



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 'Restart-Computer'

PS C:\Users\wahid> Get-Help Restart-Computer

NAME

Restart-Computer

SYNOPSIS

Restarts the operating system on local and remote computers.

SYNTAX

```
Restart-Computer [[-ComputerName] <System.String[]>] [[-Credential]
<System.Management.Automation.PSCredential>] [-AsJob] [-DcomAuthentication
{Default | None | Connect | Call | Packet | PacketIntegrity | PacketPrivacy |
Unchanged}] [-Force] [-Impersonation {Default | Anonymous | Identify |
Impersonate | Delegate}] [-ThrottleLimit <System.Int32>] [-Confirm] [-WhatIf]
[<CommonParameters>]
```

```
Restart-Computer [[-ComputerName] <System.String[]>] [[-Credential]
<System.Management.Automation.PSCredential>] [-DcomAuthentication {Default |
None | Connect | Call | Packet | PacketIntegrity | PacketPrivacy | Unchanged}]
[-Delay <System.Int16>] [-For {Wmi | WinRM | PowerShell}] [-Force]
[-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}]
[-Protocol {DCOM | WSMAN}] [-Timeout <System.Int32>] [-Wait]
```

`[-WsmanAuthentication {Basic | CredSSP | Default | Digest | Kerberos | Negotiate}] [-Confirm] [-WhatIf] [<CommonParameters>]`

DESCRIPTION

The ``Restart-Computer`` cmdlet restarts the operating system on the local and remote computers.

You can use the parameters of ``Restart-Computer`` to run the restart operations as a background job, to specify the authentication levels and alternate credentials, to limit the operations that run at the same time, and to force an immediate restart.

Starting in Windows PowerShell 3.0, you can wait for the restart to complete before you run the next command. Specify a waiting time-out and query interval, and wait for particular services to be available on the restarted computer. This feature makes it practical to use ``Restart-Computer`` in scripts and functions.

You can use the WS-Management (WSMan) protocol to restart the computer, in case Distributed Component Object Model (DCOM) calls are blocked, such as by an enterprise firewall. For more information, see [WS-Management Protocol \(/windows/desktop/WinRM/ws-management-protocol\)](/windows/desktop/WinRM/ws-management-protocol).

This cmdlet requires Windows PowerShell remoting only when you use the `AsJob` parameter in a command.

PARAMETERS

`-AsJob <System.Management.Automation.SwitchParameter>`

Indicates that ``Restart-Computer`` runs as a background job.

To use this parameter, the local and remote computers must be configured

for remoting. On Windows Vista and later versions of the Windows operating system, you must open PowerShell by using the Run as Administrator option. For more information, see `about_Remote_Requirements` (`../Microsoft.PowerShell.Core/About/about_Remote_Requirements.md`).

When you specify the `AsJob` parameter, the command immediately returns an object that represents the background job. You can continue to work in the session while the job finishes. The job is created on the local computer and the results from remote computers are automatically returned to the local computer. To manage the job, use the `Job` cmdlets. To get the job results, use the ``Receive-Job`` cmdlet.

For more information about Windows PowerShell background jobs, see `about_Jobs` (`../Microsoft.PowerShell.Core/About/about_Jobs.md`) and `about_Remote_Jobs` (`../Microsoft.PowerShell.Core/About/about_Remote_Jobs.md`).

`-ComputerName <System.String[]>`

Specifies one computer name or a comma-separated array of computer names. ``Restart-Computer`` accepts `ComputerName` objects from the pipeline or variables.

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

This parameter doesn't rely on PowerShell remoting. You can use the `ComputerName` parameter even if your computer isn't configured to run remote commands.

If the `ComputerName` parameter isn't specified, ``Restart-Computer`` restarts the local computer.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to do this action. The default is the current user.

Type a user name, such as User01 or Domain01\User01 , or enter a PSCredential object generated by the `Get-Credential` cmdlet. If you type a user name, you're prompted to enter the password.

Credentials are stored in a PSCredential

(/dotnet/api/system.management.automation.pscredential)object and the password is stored as a SecureString

(/dotnet/api/system.security.securestring).

> [!NOTE] > For more information about SecureString data protection, see >

How secure is SecureString?

(/dotnet/api/system.security.securestring#how-secure-is-securestring).

-DcomAuthentication <System.Management.AuthenticationLevel>

Specifies the authentication level that is used for the WMI connection.

`Restart-Computer` uses WMI.

Valid values are:

- Call : Call-level COM authentication - Connect :

Connect-level COM authentication - Default : Windows

Authentication - None : No COM authentication - Packet :

Packet-level COM authentication. - PacketIntegrity : Packet

Integrity-level COM authentication - PacketPrivacy : Packet

Privacy-level COM authentication. - Unchanged : The authentication level is the same as the previous command.

For more information, see AuthenticationLevel Enumeration

(/dotnet/api/system.management.authenticationlevel).

This parameter is introduced in Windows PowerShell 3.0.

-Delay <System.Int16>

Specifies the frequency of queries, in seconds. PowerShell queries the service specified by the For parameter to determine whether the service is available after the computer is restarted.

This parameter is valid only together with the Wait and For parameters.

This parameter was introduced in Windows PowerShell 3.0.

If the Delay parameter isn't specified, `Restart-Computer` uses a five second delay.

-For <Microsoft.PowerShell.Commands.WaitForServiceTypes>

Specifies the behavior of PowerShell as it waits for the specified service or feature to become available after the computer restarts. This parameter is only valid with the Wait parameter.

The acceptable values for this parameter are:

- Default : Waits for PowerShell to restart. - PowerShell : Can run commands in a PowerShell remote session on the computer. - WMI : Receives a reply to a Win32_ComputerSystem query for the computer. - WinRM : Can establish a remote session to the computer by using WS-Management.

This parameter was introduced in Windows PowerShell 3.0.

-Force <System.Management.Automation.SwitchParameter>

Forces an immediate restart of the computer.

-Impersonation <System.Management.ImpersonationLevel>

Specifies the impersonation level that this cmdlet uses to call WMI.

`Restart-Computer` uses WMI. The acceptable values for this parameter are:

- Default : Default impersonation. Despite the name, this isn't the default value. - Anonymous : Hides the identity of the caller. - Identify : Allows objects to query the credentials of the caller. - Impersonate : Allows objects to use the credentials of the caller.

-Protocol <System.String>

Specifies which protocol to use to restart the computers. The valid values are WSMAN and DCOM .

This parameter is introduced in Windows PowerShell 3.0.

-ThrottleLimit <System.Int32>

Specifies the maximum number of concurrent connections that can be established to run this command. The throttle limit applies only to the current command, not to the session or to the computer.

If the ThrottleLimit parameter isn't specified or a value of 0 is used, `Restart-Computer` uses a maximum of 32 concurrent connections.

-Timeout <System.Int32>

Specifies the duration of the wait, in seconds. When the timeout elapses, `Restart-Computer` returns to the command prompt, even if the computers aren't restarted.

The Timeout parameter is only valid with the Wait parameter. Timeout overrides the Wait parameter's indefinite waiting period.

This parameter was introduced in Windows PowerShell 3.0.

-Wait <System.Management.Automation.SwitchParameter>

``Restart-Computer`` suppresses the PowerShell prompt and blocks the pipeline until the computers have restarted. You can use this parameter in a script to restart computers and then continue to process when the restart is finished.

The `Wait` parameter waits indefinitely for the computers to restart. You can use `Timeout` to adjust the timing and the `For` and `Delay` parameters to wait for particular services to become available on the restarted computers.

The `Wait` parameter isn't valid when you're restarting the local computer. If the value of the `ComputerName` parameter contains the names of remote computers and the local computer, ``Restart-Computer`` generates a non-terminating error for `Wait` on the local computer, but waits for the remote computers to restart.

This parameter was introduced in Windows PowerShell 3.0.

`-WsmanAuthentication <System.String>`

Specifies the mechanism that is used to authenticate the user credentials.

This parameter was introduced in Windows PowerShell 3.0.

The acceptable values for this parameter are: `Basic` , `CredSSP` , `Default` , `Digest` , `Kerberos` , and `Negotiate` .

For more information, see `AuthenticationMechanism` (</dotnet/api/system.management.automation.runspaces.authenticationmechanism>).

> **[!WARNING]** > Credential Security Service Provider (CredSSP) authentication, in which the user credentials are > passed to a remote computer to be authenticated, is designed for commands that require > authentication on more than one resource, such as accessing a remote network share. This mechanism > increases the security risk of the remote

operation. If the remote computer is compromised, the > credentials that are passed to it can be used to control the network session.

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running `Restart-Computer`.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the `Restart-Computer` runs. The `Restart-Computer` cmdlet isn't 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: Restart the local computer -----

```
Restart-Computer
```

----- Example 2: Restart multiple computers -----

```
Restart-Computer -ComputerName Server01, Server02, localhost
```

----- Example 3: Restart computers as a background job -----

```
$Job = Restart-Computer -ComputerName "Server01", "Server02" -AsJob
```

```
$Job | Receive-Job
```

`Restart-Computer` uses the ComputerName parameter to specify Server01 and Server02 . The AsJob parameter runs the command as a background job. The job

object is stored in the `\$Job` variable. `\$Job` is sent down the pipeline to the `Receive-Job` cmdlet that gets the results.

----- Example 4: Restart a remote computer -----

```
Restart-Computer -ComputerName Server01 -Impersonation Anonymous  
-DcomAuthentication PacketIntegrity
```

`Restart-Computer` uses the ComputerName parameter to specify Server01 . The Impersonation parameter specifies Anonymous to hide the requester's identity. The DcomAuthentication parameter specifies PacketIntegrity as the connection's authentication level.

- Example 5: Force restart of computers listed in a text file -

```
$Names = Get-Content -Path C:\Domain01.txt  
$Creds = Get-Credential  
Restart-Computer -ComputerName $Names -Credential $Creds -Force -ThrottleLimit  
10
```

`Get-Content` uses the Path parameter to get a list of computer names from a text file, Domain01.txt . The computer names are stored in the variable `\$Names`. `Get-Credential` prompts you for a username and password and stores the values in the variable `\$Creds`. `Restart-Computer` uses the ComputerName and Credential parameters with their variables. The Force parameter causes an immediate restart of each computer. The ThrottleLimit parameter limits the command to 10 concurrent connections.

- Example 6: Restart a remote computer and wait for PowerShell -

```
Restart-Computer -ComputerName Server01 -Wait -For PowerShell -Timeout 300  
-Delay 2
```

`Restart-Computer` uses the ComputerName parameter to specify Server01 . The Wait parameter waits for the restart to finish. The For specifies that PowerShell can run commands on the remote computer. The Timeout parameter

specifies a five-minute wait. The Delay parameter queries the remote computer every two seconds to determine whether it's restarted.

-- Example 7: Restart a computer by using the WSMAN Protocol --

```
Restart-Computer -ComputerName Server01 -Protocol WSMAN -WsmanAuthentication  
Kerberos
```

`Restart-Computer` uses the ComputerName parameter to specify the remote computer, Server01 . The Protocol parameter specifies to use the WSMAN protocol. The WsmanAuthentication parameter specifies the authentication method as Kerberos .

REMARKS

To see the examples, type: "get-help Restart-Computer -examples".

For more information, type: "get-help Restart-Computer -detailed".

For technical information, type: "get-help Restart-Computer -full".

For online help, type: "get-help Restart-Computer -online"