



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 'Get-CimInstance'**

**PS C:\Users\wahid> Get-Help Get-CimInstance**

#### NAME

Get-CimInstance

#### SYNOPSIS

Gets the CIM instances of a class from a CIM server.

#### SYNTAX

```
Get-CimInstance -CimSession <Microsoft.Management.Infrastructure.CimSession[]>  
[-Filter <System.String>] [-KeyOnly] [-Namespace <System.String>]  
[-OperationTimeoutSec <System.UInt32>] [-Property <System.String[]>]  
[-ResourceUri <System.Uri>] [-Shallow] [<CommonParameters>]
```

```
Get-CimInstance -CimSession <Microsoft.Management.Infrastructure.CimSession[]>  
[-Namespace <System.String>] [-OperationTimeoutSec <System.UInt32>] -Query  
<System.String> [-QueryDialect <System.String>] [-ResourceUri <System.Uri>]  
[-Shallow] [<CommonParameters>]
```

```
Get-CimInstance [-ClassName] <System.String> -CimSession  
<Microsoft.Management.Infrastructure.CimSession[]> [-Filter <System.String>]  
[-KeyOnly] [-Namespace <System.String>] [-OperationTimeoutSec <System.UInt32>]
```

[-Property <System.String[]>] [-QueryDialect <System.String>] [-Shallow]  
[<CommonParameters>]

Get-CimInstance [-InputObject]

<Microsoft.Management.Infrastructure.CimInstance> -CimSession  
<Microsoft.Management.Infrastructure.CimSession[]> [-OperationTimeoutSec  
<System.UInt32>] [-ResourceUri <System.Uri>] [<CommonParameters>]

Get-CimInstance [-ClassName] <System.String> [-ComputerName <System.String[]>]  
[-Filter <System.String>] [-KeyOnly] [-Namespace <System.String>]  
[-OperationTimeoutSec <System.UInt32>] [-Property <System.String[]>]  
[-QueryDialect <System.String>] [-Shallow] [<CommonParameters>]

Get-CimInstance [-InputObject]

<Microsoft.Management.Infrastructure.CimInstance> [-ComputerName  
<System.String[]>] [-OperationTimeoutSec <System.UInt32>] [-ResourceUri  
<System.Uri>] [<CommonParameters>]

Get-CimInstance [-ComputerName <System.String[]>] [-Filter <System.String>]  
[-KeyOnly] [-Namespace <System.String>] [-OperationTimeoutSec <System.UInt32>]  
[-Property <System.String[]>] [-ResourceUri <System.Uri>] [-Shallow]  
[<CommonParameters>]

Get-CimInstance [-ComputerName <System.String[]>] [-Namespace <System.String>]  
[-OperationTimeoutSec <System.UInt32>] -Query <System.String> [-QueryDialect  
<System.String>] [-ResourceUri <System.Uri>] [-Shallow] [<CommonParameters>]

## DESCRIPTION

The `Get-CimInstance` cmdlet gets the CIM instances of a class from a CIM server. You can specify either the class name or a query for this cmdlet. This cmdlet returns one or more CIM instance objects representing a snapshot of the CIM instances present on the CIM server.

If the `InputObject` parameter is not specified, the cmdlet works in one of the following ways:

- If neither the `ComputerName` parameter nor the `CimSession` parameter is specified, then this cmdlet works on local Windows Management Instrumentation (WMI) using a Component Object Model (COM) session.
- If either the `ComputerName` parameter or the `CimSession` parameter is specified, then this cmdlet works against the CIM server specified by either the `ComputerName` parameter or the `CimSession` parameter.

If the `InputObject` parameter is specified, the cmdlet works in one of the following ways:

- If neither the `ComputerName` parameter nor the `CimSession` parameter is specified, then this cmdlet uses the CIM session or computer name from the input object.
- If either the `ComputerName` parameter or the `CimSession` parameter is specified, then this cmdlet uses either the `CimSession` parameter value or `ComputerName` parameter value.

## PARAMETERS

`-CimSession <Microsoft.Management.Infrastructure.CimSession[]>`

Specifies the CIM session to use for this cmdlet. Enter a variable that contains the CIM session or a command that creates or gets the CIM session, such as the ``New-CimSession`` or ``Get-CimSession`` cmdlets. For more information, see `about_CimSession` (`../Microsoft.PowerShell.Core/About/about_CimSession.md`).

`-ClassName <System.String>`

Specifies the name of the CIM class for which to retrieve the CIM instances. You can use tab completion to browse the list of classes, because PowerShell gets a list of classes from the local WMI server to

provide a list of class names.

`-ComputerName <System.String[]>`

Specifies computer on which you want to run the CIM operation. You can specify a fully qualified domain name (FQDN), a NetBIOS name, or an IP address. If you do not specify this parameter, the cmdlet performs the operation on the local computer using Component Object Model (COM).

If you specify this parameter, the cmdlet creates a temporary session to the specified computer using the WsMan protocol.

If multiple operations are being performed on the same computer, connect using a CIM session for better performance.

`-Filter <System.String>`

Specifies a where clause to use as a filter. Specify the clause in either the WQL or the CQL query language. Do not include the `WHERE` keyword in the value of the parameter.

`-InputObject <Microsoft.Management.Infrastructure.CimInstance>`

Specifies a CIM instance object to use as input.

If you are already working with a CIM instance object, you can use this parameter to pass the CIM instance object in order to get the latest snapshot from the CIM server. When you pass a CIM instance object as an input, `Get-CimInstance` returns the object from server using a get CIM operation, instead of an enumerate or query operation. Using a get CIM operation is more efficient than retrieving all instances and then filtering them.

The InputObject parameter doesn't enumerate over collections. If a collection is passed, an error is thrown. When working with collections, pipe the input to enumerate the values.

If the CIM class does not implement the get operation, then specifying the InputObject parameter returns an error.

**-KeyOnly <System.Management.Automation.SwitchParameter>**

Indicates that only objects with key properties populated are returned.

Specifying the KeyOnly parameter reduces the amount of data transferred over the network.

Use the KeyOnly parameter to return only a small portion of the object, which can be used for other operations, such as the `Set-CimInstance` or `Get-CimAssociatedInstance` cmdlets.

**-Namespace <System.String>**

Specifies the namespace of CIM class.

The default namespace is root/cimv2 . You can use tab completion to browse the list of namespaces, because PowerShell gets a list of namespaces from the local WMI server to provide the list of namespaces.

**-OperationTimeoutSec <System.UInt32>**

Specifies the amount of time that the cmdlet waits for a response from the computer. By default, the value of this parameter is 0, which means that the cmdlet uses the default timeout value for the server.

If the OperationTimeoutSec parameter is set to a value less than the robust connection retry timeout of 3 minutes, network failures that last more than the value of the OperationTimeoutSec parameter are not recoverable, because the operation on the server times out before the client can reconnect.

**-Property <System.String[]>**

Specifies a set of instance properties to retrieve. Use this parameter

when you need to reduce the size of the object returned, either in memory or over the network. The object returned also contains the key properties even if you have not listed them using the Property parameter. Other properties of the class are present but they are not populated.

#### `-Query <System.String>`

Specifies a query to run on the CIM server. If the value specified contains double quotes `"`, single quotes `'`, or a backslash `\`, you must escape those characters by prefixing them with the backslash character. If the value specified uses the WQL LIKE operator, then you must escape the following characters by enclosing them in square brackets `[]`: percent `%`, underscore `_`, or opening square bracket `[`.

You cannot use a metadata query to retrieve a list of classes or an event query. To retrieve a list of classes, use the `Get-CimClass` cmdlet. To retrieve an event query, use the `Register-CimIndicationEvent` cmdlet.

You can specify the query dialect using the QueryDialect parameter.

#### `-QueryDialect <System.String>`

Specifies the query language used for the Query parameter. The acceptable values for this parameter are: WQL or CQL . The default value is WQL .

#### `-ResourceUri <System.Uri>`

Specifies the resource uniform resource identifier (URI) of the resource class or instance. The URI is used to identify a specific type of resource, such as disks or processes, on a computer.

A URI consists of a prefix and a path to a resource. For example:

```
- `http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_LogicalDisk`
```

- `http://intel.com/wbem/wscim/1/amt-schema/1/AMT\_GeneralSettings`

By default, if you do not specify this parameter, the DMTF standard resource URI `http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/` is used and the class name is appended to it. ResourceURI can only be used with CIM sessions created using the WSMAN protocol, or when specifying the ComputerName parameter, which creates a CIM session using WSMAN. If you specify this parameter without specifying the ComputerName parameter, or if you specify a CIM session created using DCOM protocol, you will get an error, because the DCOM protocol does not support the ResourceURI parameter.

If both the ResourceUri parameter and the Filter parameter are specified, the Filter parameter is ignored.

-Shallow <System.Management.Automation.SwitchParameter>

Indicates that the instances of a class are returned without including the instances of any child classes. By default, the cmdlet returns the instances of a class and its child classes.

<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: Get the CIM instances of a specified class ----

```
Get-CimInstance -ClassName Win32_Process
```

---- Example 2: Get a list of namespaces from a WMI server ----

```
Get-CimInstance -Namespace root -ClassName __Namespace
```

Example 3: Get instances of a class filtered by using a query

```
Get-CimInstance -Query "SELECT * from Win32_Process WHERE name LIKE 'P%'"
```

Example 4: Get instances of a class filtered by using a class name and a filter expression

```
Get-CimInstance -ClassName Win32_Process -Filter "Name like 'P%'"
```

Example 5: Get the CIM instances with only key properties filled in

```
$x = New-CimInstance -ClassName Win32_Process -Namespace root\cimv2 -Property
```

```
@{ "Handle"=0 } -Key Handle -ClientOnly
```

```
Get-CimInstance -CimInstance $x
```

----- Example 6: Retrieve CIM instances and reuse them -----

```
$x,$y = Get-CimInstance -ClassName Win32_Process
```

```
$x | Format-Table -Property Name,KernelModeTime -AutoSize
```

```
Name          KernelModeTime
```

```
----
```

```
System Idle Process 157238797968750
```

----- Example 7: Get CIM instances from remote computer -----



```
Get-CimInstance -ClassName Win32_ComputerSystem -ComputerName Server01,Server02
```

Example 8: Getting only the key properties, instead of all properties

```
$x = Get-CimInstance -Class Win32_Process -KeyOnly  
$x | Invoke-CimMethod -MethodName GetOwner
```

Example 9: Getting only a subset of properties, instead of all properties

```
Get-CimInstance -Class Win32_Process -Property Name,KernelModeTime  
$x = Get-CimInstance -Class Win32_Process -Property Name,KernelModeTime  
$x | Invoke-CimMethod -MethodName GetOwner
```

The instance retrieved with the Property parameter can be used to perform other CIM operations, for example ``Set-CimInstance`` or ``Invoke-CimMethod``.

----- Example 10: Get the CIM instance using CIM session -----

```
$s = New-CimSession -ComputerName Server01,Server02  
Get-CimInstance -ClassName Win32_ComputerSystem -CimSession $s
```

## REMARKS

To see the examples, type: "get-help Get-CimInstance -examples".

For more information, type: "get-help Get-CimInstance -detailed".

For technical information, type: "get-help Get-CimInstance -full".

For online help, type: "get-help Get-CimInstance -online"