

Przekazywanie argumentów – info

Przekazywanie parametrów lokalnych do komend zdalnych

Opcja ArgumentList i sekcja Param

Zaawansowane opcje polecenia Invoke-Command

Wyświetlimy listę procesów z komputera zdalnego, ograniczymy wyniki tylko do procesu notepad

Nazwę procesu chce przechowywać w zmiennej, która mogła by się stać parametrem funkcji

```
PS C:\Users\Administrator> $name = "notepad"
PS C:\Users\Administrator> Invoke-Command -ComputerName server -Credential $cred -ScriptBlock { Get-Process -Name $name }
Cannot validate argument on parameter 'Name'. The argument is null or empty. Provide an argument that is not null or empty, and then try the command again.
+ CategoryInfo          : InvalidData: (:) [Get-Process], ParameterBindingValidationException
+ FullyQualifiedErrorId : ParameterArgumentValidationError,Microsoft.PowerShell.Commands.GetProcessCommand
+ PSComputerName        : server
```

Zmienna \$name jest zadeklarowana lokalnie a na komputerze zdalnym taka zmienna nie istnieje

Poprawiam

Do polecenia **Invoke-Command** dodałem nowy argument **-ArgumentList** parametr ten zawiera listę zmiennych, które dla maszyny zdalnej będą widoczne jako parametry, w moim przypadku jest do przekazania tylko zmienna **\$name** lista jest jednoelementowa. Wykonałem modyfikację skryptu, pojawia się nagłówek **Param** który definiuje jedną lub więcej zmiennych, tyle samo, ile w **ArgumentList**

```
Invoke-Command -ComputerName server -Credential $cred -ScriptBlock { Param($x) Get-Process -Name $x } -ArgumentList ($name)
```

```
PS C:\Users\Administrator> Invoke-Command -ComputerName server -Credential $cred -ScriptBlock { Param($x) Get-Process -Name $x } -ArgumentList ($name)
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName	PSComputerName
243	14	2932	15628	0,02	1500	2	notepad	server

W kolejnym przykładzie jak przekazywać większą ilość parametrów

```
> $log="Security"
> $num=10
```

Lokalnie:

```
PS C:\Users\Administrator> Get-EventLog $log -Newest $num
```

Index	Time	EntryType	Source	InstanceID	Message
15453	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15452	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15451	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15450	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15449	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15448	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15447	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15446	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15445	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...
15444	paź 15 17:37	SuccessA...	Microsoft-Windows...	5379	Odczytano poświadczenia Menedżera poświadczeń...

Zdalnie

```
Invoke-Command -ComputerName serwer -Credential $cred -ScriptBlock { Param($logName,$n)  
Get-EventLog $logName -Newest $n } -ArgumentList ($log,$num)
```

```
Invoke-Command -ComputerName serwer -Credential $cred -ScriptBlock { Param($logName,$n) Get-EventLog $logName -Newest $n } -ArgumentList ($log,$num)
```

Celowo użyłem innych nazw zmiennych niż te lokalne, aby nie było wrażenia, że przekazywanie zmiennych działa nawet bez wykorzystania -ArgumentList

Z praktycznego punktu widzenia lepiej używać takich samych nazw parametrów co przekazywane zmienne, parametry nazwałem \$logName,\$n

Skrypt przesyłany do wywołania na systemie zdalnym jest taki sam jak poprzednio

Index	Time	EntryType	Source	InstanceID	Message	PSComputerName
2504	paź 15 17:44	SuccessA...	Microsoft-Windows...	4634	Użytkownik wylogował się z konta....	serwer
2503	paź 15 17:44	SuccessA...	Microsoft-Windows...	4634	Użytkownik wylogował się z konta....	serwer
2502	paź 15 17:44	SuccessA...	Microsoft-Windows...	4634	Użytkownik wylogował się z konta....	serwer
2501	paź 15 17:44	SuccessA...	Microsoft-Windows...	4634	Użytkownik wylogował się z konta....	serwer
2500	paź 15 17:44	SuccessA...	Microsoft-Windows...	4624	Logowanie do konta zakończyło się po...	serwer
2499	paź 15 17:44	SuccessA...	Microsoft-Windows...	4672	Przypisano specjalne uprawnienia do ...	serwer
2498	paź 15 17:44	SuccessA...	Microsoft-Windows...	4776	Komputer podjął próbę sprawdzenia po...	serwer
2497	paź 15 17:44	SuccessA...	Microsoft-Windows...	4624	Logowanie do konta zakończyło się po...	serwer
2496	paź 15 17:44	SuccessA...	Microsoft-Windows...	4672	Przypisano specjalne uprawnienia do ...	serwer
2495	paź 15 17:44	SuccessA...	Microsoft-Windows...	4776	Komputer podjął próbę sprawdzenia po...	serwer

Jak widać to działo na komputerze zdalnym serwer

Wydaje się, że Invoke-Command mając na celu uruchomienie polecenia na komputerze zdalnym jako parametry potrzebuje tylko nazwę komputera i blok skryptu, tymczasem, gdy zajrzę do helpa to okaże się, że parametrów polecenia jest więcej.

help Invoke-Command

get-help Invoke-Command -full

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/invoke-command?view=powershell-7.1>

Przekazywanie parametrów lokalnych do komend zdalnych

Opcja ArgumentList i sekcja Param

Zaawansowane opcje polecenia Invoke-Command