PowerShell Quick Start

very quick - up to 6 hours
delivered (or not :)) by Ziemek Borowski, with some lab, homework and code review

Summary

Quick PowerShell course for people with limited experience on system administrator scripting (5 hours in person meeting + homework + online homework review session). The course is based on "Learn Windows PowerShell 3 in a Month of Lunches, Second Edition' by Don Jones and Jeffery Hicks".

Method of participation

- 5 hours in person meeting
- homework
- 1 hour online (WebEx/telco) meeting to review homework

Participant requirements

Required knowledge / skills

- basic knowledge on Windows Server administration and basic knowledge on computer programming (simple VBA macros, Lego Robotics, VBScript or cmd.exe are enough).

Required equipment

- Windows 7 or Windows 10 virtual machine
- working access to 'Laboratory' with Windows Server 2016

Agenda

- What is PowerShell
- How to apply for everyday tasks
- Running commands
- The pipeline: connecting commands
- Adding commands: function, snap-ins, modules

Agenda cont'ed

- Objects: data by another name
- Formatting: how to do it properly
- Filtering and comparison
- Simple function & script
- Homework selection: write script for specific needs

Homework

I expect one week for homework done. In middle of that time, I will organize office hours using WebEx remote conference tool. After homework submission date, we will meet and discuss selected works.

Supporting sources

- 'Using Windows PowerShell' / free
- 'Learn Windows PowerShell 3 in a Month of Lunches, Second Edition' by Don Jones and Jeffery Hicks Publisher: Manning Publications / paid, here Safair Books Online
- MikeFal/IntroToPowershell / free
- Rafał Kraik Powershell dla administratora Windows - kompletny kurs / paid, Udemy

Note:
Windows PowerShell Survival Guide @ TechNet Wikihttps://social.technet.microsoft.com/wiki/contents/articles/183.windows-powershell-survival-guide.aspx

What is PowerShell?

> - PowerShell is a command-line interface (CLI),
> - that contains a rich, yet simplified scripting language for automating complex, multi-step tasks
> - Built on the .NET Framework
> - Extensible, so various products and technologies can be managed by “snapping in” tech-specific extensions

> - Most importantly... it’s **discoverable**! It can teach you how to use itself!

Windows PowerShell - Crash Course] (https://channel9.msdn.com/Events/TechEd/NorthAmerica/2012/WSV321) by Don Jones and Jefferey Snover.

before

- DOS's `command.com`, cmd.exe, KixStart, VBScript/JScript (based on Windows Scripting Host (WHS)
- bash, python, perl (on Unicses or via Cygwin or native ports)

```
for /L %u in (1,2,99) do echo %i
```

Envisioned by Jeffery Snover - 2002

- The Monad Manifesto
- long time known as Project 'Monad'
- released as PowerShell RC1 - 2006-04
- first product requiring it was Exchange Server 2007
- PowerShell 2.0 - basic remoting
- Windows Server 2008 R2 - PowerShell 3.0
- Windows Server 2012 - PowerShell 4.0 - DSC - Desired State Configuration
- Windows Server 2016 - PowerShell 5.1

PowerShell scope of use

- PowerShell System Requirements
- Installing PowerShell v5.1 on Windows
- PowerShell Core is a cross-platform (Windows, Linux, and macOS)... You can download and install a PowerShell package for any of the platforms
- ... but do not expect exectly the same experience

Operating System Version	WMF 5.1	WMF 5.0	WMF 4.0	WMF 3.0	WMF 2.0
Windows Server 2016	Ships in-box				
Windows Server 2012 R2	Yes	Yes	Ships in-box		
Windows Server 2008 R2 SP1	Yes	Yes	Yes	Yes	Ships in-box
Windows 7 SP1	Yes	Yes	Yes	Yes	Ships in-box
Windows Server 2003					Yes
Windows XP					Yes

How to apply for everyday tasks

- interactive shell
- ad-hoc scripts
- tools
- CI/CD (Contionous Integration/Contionous Delivery)

Let's start

`powershell.exe` or `ise.exe`

```
Start-Transcript $env:Temp\GettingStarted.txt -Force

Set-ExecutionPolicy RemoteSigned -Force -Scope CurrentUser

get-host

Stop-Transcript
notepad $env:Temp\GettingStarted.txt
```

- The console window vs `Integrated Scripting Environment`
- Common Points of Confusion
  - 32- and 64-bit
  - Running as Administrator

Using help

- on fresh system execute `update-help` require Internet access and escalate shell.
- `get-help` is main command for geting help :)

```
help get-ChildItem
help get-ChildItem -examples
help get-ChildItem -detailed
help get-ChildItem -full
help get-ChildItem -examples
Get-Help Get-ChildItem -ShowWindow
```

- show-command Get-ChildItem

---?image=_Memes/CopingAndPasting.png&size=auto 90%

Discover - parameters

```
PS C:\code\bin> get-command get-member | get-member
   TypeName: System.Management.Automation.CmdletInfo
Name                MemberType     Definition
----                ----------     ----------
Equals              Method         bool Equals(System.Object ob
GetHashCode         Method         int GetHashCode()
GetType             Method         type GetType()
ResolveParameter    Method         System.Management.Automatior
ToString            Method         string ToString()
CommandType         Property       System.Management.Automatior
DefaultParameterSet Property       string DefaultParameterSet {
Definition          Property       string Definition {get;}
HelpFile            Property       string HelpFile {get;}
ImplementingType    Property       type ImplementingType {get;}
Module              Property       psmoduleinfo Module {get;}
ModuleName          Property       string ModuleName {get;}
Name                Property       string Name {get;}
Noun                Property       string Noun {get;}
Options             Property       System.Management.Automatior
OutputType          Property       System.Collections.ObjectMod
Parameters          Property       System.Collections.Generic.D
ParameterSets       Property       System.Collections.ObjectMod
PSSnapIn            Property       System.Management.Automatior
```

```
    Directory: C:\code\bin


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        22.06.2017     16:38       86325248 calc2017.exe
-a----        09.09.2017     21:31             66 script.ps1


PS C:\code\bin> calc2017.exe
calc2017.exe : The term 'calc2017.exe' is not recognized as the
the name, or if a path was included, verify that the path is co
At line:1 char:1
+ calc2017.exe
+ ~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (calc2017.exe:Str
    + FullyQualifiedErrorId : CommandNotFoundException


Suggestion [3,General]: The command calc2017.exe was not found,
PS C:\code\bin> .\calc2017.exe
```

The anatomy of a command

Adding commands: ... snap-ins ... (quite old fashion, powershell 1.0)

```
PS C:\code\powershellQuickStart> Get-PSSnapin -Registered
Name        : Microsoft.BDD.PSSnapIn
Description : This Microsoft Deployment Toolkit 2010 snap-in co

PS C:\code\powershellQuickStart> Add-PSSnapin Microsoft.BDD.PSS
```

Adding commands: ... modules ...

```
Get-Module
#What module we have locally available?
Get-Module -ListAvailable
# Starting powershell 4.0 (or 3.0) modules can be loaded automa
# but in powershell 2.0 we need do it manually
Import-Module   Defender
Remove-module   Defender
# what in module
Get-Command -Module Defender
Find-Module  PasswordsGenerator # PowerShellGallery.com

# PS C:\WINDOWS\system32> Find-Module   PasswordsGenerator
Version    Name                                   Repository
2.5.0      PasswordsGenerator                   PSGallery
Install-Module   PasswordsGenerator
Update-Module   PasswordsGenerator
UnInstall-Module   PasswordsGenerator -whatif
```

Adding commands: ... functions ...

```
. .\fx-Get-ZBFunction.ps1
```

Objects: data by another name

```
$string="This is a variable"
$string

#We can use Get-Member to find out all the information on our c
$string | Get-Member

$string.Length
$string.IndexOf('s')

# Powershell uses .Net objects.

$date=Get-Date
$date
$date | gm #gm is the alias of Get-Member

# Variables contains objects, so they has properties and methoc
$date.Day
$date.DayOfWeek
$date.DayOfYear
$date.ToUniversalTime()
$date.addDays(365)
```

Formatting: how to do it properly

```
dir | ft #ft is alias for Format-Table
dir | select-object
```

Filtering and comparison

`where-object`

Variables, input, output

Simple script

```
PS C:\code> echo "param(`$zmienna) `necho `$zmienna" >  .\scrip
PS C:\code> .\script.ps1 -zmienna "To jest argument zmiennej"
To jest argument zmiennej
```

Homework selection: write script for specific needs

- return date and time of the last restart - it should return at least two properties: name of machine and datetime of event
- test if a specified application has been installed and if it happens after a date of creating a new version of the software (stored somewhere in local machine as MSI package). if the test goes OK: install unattended that newer version of the software.
- write script which will remove all logs older than one year, and compress older than 30 days in c:\

???