Planowanie zadań – info

Jak uruchamiać zadania w tle

Jak odczytywać wyniki tych zadań

Jak sprawdzać status zadań uruchomionych w tle

Jak planować zadania do uruchomienia w przyszłości lub regularnie

Jak zarządzać zadaniami zaplanowanymi

PowerShell pozwala na uruchamianie poleceń w trybie zadań tzn. nie musisz czekać na zakończenie się wykonywania polecania a po uruchomieniu się zadania możesz zająć się innymi czynnościami tak aby w pewnym momencie sprawdzić status wcześniej uruchomionego zadania.

```
PS C:\WINDOWS\system32> Start-Job -ScriptBlock { Get-Process } -Name "Obecnie trwa proces"

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Running       True            localhost             Get-Process
```

Dzięki Start-Job zadanie Get-Process zostało uruchomione w tle i na ekranie nie widać wyników, gdyby było to polecenie długotrwałe to nie muszę czekać na jego zakończenie.

Za pewien czas można sprawdzić czy zadanie już się wykonało.

```
PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     True            localhost             Get-Process
```

W identyfikacji mogą pomóc numery Id tu 1 i nazwa zadania tu „Obecnie trwa proces”

Po -ScriptBlock znajduje się polecenie { Get-Process }

Get-WmiObject wymaga innego postępowania do znanej komendy dopisujemy parametr -AsJob

```
PS C:\WINDOWS\system32> Get-WmiObject -Class Win32_operatingSystem -ComputerName DESKTOP-VNM5DM7 -AsJob

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
3      Job3            WmiJob          Running       True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
```

Teraz będzie on uruchamiany w tle

Gdy zapytam o polecenia wykonywane w tle to mamy dwa zadania

```
PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     True            localhost             Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
```

Drugie zadanie to zadanie Wmi

Polecenie wykorzystujące remouting

Do Invoke-Command komendy dopisujemy parametr -AsJob

```
PS C:\WINDOWS\system32> Invoke-Command -ComputerName DESKTOP-VNM5DM7 { Get-Process } -AsJob

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
5      Job5            RemoteJob       Running       True            DESKTOP-VNM5DM7      Get-Process
```

Wszystkie podzadania są traktowane jako jedno główne

Aby sprawdzić stan zadania

```
PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa...  BackgroundJob   Completed     True            localhost            Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7      Get-Process
```

Zleciłem zadanie na nie uruchomionym komputerze serwer

```
PS C:\WINDOWS\system32> Invoke-Command -ComputerName serwer { Get-Process } -AsJob

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
7      Job7            RemoteJob       Running       True            serwer               Get-Process


PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa...  BackgroundJob   Completed     True            localhost            Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7      Get-Process
7      Job7            RemoteJob       Failed        False           serwer               Get-Process
```

Zadanie nie powiodło się.

Jeżeli chcesz uzyskać informacje o tylko jednym zadaniu o znanym id tu 3

```
PS C:\WINDOWS\system32> Get-Job -Id 3

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
```

Jeżeli chcesz uzyskać informacje o tylko jednym zadaniu o znanej nazwie tu "Obecnie trwa proces"

```
PS C:\WINDOWS\system32> Get-Job -Name "Obecnie trwa proces"

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa...  BackgroundJob   Completed     True            localhost            Get-Process
```

Zadania można wyświetlać ze względu na inne kryteria tylko te zadania, które zwróciły dodatkowe dane.

```
PS C:\WINDOWS\system32> Get-Job -HasMoreData $True

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa...  BackgroundJob   Completed     True            localhost            Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7      Get-Process
```

Dlatego można przejrzeć ich wyniki. Wynik można potokiem przesłać do Where-Object

Uruchomię zadanie, które nigdy się nie skończy. W blok wpisuje nieskończoną pętle

```
PS C:\WINDOWS\system32> Start-Job -ScriptBlock { while ($true) {} } -Name loop

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
9      loop            BackgroundJob   Running       True            localhost            while ($true) {}


PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     True            localhost            Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7      Get-Process
7      Job7            RemoteJob       Failed        False           serwer               Get-Process
9      loop            BackgroundJob   Running       True            localhost            while ($true) {}
```

Takie zadanie należy przerwać

```
PS C:\WINDOWS\system32> Stop-Job -Name loop

PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     True            localhost            Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7      Get-Process
7      Job7            RemoteJob       Failed        False           serwer               Get-Process
9      loop            BackgroundJob   Stopped       False           localhost            while ($true) {}
```

Zakończenie zadanie nie jest równoznaczne z usunięciem zadania.

Status Stopped wskazuje, że zadanie zakończyło się w nienaturalny sposób.

Do usuwania zadań służy

```
PS C:\WINDOWS\system32> Get-Job -Name loop | Remove-Job

PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     True            localhost            Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7      Get-Process
7      Job7            RemoteJob       Failed        False           serwer               Get-Process
```

Zadania loop niema.

Odczytywanie wyników zadań

```
PS C:\WINDOWS\system32> Receive-Job -Name "Obecnie trwa proces"

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    244      14     7592      14388       0,09    1584   0 audiodg
     83       5      856       1176       0,00    1932   0 CompatTelRunner
     95       7     2888       7312       0,00    3180   2 conhost
    156      10     6660       1012       0,02    3844   0 conhost
    481      20     1588       4808       0,28     548   0 csrss
    168      11     1548       4492       0,11     624   1 csrss
    349      15     1660       5380       0,34    2620   2 csrss
    411      17     3832      18472       0,55    1096   2 ctfmon
```

Zadanie "Obecnie trwa proces" miało wyświetlać bieżące procesy.

Jeżeli będę jeszcze raz wyświetlał wyniki tego polecenia to

```
PS C:\WINDOWS\system32> Receive-Job -Name "Obecnie trwa proces"

PS C:\WINDOWS\system32>
```

ich nie ma

dlatego że Receive-Job po zwróceniu wyniku usuwa go

można to sprawdzić

```
PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     False           localhost             Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7       Get-Process
7      Job7            RemoteJob       Failed        False           serwer                Get-Process
```

Brak nowych danych do pokazania

Jeżeli chcesz, aby Receive-Job nie usuwał wyniku

```
PS C:\WINDOWS\system32> Receive-Job -Id 3 -Keep


SystemDirectory : C:\WINDOWS\system32
Organization    :
BuildNumber     : 19042
RegisteredUser  : admin
SerialNumber    : 00378-60400-60696-AA229
Version         : 10.0.19042



PS C:\WINDOWS\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Obecnie trwa... BackgroundJob   Completed     False           localhost             Get-Process
3      Job3            WmiJob          Completed     True            DESKTOP-VNM5DM7      Get-WmiObject -Class W...
5      Job5            RemoteJob       Completed     True            DESKTOP-VNM5DM7       Get-Process
7      Job7            RemoteJob       Failed        False           serwer                Get-Process
```

Usuwamy wszystkie zadania

```
PS C:\WINDOWS\system32> Get-Job | Remove-Job

PS C:\WINDOWS\system32> Get-Job

PS C:\WINDOWS\system32>
```

Zadania można zaplanować do wykonania na później

Należy wykonać dwa kroki

Po pierwsze utwórz opcje

```
$optHIDE = New-ScheduledJobOption -WakeToRun -HideInTaskScheduler
```

-WakeToRun oznacza, że komputer w stanie wstrzymania powinien się wybudzić, gdy nadejdzie chwila uruchomienia zaplanowanego zadania

-HideInTaskScheduler oznacza, że zadanie nie ma być pokazywane w harmonogramie zadań

```
$optElev = New-ScheduledJobOption -RequireNetwork -RunElevated
```

-RequireNetwork oznacza, że zadanie wymaga dostępności sieci

-RunElevated oznacza, że zadanie ma być uruchomione z podniesionymi uprawnieniami jako administrator

Jeżeli chcesz zobaczyć jakie opcje zostały zdefiniowane to wyświetl obiekt opcji

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


StartIfOnBatteries      : False
StopIfGoingOnBatteries  : True
WakeToRun               : True
StartIfNotIdle          : True
StopIfGoingOffIdle      : False
RestartOnIdleResume     : False
IdleDuration            : 00:10:00
IdleTimeout             : 01:00:00
ShowInTaskScheduler     : False
RunElevated             : False
RunWithoutNetwork       : True
DoNotAllowDemandStart   : False
MultipleInstancePolicy  : IgnoreNew
JobDefinition           :
```

StartIfOnBatteries : False - jeśli testujesz działanie zadań na laptopie nie podłączonym do ładowarki to zmień na True

Aby określić, kiedy zadanie ma się uruchomić należy zdefiniować wyzwalacz

```
$trigWeek= New-ScheduledTaskTrigger -Weekly -DaysOfWeek: Monday,Wednesday,Friday -At 3am -RandomDelay 00:20
```

Zadanie uruchamiane

-Weekly – co tydzień

DaysOfWeek: Monday,Wednesday,Friday w poniedziałek, środę, piątek

-At 3am – o 3 w nocy

-RandomDelay 00:20 – z losowym opóźnieniem około 20 minut

Drugi

```
$trigLater = New-ScheduledTaskTrigger -Once -At (Get-Date).AddMinutes(3)
```

-Once - zadanie uruchomi się tylko raz

-At (Get-Date).AddMinutes(3) - za 3 minuty od teraz

Obiekt trigera można skontrolować

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


Enabled              : True
EndBoundary          :
ExecutionTimeLimit   :
Id                   :
Repetition           :
StartBoundary        : 2021-10-16T01:00:00Z
DaysOfWeek           : 42
RandomDelay          : PT20M
WeeksInterval        : 1
PSComputerName       :
```

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


Enabled              : True
EndBoundary          :
ExecutionTimeLimit   :
Id                   :
Repetition           :
StartBoundary        : 2021-10-16T18:25:07Z
RandomDelay          :
PSComputerName       :
```

Teraz planujemy zadanie

```
PS C:\Windows\system32> Register-ScheduledJob -ScriptBlock { Get-WmiObject -Class Win32_OperatingSystem | Select FreePhysicalMemory }
      -Trigger $trigWeek -ScheduledJobOption $optElev -Name "Memory usage report"

Id        Name            JobTriggers     Command                                  Enabled
--        ----            -----------     -------                                  -------
7         Memory usage... 1                Get-WmiObject -Class Win32_Operating... True
```

Sprawdzam zaplanowane zadania

```
PS C:\Windows\system32> Get-ScheduledJob

Id        Name            JobTriggers     Command                                  Enabled
--        ----            -----------     -------                                  -------
7         Memory usage... 1                Get-WmiObject -Class Win32_Operating... True
```

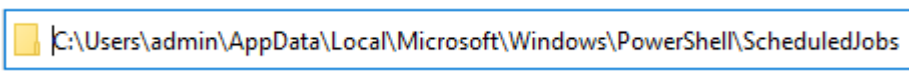
Kolejne zadanie ma pobrać listę procesów po 3 minutach

```
PS C:\Windows\system32> Register-ScheduledJob -ScriptBlock { Get-Process } -Trigger $trigLater -ScheduledJobOption $optHide -Name "Pr

Id        Name            JobTriggers     Command                                  Enabled
--        ----            -----------     -------                                  -------
8         Processes af... 1                Get-Process                              True
```

Zadania można znaleźć w harmonogramie zadań

Zadania można znaleźć w profilu użytkownika



Każde zadanie ma swój folder, można znaleźć folder zadania, a jeżeli zadanie się wykonało to także jego wynik.

Zdefiniowałem dwa zadania więc po wykonaniu

```
PS C:\Windows\system32> Get-ScheduledJob

Id          Name            JobTriggers     Command                                 Enabled
--          ----            -----------     -------                                 -------
7           Memory usage... 1                Get-WmiObject -Class Win32_Operating... True
8           Processes af... 1                Get-Process                             True
```

Poprzednio Get-Job nie pokazywało wyników a teraz

```
PS C:\Windows\system32> Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
39     Processes af... PSScheduledJob  Completed     True            localhost             Get-Pro
```

Raportuje zadanie, które pomyślnie wykonało się w tle to to do którego miało dojść w ciągu 3 minut

Aby usunąć zadanie z harmonogramu

```
PS C:\Windows\system32> Unregister-ScheduledTask -TaskName "Memory usage report"
```

Sprawdzam wynik zadania, które się wykonało

```
Receive-Job -Id 39 -Keep
```

Wiesz, jak uruchamiać zadania w tle

Jak odczytywać wyniki tych zadań

Jak sprawdzać status zadań uruchomionych w tle

Jak planować zadania do uruchomienia w przyszłości lub regularnie

Jak zarządzać zadaniami zaplanowanymi