Windows PowerShell Get-Help on Cmdlet 'Get-PSSessionConfiguration'

PS:\>Get-HELP Get-PSSessionConfiguration -Full

NAME

Get-PSSessionConfiguration

SYNOPSIS

Gets the registered session configurations on the computer.

SYNTAX

Get-PSSessionConfiguration [[-Name] <System.String[]>] [-Force] [<CommonParameters>]

DESCRIPTION

The `Get-PSSessionConfiguration` cmdlet gets the session configurations that have been registered on the local computer. This is an advanced cmdlet that is designed

to be used by system administrators to manage customized session configurations for their users.

Beginning in PowerShell 3.0, you can define the properties of a session configuration by using a session configuration (.pssc) file. This feature lets you create

customized and restricted sessions without writing a computer program. For more information about session configuration files, see about_Session_Configuration_Files

(About/about_Session_Configuration_Files.md).

Also, beginning in PowerShell 3.0, new note properties have been added to the session configuration object that `Get-PSSessionConfiguration` returns. These properties

make it easier for users and session configuration authors to examine and compare session configurations.

To create and register a session configuration, use the `Register-PSSessionConfiguration` cmdlet. For more information about session configurations, see

about_Session_Configurations (About/about_Session_Configurations.md).

PARAMETERS

-Force <System.Management.Automation.SwitchParameter>

Suppresses the prompt to restart the WinRM service, if the service is not already running.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Name <System.String[]>

Gets only the session configurations with the specified name or name pattern. Enter one or more session configuration names. Wildcards are permitted.

Required? false

Position? 0

Default value All session configurations on the local computer

Accept pipeline input? False

Accept wildcard characters? true

<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

None

You can't pipe objects to this cmdlet.

OUTPUTS

Microsoft.PowerShell.Commands.PSSessionConfigurationCommands#PSSessionConfiguration

NOTES

- To run this cmdlet, start PowerShell with the Run as administrator option.

- To view the session configurations on the computer, you must be a member of the Administrators group on the computer.

- To run a `Get-PSSessionConfiguration` command on a remote computer, Credential Security Service Provider (CredSSP) authentication must be enabled in the

client settings on the local computer (by using the `Enable-WSManCredSSP` cmdlet) and in the service settings on the remote computer. Also, you must use the

CredSSP value of the Authentication parameter when establishing the remote session. Otherwise, access is denied.

- The note properties of the object that `Get-PSSessionConfiguration` returns appear on the object only when they have a value. Only session configurations that

were created by using a session configuration file have all the defined properties.

- The properties of a session configuration object vary with the options set for the session configuration and the

values of those options. Also, session

configurations that use a session configuration file have additional properties.

- You can use commands in the WSMan: drive to change the properties of session configurations. However, you cannot use the WSMan: drive in PowerShell 2.0 to

change session configuration properties that are introduced in PowerShell 3.0, such as OutputBufferingMode . PowerShell 2.0 commands do not generate an error,

but they are ineffective. To change properties introduced in PowerShell 3.0, use the WSMan: drive in PowerShell 3.0.

- Example 1 - Get session configurations on the local computer -

```
Get-PSSessionConfiguration
```

---- Example 2 - Get the two default session configurations ----

```
Get-PSSessionConfiguration -Name Microsoft*

Name                 PSVersion  StartupScript       Permission
----                 ---------  -------------       ----------
microsoft.powershell     5.1                       BUILTIN\Administrators AccessAll...
microsoft.powershell32   5.1                       BUILTIN\Administrators AccessAll...
```

Example 3 - Get the properties and values of a session configuration

```
Get-PSSessionConfiguration -Name Full  | Format-List -Property *

Copyright                : (c) 2011 User01. All rights reserved.
AliasDefinitions         : {System.Collections.Hashtable}
SessionType              : Default
CompanyName              : Unknown
GUID                     : 1e9cb265-dae0-4bd3-89a9-8338a47698a1
```

Author : User01

ExecutionPolicy : Restricted

SchemaVersion : 1.0.0.0

LanguageMode : FullLanguage

Architecture : 64

Filename : %windir%\system32\pwrshplugin.dll

ResourceUri : http://schemas.microsoft.com/powershell/Full

MaxConcurrentCommandsPerShell : 1500

UseSharedProcess : false

ProcessIdleTimeoutSec : 0

xmlns : http://schemas.microsoft.com/wbem/wsman/1/config/PluginConfiguration

MaxConcurrentUsers : 10

lang : en-US

SupportsOptions : true

ExactMatch : true

configfilepath :

C:\WINDOWS\System32\WindowsPowerShell\v1.0\SessionConfig\Full_1e9cb265-dae0-4bd3-89a9-8338a47698a1.pssc

RunAsUser :

IdleTimeoutms : 7200000

PSVersion : 3.0

OutputBufferingMode : Block

AutoRestart : false

MaxShells : 300

MaxMemoryPerShellMB : 1024

MaxIdleTimeoutms : 43200000

SDKVersion : 1

Name : Full

XmlRenderingType : text

Capability : {Shell}

RunAsPassword :

MaxProcessesPerShell : 25

Enabled : True

MaxShellsPerUser : 30

Permission :

The example uses the `Get-PSSessionConfiguration` cmdlet to get the full session configuration. A pipeline operator sends the Full session configuration to the

`Format-List` cmdlet. The Property parameter with a value of `*` (all) directs `Format-List` to display all the properties and values of the object in a list.

The output includes useful information, including the author of the session configuration, the session type, language mode, and execution policy of sessions that are

created with this session configuration, session quotas, and the full path to the session configuration file.

This view of a session configuration is used for sessions that include a session configuration file. For more information about session configuration files, see

about_Session_Configuration_Files (About/about_Session_Configuration_Files.md).

Example 4 - Another way to look at the session configurations

```
dir wsman:\localhost\plugin

Type            Keys                            Name
----            ----                            ----
Container       {Name=Event Forwarding Plugin}      Event Forwarding Plugin
Container       {Name=Full}                         Full
Container       {Name=microsoft.powershell}         microsoft.powershell
Container       {Name=microsoft.powershell.workf... microsoft.powershell.workflow
Container       {Name=microsoft.powershell32}       microsoft.powershell32
Container       {Name=microsoft.ServerManager}      microsoft.ServerManager
Container       {Name=WMI Provider}                 WMI Provider
```

The PlugIn node contains ContainerElement objects (Microsoft.WSMan.Management.WSManConfigContainerElement) that represent the registered PowerShell session

configurations, along with other plug-ins for WS-Management.

- Example 6 - View session configurations on a remote computer -

```
Connect-WSMan -ComputerName Server01
dir WSMan:\Server01\Plugin

WSManConfig: Microsoft.WSMan.Management\WSMan::localhost\Plugin

Type            Keys                            Name
----            ----                            ----
Container       {Name=Empty}                    Empty
Container       {Name=Event Forwarding Plugin}  Event Forwarding Plugin
Container       {Name=Full}                     Full
Container       {Name=microsoft.powershell}     microsoft.powershell
Container       {Name=microsoft.powershell.workf... microsoft.powershell.workflow
Container       {Name=microsoft.powershell32}   microsoft.powershell32
Container       {Name=microsoft.ServerManager}  microsoft.ServerManager
Container       {Name=NoLanguage}               NoLanguage
Container       {Name=RestrictedLang}           RestrictedLang
Container       {Name=RRS}                      RRS
Container       {Name=SEL Plugin}               SEL Plugin
Container       {Name=WithProfile}              WithProfile
Container       {Name=WMI Provider}             WMI Provider
```

The `Connect-WSMan` cmdlet connects to the WinRM service on the Server01 remote computer. The `Get-ChildItem` cmdlet (alias `dir`) of the WSMan: drive gets the items in the Server01\Plugin path. The output shows the items in the Plugin directory on the Server01 computer. The items include the session configurations, which are a type of WSMan plug-in, along with other types of plug-ins on the computer.

Example 7 - Get detailed session configurations from a remote computer

```
Enable-WSManCredSSP -Delegate Server02
Connect-WSMan Server02
Set-Item WSMan:\Server02*\Service\Auth\CredSSP -Value $true
Invoke-Command -ScriptBlock {Get-PSSessionConfiguration} -ComputerName Server02 -Authentication CredSSP
-Credential Domain01\Admin01
```

```
Name                 PSVersion  StartupScript       Permission                    PSComputerName
----                 ---------  -------------       ----------                    --------------
microsoft.powershell 5.1                            BUILTIN\Administrators AccessAll... server02.corp.fabrikam.com
microsoft.powershell32 5.1                          BUILTIN\Administrators AccessAll... server02.corp.fabrikam.com
MyX86Shell           5.1        c:\test\x86Shell.ps1 BUILTIN\Administrators AccessAll... server02.corp.fabrikam.com
```

The `Enable-WSManCredSSP` cmdlet enables CredSSP delegation on Server01, the local computer. The `Connect-WSMan` cmdlet connects to Server02 computer. This action

adds a node for Server02 to the WSMan: drive on the local computer, allowing you to view and change the WS-Management settings on the Server02 computer. The

`Set-Item` cmdlet changes the value of the CredSSP item in the Service node of the Server02 computer to True . This configures the service settings on the remote

computer. The `Invoke-Command` cmdlet runs the`Get-PSSessionConfiguration` command on the Server02 computer. The command uses the Credential parameter, and it uses

the Authentication parameter with a value of CredSSP . The output shows the session configurations on the Server02 remote computer.

- Example 8 - Get the resource URI of a session configuration -

```
(Get-PSSessionConfiguration -Name CustomShell).resourceURI
```

```
http://schemas.microsoft.com/powershell/microsoft.CustomShell
```

The `$PSSessionConfigurationName` variable specifies the default configuration that is used when you create a session. This variable is set on the local computer, but

it specifies a configuration on the remote computer. For more information about the `$PSSessionConfiguration` variable, see about_Preference_Variables

(About/about_Preference_Variables.md).

RELATED LINKS

Online Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.core/get-pssessionconfiguration?view=powershell-5.1&WT.mc_id=ps-gethelp

Disable-PSSessionConfiguration

Enable-PSSessionConfiguration

Get-PSSessionConfiguration

New-PSSessionConfigurationFile

New-PSSessionOption

Register-PSSessionConfiguration

Set-PSSessionConfiguration

Test-PSSessionConfigurationFile

Unregister-PSSessionConfiguration

WSMan Provider

about_Session_Configurations

about_Session_Configuration_Files