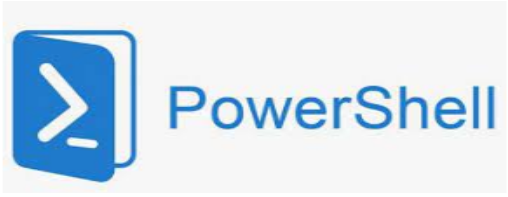

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'New-PSWorkflowSession'

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

NAME

New-PSWorkflowSession

SYNOPSIS

Creates a workflow session.

SYNTAX

New-PSWorkflowSession [[-ComputerName] <System.String[]>] [-ApplicationName <System.String>] [-Authentication {Default | Basic | Negotiate | NegotiateWithImplicitCredential | Credssp | Digest | Kerberos}] [-CertificateThumbprint <System.String>] [-Credential <System.Object>] [-EnableNetworkAccess] [-Name <System.String[]>] [-Port <System.Int32>] [-SessionOption <System.Management.Automation.Remoting.PSSessionOption>] [-ThrottleLimit <System.Int32>] [-UseSSL] [<CommonParameters>]

DESCRIPTION

The `New-PSWorkflowSession` cmdlet creates a user-managed session (PSSession) that is especially designed for running Windows PowerShell workflows. It uses the Microsoft.PowerShell.Workflow session configuration, which includes

scripts, type and formatting files, and options that are required for
workflows.

You can use `New-PSWorkflowSession` or its alias, `nwsn`.

You can also add workflow common parameters to this command. For more
information about workflow common parameters, see
about_WorkflowCommonParameters (./about/about_WorkflowCommonParameters.md)This
cmdlet was introduced in Windows PowerShell 3.0.

PARAMETERS

-ApplicationName <System.String>

Specifies the application name segment of the connection URI.

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
(../Microsoft.PowerShell.Core/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.

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

Specifies the mechanism that is used to authenticate the user 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 Enumeration (/dotnet/api/system.management.automation.runspaces.authenticationmechanism).
> [!CAUTION] > Credential Security Service Provider (CredSSP) authentication, in which the user > 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.

-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` cmdlet or the `Get-ChildItem` cmdlet in the Windows PowerShell `Cert:` drive.

-ComputerName <System.String[]>

Creates a persistent connection (PSSession) to the specified computer. If you enter multiple computer names, Windows PowerShell creates multiple PSSessions , one for each computer. The default is the local computer.

Type the NetBIOS name, an IP address, or a fully qualified domain name of one or more remote computers. To specify the local computer, type the computer name, `localhost`, or a dot (`.`). When the computer is in a different domain than the user, the fully qualified domain name is required. You can also pipe a computer name, in quotation marks to `New-PSWorkflowSession`.

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 (../Microsoft.PowerShell.Core/About/about_Remote_Troubleshooting.md).

-Credential <System.Object>

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.

-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, do not specify the ComputerName parameter or set its value to dot (`.`), `localhost`, or the name of the local computer.

By default, loopback sessions are created that have 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 specify the EnableNetworkAccess parameter 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.

To protect the computer from malicious access, disconnected loopback sessions that have interactive tokens, those created by using the EnableNetworkAccess parameter, can be reconnected only from the computer on which the session was created. Disconnected sessions that use CredSSP authentication can be reconnected from other computers. For more

information, see the `Disconnect-PSSession` cmdlet.

This parameter was introduced in Windows PowerShell 3.0.

-Name <System.String[]>

Specifies a friendly name for the workflow session. You can use the name with other cmdlets, such as `Get-PSSession` and `Enter-PSSession`. The name is not required to be unique to the computer or the current session.

-Port <System.Int32>

Specifies the network port on the remote computer that is used for this connection. To connect to a remote computer, the remote computer must be listening on the port that the connection uses. The default ports are `5985` (WinRM port for HTTP) and `5986` (WinRM port for HTTPS).

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

`winrm delete winrm/config/listener?Address=*+Transport=HTTP`

`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.

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

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

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 more information about session configurations, see about_Session_Configurations (../Microsoft.PowerShell.Core/About/about_Session_Configurations.md).

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 (../Microsoft.PowerShell.Core/About/about_Preference_Variables.md).

-ThrottleLimit <System.Int32>

Specifies the maximum number of concurrent connections that can be established to run this command. If you omit this parameter or enter a value of `0` (zero), the default value for the Microsoft.PowerShellWorkflow session configuration, `100`, is used.

The throttle limit applies only to the current command, not to the session or to the computer.

-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 specify this parameter, but SSL is not available on the port that
is used for the command, 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: Create a workflow session on a remote computer --

```
$params = @{
    ComputerName = "ServerNode01"
    Name = "WorkflowTests"
    SessionOption = (New-PSSessionOption -OutputBufferingMode Drop)
}
New-PSWorkflowSession @params
```

The value of the SessionOption parameter is a `New-PSSessionOption` command
that sets the output buffering mode in the session to Drop .
Example 2: Create workflow sessions on multiple remote computers

```
"ServerNode01", "Server12" |
    New-PSWorkflowSession -Name WorkflowSession -Credential Domain01\Admin01
-ThrottleLimit 150
```

The command uses the ThrottleLimit parameter to increase the per-command
throttle limit to `150`. This value takes precedence over the default throttle
limit of `100` that is set in the Microsoft.PowerShell.Workflow session
configuration.

REMARKS

To see the examples, type: "get-help New-PSWorkflowSession -examples".
For more information, type: "get-help New-PSWorkflowSession -detailed".

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

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