



python



PowerShell

FPDF Library
PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Resume-Job'

PS C:\Users\wahid> Get-Help Resume-Job

NAME

Resume-Job

SYNOPSIS

Restarts a suspended job.

SYNTAX

Resume-Job [-Filter] <System.Collections.Hashtable> [-Wait] [-Confirm]
[-WhatIf] [<CommonParameters>]

Resume-Job [-Id] <System.Int32[]> [-Wait] [-Confirm] [-WhatIf]
[<CommonParameters>]

Resume-Job [-InstanceId] <System.Guid[]> [-Wait] [-Confirm] [-WhatIf]
[<CommonParameters>]

Resume-Job [-Job] <System.Management.Automation.Job[]> [-Wait] [-Confirm]
[-WhatIf] [<CommonParameters>]

Resume-Job [-Name] <System.String[]> [-Wait] [-Confirm] [-WhatIf]

[<CommonParameters>]

Resume-Job [-State] {NotStarted | Running | Completed | Failed | Stopped |
Blocked | Suspended | Disconnected | Suspending | Stopping | AtBreakpoint}
[-Wait] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Resume-Job` cmdlet resumes a workflow job that was suspended, such as by using the `Suspend-Job` cmdlet or the `about_Suspend-Workflow` (`../PSWorkflow/about/about_Suspend-Workflow.md`) activity. When a workflow job resumes, the job engine reconstructs the state, metadata, and output from saved resources, such as checkpoints. The job is restarted without any loss of state or data. The job state is changed from `Suspended` to `Running`.

Use the parameters of `Resume-Job` to select jobs by name, ID, instance ID or pipe a job object, such as one returned by the `Get-Job` cmdlet, to `Resume-Job`. You can also use a property filter to select a job to be resumed.

By default, `Resume-Job` returns immediately, even though all jobs might not yet be resumed. To suppress the command prompt until all specified jobs are resumed, use the `Wait` parameter.

The `Resume-Job` cmdlet works only on custom job types, such as workflow jobs. It does not work on standard background jobs, such as those that are started by using the `Start-Job` cmdlet. If you submit a job of an unsupported type, `Resume-Job` generates a terminating error and stops running.

To identify a workflow job, look for a value of `PSWorkflowJob` in the `PSJobTypeName` property of the job. To determine whether a particular custom job type supports the `Resume-Job` cmdlet, see the help topics for the custom job type.

Before using a Job cmdlet on a custom job type, import the module that supports the custom job type, either by using the ``Import-Module`` cmdlet or getting or using a cmdlet in the module.

This cmdlet was introduced in Windows PowerShell 3.0.

PARAMETERS

`-Filter <System.Collections.Hashtable>`

Specifies a hash table of conditions. This cmdlet resumes jobs that satisfy all of the conditions in the hash table. Enter a hash table where the keys are job properties and the values are job property values.

`-Id <System.Int32[]>`

Specifies an array of IDs for jobs that this cmdlet resumes.

The ID is an integer that uniquely identifies the job in the current session. It is easier to remember and to type than the instance ID, but it is unique only in the current session. You can type one or more IDs, separated by commas. To find the ID of a job, run ``Get-Job``.

`-InstanceId <System.Guid[]>`

Specifies an array of instance IDs of jobs that this cmdlet resumes. The default is all jobs.

An instance ID is a GUID that uniquely identifies the job on the computer.

To find the instance ID of a job, run ``Get-Job``.

`-Job <System.Management.Automation.Job[]>`

Specifies the jobs to be resumed. Enter a variable that contains the jobs or a command that gets the jobs. You can also pipe jobs to the ``Resume-Job`` cmdlet.

-Name <System.String[]>

Specifies an array of friendly names of jobs that this cmdlet resumes.

Enter one or more job names. Wildcard characters are permitted.

-State <System.Management.Automation.JobState>

Specifies the state of jobs to resume. The acceptable values for this parameter are:

- NotStarted

- Running

- Completed

- Failed

- Stopped

- Blocked

- Suspended

- Disconnected

- Suspending

- Stopping

This cmdlet resumes only jobs in the Suspended state.

For more information about job states, see JobState Enumeration

(/dotnet/api/system.management.automation.jobstate).

-Wait <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet suppresses the command prompt until all job results are restarted. By default, this cmdlet immediately returns the available results.

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (<https://go.microsoft.com/fwlink/?LinkID=113216>).

----- Example 1: Resume a job by ID -----

```
PS C:\> Get-Job EventJob
```

Id	Name	PSJobTypeName	State	HasMoreData	Location
4	EventJob	PSWorkflowJob	Suspended	True	Server01

\\Script\Share\Event.ps1

```
PS C:\> Resume-Job -Id 4
```

----- Example 2: Resume a job by name -----

```
PS C:\> Resume-Job -Name WorkflowJob, InventoryWorkflow, WFTest*
```

----- Example 3: Use custom property values -----

```
PS C:\> Resume-Job -Filter @{CustomID="T091291"} -State Suspended
```

-- Example 4: Resume all suspended jobs on a remote computer --

```
PS C:\> Invoke-Command -ComputerName Srv01 -ScriptBlock {Get-Job -State  
Suspended | Resume-Job}
```

The command uses the `Invoke-Command` cmdlet to run a command on the Srv01 computer. The remote command uses the State parameter of the `Get-Job` cmdlet to get all suspended jobs on the computer. A pipeline operator (`|`) sends the suspended jobs to the `Resume-Job` cmdlet, which resumes them.

----- Example 5: Wait for jobs to resume -----

```
PS C:\> Resume-Job -Name WorkflowJob, InventoryWorkflow, WFTest* -Wait
```

----- Example 6: Resume a workflow that suspends itself -----

```
#SampleWorkflow  
Workflow Test-Suspend  
{  
    $a = Get-Date  
    Suspend-Workflow  
    (Get-Date)- $a  
}
```

```
PS C:\> Test-Suspend -PSComputerName Server01
```

Id	Name	PSJobTypeName	State	HasMoreData	Location
	Command				
8	Job8	PSWorkflowJob	Suspended	True	Server01
	Test-Suspend				

PS C:\> Resume-Job -Name "Job8" -Wait

Id	Name	PSJobTypeName	State	HasMoreData	Location
	Command				
8	Job8	PSWorkflowJob	Running	True	Server01
	Test-Suspend				

PS C:\> Receive-Job -Name Job8

```

Days          : 0
Hours         : 0
Minutes       : 0
Seconds       : 19
Milliseconds  : 823
Ticks         : 198230041
TotalDays     : 0.000229432917824074
TotalHours    : 0.00550639002777778
TotalMinutes  : 0.330383401666667
TotalSeconds  : 19.8230041
TotalMilliseconds : 19823.0041
PSComputerName : Server01

```

The `Resume-Job` cmdlet lets you resume a workflow job that was suspended by using the `Suspend-Workflow` activity. This activity suspends a workflow from within a workflow. It is valid only in workflows.

For information about the `Suspend-Workflow`, see
about_Suspend-Workflow](../PSWorkflow/about/about_Suspend-Workflow.md).

REMARKS

To see the examples, type: "get-help Resume-Job -examples".

For more information, type: "get-help Resume-Job -detailed".

For technical information, type: "get-help Resume-Job -full".

For online help, type: "get-help Resume-Job -online"