### PowerShell Get-Help on command 'Invoke-WSManAction'

***PS C:\Users\wahid> Get-Help Invoke-WSManAction***

```
NAME
    Invoke-WSManAction

SYNOPSIS
    Invokes an action on the object that is specified by the Resource URI and by
    the selectors.


SYNTAX
    Invoke-WSManAction [-ResourceURI] <System.Uri> [-Action] <System.String>
    [[-SelectorSet] <System.Collections.Hashtable>] [-ApplicationName
    <System.String>] [-Authentication {None | Default | Digest | Negotiate | Basic
    | Kerberos | ClientCertificate | Credssp}] [-CertificateThumbprint
    <System.String>] [-ComputerName <System.String>] [-Credential
    <System.Management.Automation.PSCredential>] [-FilePath <System.String>]
    [-OptionSet <System.Collections.Hashtable>] [-Port <System.Int32>]
    [-SessionOption <Microsoft.WSMan.Management.SessionOption>] [-UseSSL]
    [-ValueSet <System.Collections.Hashtable>] [<CommonParameters>]

    Invoke-WSManAction [-ResourceURI] <System.Uri> [-Action] <System.String>
    [[-SelectorSet] <System.Collections.Hashtable>] [-Authentication {None |
```

Default | Digest | Negotiate | Basic | Kerberos | ClientCertificate |
Credssp}] [-CertificateThumbprint <System.String>] [-ConnectionURI
<System.Uri>] [-Credential <System.Management.Automation.PSCredential>]
[-FilePath <System.String>] [-OptionSet <System.Collections.Hashtable>]
[-SessionOption <Microsoft.WSMan.Management.SessionOption>] [-ValueSet
<System.Collections.Hashtable>] [<CommonParameters>]

DESCRIPTION

The `Invoke-WSManAction` runs an action on the object that is specified by RESOURCE_URI, where parameters are specified by key value pairs.

This cmdlet uses the WSMan connection/transport layer to run the action.

PARAMETERS

-Action <System.String>

Specifies the method to run on the management object specified by the ResourceURI and selectors.

-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 when numerous computers establish remote connections to one computer running Windows PowerShell. In this case, IIS hosts Web Services for Management (WS-Management) for efficiency.

-Authentication <Microsoft.WSMan.Management.AuthenticationMechanism>

Specifies the authentication mechanism to be used at the server. Possible values 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   negotiation to determine whether the Kerberos protocol or NTLM is used. - `CredSSP` - Use Credential Security Support Provider (CredSSP) authentication, which allows the   user to 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's 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.

-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 Windows PowerShell Cert: drive.

-ComputerName <System.String>

Specifies the computer against which you want 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.

-ConnectionURI <System.Uri>

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

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

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

`http://Server01:8080/WSMAN`

The URI must be fully qualified.

-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, you will be prompted for a password.

-FilePath <System.String>

Specifies the path of a file that is used to update a management resource. You specify the management resource by using the ResourceURI parameter and the SelectorSet parameter. For example, the following command uses the FilePath parameter:

`Invoke-WSManAction -Action stopservice -ResourceUri wmicimv2/Win32_Service -SelectorSet @{Name="spooler"} -FilePath c:\input.xml -Authentication default`

This command calls the StopService method on the Spooler service by using input from a file. The file, `Input.xml`, contains the following content:

`<p:StopService_INPUT xmlns:p="http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service" />`

-OptionSet <System.Collections.Hashtable>

Passes a set of switches to a service to modify or refine the nature of the request. These are similar to 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}`

-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, then 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 machines.

-ResourceURI <System.Uri>

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

-SelectorSet <System.Collections.Hashtable>

Specifies a set of value pairs that are used to select particular management resource instances. SelectorSet is used when more than one instance of the resource exists. The value of SelectorSet must be a hash table.

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

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

-SessionOption <Microsoft.WSMan.Management.SessionOption>

Defines a set of 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, see `New-WSManSessionOption`.

-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 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.

-ValueSet <System.Collections.Hashtable>

Specifies a hash table that helps modify a management resource. You specify the management resource using the ResourceURI and SelectorSet parameters. The value of the ValueSet parameter must be a hash table.

<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: Invoke a method ------------------

```
Invoke-WSManAction -Action startservice -ResourceURI wmicimv2/win32_service
-SelectorSet @{name="spooler"} -Authentication default
```

```
xsi         : http://www.w3.org/2001/XMLSchema-instance
p           :
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service
cim         : http://schemas.dmtf.org/wbem/wscim/1/common
lang        : en-US
ReturnValue : 0
```

This command calls the StartService method of the Win32_Service WMI class instance that corresponds to the Spooler service.

The return value indicates whether the action was successful. In this case, a return value of 0 indicates success. A return value of 5 indicates that the service is already started.

----------------- Example 2: Invoke a method -----------------

```
Invoke-WSManAction -Action stopservice -ResourceURI wmicimv2/Win32_Service
-SelectorSet @{Name="spooler"} -FilePath:input.xml -Authentication default
```

```
xsi         : http://www.w3.org/2001/XMLSchema-instance
p           :
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service
cim         : http://schemas.dmtf.org/wbem/wscim/1/common
lang        : en-US
ReturnValue : 0
```

This command calls the StopService method on the Spooler service by using input from a file. The file, `Input.xml`, contains the following content:

```
`<p:StopService_INPUT xmlns:p="http://schemas.microsoft.com/wbem/wsman/1/wmi/ro
ot/cimv2/Win32_Service" />`
```

The return value indicates whether the action was successful. In this case, a return value of 0 indicates success. A return value of 5 indicates that the service is already started.

-- Example 3: Invoke a method with specified parameter values --

```
Invoke-WSManAction -Action create -ResourceURI wmicimv2/win32_process
-ValueSet @{commandline="notepad.exe";currentdirectory="C:\"}
```

```
xsi         : http://www.w3.org/2001/XMLSchema-instance
p           :
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Process
cim         : http://schemas.dmtf.org/wbem/wscim/1/common
lang        : en-US
ProcessId   : 6356
ReturnValue : 0
```

This command calls the Create method of the Win32_Process class. It passes the method two parameter values, `Notepad.exe` and `C:`. As a result, a new process is created to run Notepad, and the current directory of the new process is set to `C:`.

------- Example 4: Invoke a method on a remote computer -------

```
Invoke-WSManAction -Action startservice -ResourceURI wmicimv2/win32_service
-SelectorSet @{name="spooler"} -ComputerName server01 -Authentication default
```

```
xsi         : http://www.w3.org/2001/XMLSchema-instance
p           :
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service
cim         : http://schemas.dmtf.org/wbem/wscim/1/common
lang        : en-US
ReturnValue : 0
```

This command calls the StartService method of the Win32_Service WMI class instance that corresponds to the Spooler service. Because the ComputerName parameter is specified, the command runs against the remote server01 computer.

The return value indicates whether the action was successful. In this case, a return value of 0 indicates success. A return value of 5 indicates that the service is already started.

REMARKS

To see the examples, type: "get-help Invoke-WSManAction -examples".

For more information, type: "get-help Invoke-WSManAction -detailed".

For technical information, type: "get-help Invoke-WSManAction -full".

For online help, type: "get-help Invoke-WSManAction -online"