

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 'Test-DscConfiguration'

PS:\>Get-HELP Test-DscConfiguration -Full

NAME

Test-DscConfiguration

SYNOPSIS

Tests whether the actual configuration on the nodes matches the desired configuration.

SYNTAX

Test-DscConfiguration [-Path] <System.String> [-AsJob] -CimSession <Microsoft.Management.Infrastructure.CimSession[]> [-ThrottleLimit <System.Int32>] [<CommonParameters>]

Test-DscConfiguration [-AsJob] -CimSession <Microsoft.Management.Infrastructure.CimSession[]> -ReferenceConfiguration <System.String> [-ThrottleLimit <System.Int32>]

[<CommonParameters>]

Test-DscConfiguration [-AsJob] -CimSession <Microsoft.Management.Infrastructure.CimSession[]> [-Detailed] [-ThrottleLimit <System.Int32>] [<CommonParameters>]

<System.Management.Automation.PSCredential>] [-Detailed] [-ThrottleLimit

<System.Int32>] [<CommonParameters>]

Test-DscConfiguration [-Path] <System.String> [[-ComputerName] <System.String[]>] [-AsJob] [-Credential <System.Management.Automation.PSCredential>] [-ThrottleLimit

<System.Int32>] [<CommonParameters>]

Test-DscConfiguration [[-ComputerName] <System.String[]>] [-AsJob] [-Credential <System.Management.Automation.PSCredential>] -ReferenceConfiguration <System.String>

[-ThrottleLimit <System.Int32>] [<CommonParameters>]

DESCRIPTION

The `Test-DscConfiguration` cmdlet tests whether the actual configuration on the nodes matches the desired configuration. Specify which computers for which you want

to test configurations by using computer names or Common Information Model (CIM) sessions. If you do not specify a target computer, the cmdlet tests configuration of

the local computer.

If the desired and actual configurations match, the cmdlet returns a string value of 'True'. Otherwise, it returns a string value of 'False'.

PARAMETERS

-AsJob <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet runs the command as a background job.

If you specify the AsJob parameter, the command returns an object that represents the job, and then displays the command prompt. You can continue to work in the

session until the job finishes. The job is created on the local computer and the results from remote computers are automatically returned to the local computer.

To manage the job, use the Job cmdlets. To get the job results, use the `Receive-Job` cmdlet.

To use this parameter, the local and remote computers must be configured for remoting, and on Windows Vista and later versions of the Windows operating system,

you must open Windows PowerShell with the Run as administrator option. For more information, see about_Remote_Requirements

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

For more information about Windows PowerShell background jobs, see about_Jobs (../Microsoft.PowerShell.Core/About/about_Jobs.md)and about_Remote_Jobs

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

Required?	false
Position?	named
Default value	False
Accept pipeline ir	nput? False
Accept wildcard o	characters? false

-CimSession <Microsoft.Management.Infrastructure.CimSession[]>

Runs the cmdlet in a remote session or on a remote computer. Enter a computer name or a session object, such as the output of a New-CimSession

(/powershell/module/cimcmdlets/new-cimsession)or Get-CimSession

(/powershell/module/cimcmdlets/get-cimsession)cmdlet. The default is the current session on the

local computer.

Required? true Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-ComputerName <System.String[]>

Specifies an array of computer names on which this cmdlet tests the configuration. The cmdlet tests the configuration document in the location specified by the

Path parameter to these computers.

Required? false

Position? 1

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Credential <System.Management.Automation.PSCredential>

Specifies a user name and password, as a PSCredential object, for the target computer. To obtain a PSCredential object, use the `Get-Credential` cmdlet. For more

information, type `Get-Help Get-Credential`.

Required?	false	
Position?	named	
Default value	None	
Accept pipeline i	nput? False	
Accept wildcard characters? false		

-Detailed <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet returns a detailed result of comparing the configuration document with the desired state of the nodes. The result includes information

such as overall state, resources that are in the desired state, resources that are not in desired state, and computer name.

Required?	false	
Position?	named	
Default value	False	
Accept pipeline i	nput? False	
Accept wildcard characters? false		

-Path <System.String>

Specifies the path of a folder that contains configuration document files. The cmdlet tests the configuration against the

the ComputerName or CimSession parameter.

Required? true

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ReferenceConfiguration <System.String>

Specifies the path of the configuration document file. This cmdlet tests the configuration against the actual state of computers specified by the ComputerName or

CimSession parameter.

Required?	true	
Position?	named	
Default value	None	
Accept pipeline in	put? False	
Accept wildcard characters? false		

-ThrottleLimit <System.Int32>

Specifies the maximum number of concurrent operations that can be established to run the cmdlet. If this parameter is omitted or a value of `0` is entered, then

Windows PowerShell calculates an optimum throttle limit for the cmdlet based on the number of CIM cmdlets that are running on the computer. The throttle limit

applies only to the current cmdlet, not to the session or to the computer.

Required?	false	
Position?	named	
Default value	None	
Accept pipeline in	nput? False	
Accept wildcard characters? false		

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

OUTPUTS

NOTES

----- Example 1: Test configuration for the local computer -----

Test-DscConfiguration

This command tests configuration for the local computer.

---- Example 2: Test configuration for a specified computer ----

\$Session = New-CimSession -ComputerName "Server01" -Credential ACCOUNTS\PattiFuller

Test-DscConfiguration -CimSession \$Session

This example test configuration from a computer specified by a CIM session. The example creates a CIM session for a computer named Server01 for use with the cmdlet.

Alternatively, create an array of CIM sessions to apply the cmdlet to multiple specified computers.

The first command creates a CIM session by using the `New-CimSession` cmdlet, and then stores the CimSession object

in the `\$Session` variable. The command prompts

you for a password. For more information, type `Get-Help New-CimSession`.

The second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the CimSession objects stored in the second command tests configuration for the computers identified by the cimSession objects stored in the second command tests configuration for the computers identified by the cimSession objects stored in the second command tests configuration for the computers identified by the cimSession objects stored in the second command tests configuration for the computers identified by the cimSession objects stored in the second command tests configuration for the computers identified by the cimSession objects stored command tests configuration for the computers identified by the cimSession objects stored command tests configuration for the computers identified by the cimSession objects stored command tests configuration for the computers identified by the cimSession objects stored command tes

variable, in this case, the computer named

Server01.

----- Example 3: Test configurations with detailed results -----

Test-DscConfiguration -ComputerName "Server01", "Server02", "Server03" -Detailed

This command tests configurations for a set of computers specified by the ComputerName parameter and returns detailed information that includes the overall state,

resources that are in the desired state, resources that are not in the desired state and computer name.

----- Example 4: Test configurations specified in a folder -----

Test-DscConfiguration -Path "C:\Dsc\Configurations"

This command tests configurations that are defined in a folder specified by the Path parameter. The configurations are tested against a set of computers, each

identified by the file name of the configuration file.

----- Example 5: Test configurations specified in a file ------

Test-DscConfiguration -ReferenceConfiguration "C:\Dsc\Configurations\WebServer.mof" -ComputerName "Server01", "Server02", "Server03"

This command tests a configuration defined in a file against a set of computers specified by the ComputerName parameter.

RELATED LINKS

Online Version:

https://docs.microsoft.com/powershell/module/psdesiredstateconfiguration/test-dscconfiguration?view=powershell-5.1&WT. mc_id=ps-gethelp Windows PowerShell Desired State Configuration Overview Get-DscConfiguration Get-DscConfigurationStatus Restore-DscConfiguration

Start-DscConfiguration