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

PowerShell Get-Help on command 'Test-WSMan'

PS C:\Users\wahid> Get-Help Test-WSMan

NAME

Test-WSMan

SYNOPSIS

Tests whether the WinRM service is running on a local or remote computer.

SYNTAX

Test-WSMan [[-ComputerName] <System.String>] [-ApplicationName <System.String>] [-Authentication {None | Default | Digest | Negotiate | Basic | Kerberos | ClientCertificate | Credssp}] [-CertificateThumbprint <System.String>] [-Credential <System.Management.Automation.PSCredential>] [-Port <System.Int32>] [-UseSSL] [<CommonParameters>]

DESCRIPTION

The `Test-WSMan` cmdlet submits an identification request that determines whether the WinRM service is running on a local or remote computer. If the tested computer is running the service, the cmdlet displays the WS-Management identity schema, the protocol version, the product vendor, and the product version of the tested service.

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 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. 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. - 1 - 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.

> [!IMPORTANT] > If you do not specify the Authentication parameter, the `Test-WSMan` request is sent to the > remote computer anonymously, without using authentication. If the request is made anonymously, it > returns no information that is specific to the operating-system version. Instead, this cmdlet > displays null values for the operating system version and service pack level (OS: 0.0.0 SP: 0.0).

-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 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.

-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.

-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).

-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.

<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: Determine the status of the WinRM service -----

Test-WSMan

This command determines whether the WinRM service is running on the local
computer or on a remote computer.
Example 2: Determine the status of the WinRM service on a remote computer

Test-WSMan -ComputerName "server01"

This command determines whether the WinRM service is running on the server01
computer.
Example 3: Determine the status of the WinRM service and the operating system
version

Test-WSMan -Authentication default

This command tests to see whether the WS-Management (WinRM) service is running
on the local computer by using the authentication parameter.

Using the authentication parameter enables `Test-WSMan` to return the
operating system version.
Example 4: Determine the status of the WinRM service and the OS version on a
remote computer

Test-WSMan -ComputerName "server01" -Authentication default

This command tests to see whether the WS-Management (WinRM) service is running
on the computer named server01 using the authentication parameter.

Using the authentication parameter enables `Test-WSMan` to return the operating system version.

REMARKS

To see the examples, type: "get-help Test-WSMan -examples".

For more information, type: "get-help Test-WSMan -detailed".

For technical information, type: "get-help Test-WSMan -full".

For online help, type: "get-help Test-WSMan -online"