PowerShell Get-Help on command 'Suspend-Job'

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

```
NAME
    Suspend-Job

SYNOPSIS
    Temporarily stops workflow jobs.


SYNTAX
    Suspend-Job [-Filter] <System.Collections.Hashtable> [-Force] [-Wait]
    [-Confirm] [-WhatIf] [<CommonParameters>]

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

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

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

    Suspend-Job [-Name] <System.String[]> [-Force] [-Wait] [-Confirm] [-WhatIf]
    [<CommonParameters>]

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

DESCRIPTION

The `Suspend-Job` cmdlet suspends workflow jobs. Suspend means to temporarily interrupt or pause a workflow job. This cmdlet allows users who are running workflows to suspend the workflow. It complements the Suspend-Workflowhttps://go.microsoft.com/fwlink/?LinkId=267141 activity, which is a command in the workflow that suspends the workflow.

The `Suspend-Job` cmdlet works only on workflow jobs. It does not work on standard background jobs, such as those that are started by using the `Start-Job` cmdlet.

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 `Suspend-Job` cmdlet, see the help topics for the custom job type.

When you suspend a workflow job, the workflow job runs to the next checkpoint, suspends, and immediately returns a workflow job object. To wait for the suspension to complete before getting the job, use the Wait parameter of `Suspend-Job` or the `Wait-Job` cmdlet. When the workflow job is suspended, the value of the State property of the job is Suspended.

Suspending correctly relies on checkpoints. The current job state, metadata, and output are saved in the checkpoint so the workflow job can be resumed without loss of state or data. If the workflow job does not have checkpoints, it cannot be suspended correctly. To add checkpoints to a workflow that you are running, use the PSPersist workflow common parameter. You can use the Force parameter to suspend any workflow job immediately and to suspend a workflow job that does not have checkpoints, but the action could cause loss of state and data.

Before you use a Job cmdlet on a custom job type, such as a workflow job (PSWorkflowJob) import the module that supports the custom job type, either by using the `Import-Module` cmdlet or using 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 suspends jobs that satisfy all of the conditions. Enter a hash table where the keys are job properties and the values are job property values.

-Force <System.Management.Automation.SwitchParameter>

Suspends the workflow job immediately. This action could cause a loss of state and data.

By default, `Suspend-Job` lets the workflow job run until the next checkpoint and then suspends it. You can also use this parameter to suspend workflow jobs that do not have checkpoints.

-Id <System.Int32[]>

Specifies the IDs of jobs that this cmdlet suspends.

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, use the `Get-Job` cmdlet.

-InstanceId <System.Guid[]>

Specifies the instance IDs of jobs that this cmdlet suspends. 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, use `Get-Job`.

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

Specifies the workflow jobs that this cmdlet stops. Enter a variable that contains the workflow jobs or a command that gets the workflow jobs. You can also pipe workflow jobs to the `Suspend-Job` cmdlet.

-Name <System.String[]>

Specifies friendly names of jobs that this cmdlet suspends. Enter one or more workflow job names. Wildcard characters are supported.

-State <System.Management.Automation.JobState>

Specifies a job state. This cmdlet stops only jobs in the specified state. The acceptable values for this parameter are:

- NotStarted

- Running

- Completed

- Failed

- Stopped

- Blocked

- Suspended

- Disconnected

- Suspending

- Stopping

`Suspend-Job` suspends only workflow jobs in the Running 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 the workflow job is in the suspended state. By default, `Suspend-Job` returns immediately, even if the workflow job is not yet in the suspended state.

The Wait parameter is equivalent to piping a `Suspend-Job` command to the `Wait-Job` cmdlet.

-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: Suspend a workflow job by name ----------

```
#Sample Workflow

Workflow Get-SystemLog

{

   $Events = Get-WinEvent -LogName System

   CheckPoint-Workflow

   InlineScript {\\Server01\Scripts\Analyze-SystemEvents.ps1 -Events $Events}

}

Get-SystemLog -AsJob -JobName "LogflowJob"

Get-Job -Name LogflowJob


Id   Name          PSJobTypeName   State       HasMoreData     Location

Command

--   ----          -------------   -----       -----------     --------

-------

4    LogflowJob    PSWorkflowJob   Running     True            localhost

Get-SystemLog


Suspend-Job -Name LogflowJob


Id   Name          PSJobTypeName   State       HasMoreData     Location

Command

--   ----          -------------   -----       -----------     --------

-------

4    LogflowJob    PSWorkflowJob   Suspended   True            localhost

Get-SystemLog
```

--------- Example 2: Suspend and resume a workflow job ---------

```
Suspend-Job -Name LogWorkflowJob
```

```
Id   Name         PSJobTypeName      State        HasMoreData     Location
   Command
--   ----         -------------      -----        -----------     --------
   -------
67   LogflowJob    PSWorkflowJob      Running      True            localhost
   LogWorkflow
```

```
Get-Job -Name LogWorkflowJob
```

```
Id   Name         PSJobTypeName      State        HasMoreData     Location
   Command
--   ----         -------------      -----        -----------     --------
   -------
67   LogflowJob    PSWorkflowJob      Suspended     True            localhost
   LogWorkflow
```

```
Get-Job -Name LogWorkflowJob | Resume-Job
```

```
Id     Name         PSJobTypeName      State      HasMoreData     Location
   Command
--     ----         -------------      -----      -----------     --------
   -------
67     LogflowJob    PSWorkflowJob      Running     True            localhost
   LogWorkflow
```

---- Example 3: Suspend a workflow job on a remote computer ----

```
Invoke-Command -ComputerName Srv01 -Scriptblock {Suspend-Job -Filter
@{CustomID="031589"}
```

This command uses the `Invoke-Command` cmdlet to suspend a workflow job on the

Srv01 remote computer. The value of the Filter parameter is a hash table that specifies a CustomID value. This CustomID is job metadata (PSPrivateMetadata).

------- Example 4: Wait for the workflow job to suspend -------

```
Suspend-Job VersionCheck -Wait


Id     Name          PSJobTypeName      State         HasMoreData     Location
     Command
--     ----          -------------      -----         -----------     --------
     -------
 5     VersionCheck  PSWorkflowJob      Suspended     True
localhost     LogWorkflow
```

This command suspends the VersionCheck workflow job. The command uses the Wait parameter to wait until the workflow job is suspended. When the workflow job runs to the next checkpoint and is suspended, the command finishes and returns the job object.

---------- Example 5: Force a workflow job to suspend ----------

```
Suspend-Job Maintenance -Force
```

This command suspends the Maintenance workflow job forcibly. The Maintenance job does not have checkpoints. It cannot be suspended correctly and might not resume correctly.

REMARKS

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

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

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

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