Windows PowerShell Get-Help on Cmdlet 'Remove-PSSession'

PS:\>Get-HELP Remove-PSSession -Full

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 (`.`).

      Required?   true

      Position?   0

      Default value   None

      Accept pipeline input?   True (ByPropertyName)

      Accept wildcard characters? true

-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.

Required?   true

Position?   named

Default value   None

Accept pipeline input?   True (ByPropertyName)

Accept wildcard characters? false

-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.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByPropertyName)

Accept wildcard characters? false

-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`.

Required?   true

Position?   named

Default value   None

Accept pipeline input?   True (ByPropertyName)

Accept wildcard characters? false

-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.

Required?   true

Position?   named

Default value   None

Accept pipeline input?   True (ByPropertyName)

Accept wildcard characters? true

-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`.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   True (ByPropertyName, ByValue)

Accept wildcard characters? false

-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`

  Required?   true

  Position?   named

  Default value   None

  Accept pipeline input?   True (ByPropertyName)

  Accept wildcard characters? false

-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.

  Required?   true

  Position?   named

  Default value   None

  Accept pipeline input?   True (ByPropertyName)

  Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

  Prompts you for confirmation before running the cmdlet.

  Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

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

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

<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).

INPUTS

System.Management.Automation.Runspaces.PSSession

You can pipe a session object to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

Windows PowerShell includes the following aliases for `Remove-PSSession`:

- `rsn`

- The Id parameter is mandatory. To delete all the PSSessions in the current session, type   `Get-PSSession | Remove-PSSession`. - A PSSession uses a persistent

connection to a remote computer. Create a PSSession to run a   series of commands that share data. For more information, type `Get-Help about_PSSessions`. -

PSSessions are specific to the current session. When you end a session, the PSSessions that you created in that session are forcibly closed.

--------------- 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   4699cdbc-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`.

RELATED LINKS

Connect-PSSession

Disconnect-PSSession

Enter-PSSession

Exit-PSSession

Get-PSSession

Invoke-Command

New-PSSession

Receive-PSSession

about_PSSessions

about_Remote