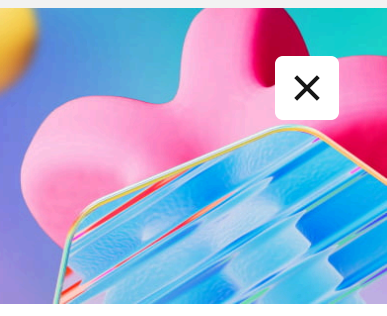

Get-WmiObject

Reference

Feedback

Module: Microsoft.PowerShell.Management

In this article

Gets instances of Windows Management Instrumentation (WMI) classes or information about the available classes.

Syntax

PowerShell Copy

```
Get-WmiObject
   [-Class] <String>
   [[-Property] <String[]>]
   [-Filter <String>]
   [-Amended]
   [-DirectRead]
   [-AsJob]
   [-Impersonation <ImpersonationLevel>]
   [-Authentication <AuthenticationLevel>]
   [-Locale <String>]
   [-EnableAllPrivileges]
   [-Authority <String>]
   [-Credential <PSCredential>]
   [-ThrottleLimit <Int32>]
   [-ComputerName <String[]>]
   [-Namespace <String>]
   [<CommonParameters>]
```

PowerShell Copy

```
Get-WmiObject
   [[-Class] <String>]
   [-Recurse]
   [-Amended]
   [-List]
```

Download PDF

```
    [-AsJob]
    [-Impersonation <ImpersonationLevel>]
    [-Authentication <AuthenticationLevel>]
    [-Locale <String>]
    [-EnableAllPrivileges]
    [-Authority <String>]
    [-Credential <PSCredential>]
    [-ThrottleLimit <Int32>]
    [-ComputerName <String[]>]
    [-Namespace <String>]
    [<CommonParameters>]
```

PowerShell Copy

```
Get-WmiObject
    [-Amended]
    [-DirectRead]
    -Query <String>
    [-AsJob]
    [-Impersonation <ImpersonationLevel>]
    [-Authentication <AuthenticationLevel>]
    [-Locale <String>]
    [-EnableAllPrivileges]
    [-Authority <String>]
    [-Credential <PSCredential>]
    [-ThrottleLimit <Int32>]
    [-ComputerName <String[]>]
    [-Namespace <String>]
    [<CommonParameters>]
```

PowerShell Copy

```
Get-WmiObject
    [-Amended]
    [-AsJob]
    [-Impersonation <ImpersonationLevel>]
    [-Authentication <AuthenticationLevel>]
    [-Locale <String>]
    [-EnableAllPrivileges]
    [-Authority <String>]
    [-Credential <PSCredential>]
    [-ThrottleLimit <Int32>]
    [-ComputerName <String[]>]
    [-Namespace <String>]
    [<CommonParameters>]
```

PowerShell Copy

```
Get-WmiObject
    [-Amended]
    [-AsJob]
    [-Impersonation <ImpersonationLevel>]
    [-Authentication <AuthenticationLevel>]
    [-Locale <String>]
    [-EnableAllPrivileges]
    [-Authority <String>]
    [-Credential <PSCredential>]
    [-ThrottleLimit <Int32>]
    [-ComputerName <String[]>]
    [-Namespace <String>]
    [<CommonParameters>]
```

Description

Starting in PowerShell 3.0, this cmdlet has been superseded by `Get-CimInstance`.

The `Get-WmiObject` cmdlet gets instances of WMI classes or information about the available WMI classes. To specify a remote computer, use the **ComputerName** parameter. If the **List** parameter is specified, the cmdlet gets information about the WMI classes that are available in

a specified namespace. If the **Query** parameter is specified, the cmdlet runs a WMI query language (WQL) statement.

The `Get-WmiObject` cmdlet does not use Windows PowerShell remoting to perform remote operations. You can use the **ComputerName** parameter of the `Get-WmiObject` cmdlet even if your computer does not meet the requirements for Windows PowerShell remoting or is not configured for remoting in Windows PowerShell.

Beginning in Windows PowerShell 3.0, the **__Server** property of the object that `Get-WmiObject` returns has a **PSComputerName** alias. This makes it easier to include the source computer name in output and reports.

Examples

Example 1: Get processes on the local computer

This example get the processes on the local computer.

PowerShell Copy

```
Get-WmiObject -Class Win32_Process
```

Example 2: Gets services on a remote computer

This example gets the services on a remote computer. The **ComputerName** parameter specifies the IP address of a remote computer. By default, the current user account must be a member of the **Administrators** group on the remote computer.

PowerShell Copy

```
Get-WmiObject -Class Win32_Service -ComputerName 10.1.4.62
```

Example 3: Get WMI classes in the root or default namespace of the local computer

This example gets the WMI classes in the root or default namespace of the local computer.

PowerShell Copy

```
Get-WmiObject -Namespace "root/default" -List
```

Example 4: Get a named service on multiple computers

This example gets the WinRM service on the computers specified by the value of the **ComputerName** parameter.

PowerShell Copy

```
Get-WmiObject -Query "select * from win32_service where name='WinRM'" -ComputerN
  Format-List -Property PSComputerName, Name, ExitCode, Name, ProcessID, StartMo

PSComputerName : SERVER01
Name           : WinRM
ExitCode       : 0
Name           : WinRM
ProcessID      : 844
StartMode      : Auto
State          : Running
```

```
Status         : OK

PSComputerName : SERVER02
Name           : WinRM
ExitCode       : 0
Name           : WinRM
ProcessID      : 932
StartMode      : Auto
State          : Running
Status         : OK
```

A pipeline operator (`|`) sends the output to the `Format-List` cmdlet, which adds the **PSComputerName** property to the default output. **PSComputerName** is an alias of the **__Server** property of the objects that `Get-WmiObject` returns. This alias was introduced in PowerShell 3.0.

Example 5: Stop a service on a remote computer

This example stops the WinRM service on a remote computer. `Get-WmiObject` gets the instance of the WinRM service object on Server01. Then, it invokes the **StopService** method of the **Win32_Service** WMI class on that object.

PowerShell Copy

```
(Get-WmiObject -Class Win32_Service -Filter "name='WinRM'" -ComputerName Server0
```

This is equivalent to using the `Stop-Service` cmdlet.

Example 6: Get the BIOS on the local computer

This example gets the BIOS information from the local computer. The **Property** parameter of the `Format-List` cmdlet is used to display all properties of the returned object in a list. By default, only the subset of properties defined in the `Types.ps1xml` configuration file are displayed.

PowerShell Copy

```
Get-WmiObject -Class Win32_Bios | Format-List -Property *

Status                : OK
Name                  : Phoenix ROM BIOS PLUS Version 1.10 A05
Caption               : Phoenix ROM BIOS PLUS Version 1.10 A05
SMBIOSPresent         : True
__GENUS               : 2
__CLASS               : Win32_BIOS
__SUPERCLASS          : CIM_BIOSElement
__DYNASTY             : CIM_ManagedSystemElement
__RELPATH             : Win32_BIOS.Name="Phoenix ROM BIOS PLUS Version 1.10
__PROPERTY_COUNT      : 27
__DERIVATION          : {CIM_BIOSElement, CIM_SoftwareElement, CIM_LogicalElemer
__SERVER              : Server01
__NAMESPACE           : root\cimv2
__PATH                : \\SERVER01\root\cimv2:Win32_BIOS.Name="Phoenix ROM BIOS
BiosCharacteristics   : {7, 9, 10, 11...}
BIOSVersion           : {DELL   - 15, Phoenix ROM BIOS PLUS Version 1.10 A05}
BuildNumber           :
CodeSet               :
CurrentLanguage       : en|US|iso8859-1
Description           : Phoenix ROM BIOS PLUS Version 1.10 A05
IdentificationCode    :
InstallableLanguages  : 1
InstallDate           :
LanguageEdition       :
ListOfLanguages       : {en|US|iso8859-1}
Manufacturer          : Dell Inc.
OtherTargetOS         :
PrimaryBIOS           : True
ReleaseDate           : 20101103000000.000000+000
```

```
SerialNumber          : 8VDM9P1
SMBIOSBIOSVersion     : A05
SMBIOSMajorVersion    : 2
SMBIOSMinorVersion    : 6
SoftwareElementID     : Phoenix ROM BIOS PLUS Version 1.10 A05
SoftwareElementState  : 3
TargetOperatingSystem : 0
Version               : DELL   - 15
Scope                 : System.Management.ManagementScope
Path                  : \\SERVER01\root\cimv2:Win32_BIOS.Name="Phoenix ROM BIOS
Options               : System.Management.ObjectGetOptions
ClassPath             : \\JUNE-PC\root\cimv2:Win32_BIOS
Properties            : {BiosCharacteristics, BIOSVersion, BuildNumber, Caption.
SystemProperties      : {__GENUS, __CLASS, __SUPERCLASS, __DYNASTY...}
Qualifiers            : {dynamic, Locale, provider, UUID}
Site                  :
Container             :
```

Example 7: Get the services on a remote computer

This example uses the **Credential** parameter of the `Get-WmiObject` cmdlet to get the services on a remote computer. The value of the **Credential** parameter is a user account name. The user is prompted for a password.

PowerShell Copy

```
Get-WmiObject Win32_Service -Credential FABRIKAM\administrator -ComputerName Fab
```

> **Note**
>
> Credentials cannot be used when targeting the local computer.

Parameters

-Amended

Gets or sets a value that indicates whether the objects that are returned from WMI should contain amended information. Typically, amended information is localizable information, such as object and property descriptions, that is attached to the WMI object.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-AsJob

Runs the command as a background job. Use this parameter to run commands that take a long time to finish.

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 `Get-WmiObject` is used with the **ComputerName** parameter, 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. To get the job results, use the `Receive-Job` cmdlet.

For more information about Windows PowerShell background jobs, see about_Jobs and about_Remote_Jobs.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Authentication

Specifies the authentication level to be used with the WMI connection. Valid values are:

- `-1`: **Unchanged**
- `0`: **Default**
- `1`: **None** (No authentication in performed.)
- `2`: **Connect** (Authentication is performed only when the client establishes a relationship with the application.)
- `3`: **Call** (Authentication is performed only at the beginning 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.)

Expand table

Type:	AuthenticationLevel
Accepted values:	Default, None, Connect, Call, Packet, PacketIntegrity, PacketPrivacy, Unchanged
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Authority

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.

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Class

Specifies the name of a WMI class. When this parameter is used, the cmdlet retrieves instances of the WMI class.

Expand table

Type:	String
Aliases:	ClassName
Position:	1
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-ComputerName

Specifies the target computer for the management operation. Enter a fully qualified domain name (FQDN), a NetBIOS name, or an IP address. When the remote computer is in a different domain than the local computer, the fully qualified domain name is required.

The default is the local computer. To specify the local computer, such as in a list of computer names, use `localhost`, the local computer name, or a dot (`.`).

When specifying a remote computer, your current account or the one you specify with the **Credential** parameter must have appropriate permissions to access the information.

This parameter does not rely on Windows PowerShell remoting, which uses WS-Management. You can use the **ComputerName** parameter of `Get-WmiObject` even if your computer is not configured to run WS-Management remote commands.

Expand table

Type:	String[]
Aliases:	Cn
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Credential

Specifies a user account that has permission to perform this action. The default is the current user. Type a user name, such as `User01`, `Domain01\User01`, or `User@Contoso.com`. Or, enter a **PSCredential** object, such as an object that is returned by the `Get-Credential` cmdlet. When you type a user name, you are prompted for a password. Credentials cannot be used when targeting the local computer.

Expand table

Type:	PSCredential
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-DirectRead

Specifies whether direct access to the WMI provider is requested for the specified class without any regard to its base class or to its derived classes.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-EnableAllPrivileges

Enables all the privileges of the current user before the command makes the WMI call.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Filter

Specifies a **Where** clause to use as a filter. Uses the syntax of the WMI Query Language (WQL).

> **Important**
>
> Do not include the **Where** keyword in the value of the parameter. For example, the following commands return only the logical disks that have a **DeviceID** of `c:` and

services that have the name 'WinRM' without using the **Where** keyword.

```
Get-WmiObject Win32_LogicalDisk -filter "DeviceID = 'c:' "
```

```
Get-WmiObject win32_service -filter "name='WinRM'"
```

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Impersonation

Specifies the impersonation level to use.

The acceptable values for this parameter are:

- `0`: **Default**. Reads the local registry for the default impersonation level. The default is usually set to **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.

Expand table

Type:	ImpersonationLevel
Accepted values:	Default, Anonymous, Identify, Impersonate, Delegate
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-List

Gets the names of the WMI classes in the WMI repository namespace that is specified by the **Namespace** parameter.

If you specify the **List** parameter, but not the **Namespace** parameter, `Get-WmiObject` uses the **Root\Cimv2** namespace by default. This cmdlet does not use the **Default Namespace** registry entry in the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBEM\Scripting` registry key to determine the default namespace.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None

Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Locale

Specifies the preferred locale for WMI objects. Enter a value in `MS_<LCID>` format.

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Namespace

When used with the **Class** parameter, the **Namespace** parameter specifies the WMI repository namespace where the specified WMI class is located. When used with the **List** parameter, it specifies the namespace from which to gather WMI class information.

Expand table

Type:	String
Aliases:	NS
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Property

Specifies the WMI class properties that this cmdlet gets information from. Enter the property names.

Expand table

Type:	String[]
Position:	1
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Query

Runs the specified WMI Query Language (WQL) statement. This parameter does not support event queries.

Expand table

Type:	String
Position:	Named
Default value:	None
Required:	True
Accept pipeline input:	False
Accept wildcard characters:	False

-Recurse

Searches the current namespace and all other namespaces for the class name that is specified by the **Class** parameter.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-ThrottleLimit

Specifies the maximum number of WMI operations that can be executed simultaneously. This parameter is valid only when the **AsJob** parameter is used in the command.

Expand table

Type:	Int32
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

Inputs

None

You cannot pipe input to `Get-WmiObject`.

Outputs

PSObject or System.Management.Automation.RemotingJob

When you use the **AsJob** parameter, the cmdlet returns a job object. Otherwise, the object that `Get-WmiObject` returns depends on the value of the **Class** parameter.

Notes

Windows PowerShell includes the following aliases for `Get-WmiObject`:

- `gwmi`

To access WMI information on a remote computer, the cmdlet must run under an account that is a member of the local administrators group on the remote computer. Or, the default access control on the WMI namespace of the remote repository can be changed to give access rights to other accounts.

Only some of the properties of each WMI class are displayed by default. The set of properties that is displayed for each WMI class is specified in the `Types.ps1xml` configuration file. To get all properties of a WMI object, use the `Get-Member` or `Format-List` cmdlets.

Related Links

- Invoke-WmiMethod
- Remove-WmiObject
- Set-WmiInstance
- Get-WSManInstance
- Invoke-WSManAction
- New-WSManInstance
- Remove-WSManInstance