

Working Smarter with PowerShell Background Jobs

MULTITASKING IN POWERSHELL



Liam Cleary

CEO / MICROSOFT MVP / MICROSOFT CERTIFIED TRAINER

@shareplicity www.shareplicity.com | @helloitsliam www.helloitsliam.com



Overview



What is Multitasking?

Multitasking within PowerShell

- Asynchronous versus Synchronous

Why use PowerShell Jobs?



PowerShell Multitasking and Multithreading



Multitasking

CPU can execute multiple tasks at the same time. CPU switches between tasks to allow collaboration between applications and tasks. Each process uses separate memory.



Multithreading

CPU uses multiple threads for tasks. CPU can execute multiple threads for a process at a time. Each process uses shared memory. Faster than multitasking.



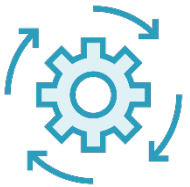
Multitasking within PowerShell



PowerShell Multitasking



PowerShell is a single-threaded application



Cannot start running another command in the same console until the first finishes



Make the decision up-front to move to a long-running command



Asynchronous versus Synchronous



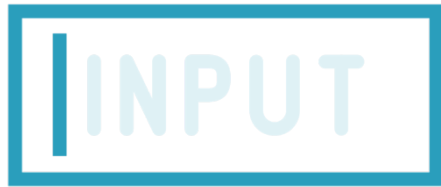
Asynchronous versus Synchronous

PowerShell runs normal commands *synchronously*, meaning you hit return and then wait for the command to complete

Moving a job into the background allows it to run *asynchronously*, meaning that you can continue using the shell for other tasks while the command completes



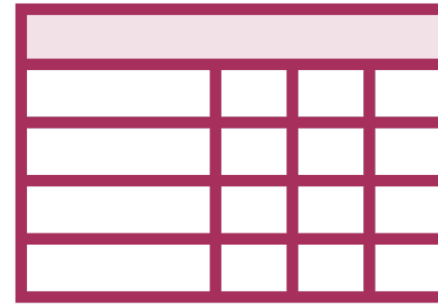
Synchronous Commands



Allow
responding to
input requests



Produce error
messages when
something goes
wrong



Display results
as soon as they
become
available



Prompt you for
any missing
information



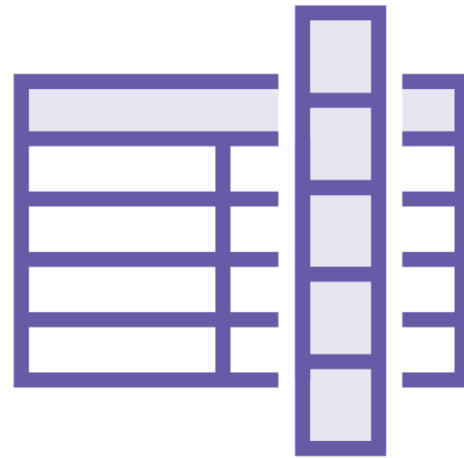
Asynchronous Commands



Not allowed to
see input
requests



Produce error
messages but
unable to see
them



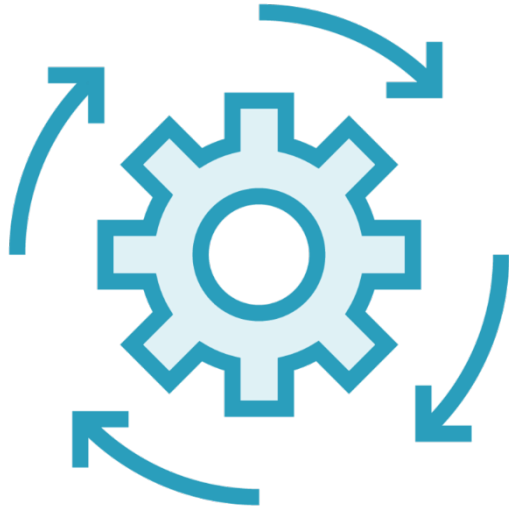
Display results
at the end of the
running task



No prompt for
any missing
information,
command will
fail



Multitasking / Multithreading Options



Background Jobs



Runspaces

Why use PowerShell Jobs?



Why use PowerShell Jobs?

PowerShell Background Jobs allow you to move singular tasks, into batches of tasks for easier management



Increase Performance

Jobs execute separately allowing for better performance in most instances



Execute Multiple Tasks

Without locking the PowerShell console, further tasks can be executed



Can Schedule Tasks

Jobs can be scheduled for specific times.



Why use PowerShell Jobs?



Long Running Tasks



PowerShell Remoting Tasks



Scheduled Jobs are
NOT supported
within PowerShell 7



PowerShell Job Scenario Examples

In a large organization where there are hundreds of servers to manage. Need to query total RAM for all servers

Using a On-premises and Cloud Services. Need to update specific details within Azure Active Directory from an On-premises application



Summary



Reviewed multitasking within PowerShell

Understood how PowerShell Jobs can help in normal administration tasks



Up Next:

Understanding PowerShell Background Jobs

