Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Get-CimClass'

PS C:\Users\wahid> Get-Help Get-CimClass

NAME

Get-CimClass

SYNOPSIS

Gets a list of CIM classes in a specific namespace.

SYNTAX

Get-CimClass [[-ClassName] <System.String>] [[-Namespace] <System.String>]

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

<System.String>] [-OperationTimeoutSec <System.UInt32>] [-PropertyName

<System.String>] [-QualifierName <System.String>] [<CommonParameters>]

Get-CimClass [[-ClassName] <System.String>] [[-Namespace] <System.String>]

[-ComputerName <System.String[]>] [-MethodName <System.String>]

[-OperationTimeoutSec <System.UInt32>] [-PropertyName <System.String>]

[-QualifierName <System.String>] [<CommonParameters>]

DESCRIPTION

The `Get-CimClass` cmdlet retrieves a list of CIM classes in a specific

namespace. If there is no class name supplied, then the cmdlet returns all the classes in the namespace. Unlike a CIM instance, CIM classes do not contain the CIM session or computer name from which they are retrieved.

PARAMETERS

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

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a `New-CimSession` or `Get-CimSession` cmdlet. The default is the current session on the local computer.

-ClassName <System.String>

Specifies the name of the CIM class for which to perform the operation. 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 the 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 specify this parameter, the cmdlet creates a temporary session to the specified computer using the WsMan protocol.

If you do not specify this parameter, the cmdlet performs the operation on the local computer using Component Object Model (COM).

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

-MethodName <System.String>

Finds the classes that have a method matching this name. You can use wildcard characters with this parameter.

-Namespace <System.String>

Specifies the namespace for CIM operation. 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.

-PropertyName <System.String>

Finds the classes which have a property matching this name. You can use wildcard characters with this parameter.

-QualifierName <System.String>

Filters the classes by class level qualifier name. You can use wildcard characters with this parameter.

<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 all the class definitions ----------

Get-CimClass

------- Example 2: Get the classes with a specific name -------

Get-CimClass -ClassName *disk*

---- Example 3: Get the classes with a specific method name ----

Get-CimClass -ClassName Win32* -MethodName Term*

--- Example 4: Get the classes with a specific property name ---

Get-CimClass -ClassName Win32* -PropertyName Handle

-- Example 5: Get the classes with a specific qualifier name --

Get-CimClass -ClassName Win32*Disk* -QualifierName Association

Example 6: Get the class definitions from a specific namespace

Get-CimClass -Namespace root/standardCimv2 -ClassName *Net*

-- Example 7: Get the class definitions from a remote server --

```
    Get-CimClass -ClassName *disk* -ComputerName Server01, Server02


    ------ Example 8: Get the classes by using a CIM session ------

    $s = New-CimSession -ComputerName Server01, Server02
    Get-CimClass -ClassName *disk* -CimSession $s

    This set of commands creates a session with multiple computers and stores it
    into a variable `$s` using the `New-CimSession` cmdlet, and then gets the
    classes using the `Get-CimClass` cmdlet.
REMARKS
    To see the examples, type: "get-help Get-CimClass -examples".
    For more information, type: "get-help Get-CimClass -detailed".
    For technical information, type: "get-help Get-CimClass -full".
    For online help, type: "get-help Get-CimClass -online"
```