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-CimInstance'

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

NAME

Remove-CimInstance

SYNOPSIS

Removes a CIM instance from a computer.

SYNTAX

Remove-CimInstance [-InputObject]

<Microsoft.Management.Infrastructure.CimInstance> -CimSession

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

<System.UInt32>] [-ResourceUri <System.Uri>] [-Confirm] [-WhatIf]

[<CommonParameters>]

Remove-CimInstance [-Query] <System.String> [[-Namespace] <System.String>]

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

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

[-Confirm] [-WhatIf] [<CommonParameters>]

Remove-CimInstance [-InputObject]

<Microsoft.Management.Infrastructure.CimInstance> [-ComputerName

<System.String[]>] [-OperationTimeoutSec <System.UInt32>] [-ResourceUri <System.Uri>] [-Confirm] [-WhatIf] [<CommonParameters>]

Remove-CimInstance [-Query] <System.String> [[-Namespace] <System.String>] [-ComputerName <System.String[]>] [-OperationTimeoutSec <System.UInt32>] [-QueryDialect <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

This cmdlet removes a CIM instance from a CIM server. You can specify the CIM instance to remove by using either a CIM instance object retrieved by the `Get-CimInstance` cmdlet, or by specifying a query.

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.

PARAMETERS

-CimSession <Microsoft.Management.Infrastructure.CimSession[]>
    Runs the command using the specified CIM session. 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).

-ComputerName <System.String[]>

Specifies the name of the computer on which you want to run the CIM operation. You can specify a fully qualified domain name (FQDN) or a NetBIOS name.

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, connecting using a CIM session gives better performance.

-InputObject <Microsoft.Management.Infrastructure.CimInstance>
Specifies a CIM instance object to be removed from the CIM server. The object passed to the cmdlet is not changed, only the instance in the CIM server is removed.

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.

-Namespace <System.String>
Specifies the namespace for the 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.

-Query <System.String>

Specifies a query to run on the CIM server. You can specify the query dialect using the QueryDialect parameter.

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 (`[`).

-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 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.

-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).

-------------- Example 1: Remove the CIM instance --------------

Remove-CimInstance -Query 'Select * from Win32_Environment where name LIKE

"testvar%"'

- Example 2: Remove the CIM instance using CIM instance object -

notepad.exe

$var = Get-CimInstance -Query 'Select * from Win32_Process where name LIKE

"notepad%"'

Remove-CimInstance -InputObject $var

REMARKS

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

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

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

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