

# Windows PowerShell

**Tomáš Herceg**

Microsoft Student Partner

<http://www.vbnet.cz>

# Agenda

- Historie
- Základní příkazy
- Skriptování
  - Podmínky
  - Cykly
- Rozšiřitelnost
- Budoucnost

# Problémy s administrací

- Nastavování „klikáním“
- Potřeba automatizace operací
- Nejednotné prostředí
- MMC – snaha o sjednocení
- Nedostatečné nástroje, dokumentace

# Historie

- Příkazový řádek MS-DOS a Windows
  - `command.com` a `cmd.exe`
- Windows Script Host
- Monad Shell → PowerShell
  - PowerShell 1.0 (2006)
  - PowerShell 2.0 Beta (2007)

# PowerShell

- Objektově orientovaný
  - Postaven nad .NET Frameworkem
- Nová sada příkazů (cmdlety)
  - Přehledné a systematické názvy (get-\*, add-\* ...)
  - Aliasy pro zkracování
  - Rozšiřitelnost, snadné vytváření vlastních cmdletů
- Dokumentace a nápověda
  - **get-help *cmdlet***

# Oblíbené cmdlety

- get-help, get-command
- get-process, stop-process
- get-service, start-service, stop-service, suspend-service, resume-service, restart-service
- format-list, format-table, export-csv, convertTo-html
- get-member, sort-object, where-object, group-object
- get-childitem, remove-item, new-item, copy-item, rename-item, move-item
- get-alias, set-alias, new-alias
- read-host

# Aliasy

- zkratky pro příkazy
- bez parametrů
- ps (get-process), gsv (get-service)
- dir, cd, cp, rm, mkdir, type
- sort, where, ft, fl



Seznámení s PowerShellem

**DEMO**

# Proměnné

- začínají symbolem \$
- stačí do nich přiřadit
- `$var = "Hello World!"`
- `$procesy = get-process`
- `$pole = @("jedna", "dvě", "tři")`
- `$asoc_pole = @{1="pondělí", 2="úterý"}`

# Předávání mezivýsledků

- *Pipeline* – znak | (pipe, roura)
- Předávání výstupu jedné aplikace na vstup aplikace druhé
- Předávání ve formě objektů nebo v textové formě (kompatibilita)
- Přesměrování
  - `ps > vysledek.txt ...` zapíše výstup do souboru
  - `ps >> vysledek.txt ...` připíše výstup do souboru

# Pipelines v praxi

- `get-process`
  - vypíše seznam procesů
- `get-process | sort name`
  - vypíše seznam procesů a předá jej programu `sort`, který je setřídí podle názvu
- Zkrácení díky aliasům (vypsání procesů řazených podle názvu a vypsání jako seznam):
  - **`get-process | sort-object name | format-list`**
  - **`ps | sort name | fl`**

The image features three blue, rounded rectangular shapes with a slight 3D effect, arranged in a cluster on a dark grey background. One shape is positioned on the left, another is slightly below and to the right of it, and a third is further to the right and slightly higher. The text 'Pipelines' is overlaid on the leftmost shape.

**Pipelines**

**DEMO**

# Relační operátory

- -eq, -ne ... rovná se, nerovná se
- -lt, -le ... menší, menší nebo rovno
- -gt, -ge ... větší, větší nebo rovno
  
- -match, -notmatch ... porovnávání reg. výrazy
- -like, -notlike ... porovnávání s maskou (\*, ?)
- -contains, -notcontains ... hledání v kolekci

# Podmínky

- `if (podmínka) {příkazy}`
- `if (podmínka) {příkazy}`  
`elseif (podmínka) {příkazy}`  
`else {příkazy}`

```
$var = 1; `  
if ($var -eq 1) {"var je jedna"} `  
elseif ($var -lt 0) {"var je záporné"} `  
else {cls}
```

Pokračování na dalším řádku ... zpětná čárka ` (Alt-7)

# Smyčky

- *While* cyklus
  - `$i = 1; while ($i -le 15) {$i; $i += 3}`
- *Do...While* cyklus
  - `$i = 1; do {$i; $i += 2} while ($i -le 15)`
- *ForEach* cyklus
  - `foreach ($i in 1,2,3,4,5) {$i * 5}`
  - `foreach ($p in ps) {$p.name + " " + $p.id}`
- *For* cyklus
  - `for ($i = 1; $i -le 10; $i += 1) {- $i}`

# Filtrování pomocí where-object

- Filtruje záznamy v kolekci
- Aktuální záznam ... `$_`
- Příklady:
  - `ps | where {$_.name -match "explor"}`
  - `dir C:\Windows | ? {$_.length -ge 65536}`

# Formátování výstupu (1)

- format-table (ft)

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a---	19.1.2008 9:00	65536	bfsvc.exe
-a--s	25.3.2008 18:12	67584	bootstat.dat
-----	10.2.2008 15:09	525792	DIFxAPI.dll
-a---	20.2.2008 19:36	394827	DirectX.log
-a---	11.10.2007 22:26	2686232	EPSTPLOG.TXT
-a---	19.1.2008 9:00	3080704	explorer.exe

- format-list (fl)

```
Name           : bfsvc.exe
Length          : 65536
CreationTime    : 29.1.2008 14:17:15
LastWriteTime   : 19.1.2008 9:00:04
LastAccessTime  : 29.1.2008 14:17:15
VersionInfo     :

Name           : bootstat.dat
Length          : 67584
```

## Formátování výstupu (2)

- `export-csv soubor ...` výstup do CSV
- `convertTo-html ...` výstup do HTML

- Příklad:

```
dir | convertTo-html -property `
name,extension,creationtime `
> test.html; .\test.html
```

# Formátování výstupu (3)

- Operátor `-f` pro zarovnávání do sloupců
  - `"{0,-5} {1,5}" -f "ahoj","14"`
    - 1. sloupec (šířka 5, vlevo), 2. sloupec (šířka 5, vpravo)
- Formátování (klasické z .NETu)
  - `"{0:C}" -f 152.43 ...` zformátuje jako měnu
- Příklad:

```
$files = dir; foreach ($file in $files) { `  
"{0,-20} {1,-40} {2,8}" -f `  
$file.name,$file.fullname,$file.length}
```

# Spolupráce s .NET Frameworkem

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Windows.Forms")  
$r = new-object System.Windows.Forms.Form  
$r.ShowInTaskBar = $True  
$b = new-object System.Windows.Forms.Button  
$b.Text = "nazdar"  
$b.Top = 50  
$b.Left = 30  
$b.Add_Click( {$r.Close()} )  
$r.Controls.Add($b)  
$r.ShowDialog()
```

# Skripty

- Přípona **.ps1**
- Podepisování skriptů
  - **set-executionPolicy Unrestricted ... vypnutí**
    - je třeba spustit s právy administrátora, stačí jednou
- Více informací
  - **get-help about\_signing**

# Vlastní cmdlet (1)

- Visual Studio – projekt **ClassLibrary**
- Přidáme reference na knihovny
  - **System.Configuration.Install**
  - **System.Management.Automation**
    - *C:\Program Files\Reference Assemblies\Microsoft\WindowsPowerShell\v1.0\*
- Vytvoříme třídu cmdletu odvozenou od **Cmdlet** s atributem **Cmdlet**

## Vlastní cmdlet (2)

- Přetížíme metodu **ProcessRecord**
- Pro každý parametr přidáme property
  - označíme atributy **Parameter**
  - pro validaci atribut **ValidateNotNullOrEmpty**
- Vytvoříme třídu odvozenou od **PSSnapIn**
- Přetížíme **Name**, **Vendor** a **Description**
- Vše umístíme do namespace  
**Microsoft.PowerShell.Commands**

# Vlastní cmdlet (3)

- Zaregistrujeme snap-in utilitou **InstallUtil**
  - `%systemroot%\Microsoft.NET\Framework\v2.0.50727\installutil.exe MySnapIn.dll`
  - Pro 64bitové platformy `...\Framework64\...`
  - Je třeba mít práva administrátora
- Před použitím musíme cmdlet načíst
  - `add-pssnapin jménoTřídy`
- Vhodné uložit do konfigurace PowerShellu



**Mailovací cmdlet**

**DEMO**

# Zdroje a odkazy

- <http://www.microsoft.com/powershell>
- Tutoriály
  - <http://www.powershellpro.com>
  - <http://www.computerperformance.co.uk/powershell/>
- Vývoj vlastních cmdletů
  - [http://msdn2.microsoft.com/en-us/library/cc136006\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/cc136006(VS.85).aspx)
- Záznam a dema z přednášky
  - <http://download.herceg.cz>

# Windows PowerShell

**Tomáš Herceg**

Microsoft Student Partner

<http://www.vbnet.cz>