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

PowerShell Get-Help on command 'New-PSSessionOption'

PS C:\Users\wahid> Get-Help New-PSSessionOption

NAME

New-PSSessionOption

SYNOPSIS

Creates an object that contains advanced options for a PSSession.

SYNTAX

New-PSSessionOption [-ApplicationArguments
<System.Management.Automation.PSPrimitiveDictionary>] [-CancelTimeout
<System.Int32>] [-Culture <System.Globalization.CultureInfo>] [-IdleTimeout
<System.Int32>] [-IncludePortInSPN] [-MaxConnectionRetryCount <System.Int32>]
[-MaximumReceivedDataSizePerCommand <System.Int32>]
[-MaximumReceivedObjectSize <System.Int32>] [-MaximumRedirection
<System.Int32>] [-NoCompression] [-NoEncryption] [-NoMachineProfile]
[-OpenTimeout <System.Int32>] [-OperationTimeout <System.Int32>]
[-OutputBufferingMode {None | Drop | Block}] [-ProxyAccessType {None |
IEConfig | WinHttpConfig | AutoDetect | NoProxyServer}] [-ProxyAuthentication
{Default | Basic | Negotiate | NegotiateWithImplicitCredential | Credssp |
Digest | Kerberos}] [-ProxyCredential
<System.Management.Automation.PSCredential>] [-SkipCACheck] [-SkipCNCheck]

[-SkipRevocationCheck] [-UICulture <System.Globalization.CultureInfo>]
[-UseUTF16] [<CommonParameters>]

DESCRIPTION

The `New-PSSessionOption` cmdlet creates an object that contains advanced options for a user-managed session (PSSession). You can use the object as the value of the SessionOption parameter of cmdlets that create a PSSession , such as `New-PSSession`, `Enter-PSSession`, and `Invoke-Command`.

Without parameters, `New-PSSessionOption` generates an object that contains the default values for all of the options. Because every property can be edited, you can use the resulting object as a template and create standard option objects for your enterprise.

You can also save a SessionOption object in the `$PSSessionOption` preference variable. The values of this variable establish new default values for the session options. They are effective when no session options are set for the session and they take precedence over options set in the session configuration, but you can override them by specifying session options or a SessionOption object in a cmdlet that creates a session. For more information about the `$PSSessionOption` preference variable, see about_Preference_Variables (About/about_Preference_Variables.md).

When you use a SessionOption object in a cmdlet that creates a session, 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 more information about session configurations, see about_Session_Configurations (About/about_Session_Configurations.md).

PARAMETERS

-ApplicationArguments <System.Management.Automation.PSPrimitiveDictionary>

Specifies a PrimitiveDictionary that is sent to the remote session.
Commands and scripts in the remote session, including startup scripts in
the session configuration, can find this dictionary in the
ApplicationArguments property of the `$PSSenderInfo` automatic variable.
You can use this parameter to send data to the remote session.

For more information, see about_Hash_Tables (about/about_Hash_Tables.md),
about_Session_Configurations (About/about_Session_Configurations.md), and
about_Automatic_Variables (about/about_Automatic_Variables.md).

-CancelTimeout <System.Int32>

Determines how long PowerShell waits for a cancel operation
(<kbd>CTRL</kbd>+<kbd>C</kbd>) to finish before ending it. Enter a value
in milliseconds.

The default value is `60000` (one minute). A value of `0` (zero) means no
time-out; the command continues indefinitely.

-Culture <System.Globalization.CultureInfo>

Specifies the culture to use for the session. Enter a culture name in
`<languagecode2>-<country/regioncode2>` format (like `ja-JP`), a variable
that contains a CultureInfo object, or a command that gets a CultureInfo
object.

The default value is `$Null`, and the culture that is set in the operating
system is used in the session.

-IdleTimeout <System.Int32>

Determines how long the session stays open if the remote computer does not
receive any communication from the local computer. This includes the
heartbeat signal. When the interval expires, the session closes.

The idle time-out value is of significant importance if you intend to disconnect and reconnect to a session. You can reconnect only if the session has not timed out.

Enter a value in milliseconds. The minimum value is `60000` (1 minute). The maximum is the value of the MaxIdleTimeoutms property of the session configuration. The default value, `-1`, does not set an idle time-out.

The session uses the idle time-out that is set in the session options, if any. If none is set (`-1`), the session uses the value of the IdleTimeoutMs property of the session configuration or the WSMan shell time-out value (`WSMan:<ComputerName>\Shell\IdleTimeout`), whichever is shortest.

If the idle time-out set in the session options exceeds the value of the MaxIdleTimeoutMs property of the session configuration, the command to create a session fails.

The IdleTimeoutMs value of the default Microsoft.PowerShell session configuration is `7200000` milliseconds (2 hours). Its MaxIdleTimeoutMs value is `2147483647` milliseconds (>24 days). The default value of the WSMan shell idle time-out (`WSMan:<ComputerName>\Shell\IdleTimeout`) is `7200000` milliseconds (2 hours).

The idle time-out value of a session can also be changed when disconnecting from a session or reconnecting to a session. For more information, see `Disconnect-PSSession` and `Connect-PSSession`.

In Windows PowerShell 2.0, the default value of the IdleTimeout parameter is `240000` (4 minutes).

Includes the port number in the Service Principal Name (SPN) used for Kerberos authentication, for example, `HTTP://<ComputerName>:5985`. This option allows a client that uses a non-default SPN to authenticate against a remote computer that uses Kerberos authentication.

The option is designed for enterprises where multiple services that support Kerberos authentication are running under different user accounts. For example, an IIS application that allows for Kerberos authentication can require the default SPN to be registered to a user account that differs from the computer account. In such cases, PowerShell remoting cannot use Kerberos to authenticate because it requires an SPN that is registered to the computer account. To resolve this problem, administrators can create different SPNs, such as by using `Setspn.exe`, that are registered to different user accounts and can distinguish between them by including the port number in the SPN.

For more information, see Setspn Overview (/previous-versions/windows/it-pro/windows-server-2003/cc773257(v=ws.10)).

This parameter was introduced in Windows PowerShell 3.0.

-MaxConnectionRetryCount <System.Int32>

Specifies the number of times that PowerShell attempts to make a connection to a target machine if the current attempt fails due to network issues. The default value is `5`.

This parameter was added for PowerShell version 5.0.

-MaximumReceivedDataSizePerCommand <System.Int32>

Specifies the maximum number of bytes that the local computer can receive from the remote computer in a single command. Enter a value in bytes. By default, there is no data size limit.

This option is designed to protect the resources on the client computer.

-MaximumReceivedObjectSize <System.Int32>
Specifies the maximum size of an object that the local computer can receive from the remote computer. This option is designed to protect the resources on the client computer. Enter a value in bytes.

In Windows PowerShell 2.0, if you omit this parameter, there is no object size limit. Beginning in Windows PowerShell 3.0, if you omit this parameter, the default value is `209715200` bytes (or `200MB`).

-MaximumRedirection <System.Int32>
Determines how many times PowerShell redirects a connection to an alternate Uniform Resource Identifier (URI) before the connection fails. The default value is `5`. A value of `0` (zero) prevents all redirection.

This option is used in the session only when the AllowRedirection parameter is used in the command that creates the session.

-NoCompression <System.Management.Automation.SwitchParameter>
Turns off packet compression in the session. Compression uses more processor cycles, but it makes transmission faster.

-NoEncryption <System.Management.Automation.SwitchParameter>
Turns off data encryption.

-NoMachineProfile <System.Management.Automation.SwitchParameter>
Prevents loading the user's Windows user profile. As a result, the session might be created faster, but user-specific registry settings, items such as environment variables, and certificates are not available in the session.

-OpenTimeout <System.Int32>

Determines how long the client computer waits for the session connection to be established. When the interval expires, the command to establish the connection fails. Enter a value in milliseconds.

The default value is `180000` (3 minutes). A value of `0` (zero) means no time-out; the command continues indefinitely.

-OperationTimeout <System.Int32>

Determines the maximum time WinRM waits for positive connection tests from a live connection before initiating a connection time-out. For more information on WinRM, see the Windows Remote Management Documentation (/windows/win32/winrm/portal). OperationTimeout does not impose a time limit on commands or processes running in a remote session and does not affect other remoting protocols like SSH.

The default value is `180000` (3 minutes). A value of `0` (zero) means no time-out.

-OutputBufferingMode
<System.Management.Automation.Runspaces.OutputBufferingMode>

Determines how command output is managed in disconnected sessions when the output buffer becomes full.

If the output buffering mode is not set in the session or in the session configuration, the default value is `Block`. Users can also change the output buffering mode when disconnecting the session.

If you omit this parameter, the value of the OutputBufferingMode of the SessionOption object is `None`. A value of `Block` or `Drop` overrides the output buffering mode transport option set in the session configuration. The acceptable values for this parameter are:

- `Block`. When the output buffer is full, execution is suspended until

the buffer is clear.

- `Drop`. When the output buffer is full, execution continues. As new output is saved, the oldest

output is discarded. - `None`. No output buffering mode is specified.

For more information about the output buffering mode transport option, see `New-PSTransportOption`.

This parameter was introduced in Windows PowerShell 3.0.

-ProxyAccessType <System.Management.Automation.Remoting.ProxyAccessType>
Determines which mechanism is used to resolve the hostname. The acceptable values for this parameter are:

- `IEConfig`

- `WinHttpConfig`

- `AutoDetect`

- `NoProxyServer`

- `None`

The default value is `None`.
For information about the values of this parameter, see ProxyAccessType Enumeration (/dotnet/api/system.management.automation.remoting.proxyaccesstype).

-ProxyAuthentication
<System.Management.Automation.Runspaces.AuthenticationMechanism>
Specifies the authentication method that is used for proxy resolution. The acceptable values for this parameter are:

- `Basic`

- `Digest`

- `Negotiate`

The default value is `Negotiate`.
For more information about the values of this parameter, see AuthenticationMechanism Enumeration (/dotnet/api/system.management.automation.runspaces.authenticationmechanism).

-ProxyCredential <System.Management.Automation.PSCredential>
Specifies the credentials to use for proxy authentication. Enter a variable that contains a PSCredential object or a command that gets a PSCredential object, such as a `Get-Credential` command. If this option is not set, no credentials are specified.

-SkipCACheck <System.Management.Automation.SwitchParameter>
Specifies that when it connects over HTTPS, the client does not validate that the server certificate is signed by a trusted certification authority (CA).

Use this option only when the remote computer is trusted by using another mechanism, such as when the remote computer is part of a network that is physically secure and isolated or when the remote computer is listed as a trusted host in a WinRM configuration.

-SkipCNCheck <System.Management.Automation.SwitchParameter>
Specifies that the certificate common name (CN) of the server does not have to match the hostname of the server. This option is used only in remote operations that use the HTTPS protocol.

Use this option only for trusted computers.

-SkipRevocationCheck <System.Management.Automation.SwitchParameter>
Does not validate the revocation status of the server certificate.

-UICulture <System.Globalization.CultureInfo>
Specifies the UI culture to use for the session.

Valid values include:

- A culture name in `<languagecode2>-<country/regioncode2>` format, such as `ja-JP`

- A variable that contains a CultureInfo object - A command that gets a CultureInfo object, such as `Get-Culture`

The default value is `$null`, and the UI culture that is set in the operating system when the session is created.

-UseUTF16 <System.Management.Automation.SwitchParameter>
Indicates that this cmdlet encodes the request in UTF16 format instead of UTF8 format.

<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: Create a default session option ----------

New-PSSessionOption

MaximumConnectionRedirectionCount : 5
NoCompression : False
NoMachineProfile : False
ProxyAccessType : IEConfig
ProxyAuthentication : Negotiate
ProxyCredential :
SkipCACheck : False
SkipCNCheck : False
SkipRevocationCheck : False
OperationTimeout : 00:03:00
NoEncryption : False
UseUTF16 : False
Culture :
UICulture :
MaximumReceivedDataSizePerCommand :
MaximumReceivedObjectSize :
ApplicationArguments :
OpenTimeout : 00:03:00
CancelTimeout : 00:01:00
IdleTimeout : 00:04:00

Example 2: Configure a session by using a session option object

$pso = New-PSSessionOption -Culture "fr-fr" -MaximumReceivedObjectSize 10MB
New-PSSession -ComputerName Server01 -SessionOption $pso

The first command creates a new SessionOption object and saves it in the value of the `$pso` variable. The second command uses the `New-PSSession` cmdlet to create a session on the Server01 remote computer. The command uses the SessionOption object in the value of the `$pso` variable as the value of the SessionOption parameter of the command.

---------- Example 3: Start an interactive session ----------

```
Enter-PSSession -ComputerName Server01 -SessionOption (New-PSSessionOption
-NoEncryption -NoCompression)
```

The value of the SessionOption parameter is a `New-PSSessionOption` command that has the NoEncryption and NoCompression parameters.

The `New-PSSessionOption` command is enclosed in parentheses to make sure that it runs before the `Enter-PSSession` command.

---------- Example 4: Modify a session option object ----------

```
$a = New-PSSessionOption
$a.OpenTimeout

Days              : 0
Hours             : 0
Minutes           : 3
Seconds           : 0
Milliseconds      : 0
Ticks             : 1800000000
TotalDays         : 0.00208333333333333
TotalHours        : 0.05
TotalMinutes      : 3
TotalSeconds      : 180
TotalMilliseconds : 180000

$a.UICulture = (Get-UICulture)
```

```
$a.OpenTimeout = (New-Timespan -Minutes 4)
$a.MaximumConnectionRedirectionCount = 1
$a

MaximumConnectionRedirectionCount : 1
NoCompression                     : False
NoMachineProfile                  : False
ProxyAccessType                   : IEConfig
ProxyAuthentication               : Negotiate
ProxyCredential                   :
SkipCACheck                       : False
SkipCNCheck                       : False
SkipRevocationCheck               : False
OperationTimeout                  : 00:03:00
NoEncryption                      : False
UseUTF16                          : False
Culture                           :
UICulture                         : en-US
MaximumReceivedDataSizePerCommand :
MaximumReceivedObjectSize         :
ApplicationArguments              :
OpenTimeout                       : 00:04:00
CancelTimeout                     : 00:01:00
IdleTimeout                       : 00:04:00
```

Use this method to create a standard session object for your enterprise, and then create customized versions of it for particular uses.

---------- Example 5: Create a preference variable ----------

```
$PSSessionOption = New-PSSessionOption -OpenTimeOut 120000
```

When the `$PSSessionOption` preference variable is set in the session, it establishes default values for options in the sessions that are created with

the `New-PSSession`, `Enter-PSSession`, and `Invoke-Command` cmdlets.

To make the `$PSSessionOption` variable available in all sessions, add it to your PowerShell session and to your PowerShell profile.

For more information about the `$PSSessionOption` preference variable, see about_Preference_Variables (About/about_Preference_Variables.md). For more information about profiles, see about_Profiles (About/about_Profiles.md).
Example 6: Fulfill the requirements for a remote session configuration

```
$skipCN = New-PSSessionOption -SkipCNCheck
New-PSSession -ComputerName 171.09.21.207 -UseSSL -Credential Domain01\User01
-SessionOption $SkipCN
```

The first command uses the `New-PSSessionOption` cmdlet to create a SessionOption object that has the SkipCNCheck property. The command saves the resulting session object in the `$skipCN` variable.

The second command uses the `New-PSSession` cmdlet to create a new session on a remote computer. The `$skipCN` check variable is used in the value of the SessionOption parameter.

Because the computer is identified by its IP address, the value of the ComputerName parameter does not match any of the common names in the certificate that is used for Secure Sockets Layer (SSL). As a result, the SkipCNCheck option is required.
--- Example 7: Make arguments available to a remote session ---

```
$team = @{Team="IT"; Use="Testing"}
$TeamOption = New-PSSessionOption -ApplicationArguments $team
$s = New-PSSession -ComputerName Server01 -SessionOption $TeamOption
Invoke-Command -Session $s {$PSSenderInfo.ApplicationArguments}
```

```
Name                Value
----                -----
Team                IT
Use                 Testing
PSVersionTable      {CLRVersion, BuildVersion, PSVersion,
WSManStackVersion...}

Invoke-Command -Session $s {
  if ($PSSenderInfo.ApplicationArguments.Use -ne "Testing") {
    .\logFiles.ps1
  }
  else {
    "Just testing."
  }
}

Just testing.
```

The first command creates a hash table with two keys, Team and Use . The command saves the hash table in the `$team` variable. For more information about hash tables, see about_Hash_Tables (about/about_Hash_Tables.md).

Next, the `New-PSSessionOption` cmdlet, using the ApplicationArguments parameter, creates a SessionOption object saved in the `$team` variable. When `New-PSSessionOption` creates the session option object, it automatically converts the hash table in the value of the ApplicationArguments parameter to a PrimitiveDictionary so the data can be reliably transmitted to the remote session.

The `New-PSSession` cmdlet starts a session on the Server01 computer. It uses the SessionOption parameter to include the options in the `$teamOption` variable.

The `Invoke-Command` cmdlet demonstrates that the data in the `$team` variable is available to commands in the remote session. The data appears in the ApplicationArguments property of the `$PSSenderInfo` automatic variable.

The final `Invoke-Command` shows how the data might be used.

REMARKS

To see the examples, type: "get-help New-PSSessionOption -examples".

For more information, type: "get-help New-PSSessionOption -detailed".

For technical information, type: "get-help New-PSSessionOption -full".

For online help, type: "get-help New-PSSessionOption -online"