

Zmienne – info

Jak korzystać ze zmiennych

Tworzenie i usuwanie zmiennych

Napęd Variables

Zmienne napisowe definiowane apostrofem i cudzysłowem

Definiujemy zmienną

```
PS C:\Windows\system32> $compName = 'Desktop007'

PS C:\Windows\system32> $compName
Desktop007
```

Jak korzystać z zmiennej

```
1 $compName = 'Desktop007'
2 $compName
3 Get-Service -ComputerName Desktop007 | where Name -Like "*net*"
4 Get-Service -ComputerName stacja | where Name -Like "*net*"

PS C:\Windows\system32> Get-Service -ComputerName Desktop007 | where Name -Like "*net*"
Get-Service : Nie można otworzyć Menedżera sterowania usługami na komputerze 'Desktop007'. Ta operacja może wymagać innych uprawnień.
At line:1 char:1
+ Get-Service -ComputerName Desktop007 | where Name -Like "*net*"
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Get-Service], InvalidOperationException
+ FullyQualifiedErrorId : System.InvalidOperationException,Microsoft.PowerShell.Commands.GetServiceCommand

PS C:\Windows\system32> hostname
stacja

PS C:\Windows\system32> Get-Service -ComputerName stacja | where Name -Like "*net*"

Status Name DisplayName
-----
Stopped Netlogon Netlogon
Stopped Netman Połączenia sieciowe
Running netprofm Usługa listy sieci
Stopped NetSetupSvc Usługa konfiguracji sieci
Stopped NetTcpPortSharing Usługa udostępniania portów Net.Tcp
Stopped WMPNetworkSvc Usługa udostępniania w sieci progra...
Stopped XboxNetApiSvc Usługa sieciowa Xbox Live
```

Polecenie ma za zadanie wyświetlić informacje o usługach wybranego komputera o ile w nawie takiej usługi jest ciąg znaków **net** posłuży do tego Get-Service -ComputerName DESKTOP-DESKTOP007 następnie (|) potok wysyła je do where które zajmu się filtrowaniem według zawartości w nazwie net (Name -Like "*net*")

Aby polecenie mogło sprawdzać usługi na różnych komputerach bez dalszej modyfikacji komendy zastępujemy nazwę **DESKTOP-007** zmienną **\$hostname** to nazwa lokalnego komputera Tutaj, "stacja" więc nazwa hosta jest nieprawidłowa i należy ją zmienić na bieżącą nazwę.

Znak\$ mówi, że PowerShell ma poszukać zmiennej **hostname** i z tej zmiennej pobrać wartość, faktyczna nazwa zmiennej to **hostname**

```
$hostname = hostname
```

```
Get-Service -ComputerName $hostname | where Name -Like "*net*"
```

```

PS C:\Windows\system32> $hostname = hostname

PS C:\Windows\system32> Get-Service -ComputerName $hostname | where Name -Like "*net*"

```

Status	Name	DisplayName
Stopped	Netlogon	Netlogon
Stopped	Netman	Połączenia sieciowe
Running	netprofm	Usługa listy sieci
Stopped	NetSetupSvc	Usługa konfiguracji sieci
Stopped	NetTcpPortSharing	Usługa udostępniania portów Net.Tcp
Stopped	WMPNetworkSvc	Usługa udostępniania w sieci progra...
Stopped	XboxNetApiSvc	Usługa sieciowa Xbox Live

Aby sprawdzić jakie zmienne występują w aktualnej sesji posłuż się poleceniem

Get-ChildItem VARIABLE:

i wyświetl wszystkie elementy z wirtualnego napędu **VARIABLE:**

Można sprawdzić, czy zmienna istnieje lub ją usunąć

```

PS C:\Windows\system32> Get-ChildItem VARIABLE:\

```

Name	Value
\$	compMame
?	True
^	Remove-Variable
args	{}
CIMSession	Sesja CIM: STACJA
CIMSessionOption	Microsoft.Management.Infrastructure.Options.DComSessionOptions
compName	Desktop007
ConfirmPreference	High
ConsoleFileName	
DebugPreference	SilentlyContinue
Error	{Cannot validate argument on parameter 'ComputerName'. The argument is null or
ErrorActionPreference	Continue
ErrorView	NormalView
ExecutionContext	System.Management.Automation.EngineIntrinsics
false	False
FormatEnumerationLimit	4
HOME	C:\Users\admin
Host	System.Management.Automation.Internal.Host.InternalHost
hostname	stacja
InformationPreference	SilentlyContinue
input	System.Collections.ArrayList+ArrayListEnumeratorSimple
LASTEXITCODE	0
MaximumAliasCount	4096
MaximumDriveCount	4096
MaximumErrorCount	256
MaximumFunctionCount	4096
MaximumHistoryCount	4096
MaximumVariableCount	4096
MyInvocation	System.Management.Automation.InvocationInfo
NestedPromptLevel	0
null	
option	Microsoft.Management.Infrastructure.Options.DComSessionOptions
OutputEncoding	System.Text.UTF8Encoding
PID	1792
profile	C:\Users\admin\Documents\WindowsPowerShell\Microsoft.PowerShellISE_profile.ps1
ProgressPreference	Continue
PSBoundParameters	{}
PSCommandPath	
PSCulture	pl-PL
PSDefaultParameterValues	{}
PSEdition	Desktop
PSEmailServer	
PSHOME	C:\Windows\SysWOW64\WindowsPowerShell\v1.0
psISE	Microsoft.PowerShell.Host.ISE.ObjectModelRoot

Poza hostname mamy mnóstwo innych zmiennych

Zamknięcie sesji PowerShell spowoduje, że zmienna zostanie utracona

Zmienne definiowane przez PowerShell będą dostępne zawsze

Wyświetlenie zmiennych pasujących do maski

```
7 Get-ChildItem VARIABLE:\comp*

PS C:\Windows\system32> Get-ChildItem VARIABLE:\comp*

Name                                     Value
----                                     -
compName                               Desktop007
```

Do pracy ze zmiennymi mamy cmdlety

Poniżej identyfikacja tych poleceń (w części rzeczownikowej mają VARIABLE)

```
8 Get-Command -Noun VARIABLE

CommandType      Name                                Version      Source
-----
Cmdlet           Clear-Variable                     3.1.0.0      Microsoft.PowerShell.Utility
Cmdlet           Get-Variable                       3.1.0.0      Microsoft.PowerShell.Utility
Cmdlet           New-Variable                       3.1.0.0      Microsoft.PowerShell.Utility
Cmdlet           Remove-Variable                   3.1.0.0      Microsoft.PowerShell.Utility
Cmdlet           Set-Variable                       3.1.0.0      Microsoft.PowerShell.Utility
```

Poniższe polecenie wyczyści wartość zmiennej, ale zmienna pozostaje

```
PS C:\Windows\system32> Clear-Variable compName
PS C:\Windows\system32> $compName
PS C:\Windows\system32> Get-ChildItem VARIABLE:\comp*

Name                                     Value
----                                     -
compName

PS C:\Windows\system32> Remove-Variable compName
PS C:\Windows\system32> |
```

Nie używamy \$ ponieważ chodzi o zmienną a nie o jej wartość

Odwołanie do **\$hostname** daje wartość pustą

```
PS C:\WINDOWS\system32> $compNames
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
```

Zmiennej nie ma

```
PS C:\WINDOWS\system32> Get-ChildItem VARIABLE:\comp*
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> |
```

Domyślnie odwołanie do nieistniejącej zmiennej nie generuje komunikatu o błędzie i to utrudnia odnalezienie literówki w skrypcie

Można wpisać polecenie Set-PSDebug -Strict co powoduje zmianę działania shella w taki sposób, że użycie niezainicjowanej zmiennej powoduje błąd

Ustawienie wartości zmiennej dla compNames

```
PS C:\Windows\system32> Set-Variable compName Desktop01  
PS C:\Windows\system32> $compName  
Desktop01
```

Tworzenie nowej zmiennej bez wartości

New-Variable filterExpression

```
PS C:\WINDOWS\system32> New-Variable filterExpression  
PS C:\WINDOWS\system32>
```

Ustawienie wartości zmiennej dla net

Set-Variable filterExpression net

```
PS C:\Windows\system32> Set-Variable filterExpression net
```

W PowerShell zmienne nie muszą mieć określonego typu on sam się domyśla o jaki typ danych chodzi, chociaż można określać typ zmiennych

Czy zmienna zmieniła wartość

```
PS C:\WINDOWS\system32> $filterExpression  
net
```

Zmodyfikujmy polecenie z początku

```
13 Set-Variable filterExpression net  
14 Get-Service -ComputerName $compNames | where Name -Like "$filterExpression"
```

```
< net  
PS C:\WINDOWS\system32> Get-Service -ComputerName $compNames | where Name -Like "$filterExpression"  


| Status  | Name              | DisplayName                            |
|---------|-------------------|----------------------------------------|
| Stopped | NetLogon          | NetLogon                               |
| Stopped | Netman            | Połączenia sieciowe                    |
| Running | netprofm          | Usługa listy sieci                     |
| Stopped | NetSetupSvc       | Usługa konfiguracji sieci              |
| Stopped | NetTcpPortSharing | Usługa udostępniania portów Net.Tcp    |
| Stopped | WMPNetworkSvc     | Usługa udostępniania w sieci progra... |
| Stopped | XboxNetApiSvc     | Usługa sieciowa Xbox Live              |


```

Budując wyrażenie filtrujące użyliśmy” (cudzysłów) a co by się stało aby w tym miejscu użyć ‘ (apostrofu)

Tu nie ma dowolności

```
14 Get-Service -ComputerName $compNames | where Name -Like "$filterExpression"  
15 Get-Service -ComputerName $compNames | where Name -Like '$filterExpression'  
16  
PS C:\WINDOWS\system32> Get-Service -ComputerName $compNames | where Name -Like '$filterExpression'  
PS C:\WINDOWS\system32>
```

Jeśli tekst zamykasz w cudzysłowie a w tekście znajduje się odwołanie do zmiennej to PowerShell stara się zinterpretować tą zmienną tzn podstawić za zmienną jej wartość

```

15 Get-Service -ComputerName $compNames | where Name -Like $filterExpression
16 "Displaying services like $filterExpression on computer(s) $compNames"

```

```

PS C:\WINDOWS\system32> "Displaying services like $filterExpression on computer(s) $compNames"
Displaying services like net on computer(s) DESKTOP-VNM5DM7

```

Jeśli tekst zamykasz w apostrof to PowerShell nie zajmie się podmianą zmiennych na wartości pozostawia tekst w apostrofie

```

17 'Displaying services like $filterExpression on computer(s) $compNames'

```

```

PS C:\WINDOWS\system32> 'Displaying services like $filterExpression on computer(s) $compNames'
Displaying services like $filterExpression on computer(s) $compNames

```

Teraz wiadomo dlaczego:

```

14 Get-Service -ComputerName $compNames | where Name -Like $filterExpression
15 Get-Service -ComputerName $compNames | where Name -Like '*$filterExpression*'
16 "Displaying services like $filterExpression on computer(s) $compNames"
17 'Displaying services like $filterExpression on computer(s) $compNames'

```

```

PS C:\WINDOWS\system32> Get-Service -ComputerName $compNames | where Name -Like '*$filterExpression*'
PS C:\WINDOWS\system32> |

```

oraz

```

14 Get-Service -ComputerName $compNames | where Name -Like "$filterExpression*"
15 Get-Service -ComputerName $compNames | where Name -Like '*$filterExpression*'
16 "Displaying services like $filterExpression on computer(s) $compNames"
17 'Displaying services like $filterExpression on computer(s) $compNames'

```

```

PS C:\WINDOWS\system32> Get-Service -ComputerName $compNames | where Name -Like "$filterExpression*"

```

Status	Name	DisplayName
Stopped	Netlogon	Netlogon
Stopped	Netman	Połączenia sieciowe
Running	netprofm	Usługa listy sieci
Stopped	NetSetupSvc	Usługa konfiguracji sieci
Stopped	NetTcpPortSharing	Usługa udostępniania portów Net.Tcp
Stopped	WMPNetworkSvc	Usługa udostępniania w sieci progra...
Stopped	XboxNetApiSvc	Usługa sieciowa Xbox Live

Jak korzystać ze zmiennych

Tworzenie i usuwanie zmiennych

Napęd Variables

Zmienne napisowe definiowane apostrofem i cudzysłowem