

# Creating and Managing PowerShell Background Jobs

---



**Liam Cleary**

CEO / MICROSOFT MVP / MICROSOFT CERTIFIED TRAINER

@shareplicity [www.shareplicity.com](http://www.shareplicity.com) | @helloitsliam [www.helloitsliam.com](http://www.helloitsliam.com)



# Overview



## Creating Background Jobs

- Start-Job
- & Operator

## Remove Background Jobs

## Wait for Background Jobs to Complete

## Retrieve Background Job Information

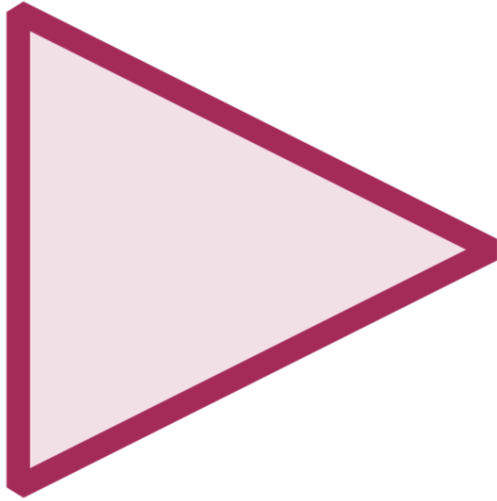


# Creating Background Jobs

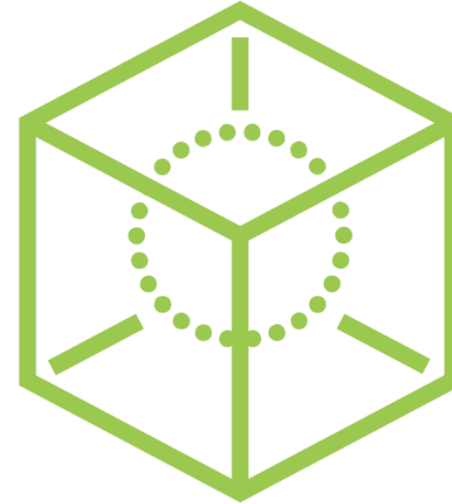
---



# Creating Background Jobs



PowerShell command `Start-Job`



Ampersand (`&`)



# Start-Job

Starts a PowerShell background job



# Creating a Background Job

**# Create Background Job using "Start-Job"**

```
Start-Job -ScriptBlock { Get-Process -Name notepad }
```

**# Create Background Job using "&" Operator**

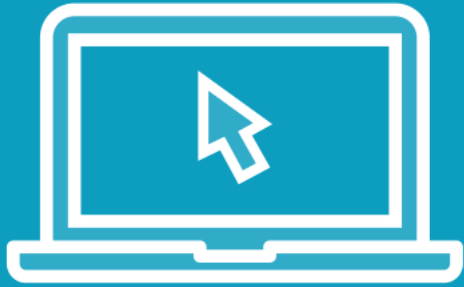
```
Get-Process -Name notepad &
```

**# Create Background Job using "Invoke-Command" and "-AsJob" Parameter**

```
$job = Invoke-Command `
    -ComputerName (Get-Content -Path "C:\Servers.txt") `
    -ScriptBlock { Get-Service -Name WinRM } `
    -JobName WinRM -ThrottleLimit 16 -AsJob
```



# Demo



## Create Background Jobs



# Remove Background Jobs

---





# Remove-Job

Deletes a PowerShell background job



# Removing Background Jobs

Delete by Name

Delete All Jobs

Delete by State

Delete  
"Invoke-Command" Jobs

Delete by Instance ID



# Removing Jobs

## # Remove Job by Name

```
$job = Get-Job -Name BatchJob  
$job | Remove-Job
```

## # Remove Job by Instance ID

```
$job = Start-Job -ScriptBlock {Get-Process PowerShell}  
$job | Format-List -Property *  
Remove-Job -InstanceId cf02b942-9807-4407-87f3-d23e72055872
```

## # Delete Job using "Invoke-Command"

```
$session = New-PSSession -ComputerName "Server"  
Invoke-Command -Session $session -ScriptBlock {Start-Job `br/>    -ScriptBlock {Get-Process} -Name "Job"}  
Invoke-Command -Session $session `br/>    -ScriptBlock {Remove-Job -Name "Job"}
```



# Demo



## Remove Completed Background Jobs



# Wait for Background Jobs to Complete

---



# Wait-Job

Suppresses the command prompt until one or all of the PowerShell background jobs running in the session are completed



# Waiting for Background Jobs

**# Wait for All Jobs**

```
Get-Job | Wait-Job
```

**# Wait for Several Jobs by ID to Finish**

```
Wait-Job -Id 1,2,5 -Any
```

**# Waiting for "Invoke-Command" Jobs**

```
$job = Invoke-Command -Session $session -ScriptBlock {Get-Process} -AsJob
```

```
$job | Wait-Job
```



# Demo



## Waiting for Background Jobs to Complete

- Wait-Job



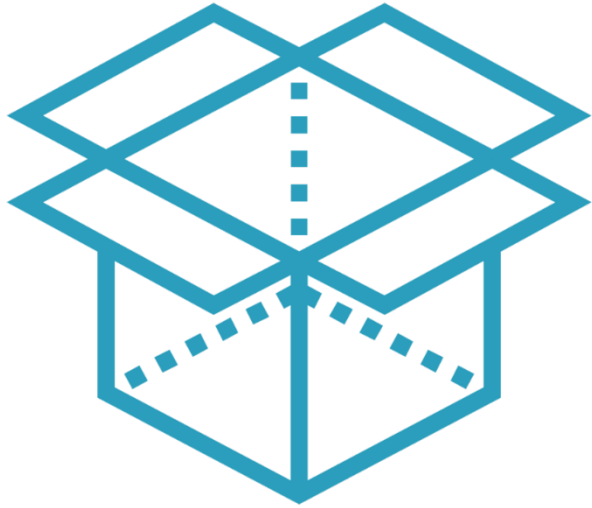


# Retrieve Background Job Information

---



# Retrieve Background Job Details



## Get-Job

Gets PowerShell background jobs that are running in the current session



## Receive-Job

Gets the results of the PowerShell background jobs in the current session



# Get-Job

## # Retrieve Job Details

```
Start-Job -ScriptBlock {Get-Process} -Name "Job"  
$job = Get-Job -Name "Job"  
$job
```

## # Get Child Job Details

```
Get-Job -IncludeChildJob
```

## # Get Non-Started Jobs

```
Get-Job -State NotStarted
```



# Receive-Job

## # Retrieve Specific Job Results

```
$job = Start-Job -ScriptBlock {Get-Process}  
Receive-Job -Job $job
```

## # Retrieve Specific Job Results from Multiple Computers

```
$session = New-PSSession -ComputerName DC01, SQL02, WEB03  
$job = Invoke-Command -Session $session `  
    -ScriptBlock {Start-Job -ScriptBlock {$env:COMPUTERNAME}}  
Get-Job
```



# Demo



## Retrieve Background Job Information



# Summary



Created, and Removed Background Jobs

Reviewed using both the Start-Job and the & Operator for creating jobs

Retrieved details about the background jobs

Removed completed jobs



Up Next:

Executing Background Jobs on Remote  
Computers

---

