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

Windows PowerShell Get-Help on Cmdlet 'Enter-PSSession'

PS:\>Get-HELP Enter-PSSession -Full

NAME

Enter-PSSession

SYNOPSIS

Starts an interactive session with a remote computer.

SYNTAX

Enter-PSSession [[-ConnectionUri] <System.Uri>] [[-Credential] <System.Management.Automation.PSCredential>] [-AllowRedirection] [-Authentication {Default | Basic | Negotiate | NegotiateWithImplicitCredential | Credssp | Digest | Kerberos}] [-CertificateThumbprint <System.String>] [-ConfigurationName <System.String>] [-EnableNetworkAccess] [-SessionOption <System.Management.Automation.Remoting.PSSessionOption>] [<CommonParameters>]

Enter-PSSession [-ComputerName] <System.String> [[-Credential] <System.Management.Automation.PSCredential>] [-ApplicationName <System.String>] [-Authentication {Default | Basic | Negotiate | NegotiateWithImplicitCredential | Credssp | Digest | Kerberos}] [-CertificateThumbprint <System.String>] [-ConfigurationName <System.String>] [-EnableNetworkAccess] [-Port <System.Int32>] [-SessionOption

<System.Management.Automation.Remoting.PSSessionOption>] [-UseSSL] [<CommonParameters>]

Enter-PSSession [-VMId] <System.Guid> [[-Credential] <System.Management.Automation.PSCredential>] [-ConfigurationName <System.String>] [<CommonParameters>]

Enter-PSSession [-VMName] <System.String> [[-Credential] <System.Management.Automation.PSCredential>] [-ConfigurationName <System.String>] [<CommonParameters>]

Enter-PSSession [-ContainerId] <System.String> [-ConfigurationName <System.String>] [-RunAsAdministrator] [<CommonParameters>]

Enter-PSSession [[-Id] <System.Int32>] [<CommonParameters>]

Enter-PSSession [-InstanceId <System.Guid>] [<CommonParameters>]

Enter-PSSession [-Name <System.String>] [<CommonParameters>]

Enter-PSSession [[-Session] <System.Management.Automation.Runspaces.PSSession>] [<CommonParameters>]

DESCRIPTION

The `Enter-PSSession` cmdlet starts an interactive session with a single remote computer. During the session, the commands that you type run on the remote computer,

just as if you were typing directly on the remote computer. You can have only one interactive session at a time.

Typically, you use the ComputerName parameter to specify the name of the remote computer. However, you can also use a session that you create by using the

`New-PSSession` cmdlet for the interactive session. However, you cannot use the `Disconnect-PSSession`, `Connect-PSSession`, or `Receive-PSSession` cmdlets to

disconnect from or re-connect to an interactive session.

To end the interactive session and disconnect from the remote computer, use the `Exit-PSSession` cmdlet, or type `exit`.

PARAMETERS

-AllowRedirection <System.Management.Automation.SwitchParameter>

Allows redirection of this connection to an alternate Uniform Resource Identifier (URI). By default, redirection is not allowed.

When you use the ConnectionURI parameter, the remote destination can return an instruction to redirect to a different URI. By default, PowerShell does not

redirect connections, but you can use this parameter to allow it to redirect the connection.

You can also limit the number of times the connection is redirected by changing the MaximumConnectionRedirectionCount session option value. Use the

MaximumRedirection parameter of the `New-PSSessionOption` cmdlet or set the MaximumConnectionRedirectionCount property of the `$PSSessionOption` preference

variable. The default value is 5.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-ApplicationName <System.String>

Specifies the application name segment of the connection URI. Use this parameter to specify the application name when you are not using the ConnectionURI

parameter in the command.

The default value is the value of the `$PSSessionApplicationName` preference variable on the local computer. If this preference variable is not defined, the

default value is WSMAN. This value is appropriate for most uses. For more information, see about_Preference_Variables (About/about_Preference_Variables.md).

The WinRM service uses the application name to select a listener to service the connection request. The value of this

parameter should match the value of the

URLPrefix property of a listener on the remote computer.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Authentication <System.Management.Automation.Runspaces.AuthenticationMechanism>

Specifies the mechanism that is used to authenticate the user's credentials. The acceptable values for this parameter are:

- Default

- Basic

- Credssp

- Digest

- Kerberos

- Negotiate

- NegotiateWithImplicitCredential

The default value is Default.

CredSSP authentication is available only in Windows Vista, Windows Server 2008, and later versions of the Windows operating system.

For more information about the values of this parameter, see AuthenticationMechanism Enum (/dotnet/api/system.management.automation.runspaces.authenticationmechanism).

> [!CAUTION] > Credential Security Support Provider (CredSSP) authentication, in which the user's credentials are > passed to a remote computer to be authenticated, is designed for commands that require > authentication on more than one resource, such as accessing a remote network share. This mechanism > increases the security risk of the remote operation. If the remote computer is compromised, the > credentials that are passed to it can be used to control the network session.

Required? false
Position? named
Default value None
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, 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 a computer name. This cmdlet starts an interactive session with the specified remote computer. Enter only one computer name. The default is the local

computer.

Type the NetBIOS name, the IP address, or the fully qualified domain name of the computer. You can also pipe a computer name to `Enter-PSSession`.

To use an IP address in the value of the ComputerName parameter, the command must include the Credential parameter. Also, the computer must be configured for

HTTPS transport or the IP address of the remote computer must be included in the WinRM TrustedHosts list on the local computer. For instructions for adding a

computer name to the TrustedHosts list, see "How to Add a Computer to the Trusted Host List" in about_Remote_Troubleshooting

(About/about_Remote_Troubleshooting.md).

> [!NOTE] > On the Windows operating system, to include the local computer in the value of the > ComputerName parameter, you must start PowerShell with the Run as

administrator option.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-ConfigurationName <System.String>

Specifies the session configuration that is used for the interactive session.

Enter a configuration name or the fully qualified resource URI for a session configuration. If you specify only the configuration name, the following schema URI

is prepended: `http://schemas.microsoft.com/powershell`.

The session configuration for a session is located on the remote computer. If the specified session configuration does not exist on the remote computer, the

command fails.

The default value is the value of the `$PSSessionConfigurationName` preference variable on the local computer. If this preference variable is not set, the default

is Microsoft.PowerShell. For more information, see about_Preference_Variables (About/about_Preference_Variables.md).

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ConnectionUri <System.Uri>

Specifies a URI that defines the connection endpoint for the session. The URI must be fully qualified. The format of this string is as follows:

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

The default value is as follows:

`http://localhost:5985/WSMAN`

If you do not specify a ConnectionURI , you can use the UseSSL , ComputerName , Port , and ApplicationName parameters to specify the ConnectionURI values.

Valid values for the Transport segment of the URI are HTTP and HTTPS. If you specify a connection URI with a Transport segment, but do not specify a port, the

session is created by using standards ports: 80 for HTTP and 443 for HTTPS. To use the default ports for PowerShell remoting, specify port 5985 for HTTP or 5986

for HTTPS.

If the destination computer redirects the connection to a different URI, PowerShell prevents the redirection unless you use the AllowRedirection parameter in the

command.

Required? false

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ContainerId <System.String>

Specifies the ID of a container.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

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 or Domain01\User01 , or enter a PSCredential object generated by the `Get-Credential` cmdlet. If you type a user name, you're

prompted to enter the password.

Credentials are stored in a PSCredential (/dotnet/api/system.management.automation.pscredential)object and the password is stored as a SecureString

(/dotnet/api/system.security.securestring).

> [!NOTE] > For more information about SecureString data protection, see > How secure is SecureString?

(/dotnet/api/system.security.securestring#how-secure-is-securestring).

Required? false

Position? 1

Default value Current user

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-EnableNetworkAccess <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet adds an interactive security token to loopback sessions. The interactive token lets you run commands in the loopback session that get

data from other computers. For example, you can run a command in the session that copies XML files from a remote computer to the local computer.

A loopback session is a PSSession that originates and ends on the same computer. To create a loopback session, omit the ComputerName parameter or set its value to

. (dot), localhost, or the name of the local computer.

By default, loopback sessions are created by using a network token, which might not provide sufficient permission to authenticate to remote computers.

The EnableNetworkAccess parameter is effective only in loopback sessions. If you use EnableNetworkAccess when you create a session on a remote computer, the

command succeeds, but the parameter is ignored.

You can also allow remote access in a loopback session by using the CredSSP value of the Authentication parameter, which delegates the session credentials to

other computers.

This parameter was introduced in Windows PowerShell 3.0.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Id <System.Int32>

Specifies the ID of an existing session. `Enter-PSSession` uses the specified session for the interactive session.

To find the ID of a session, use the `Get-PSSession` cmdlet.

Required? false

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-InstanceId <System.Guid>

Specifies the instance ID of an existing session. `Enter-PSSession` uses the specified session for the interactive session.

The instance ID is a GUID. To find the instance ID of a session, use the `Get-PSSession` cmdlet. You can also use the Session , Name , or ID parameters to specify

an existing session. Or, you can use the ComputerName parameter to start a temporary session.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Name <System.String>

Specifies the friendly name of an existing session. `Enter-PSSession` uses the specified session for the interactive session.

If the name that you specify matches more than one session, the command fails. You can also use the Session ,

InstanceID , or ID parameters to specify an existing

session. Or, you can use the ComputerName parameter to start a temporary session.

To establish a friendly name for a session, use the Name parameter of the `New-PSSession` cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Port <System.Int32>

Specifies the network port on the remote computer that is used for this command. To connect to a remote computer, the remote computer must be listening on the

port that the connection uses. The default ports are 5985, which is the WinRM port for HTTP, and 5986, which is the WinRM port for HTTPS.

Before using an alternate port, you must configure the WinRM listener on the remote computer to listen at that port. Use the following commands to configure the

listener:

1. `winrm delete winrm/config/listener?Address=*+Transport=HTTP` 2. `winrm create winrm/config/listener?Address=*+Transport=HTTP @{Port="<port-number>"}`

Do not use the Port parameter unless you must. The port setting in the command applies to all computers or sessions on which the command runs. An alternate port

setting might prevent the command from running on all computers.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-RunAsAdministrator <System.Management.Automation.SwitchParameter>

Indicates that the PSSession runs as administrator.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Session <System.Management.Automation.Runspaces.PSSession>

Specifies a Windows PowerShell session (PSSession) to use for the interactive session. This parameter takes a session object. You can also use the Name ,

InstanceID , or ID parameters to specify a PSSession .

Enter a variable that contains a session object or a command that creates or gets a session object, such as a `New-PSSession` or `Get-PSSession` command. You can

also pipe a session object to `Enter-PSSession`. You can submit only one PSSession by using this parameter. If you enter a variable that contains more than one

PSSession , the command fails.

When you use `Exit-PSSession` or the EXIT keyword, the interactive session ends, but the PSSession that you created remains open and available for use.

Required? false

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-SessionOption <System.Management.Automation.Remoting.PSSessionOption>

Sets advanced options for the session. Enter a SessionOption object, such as one that you create by using the `New-PSSessionOption` cmdlet, or a hash table in

which the keys are session option names and the values are session option values.

The default values for the options are determined by the value of the `$PSSessionOption` preference variable, if it is set. Otherwise, the default values are

established by options set in the session configuration.

The session option values take precedence over default values for sessions set in the `$PSSessionOption` preference variable and in the session configuration.

However, they do not take precedence over maximum values, quotas or limits set in the session configuration.

For a description of the session options, including the default values, see `New-PSSessionOption`. For information about the `$PSSessionOption` preference

variable, see about_Preference_Variables (About/about_Preference_Variables.md). For more information about session configurations, see

about_Session_Configurations (About/about_Session_Configurations.md).

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-UseSSL <System.Management.Automation.SwitchParameter>

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

WS-Management encrypts all Windows PowerShell content transmitted over the network. The UseSSL parameter is an additional protection that sends the data across an

HTTPS connection instead of an HTTP connection.

If you use this parameter, but SSL is not available on the port that is used for the command, the command fails.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-VMId <System.Guid>

Specifies the ID of a virtual machine.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-VMName <System.String>

Specifies the name of a virtual machine.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

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

System.String

You can pipe a computer name as a string to this cmdlet.

System.Management.Automation.Runspaces.PSSession

You can pipe a session object to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

Windows PowerShell includes the following aliases for `Enter-PSSession`:

- `etsn`

To connect to a remote computer, you must be a member of the Administrators group on the remote computer. To start an interactive session on the local computer,
you must start PowerShell with the Run as administrator option.

When you use `Enter-PSSession`, your user profile on the remote computer is used for the interactive session. The commands in the remote user profile, including
commands to add PowerShell modules and to change the command prompt, run before the remote prompt is displayed.

`Enter-PSSession` uses the UI culture setting on the local computer for the interactive session. To find the local UI culture, use the `$UICulture` automatic
variable.

`Enter-PSSession` requires the `Get-Command`, `Out-Default`, and `Exit-PSSession` cmdlets. If these cmdlets are not included in the session configuration on the
remote computer, the `Enter-PSSession` commands fails.

Unlike `Invoke-Command`, which parses and interprets the commands before it sends them to the remote computer, `Enter-PSSession` sends the commands directly to

the remote computer without interpretation.

If the session you want to enter is busy processing a command, there might be a delay before PowerShell responds to the `Enter-PSSession` command. You are

connected as soon as the session is available. To cancel the `Enter-PSSession` command, press <kbd>CTRL</kbd>+<kbd>C</kbd>.

---------- Example 1: Start an interactive session ----------

```
PS C:\> Enter-PSSession
[localhost]: PS C:\>
```

This command starts an interactive session on the local computer. The command prompt changes to indicate that you are now running commands in a different session.

The commands that you enter run in the new session, and the results are returned to the default session as text.

--------- Example 2: Work with an interactive session ---------

```
PS C:\> Enter-PSSession -ComputerName Server01
[Server01]: PS C:\>
[Server01]: PS C:\> Get-Process PowerShell > C:\ps-test\Process.txt
[Server01]: PS C:\> exit
PS C:\>
PS C:\> dir C:\ps-test\Process.txt
Get-ChildItem : Cannot find path 'C:\ps-test\Process.txt' because it does not exist.
At line:1 char:4
+ dir <<<<  c:\ps-test\Process.txt
```

This command shows how to work in an interactive session with a remote computer.

---------- Example 3: Use the Session parameter ----------

PS> $s = New-PSSession -ComputerName Server01

PS> Enter-PSSession -Session $s

[Server01]: PS>

These commands use the Session parameter of `Enter-PSSession` to run the interactive session in an existing PowerShell session (PSSession).

Example 4: Start an interactive session and specify the Port and Credential parameters

PS> Enter-PSSession -ComputerName Server01 -Port 90 -Credential Domain01\User01

[Server01]: PS>

This command starts an interactive session with the Server01 computer. It uses the Port parameter to specify the port and the Credential parameter to specify the

account of a user who has permission to connect to the remote computer.

---------- Example 5: Stop an interactive session ----------

PS> Enter-PSSession -ComputerName Server01

[Server01]: PS> Exit-PSSession

PS>

This example shows how to start and stop an interactive session. The first command uses the `Enter-PSSession` cmdlet to start an interactive session with the Server01

computer.

The second command uses the `Exit-PSSession` cmdlet to end the session. You can also use the Exit keyword to end the interactive session. `Exit-PSSession` and Exit

have the same effect.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/enter-pssession?view=powershell-5.1&WT.mc_id=ps-gethelp

Exit-PSSession

Get-PSSession

Invoke-Command

New-PSSession

Remove-PSSession

Connect-PSSession

Disconnect-PSSession

Receive-PSSession

about_PSSessions

about_Remote