Process]::Start("notepad.exe")*
 - ❑ *\$p | Get-Member*
 - ❑ *[System.Int32]::Parse("254");*
- ▶ Link-uri utile biblioteca .NET
 - ❑ <http://msdn.microsoft.com/en-us/library/ms229335%28v=VS.90%29.aspx>

Scripturi powershell

- ▶ Windows PowerShell Integrated Scripting Environment (ISE)
- ▶ Extensie .ps1
- ▶ Nu pot fi rulate extern
- ▶ Necesita full-path
- ▶ Controlate prin Execution-Policy sau Group Policy
 - *Set-ExecutionPolicy*
 - Nivele de securitate pentru scripturi
 - Restricted (implicit)
 - AllSigned
 - RemoteSigned
 - Unrestricted (nerecomandat)

Exemplu script

```
▶ $NICs = Get-WMIObject Win32_NetworkAdapterConfiguration  
  / where{$_ .IPEnabled -eq "TRUE"}
```

```
$i = 1
```

```
Foreach($NIC in $NICs)
```

```
{  
  $NIC.EnableStatic("192.168."+$i+".42", "255.255.255.0")
```

```
$NIC.SetGateways("192.168."+$i+".1")
```

```
$DNSServers = "198.102.234.125","198.102.234.126"
```

```
$NIC.SetDNSServerSearchOrder($DNSServers)
```

```
$NIC.SetDynamicDNSRegistration("TRUE")
```

```
$NIC.SetWINSserver("198.102.234.125", "198.102.234.126")
```

```
$i ++
```

```
}
```

Background jobs

- ▶ Start-job

- *start-job -scriptblock {get-process} -name MyJob*

- ▶ Get-job

- *\$j = get-job -name MyJob*

- ▶ Receive-job

- *receive-job -job \$j*

- ▶ Stop-job

- *stop-job MyJob*

- ▶ Remove-job

- *remove-job MyJob*

Permisuni pentru fisiere

- ▶ Listarea permisiunilor existente
 - ❑ *[enum]::GetNames([System.Security.AccessControl.FileSystemRights])*
- ▶ Setarea permisiunilor in patru pasi :
 - ❑ Descriptorul curent de securitate
 - *\$SD = get-acl c:\2k8\Powershell\demo1.ps*
 - ❑ Construirea unui ACE (Access Control Entities)
 - *\$rule = new-object System.Security.AccessControl.FileSystemAccessRule("guest","FullControl", "Allow")*
 - ❑ Adaugarea ACE la DACL
 - *\$SD.AddAccessRule(\$Rule)*
 - ❑ Setarea noului nivel de permisuni
 - *Set-Acl c:\2k8\Powershell\demo1.ps \$SD*

Accesarea registrilor

- ▶ Accesarea unui anumit set de registri
 - ❑ *Set-location HKCU:*
- ▶ Afisarea continutului
 - ❑ *Get-ChildItem*
- ▶ Afisarea proprietatilor
 - ❑ *\$path="HKLM:\Software\Microsoft\Windows NT\CurrentVersion"*
 - ❑ *\$key = Get-ItemProperty \$path*
 - ❑ *\$key.ProductName*
- ▶ Setarea unor noi proprietati
 - ❑ *Set-ItemProperty -\$path -name "PSinfo" -type "String" -value "Razvan was here"*

Interesting scripting

- ▶ Crearea unui fisier de 1GB

- ❑ *`$path = "$env:temp\testfile.txt";`*
`$file = [io.file]::Create($path);`
`$file.SetLength(1gb);`
`$file.Close();`
`Get-Item $path;`

- ▶ Ping catre toate calculatoarele din retea

- ❑ *`1..254 | ForEach-Object -Process { Get-WmiObject`*
`Win32_PingStatus -Filter ("Address='172.16.1.'" + $_ + "'") } |`
`Select-Object -Property Address,ResponseTime,StatusCode`