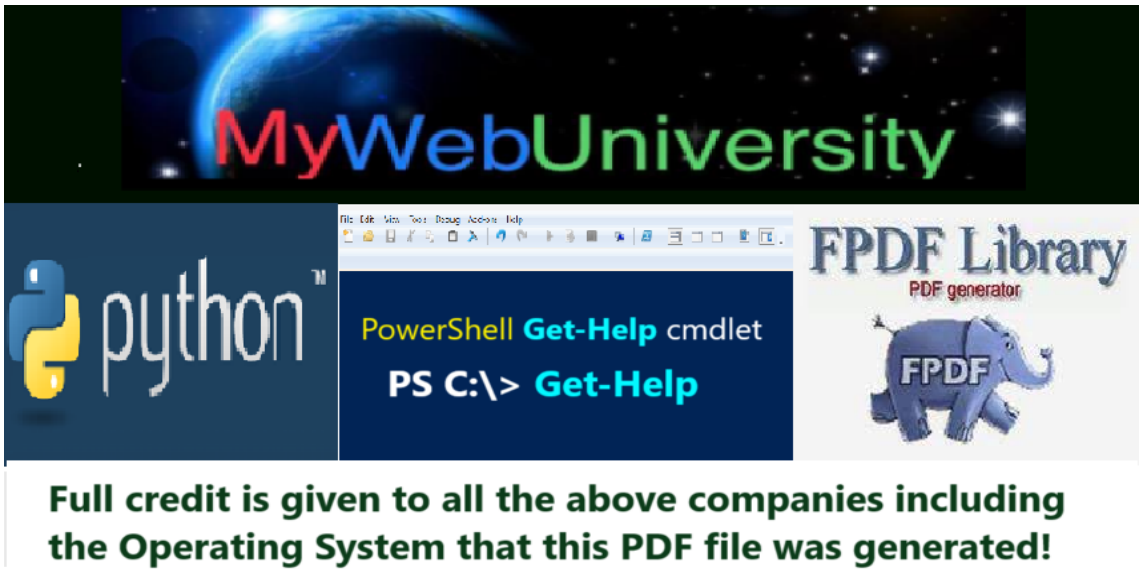

Windows PowerShell Get-Help on Cmdlet 'Get-WSManInstance'

PS:\>Get-HELP Get-WSManInstance -Full

NAME

Get-WSManInstance

SYNOPSIS

Displays management information for a resource instance specified by a Resource URI.

SYNTAX

Get-WSManInstance [-ResourceURI] <System.Uri> [-ApplicationName <System.String>] [-Associations] [-Authentication {None | Default | Digest | Negotiate | Basic |

Kerberos | ClientCertificate | Credssp}] [-BasePropertiesOnly] [-CertificateThumbprint <System.String>] [-ComputerName <System.String>] [-ConnectionURI <System.Uri>]

[-Credential <System.Management.Automation.PSCredential>] [-Dialect <System.Uri>] -Enumerate [-Filter <System.String>] [-OptionSet <System.Collections.Hashtable>]

[-Port <System.Int32>] [-ReturnType {object | epr | objectandepr}] [-SessionOption <Microsoft.WSMan.Management.SessionOption>] [-Shallow] [-UseSSL]

[<CommonParameters>]

Get-WSManInstance [-ResourceURI] <System.Uri> [-ApplicationName <System.String>] [-Authentication {None | Default | Digest | Negotiate | Basic | Kerberos |

ClientCertificate | Credssp}] [-CertificateThumbprint <System.String>] [-ComputerName <System.String>] [-ConnectionURI <System.Uri>] [-Credential

<System.Management.Automation.PSCredential>] [-Dialect <System.Uri>] [-Fragment <System.String>] [-OptionSet <System.Collections.Hashtable>] [-Port <System.Int32>]

[-SelectorSet <System.Collections.Hashtable>] [-SessionOption <Microsoft.WSMan.Management.SessionOption>] [-UseSSL] [<CommonParameters>]

DESCRIPTION

The `Get-WSManInstance` cmdlet retrieves an instance of a management resource that is specified by a resource Uniform Resource Identifier (URI). The information that

is retrieved can be a complex XML information set, which is an object, or a simple value. This cmdlet is the equivalent to the standard Web Services for Management

(WS-Management) Get command.

This cmdlet uses the WS-Management connection/transport layer to retrieve information.

PARAMETERS

-ApplicationName <System.String>

Specifies the application name in the connection. The default value of the ApplicationName parameter is WSMAN. The complete identifier for the remote endpoint is

in the following format:

<transport>://<server>:<port>/<ApplicationName>

For example: `http://server01:8080/WSMAN`

Internet Information Services (IIS), which hosts the session, forwards requests with this endpoint to the specified application. This default setting of WSMAN is

appropriate for most uses. This parameter is designed to be used if many computers establish remote connections to one computer that is running PowerShell. In

this case, IIS hosts WS-Management for efficiency.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Associations <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet gets association instances, not associated instances. You can use this parameter only when the Dialect parameter has a value of

Association.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Authentication <Microsoft.WSMan.Management.AuthenticationMechanism>

Specifies the authentication mechanism to be used at the server. The acceptable values for this parameter are:

- `Basic` - Basic is a scheme in which the user name and password are sent in clear text to the server or proxy. - `Default` - Use the authentication method

implemented by the WS-Management protocol. This is the default. - `Digest` - Digest is a challenge-response scheme that uses a server-specified data string for

the challenge. - `Kerberos` - The client computer and the server mutually authenticate by using Kerberos certificates. - `Negotiate` - Negotiate is a

challenge-response scheme that negotiates with the server or proxy to determine the scheme to use for authentication. For example, this parameter value allows

for negotiation to determine whether the Kerberos protocol or NTLM is used. - `CredSSP` - Use Credential Security Support Provider (CredSSP) authentication,

which lets the user delegate credentials. This option is designed for commands that run on one remote computer but collect data from or run additional

commands on other remote computers.

> [!CAUTION] > CredSSP delegates the user credentials from the local computer to a remote computer. This practice > increases the security risk of the remote

operation. If the remote computer is compromised, when > credentials are passed to it, the credentials can be used to control the network session.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-BasePropertiesOnly <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet enumerates only the properties that are part of the base class that is specified by the ResourceURI parameter. This parameter has no

effect if the Shallow parameter is specified.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-CertificateThumbprint <System.String>

Specifies the digital public key certificate (X509) of a user account that has permission to perform this action. Enter the certificate thumbprint of the

certificate.

Certificates are used in client certificate-based authentication. They can be mapped only to local user accounts; they do not work with domain accounts.

To get a certificate thumbprint, use the `Get-Item` or `Get-ChildItem` command in the PowerShell Cert: drive.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ComputerName <System.String>

Specifies the computer against which to run the management operation. The value can be a fully qualified domain name, a NetBIOS name, or an IP address. Use the

local computer name, use localhost, or use a dot (`.`) to specify the local computer. The local computer is the default. When the remote computer is in a

different domain from the user, you must use a fully qualified domain name must be used. You can pipe a value for this parameter to the cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ConnectionURI <System.Uri>

Specifies the connection endpoint. The format of this string is as follows:

<Transport>://<Server>:<Port>/<ApplicationName>

The following string is a correctly formatted value for this parameter:

`http://Server01:8080/WSMAN`

The URI must be fully qualified.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-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, Domain01\User01, or

User@Domain.com. Or, enter a PSCredential object, such as one returned by the `Get-Credential` cmdlet. When you type a user name, this cmdlet prompts you for a

password.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Dialect <System.Uri>

Specifies the dialect to use in the filter predicate. This can be any dialect that is supported by the remote service. The following aliases can be used for the

dialect URI:

- `WQL` - `http://schemas.microsoft.com/wbem/wsman/1/WQL`

- Selector - `http://schemas.microsoft.com/wbem/wsman/1/wsman/SelectorFilter`

- Association - `http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter`

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Enumerate <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns all of the instances of a management resource.

Required? true

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Filter <System.String>

Specifies the filter expression for the enumeration. If you specify this parameter, you must also specify Dialect .

The valid values of this parameter depend on the dialect that is specified in Dialect . For example, if Dialect is WQL, the Filter parameter must contain a

string, and the string must contain a valid WQL query such as the following query:

`"Select * from Win32_Service where State != Running"`

If Dialect is Association, Filter must contain a string, and the string must contain a valid filter, such as the following filter:

`-filter:Object=EPR[;AssociationClassName=AssocClassName][;ResultClassName=ClassName][;Role=RefPropertyName][;ResultRole=RefPropertyName]}`

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Fragment <System.String>

Specifies a section inside the instance that is to be updated or retrieved for the specified operation. For example, to get the status of a spooler service,

specify the following:

`-Fragment Status`

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-OptionSet <System.Collections.Hashtable>

Specifies a set of switches to a service to modify or refine the nature of the request. These resemble switches used in command-line shells because they are

service specific. Any number of options can be specified.

The following example demonstrates the syntax that passes the values 1, 2, and 3 for the a, b, and c parameters:

`-OptionSet @{a=1;b=2;c=3}`

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Port <System.Int32>

Specifies the port to use when the client connects to the WinRM service. When the transport is HTTP, the default port is 80. When the transport is HTTPS, the

default port is 443.

When you use HTTPS as the transport, the value of the ComputerName parameter must match the server's certificate common name (CN). However, if the SkipCNCheck

parameter is specified as part of the SessionOption parameter, the certificate common name of the server does not have to match the host name of the server. The

SkipCNCheck parameter should be used only for trusted computers.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ResourceURI <System.Uri>

Specifies the URI of the resource class or instance. The URI identifies a specific type of resource, such as disks or processes, on a computer.

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

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

`http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_NumericSensor`

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-ReturnType <System.String>

Specifies the type of data to be returned. The acceptable values for this parameter are:

- `Object`

- `EPR`

- `ObjectAndEPR`

The default value is `Object`.

If you specify `Object` or do not specify this parameter, this cmdlet returns only objects. If you specify endpoint reference (EPR) this cmdlet returns only the

endpoint references of the objects. Endpoint references contain information about the resource URI and the selectors for the instance. If you specify

`ObjectAndEPR`, this cmdlet returns both the object and its associated endpoint references.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SelectorSet <System.Collections.Hashtable>

Specifies a set of value pairs that are used to select particular management resource instances. The SelectorSet parameter is used when more than one instance of

the resource exists. The value of the SelectorSet parameter must be a hash table.

The following example shows how to enter a value for this parameter:

`-SelectorSet @{Name="WinRM";ID="yyy"}`

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SessionOption <Microsoft.WSMan.Management.SessionOption>

Specifies extended options for the WS-Management session. Enter a SessionOption object that you create by using the New-WSManSessionOption cmdlet. For more

information about the options that are available, type `Get-Help New-WSManSessionOption`.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Shallow <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns only instances of the base class that is specified in the resource URI. If you do not specify this parameter, this cmdlet

returns instances of the base class that is specified in the URI and in all its derived classes.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-UseSSL <System.Management.Automation.SwitchParameter>

Specifies that the Secure Sockets Layer (SSL) protocol is used to establish a connection to the remote computer. By default, SSL is not used.

WS-Management encrypts all the Windows PowerShell content that is transmitted over the network. The UseSSL parameter lets you specify the additional protection of

HTTPS instead of HTTP. If SSL is not available on the port that is used for the connection, and you specify this parameter, the command fails.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<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).

INPUTS

None

You can't pipe objects to this cmdlet.

OUTPUTS

System.Xml.XmlElement

This cmdlet returns an XMLElement object.

NOTES

---------- Example 1: Get all information from WMI ----------

Get-WSManInstance -ResourceURI wmicimv2/win32_service -SelectorSet @{name="winrm"} -ComputerName "Server01"

This command returns all of the information that Windows Management Instrumentation (WMI) exposes about the WinRM service on the remote server01 computer.

------- Example 2: Get the status of the Spooler service -------

Get-WSManInstance -ResourceURI wmicimv2/win32_service -SelectorSet @{name="spooler"} -Fragment Status -ComputerName "Server01"

This command returns only the status of the Spooler service on the remote server01 computer.

----- Example 3: Get endpoint references for all services -----

Get-WSManInstance -Enumerate -ResourceURI wmicimv2/win32_service -ReturnType EPR

This command returns endpoint references that correspond to all the services on the local computer.

----- Example 4: Get services that meet specified criteria -----

Get-WSManInstance -Enumerate -ResourceURI wmicimv2/* -Filter "select * from win32_service where StartMode = 'Auto' and State = 'Stopped'" -ComputerName "Server01"

This command lists all of the services that meet the following criteria on the remote Server01 computer:

- The startup type of the service is Automatic.

- The service is stopped.

Example 5: Get listener configuration that matches criteria on the local computer

Get-WSManInstance -ResourceURI winrm/config/listener -SelectorSet @{Address="*";Transport="http"}

This command lists the WS-Management listener configuration on the local computer for the listener that matches the criteria in the selector set.

Example 6: Get listener configuration that matches criteria on a remote computer

Get-WSManInstance -ResourceURI winrm/config/listener -SelectorSet @{Address="*";Transport="http"} -ComputerName "Server01"

This command lists the WS-Management listener configuration on the remote server01 computer for the listener that

matches the criteria in the selector set.

Example 7: Get associated instances related to a specified instance

```
Get-WSManInstance -Enumerate -Dialect Association -Filter "{Object=win32_service?name=winrm}" -ResourceURI wmicimv2/*
```

This command gets the associated instances that are related to the specified instance (winrm).

You must enclose the filter in quotation marks, as shown in the example.

Example 8: Get association instances related to a specified instance

```
Get-WSManInstance -Enumerate -Dialect Association -Associations -Filter "{Object=win32_service?name=winrm}" -ResourceURI wmicimv2/*
```

This command gets association instances that are related to the specified instance (winrm). Because the Dialect value is association and the Associations parameter is

used, this command returns association instances, not associated instances.

You must enclose the filter in quotation marks, as shown in the example.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.wsman.management/get-wsmaninstance?view=powershell-5.1&WT.mc_id=ps-gethelp

Connect-WSMan

Disable-WSManCredSSP

Disconnect-WSMan

Enable-WSManCredSSP

Get-WSManCredSSP

Invoke-WSManAction

New-WSManInstance

New-WSManSessionOption

Remove-WSManInstance

Set-WSManInstance

Set-WSManQuickConfig

Test-WSMan