Moduły PowerShell – info

Wirtualne napędy w PowerShell

Jak pracować z plikami, dyskami i rejestrem

Jak załadować i usunąć moduł do pamięci

Jak samodzielnie tworzyć nowe napędy wirtualne

W tej lekcji opowiem o modułowości PowerShell, pokaże, jak pracować z rejestrem.

Do przejścia pomiędzy poszczególnymi napędami można posługiwać się poleceniem

```
PS C:\Windows\system32> c:
```

Mamy dodatkowe dyski, niby napędy które mają pozytywne znaczenie

```
PS C:\Windows\system32> cd alias:
PS Alias:\> _
```

Po przejściu na ten napęd ls wyświetli wszystkie zdefiniowane aliasy.

Możesz też sprawdzić co oznacza jakiś istniejący alias

```
Alias           rm -> Remove-Item
Alias           rmdir -> Remove-Item
Alias           rmo -> Remove-Module
```

Poniżej wartości wszystkich zmiennych zdefiniowanych w sesji

```
PS Alias:\> cd variable:
```

Get-Variable

```
PS Variable:\> Get-Variable

Name                           Value
----                           -----
$                              variable:
?                              True
^                              cd
args                           {}
ConfirmPreference              High
ConsoleFileName
DebugPreference                SilentlyContinue
VerbosePreference              SilentlyContinue
WarningPreference              Continue
WhatIfPreference               False
```

Wpisanie alias:

```
PS Variable:\> alias:
alias: : The term 'alias:' is not recognized as the name of a cmdlet, function, script file, or operable program. Check
 the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ alias:
+ ~~~~~~
    + CategoryInfo          : ObjectNotFound: (alias::String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```

```
PS Variable:\> c:
PS C:\Windows\system32>
```

Przechodzi na c:

Tak naprawdę napis k: to definiowana w PowerShell funkcja, która przechodzi na k:

```
PS C:\Windows\system32> cd C:\Users\
PS C:\Users> dir


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08.07.2024     14:42                admin
d-r---        26.06.2024     10:15                Public
-a----        08.07.2024     15:08           1014 mailscript.ps1
```

Każde z tych poleceń to cmdlet lub funkcja używająca innego cmdleta do wykonania określonej czynności.

Set-Location C:\Users\admin to cd C:\Users\admin

```
PS C:\Users> Set-Location C:\Users\admin
PS C:\Users\admin>
```

```
PS C:\Users\admin> cd ..
PS C:\Users> Get-ChildItem


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08.07.2024     14:42                admin
d-r---        26.06.2024     10:15                Public
-a----        08.07.2024     15:08           1014 mailscript.ps1
```

```
PS C:\Users> New-Item nowyplik.txt


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        08.07.2024     16:29              0 nowyplik.txt
```

```
PS C:\Users> Get-ItemProperty nowyplik.txt


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        08.07.2024     16:29              0 nowyplik.txt
```

```
PS C:\Users> Remove-Item nowyplik.txt
PS C:\Users> dir


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08.07.2024     14:42                admin
d-r---        26.06.2024     10:15                Public
-a----        08.07.2024     15:08           1014 mailscript.ps1


PS C:\Users> ls


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08.07.2024     14:42                admin
d-r---        26.06.2024     10:15                Public
-a----        08.07.2024     15:08           1014 mailscript.ps1
```

To jakie napędy będą widoczne dla PowerShell zależy od tego jacy dostawcy będą zainstalowani. Jeżeli zainstalowany jest np. SQL Server

Aby zobaczyć listę prowaiderów

```
PS C:\Users> Get-PSProvider

Name                 Capabilities                                    Drives
----                 ------------                                    ------
Registry             ShouldProcess, Transactions                     {HKLM, HKCU}
Alias                ShouldProcess                                   {Alias}
Environment          ShouldProcess                                   {Env}
FileSystem           Filter, ShouldProcess, Credentials              {C}
Function             ShouldProcess                                   {Function}
Variable             ShouldProcess                                   {Variable}
```

```
PS C:\Users> Set-Location HKLM:
PS HKLM:\> Set-Location HKLM:\SOFTWARE
PS HKLM:\SOFTWARE>
```

Get-ChildItem Microsoft - wyświetli zawartość wszystkich kluczy w katalogu Microsoft

Zakładam nowy klucz:

```
PS HKLM:\SOFTWARE> New-Item -Name NewSoftware


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE


Name                           Property
----                           --------
NewSoftware
```

Poniższe polecenie zdefiniuje nową właściwość

```
PS HKLM:\SOFTWARE> New-ItemProperty -Path NewSoftware -Name mode -Value "Service"


mode         : Service
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\NewSoftware
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE
PSChildName  : NewSoftware
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry
```

-Name mode (nazwa właściwości) -Value wartość dla właściwości "Service"

Powyższy skrypt może być wykorzystany podczas instalacji aplikacji, który będzie konfigurował fikcyjną aplikację pracującą w trybie aplikacji lub usługi

Aby wyświetlić właściwości dla już istniejącego obiektu:

```
PS HKLM:\SOFTWARE> Get-ItemProperty -Path NewSoftware


mode         : Service
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\NewSoftware
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE
PSChildName  : NewSoftware
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry
```

Aby zmienić wartość istniejącej właściwości użyj polecenia

```
PS HKLM:\SOFTWARE> Set-ItemProperty -Path NewSoftware -Name mode -Value "Application"
```

Sprawdzam

```
PS HKLM:\SOFTWARE> Get-ItemProperty -Path NewSoftware


mode         : Application
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\NewSoftware
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE
PSChildName  : NewSoftware
PSDrive      : HKLM
PSProvider   : Microsoft.PowerShell.Core\Registry
```

Usuwam klucz

```
PS HKLM:\SOFTWARE> Remove-Item NewSoftware
```

Teraz widać, dlaczego domyślnym poleceniem usuwającym pliki jest w PowerShell Remove-Item

Autorzy PowerShell chcieli abyśmy tych samy komend używali zarówno z systemem plików, z rejestrami, AD, sql server

Nazwy cmdletów nie są powiązane z nomenklaturą plikową mają być abstrakcyjne i uniwersalne dzięki czemu można je zastosować w różnych sytuacjach.

Czynności, które wykonujemy w PowerShell zależą od tego jakie aplikacje są zainstalowane na tym komputerze a dokładniej jakie moduły zostały zainstalowane.

Aby sprawdzić listę modułów zainstalowanych na komputerze wykonaj

```
PS HKLM:\SOFTWARE> Get-Module -ListAvailable
```

Import modułu SqlServer

```
PS HKLM:\SOFTWARE> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the exe
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y
PS HKLM:\SOFTWARE> Import-Module SqlServer
PS HKLM:\SOFTWARE>
```

Sprawdzam jacy dostawcy są dostępni

```
PS HKLM:\SOFTWARE> Get-PSProvider

Name                 Capabilities                                    Drives
----                 ------------                                    ------
Registry             ShouldProcess, Transactions                     {HKLM, HKCU}
Alias                ShouldProcess                                   {Alias}
Environment          ShouldProcess                                   {Env}
FileSystem           Filter, ShouldProcess, Credentials              {C}
Function             ShouldProcess                                   {Function}
Variable             ShouldProcess                                   {Variable}
Certificate          ShouldProcess                                   {Cert}
SqlServer●           Credentials                                     {SQLSERVER}
```

```
PS HKLM:\SOFTWARE> c:
PS C:\Users> SQLSERVER:
PS SQLSERVER:\> Set-Location SQLSERVER:\SQL\stacja
PS SQLSERVER:\SQL\stacja>
```

Wyświetla jakie mam bazy danych

```
PS SQLSERVER:\SQL\stacja> dir
PS SQLSERVER:\SQL\stacja>
```

Brak

Spróbuje stworzyć nową bazę danych poleceniem

```
PS SQLSERVER:\SQL\stacja> New-Item testowabaza
New-Item : Cannot retrieve the dynamic parameters for the cmdlet. SQL Server PowerShell provider error: Could not conne
ct to 'stacja\testowabaza'. [Failed to connect to server stacja\testowabaza. --> A network-related or instance-specific
 error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify t
hat the instance name is correct and that SQL Server is configured to allow remote connections. (provider: SQL Network
Interfaces, error: 26 - Error Locating Server/Instance Specified)]
At line:1 char:1
+ New-Item testowabaza
+ ~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidArgument: (:) [New-Item], ParameterBindingException
    + FullyQualifiedErrorId : GetDynamicParametersException,Microsoft.PowerShell.Commands.NewItemCommand
```

Nie utworzyłem

To polecenie nie zostało zaimplementowane po stronie SQL, nie każde polecenie może być używane w każdej sytuacji, zależy to od dostawcy

```
PS SQLSERVER:\SQL\stacja> c:
```

Aby sprawdzić dostępność napędów używam:

```
PS C:\Users> Get-PSDrive

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
Alias                                  Alias
C                  34,05        989,07 FileSystem    C:\                                                Users
Cert                                   Certificate   \
Env                                    Environment
Function                               Function
HKCU                                   Registry      HKEY_CURRENT_USER
HKLM                                   Registry      HKEY_LOCAL_MACHINE                                 SOFTWARE
SQLSERVER                              SqlServer     SQLSERVER:\                                        SQL\stacja
Variable                               Variable
WSMan                                  WSMan
```

Usuwam moduł SqlServer

```
PS C:\Users> Remove-Module SqlServer
PS C:\Users> Get-PSDrive

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
Alias                                  Alias
C                  34,05        989,07 FileSystem    C:\                                                Users
Cert                                   Certificate   \
Env                                    Environment
Function                               Function
HKCU                                   Registry      HKEY_CURRENT_USER
HKLM                                   Registry      HKEY_LOCAL_MACHINE                                 SOFTWARE
Variable                               Variable
WSMan                                  WSMan
```

Można samodzielnie definiować napędy

```
PS C:\Users> New-PSDrive Scripts filesystem C:\Users

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
Scripts            0,00        989,07 FileSystem    C:\Users
```

Sprawdzam czy dział, czy jest nowy napęd

```
PS C:\Users> cd Scripts:
PS Scripts:\>
```

```
PS Scripts:\> dir


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08.07.2024     14:42                admin
d-r---        26.06.2024     10:15                Public
-a----        08.07.2024     15:08           1014 mailscript.ps1
```

Próba usunięcia napędu

```
PS Scripts:\> Remove-PSDrive Scripts
Remove-PSDrive : Cannot remove drive 'Scripts' because it is in use.
At line:1 char:1
+ Remove-PSDrive Scripts
+ ~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [Remove-PSDrive], PSInvalidOperationException
    + FullyQualifiedErrorId : InvalidOperation,Microsoft.PowerShell.Commands.RemovePSDriveCommand
```

Dlaczego nie dział?

Bo jest w tym napędzie.

Rozwiązanie

```
PS Scripts:\> c:
PS C:\Users> Remove-PSDrive Scripts
PS C:\Users>
```

Działa

Przyjrzałeś się wirtualnym napędom w PowerShell

Zobaczyłeś, że praca z plikami, dyskami, rejestrem odbywa się z wykorzystaniem tych samych poleceń

Wiesz, jak załadować i usunąć moduł do pamięci

Wiesz jak samodzielnie tworzyć nowe napędy wirtualne