PowerShell Get-Help on command 'Set-WmiInstance'

PS C:\Users\wahid> Get-Help Set-WmiInstance

```
NAME
    Set-WmiInstance


SYNOPSIS
    Creates or updates an instance of an existing Windows Management
    Instrumentation (WMI) class.



SYNTAX
    Set-WmiInstance [-Class] <System.String> [-Arguments
    <System.Collections.Hashtable>] [-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>] [-PutType {None |
    UpdateOnly | CreateOnly | UpdateOrCreate}] [-ThrottleLimit <System.Int32>]
    [-Confirm] [-WhatIf] [<CommonParameters>]

    Set-WmiInstance [-Arguments <System.Collections.Hashtable>] [-AsJob]
    -InputObject <System.Management.ManagementObject> [-PutType {None | UpdateOnly
| CreateOnly | UpdateOrCreate}] [-ThrottleLimit <System.Int32>] [-Confirm]
[-WhatIf] [<CommonParameters>]

Set-WmiInstance [-Arguments <System.Collections.Hashtable>] [-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> [-PutType {None | UpdateOnly |
CreateOnly | UpdateOrCreate}] [-ThrottleLimit <System.Int32>] [-Confirm]
[-WhatIf] [<CommonParameters>]

Set-WmiInstance [-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>] [-PutType {None |
UpdateOnly | CreateOnly | UpdateOrCreate}] [-ThrottleLimit <System.Int32>]
[-Confirm] [-WhatIf] [<CommonParameters>]

Set-WmiInstance [-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>] [-PutType {None |
UpdateOnly | CreateOnly | UpdateOrCreate}] [-ThrottleLimit <System.Int32>]
[-Confirm] [-WhatIf] [<CommonParameters>]

Set-WmiInstance [-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>] [-PutType {None | UpdateOnly | CreateOnly | UpdateOrCreate}] [-ThrottleLimit <System.Int32>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Set-WmiInstance` cmdlet creates or updates an instance of an existing Windows Management Instrumentation (WMI) class. The created or updated instance is written to the WMI repository.

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. this enables cmdlets to use the same techniques to manage Windows-based computers and those running other operating systems. Instead of using `Set-WmiInstance`, consider using the Set-CimInstance (/powershell/module/cimcmdlets/set-ciminstance)or New-CimInstance (/powershell/module/cimcmdlets/new-ciminstance)cmdlets.

PARAMETERS

-Arguments <System.Collections.Hashtable>

Specifies the name of the property to be changed and the new value for that property. The name and value must be a name-value pair. The name-value pair is passed on the command line as a hash table. For example:

`@{Setting1=1; Setting2=5; Setting3="test"}`

-AsJob <System.Management.Automation.SwitchParameter>

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

When you specify 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 is used for 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 together with remote computers, the local and remote computers must be configured for remoting. Additionally, you must start Windows PowerShell by using the Run as administrator option in Windows Vista and later versions of the Windows operating system. For more information, see about_Remote_Requirements (../Microsoft.PowerShell.Core/About/about_Remote_Requirements.md).

For more information about Windows PowerShell background jobs, see about_Jobs (../Microsoft.PowerShell.Core/About/about_Jobs.md) and [about_Remote_Jobs](../Microsoft.PowerShell.Core/About/about_Remote_Jobs.md).

-Authentication <System.Management.AuthenticationLevel>

Specifies the authentication level that must be used with the WMI connection. The acceptable values for this parameter 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 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.

-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 for a password.

This parameter is not supported by any providers installed with parameter is not supported by any providers installed with Windows PowerShell.

-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.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 ,except the Arguments parameter, are ignored.

-Locale <System.String>

Specifies the preferred locale for WMI objects. The Locale parameter is specified in 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 a WMI object path of the instance that you want to create or update.

-PutType <System.Management.PutType>

Indicates whether to create or update the WMI instance. The acceptable values for this parameter are:

- `UpdateOnly`   Updates an existing WMI instance. - `CreateOnly` Creates a new WMI instance. - `UpdateOrCreate`   Updates the WMI instance if it exists or creates a new instance if an instance does not exist.

-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).

--------------- Example 1: Set WMI logging level ---------------

```
Set-WmiInstance -Class Win32_WMISetting -Argument @{LoggingLevel=2}


__GENUS                        : 2
__CLASS                        : Win32_WMISetting
__SUPERCLASS                   : CIM_Setting
__DYNASTY                      : CIM_Setting
__RELPATH                      : Win32_WMISetting=@
__PROPERTY_COUNT               : 27
__DERIVATION                   : {CIM_Setting}
__SERVER                       : SYSTEM01
__NAMESPACE                    : root\cimv2
__PATH                         : \\SYSTEM01\root\cimv2:Win32_WMISetting=@
ASPScriptDefaultNamespace      : \\root\cimv2
ASPScriptEnabled               : False
AutorecoverMofs                : {%windir%\system32\wbem\cimwin32.mof,
%windir%\system32\wbem\ncprov.mof, %windir%\syst
em32\wbem\wmipcima.mof, %windir%\system32\wbem\secrcw32.mof...}
AutoStartWin9X                 :
BackupInterval                 :
BackupLastTime                 :
BuildVersion                   : 6001.18000
Caption                        :
DatabaseDirectory              : C:\Windows\system32\wbem\repository
DatabaseMaxSize                :
Description                    :
EnableAnonWin9xConnections     :
EnableEvents                   : False
EnableStartupHeapPreallocation : False
HighThresholdOnClientObjects   :
HighThresholdOnEvents          : 20000000
InstallationDirectory          : C:\Windows\system32\wbem
```

```
LastStartupHeapPreallocation   :
LoggingDirectory               : C:\Windows\system32\wbem\Logs\
LoggingLevel                   : 2
LowThresholdOnClientObjects    :
LowThresholdOnEvents           : 10000000
MaxLogFileSize                 : 65536
MaxWaitOnClientObjects         :
MaxWaitOnEvents                : 2000
MofSelfInstallDirectory        :
SettingID                      :
```

This command sets the WMI logging level to 2. The command passes the property to be set and the value, together considered a value pair, in the argument parameter. The parameter takes a hash table that is defined by the `@{property = value}` construction. The class information that is returned reflects the new value.

--- Example 2: Create an environment variable and its value ---

```
Set-WmiInstance -Class win32_environment -Argument
@{Name="testvar";VariableValue="testvalue";UserName="<SYSTEM>"}

__GENUS          : 2
__CLASS          : Win32_Environment
__SUPERCLASS     : CIM_SystemResource
__DYNASTY        : CIM_ManagedSystemElement
__RELPATH        : Win32_Environment.Name="testvar",UserName="<SYSTEM>"
__PROPERTY_COUNT : 8
__DERIVATION     : {CIM_SystemResource, CIM_LogicalElement,
CIM_ManagedSystemElement}
__SERVER         : SYSTEM01
__NAMESPACE      : root\cimv2
__PATH           :
\\SYSTEM01\root\cimv2:Win32_Environment.Name="testvar",UserName="<SYSTEM>"
```

```
Caption          : <SYSTEM>\testvar
Description      : <SYSTEM>\testvar
InstallDate      :
Name             : testvar
Status           : OK
SystemVariable   : True
UserName         : <SYSTEM>
VariableValue    : testvalue
```

This command creates the testvar environment variable that has the value testvalue. It does this by creating a new instance of the Win32_Environment WMI class. This operation requires appropriate credentials and that you may have to restart Windows PowerShell to see the new environment variable.

Example 3: Set WMI logging level for several remote computers

```
Set-WmiInstance -Class Win32_WMISetting -Argument @{LoggingLevel=2}
-Computername "system01", "system02", "system03"
```

```
__GENUS                        : 2
__CLASS                        : Win32_WMISetting
__SUPERCLASS                   : CIM_Setting
__DYNASTY                      : CIM_Setting
__RELPATH                      : Win32_WMISetting=@
__PROPERTY_COUNT               : 27
__DERIVATION                   : {CIM_Setting}
__SERVER                       : SYSTEM01
__NAMESPACE                    : root\cimv2
__PATH                         : \\SYSTEM01\root\cimv2:Win32_WMISetting=@
ASPScriptDefaultNamespace      : \\root\cimv2
ASPScriptEnabled               : False
AutorecoverMofs                : {%windir%\system32\wbem\cimwin32.mof,
%windir%\system32\wbem\ncprov.mof, %windir%\syst
em32\wbem\wmipcima.mof, %windir%\system32\wbem\secrcw32.mof...}
```

```
    AutoStartWin9X                   :
    BackupInterval                   :
    BackupLastTime                   :
    BuildVersion                     : 6001.18000
    Caption                          :
    DatabaseDirectory                : C:\Windows\system32\wbem\repository
    DatabaseMaxSize                  :
    Description                      :
    EnableAnonWin9xConnections       :
    EnableEvents                     : False
    EnableStartupHeapPreallocation : False
    HighThresholdOnClientObjects     :
    HighThresholdOnEvents            : 20000000
    InstallationDirectory            : C:\Windows\system32\wbem
    LastStartupHeapPreallocation     :
    LoggingDirectory                 : C:\Windows\system32\wbem\Logs\
    LoggingLevel                     : 2
    LowThresholdOnClientObjects      :
    LowThresholdOnEvents             : 10000000
    MaxLogFileSize                   : 65536
    MaxWaitOnClientObjects           :
    MaxWaitOnEvents                  : 2000
    MofSelfInstallDirectory          :
    SettingID                        :
    ...
```

This command sets the WMI logging level to 2. The command passes the property to be set and the value, together considered a value pair, in the argument parameter. The parameter takes a hash table that is defined by the `@{property = value}` construction. The returned class information reflects the new value.

REMARKS

To see the examples, type: "get-help Set-WmiInstance -examples".

For more information, type: "get-help Set-WmiInstance -detailed".

For technical information, type: "get-help Set-WmiInstance -full".

For online help, type: "get-help Set-WmiInstance -online"