



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-WmiObject'

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

NAME

Remove-WmiObject

SYNOPSIS

Deletes an instance of an existing Windows Management Instrumentation (WMI) class.

SYNTAX

```
Remove-WmiObject [-Class] <System.String> [-AsJob] [-Authentication {Default | None | Connect | Call | Packet | PacketIntegrity | PacketPrivacy | Unchanged}] [-Authority <System.String>] [-ComputerName <System.String[]>] [-Credential <System.Management.Automation.PSCredential>] [-EnableAllPrivileges] [-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}] [-Locale <System.String>] [-Namespace <System.String>] [-ThrottleLimit <System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]
```

```
Remove-WmiObject [-AsJob] [-Authentication {Default | None | Connect | Call | Packet | PacketIntegrity | PacketPrivacy | Unchanged}] [-Authority <System.String>] [-ComputerName <System.String[]>] [-Credential <System.Management.Automation.PSCredential>] [-EnableAllPrivileges]
```

[-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}]
[-Locale <System.String>] [-Namespace <System.String>] -Path <System.String>
[-ThrottleLimit <System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

Remove-WmiObject [-AsJob] [-Authentication {Default | None | Connect | Call |
Packet | PacketIntegrity | PacketPrivacy | Unchanged}] [-Authority
<System.String>] [-ComputerName <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-EnableAllPrivileges]
[-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}]
[-Locale <System.String>] [-Namespace <System.String>] [-ThrottleLimit
<System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

Remove-WmiObject [-AsJob] [-Authentication {Default | None | Connect | Call |
Packet | PacketIntegrity | PacketPrivacy | Unchanged}] [-Authority
<System.String>] [-ComputerName <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-EnableAllPrivileges]
[-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}]
[-Locale <System.String>] [-Namespace <System.String>] [-ThrottleLimit
<System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

Remove-WmiObject [-AsJob] [-Authentication {Default | None | Connect | Call |
Packet | PacketIntegrity | PacketPrivacy | Unchanged}] [-Authority
<System.String>] [-ComputerName <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-EnableAllPrivileges]
[-Impersonation {Default | Anonymous | Identify | Impersonate | Delegate}]
[-Locale <System.String>] [-Namespace <System.String>] [-ThrottleLimit
<System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

Remove-WmiObject [-AsJob] -InputObject <System.Management.ManagementObject>
[-ThrottleLimit <System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

The ``Remove-WmiObject`` cmdlet deletes an instance of an existing Windows Management Instrumentation (WMI) class.

PARAMETERS

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

Indicates that this cmdlet run as a background job. Use this parameter to run commands that take a long time to finish.

New CIM cmdlets, introduced Windows PowerShell 3.0, perform the same tasks as the WMI cmdlets. The CIM cmdlets comply with WS-Management (WSMan) standards and with the Common Information Model (CIM) standard, which enables the cmdlets to use the same techniques to manage computers that run the Windows operating system and those running other operating systems. Instead of using ``Remove-WmiObject``, consider using the `Remove-CimInstance` cmdlet.

When you use the `AsJob` parameter, the command returns an object that represents the background job and then displays the command prompt. You can continue to work in the session while the job finishes. If ``Remove-WmiObject`` is used against a remote computer, 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 cmdlets that contain the `Job` noun (the `Job` cmdlets). To get the job results, use the ``Receive-Job`` cmdlet.

To use this parameter for remote computers, the local and remote computers must be configured for remoting. Start Windows PowerShell by using the `Run as administrator` option. For more information, see `about_Remote_Requirements` (`../Microsoft.PowerShell.Core/About/about_Remote_Requirements.md`).

about_Jobs (../Microsoft.PowerShell.Core/About/about_Jobs.md)and
about_Remote_Jobs.

-Authentication <System.Management.AuthenticationLevel>

Specifies the authentication level to use for the WMI connection. The acceptable values for this parameter are:

- `1`: Unchanged.

- `0`: Default.

- `1`: None.

No authentication is performed. - `2`: Connect. Authentication is performed only when the client establishes a relationship with the application. - `3`: Call. Authentication is performed only at the start of each call when the application receives the request. - `4`: Packet. Authentication is performed on all the data that is received from the client. - `5`: PacketIntegrity. All the data that is transferred between the client and the application is authenticated and verified. - `6`: PacketPrivacy. The properties of the other authentication levels are used, and all the data is encrypted.

-Authority <System.String>

Specifies the authority to use to authenticate the WMI connection. You can specify standard NTLM or Kerberos authentication. To use NTLM, set the authority setting to ntlmdomain:<DomainName>, where <DomainName> identifies a valid NTLM domain name. To use Kerberos, specify kerberos:<DomainName>\<ServerName>. You cannot include the authority setting when you connect to the local computer.

-Class <System.String>

Specifies the name of a WMI class that this cmdlet deletes.

-ComputerName <System.String[]>

Specifies the name of the computer on which this cmdlet runs. The default is the local computer.

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

This parameter does not rely on Windows PowerShell remoting. You can use the ComputerName parameter even if your computer is not configured to run remote commands.

-Credential <System.Management.Automation.PSCredential>

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

Type a user name, such as User01 or Domain01\User01, or enter a PSCredential object, such as one generated by the `Get-Credential` cmdlet. If you type a user name, this cmdlet prompts you for a password.

-EnableAllPrivileges <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet enables all the permissions of the current user before the command it makes the WMI call.

-Impersonation <System.Management.Automation.ImpersonationLevel>

Specifies the impersonation level to use. The acceptable values for this parameter are:

- `0`: Default. Reads the local registry for the default impersonation level, which is usually set to 3: Impersonate. - `1`: Anonymous. Hides the credentials of the caller. - `2`: Identify. Allows objects to query the credentials of the caller. - `3`: Impersonate. Allows objects to use

the credentials of the caller. - `4`: Delegate. Allows objects to permit other objects to use the credentials of the caller.

-InputObject <System.Management.ManagementObject>

Specifies a ManagementObject object to use as input. When this parameter is used, all other parameters are ignored.

-Locale <System.String>

Specifies the preferred locale for WMI objects. The Locale parameter is specified as an array in the MS_<LCID> format in the preferred order.

-Namespace <System.String>

Specifies the WMI repository namespace where the referenced WMI class is located when it is used with the Class parameter.

-Path <System.String>

Specifies the WMI object path of a WMI class, or specifies the WMI object path of an instance of a WMI class to delete.

-ThrottleLimit <System.Int32>

Specifies the maximum number of concurrent connections that can be established to run this command. This parameter is used together with the AsJob parameter. The throttle limit applies only to the current command, not to the session or to the computer.

-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: Close all instances of a Win32 process -----

```
notepad  
  
$np = Get-WmiObject -Query "select * from win32_process where  
name='notepad.exe'"  
  
$np | Remove-WmiObject
```

This example closes all the instances of Notepad.exe.

The first command starts an instance of Notepad.

The second command uses the Get-WmiObject cmdlet to retrieve the instances of the Win32_Process that correspond to Notepad.exe, and then stores them in the `\$np` variable.

The third command passes the object in the \$np variable to `Remove-WmiObject`, which deletes all the instances of Notepad.exe.

----- Example 2: Delete a folder -----

```
$a = Get-WMIObject -Query "Select * From Win32_Directory Where Name  
='C:\\Test'"  
  
$a | Remove-WMIObject
```

The first command uses `Get-WMIObject` to query for the `C:\Test` folder, and then stores the object in the `\$a` variable.

The second command pipes the `\$a` variable to `Remove-WMIObject`, which deletes the folder.

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

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

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

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