



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 'Remove-PSSession'

PS C:\Users\wahid> Get-Help Remove-PSSession

NAME

Remove-PSSession

SYNOPSIS

Closes one or more PowerShell sessions (PSSessions).

SYNTAX

Remove-PSSession [-ComputerName] <System.String[]> [-Confirm] [-WhatIf]
[<CommonParameters>]

Remove-PSSession -ContainerId <System.String[]> [-Confirm] [-WhatIf]
[<CommonParameters>]

Remove-PSSession [-Id] <System.Int32[]> [-Confirm] [-WhatIf]
[<CommonParameters>]

Remove-PSSession -InstanceId <System.Guid[]> [-Confirm] [-WhatIf]
[<CommonParameters>]

Remove-PSSession -Name <System.String[]> [-Confirm] [-WhatIf]

[<CommonParameters>]

Remove-PSSession [-Session]

<System.Management.Automation.Runspaces.PSSession[]> [-Confirm] [-WhatIf]

[<CommonParameters>]

Remove-PSSession -VMId <System.Guid[]> [-Confirm] [-WhatIf]

[<CommonParameters>]

Remove-PSSession -VMName <System.String[]> [-Confirm] [-WhatIf]

[<CommonParameters>]

DESCRIPTION

The `Remove-PSSession` cmdlet closes PowerShell sessions (PSSessions) in the current session. It stops any commands that are running in the PSSessions , ends the PSSession , and releases the resources that the PSSession was using. If the PSSession is connected to a remote computer, this cmdlet also closes the connection between the local and remote computers.

To remove a PSSession , enter the Name , ComputerName , ID , or InstanceID of the session.

If you have saved the PSSession in a variable, the session object remains in the variable, but the state of the PSSession is Closed.

PARAMETERS

-ComputerName <System.String[]>

Specifies an array of names of computers. This cmdlet closes the PSSessions that are connected to the specified computers. Wildcard characters are permitted.

Type the NetBIOS name, an IP address, or a fully qualified domain name of one or more remote computers. To specify the local computer, type the computer name, `localhost`, or a dot (`.`).

`-ContainerId <System.String[]>`

Specifies an array of IDs of containers. This cmdlet removes sessions for each of the specified containers. Use the `docker ps` command to get a list of container IDs. For more information, see the help for the docker ps (<https://docs.docker.com/engine/reference/commandline/ps/>) command.

`-Id <System.Int32[]>`

Specifies an array of IDs of sessions. This cmdlet closes the PSSessions with the specified IDs. Type one or more IDs, separated by commas, or use the range operator (`. .`) to specify a range of IDs.

An ID is an integer that uniquely identifies the PSSession in the current session. It is easier to remember and type than the InstanceId, but it is unique only in the current session. To find the ID of a PSSession, run the `Get-PSSession` cmdlet without parameters.

`-InstanceId <System.Guid[]>`

Specifies an array of instance IDs. This cmdlet closes the PSSessions that have the specified instance IDs.

The instance ID is a GUID that uniquely identifies a PSSession in the current session. The instance ID is unique, even when you have multiple sessions running on a single computer.

The instance ID is stored in the InstanceID property of the object that represents a PSSession. To find the InstanceID of the PSSessions in the current session, type `Get-PSSession | Format-Table Name, ComputerName, InstanceId`.

`-Name <System.String[]>`

Specifies an array of friendly names of sessions. This cmdlet closes the PSSessions that have the specified friendly names. Wildcard characters are permitted.

Because the friendly name of a PSSession might not be unique, when you use the Name parameter, consider also using the WhatIf or Confirm parameter in the ``Remove-PSSession`` command.

`-Session <System.Management.Automation.Runspaces.PSSession[]>`

Specifies the session objects of the PSSessions to close. Enter a variable that contains the PSSessions or a command that creates or gets the PSSessions , such as a ``New-PSSession`` or ``Get-PSSession`` command. You can also pipe one or more session objects to ``Remove-PSSession``.

`-VMId <System.Guid[]>`

Specifies an array of ID of virtual machines. This cmdlet starts an interactive session with each of the specified virtual machines. To see the virtual machines that are available to you, use the following command:

```
`Get-VM | Select-Object -Property Name, ID`
```

`-VMName <System.String[]>`

Specifies an array of names of virtual machines. This cmdlet starts an interactive session with each of the specified virtual machines. To see the virtual machines that are available to you, use the ``Get-VM`` 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 \(https://go.microsoft.com/fwlink/?LinkID=113216\)](https://go.microsoft.com/fwlink/?LinkID=113216).

----- Example 1: Remove sessions by ID -----

```
Remove-PSSession -Id 1, 2
```

This command removes the PSSessions that have IDs `1` and `2`.

-- Example 2: Remove all the sessions in the current session --

```
Get-PSSession | Remove-PSSession
```

```
Remove-PSSession -Session (Get-PSSession)
```

```
$s = Get-PSSession
```

```
Remove-PSSession -Session $s
```

These commands remove all of the PSSessions in the current session. Although the three command formats look different, they have the same effect.

----- Example 3: Close sessions by name -----

```
$r = Get-PSSession -ComputerName Serv*
```

```
$r | Remove-PSSession
```

These commands close the PSSessions that are connected to computers that have names that begin with `Serv`.

----- Example 4: Close sessions connected to a port -----

```
Get-PSSession | where {$_.port -eq 90} | Remove-PSSession
```

This command closes the PSSessions that are connected to port 90. You can use this command format to identify PSSessions by properties other than

ComputerName , Name , InstanceID , and ID .

----- Example 5: Close a session by instance ID -----

```
Get-PSSession | Format-Table ComputerName, InstanceID -AutoSize
```

```
ComputerName InstanceId
```

```
-----
```

```
Server01 875d231b-2788-4f36-9f67-2e50d63bb82a
```

```
localhost c065ffa0-02c4-406e-84a3-dacb0d677868
```

```
Server02 4699cdbbc-61d5-4e0d-b916-84f82ebede1f
```

```
Server03 4e5a3245-4c63-43e4-88d0-a7798bfc2414
```

```
TX-TEST-01 fc4e9dfa-f246-452d-9fa3-1adbdd64ae85
```

```
Remove-PSSession -InstanceID fc4e9dfa-f246-452d-9fa3-1adbdd64ae85
```

These commands show how to close a PSSession based on its instance ID, or RemoteRunspaceID .

The first command uses the `Get-PSSession` cmdlet to get the PSSessions in the current session. It uses a pipeline operator (`|`) to send the PSSessions to the `Format-Table` cmdlet, which formats their ComputerName and InstanceID properties in a table. The `AutoSize` parameter compresses the columns for display.

From the resulting display, you can identify the PSSession to be closed, and copy and paste the InstanceID of that PSSession into the second command.

The second command uses the `Remove-PSSession` cmdlet to remove the PSSession with the specified instance ID.

Example 6: Create a function that deletes all sessions in the current session

```
Function EndPSS { Get-PSSession | Remove-PSSession }
```

This function closes every PSSession in the current session. After you add this function to your PowerShell profile, to delete all sessions, type ``EndPSS``.

REMARKS

To see the examples, type: "get-help Remove-PSSession -examples".

For more information, type: "get-help Remove-PSSession -detailed".

For technical information, type: "get-help Remove-PSSession -full".

For online help, type: "get-help Remove-PSSession -online"