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

PowerShell Get-Help on command 'Get-NetTCPSetting'

PS C:\Users\wahid> Get-Help Get-NetTCPSetting

NAME

    Get-NetTCPSetting

SYNOPSIS

    Gets information about TCP settings and configuration.

SYNTAX

    Get-NetTCPSetting [[-SettingName] <String[]>] [-AsJob]

    [-AssociatedTransportFilter <CimInstance>] [-AutoReusePortRangeNumberOfPorts

    <UInt16[]>] [-AutoReusePortRangeStartPort <UInt16[]>]

    [-AutoTuningLevelEffective {Local | GroupPolicy}] [-AutoTuningLevelGroupPolicy

    {Disabled | HighlyRestricted | Restricted | Normal | Experimental |

    NotConfigured | NotChanged}] [-AutoTuningLevelLocal {Disabled |

    HighlyRestricted | Restricted | Normal | Experimental}] [-AutomaticUseCustom

    {Disabled | Enabled}] [-CimSession <CimSession[]>] [-CongestionProvider

    {Default | CTCP | DCTCP}] [-CwndRestart {False | True}] [-DelayedAckFrequency

    <Byte[]>] [-DelayedAckTimeoutMs <UInt32[]>] [-DynamicPortRangeNumberOfPorts

    <UInt16[]>] [-DynamicPortRangeStartPort <UInt16[]>] [-EcnCapability {Disabled

    | Enabled}] [-ForceWS {Disabled | Enabled}] [-InitialCongestionWindowMss

    <UInt32[]>] [-InitialRtoMs <UInt32[]>] [-MaxSynRetransmissions <Byte[]>]

[-MemoryPressureProtection {Disabled | Enabled | Default}] [-MinRtoMs
<UInt32[]>] [-NonSackRttResiliency {Disabled | Enabled}] [-ScalingHeuristics
{Disabled | Enabled}] [-ThrottleLimit <Int32>] [-Timestamps {Disabled |
Enabled}] [<CommonParameters>]

DESCRIPTION

The Get-NetTCPSetting cmdlet gets TCP settings. TCP settings are optimized for
different network conditions including latency and congestion. To apply a TCP
setting to a port number or destination IP address range, create a transport
filter by using the New-NetTransportFilter cmdlet.

Specify a setting to get by using the SettingName parameter, or by specifying
a transport filter that is associated with a setting. Specify parameter values
to determine which settings to get, or do not include any parameters to get
all TCP settings.

PARAMETERS

-AsJob [<SwitchParameter>]

Runs the cmdlet as a background job. Use this parameter to run commands
that take a long time to complete.

-AssociatedTransportFilter <CimInstance>

Specifies CIM object for a transport filter associated with a setting. The
cmdlet gets the setting for this transport filter. To obtain a transport
filter, use the Get-NetTransportFilter cmdlet.

-AutoReusePortRangeNumberOfPorts <UInt16[]>

Specifies the number of ports for the auto-reuse port range, which is a
port range used for local ephemeral port selection by outbound TCP
connections for which either SO_REUSE_UNICASTPORT has been set on the
socket, or for which connect() has been called without calling bind() on

the socket.

If you specify 0, the auto-reuse feature is disabled and ephemeral ports
are drawn instead from the dynamic port range as specified by
DynamicPortRangeStartPort and DynamicPortRangeNumberOfPorts , even if
SO_REUSE_UNICASTPORT is set on a socket.

-AutoReusePortRangeStartPort <UInt16[]>
   Specifies the number of ports for the auto-reuse port range, which is a
   port range used for local ephemeral port selection by outbound TCP
   connections for which either SO_REUSE_UNICASTPORT has been set on the
   socket, or for which connect() has been called without calling bind() on
   the socket.

   If you specify 0, the auto-reuse feature is disabled and ephemeral ports
   are drawn instead from the dynamic port range as specified by
   DynamicPortRangeStartPort and DynamicPortRangeNumberOfPorts , even if
   SO_REUSE_UNICASTPORT is set on a socket.

-AutoTuningLevelEffective <AutoTuningLevelEffective[]>
   Specifies an array of values of the TCP auto-tuning level effective
   property. The cmdlet gets settings that the values that you specify. This
   property can improve throughput for high thoughput, high latency networks.
   The acceptable values for this parameter are:

   - Local

   - GroupPolicy

-AutoTuningLevelGroupPolicy <AutoTuningLevelGroupPolicy[]>
   Specifies an array of values for the group policy for the auto-tuning
   level. The cmdlet gets settings that have the values that you specify.
   This property determines the setting of the TCP auto-tuning level that

group policy configures. A group policy applies uniformly across all
NetTcpSetting objects. The auto-tuning level can improve throughput for
high thoughput, high latency networks. The acceptable values for this
parameter are:

- Disabled

- HighlyRestricted

- Restricted

- Normal

- Experimental

- NotConfigured

- NotChanged

-AutoTuningLevelLocal <AutoTuningLevelLocal[]>
   Specifies an array of TCP auto-tuning levels for the host computer. The
   cmdlet gets settings that have the values that you specify. TCP
   auto-tuning can improve throughput on high throughput, high latency
   networks. The acceptable values for this parameter are:

   - Disabled. Sets the TCP receive window to the default value. -
   HighlyRestricted. Sets the TCP receive window to grow beyond the default
   value, but very conservatively. - Restricted. Sets the TCP receive window
   to grow beyond the default value, but less conservatively than
   HighlyRestricted. - Normal. Sets the TCP receive window to grow to
   accommodate almost all scenarios. - Experimental. Sets the TCP receive
   window to grow to accommodate extreme scenarios.

-AutomaticUseCustom <AutomaticUseCustom[]>

Specifies an array of values for whether the automatic profile assigns a

custom template, either Datacenter Custom or Internet Custom, to a

connection. The cmdlet gets settings that have the values you specify. The

acceptable values for this parameter are:

- True

- False

-CimSession <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

(https://go.microsoft.com/fwlink/p/?LinkId=227967) or

[Get-CimSession](https://go.microsoft.com/fwlink/p/?LinkId=227966)cmdlet.

The default is the current session on the local computer.

-CongestionProvider <CongestionProvider[]>

Specifies an array of congestion provider properties that TCP uses. The

cmdlet gets settings that have the congestion provider values that you

specify. The acceptable values for this parameter are:

- CTCP. Compound TCP increases the receive window and amount of data sent.

This value can improve throughput on higher latency connections. - DCTCP.

Data Center TCP adjusts the TCP window based on network congestion

feedback based on Explicit Congestion Notification (ECN) signaling. This

value can improve throughput on low latency links. - Default. Servers use

Data Center TCP by default. Clients use CUBIC (https://techcommunity.micros

oft.com/t5/Networking-Blog/Top-10-Networking-Features-in-Windows-Server-201

9-8-A-Faster/ba-p/339749).

-CwndRestart <CwndRestart[]>

Specifies an array of values that determine whether to enable congestion

window restart. The cmdlet gets settings that have the congestion window restart values that you specify. Congestion window restart can avoid slow start to optimize throughput on low latency networks. For more information about congestion window restart, see RFC 2581 (http://www.ietf.org/rfc/rfc2581.txt). The acceptable values for this parameter are:

- True. TCP uses congestion window restart. - False. TCP uses the default setting of the connection.

-DelayedAckFrequency <Byte[]>
   Specifies an array of numbers of acknowledgments (ACKs) received before the computer sends a response. The cmdlet gets settings that have the delayed ACK frequency values that you specify.

-DelayedAckTimeoutMs <UInt32[]>
   Specifies an array of the time to wait, in milliseconds, before the computer sends an ACK if the computer receives fewer than delayed acknowledgment frequency of packets. The cmdlet gets settings that have the delayed ACK timeout values that you specify. Use the DelayedAckFrequency parameter to specify the delayed ACK frequency value. Reducing the time to wait can increase throughput on low latency networks by accelerating growth in TCP window size. The acceptable values for this parameter are: increments of 10, from 10 through 600.

-DynamicPortRangeNumberOfPorts <UInt16[]>
   Specifies an array of the number of ports for the dynamic port range that starts from the port that you specify for the DynamicPortRangeStartPort parameter. The cmdlet gets settings that have the numbers of ports that you specify.

-DynamicPortRangeStartPort <UInt16[]>
   Specifies an array of starting ports for dynamic port ranges. The

acceptable values for this parameter are: 1 through 65535. The cmdlet gets settings that have the starting ports that you specify.

-EcnCapability <EcnCapability[]>

Specifies an array of values that determine whether to enable ECN capability. The cmdlet gets settings that have ECN capability value that you specify. The acceptable values for this parameter are:

- Enabled. Uses ECN capability. - Disabled. Does not use ECN capability.

-ForceWS <ForceWS[]>

Specifies an array of values that determine whether to force window scaling for retransmission. The cmdlet gets settings that have the force window scaling value that you specify. The acceptable values for this parameter are:

- Enabled. Requires window scaling for retransmission. - Disabled. Does not require window scaling for retransmission.

-InitialCongestionWindowMss <UInt32[]>

Specifies an array of initial sizes of the congestion window. The cmdlet gets settings that have the initial congestion window value that you specify. Provide a value to multiply by the maximum segment size (MMS). The acceptable values for this parameter are: an even number from 2 through 64.

-InitialRtoMs <UInt32[]>

Specifies the period, in milliseconds, before connect, or SYN, retransmit. The cmdlet gets settings that have the initial connect retransmit values that you specify. The acceptable values for this parameter are: increments of 10, from 300 ms through 3000 ms.

-MaxSynRetransmissions <Byte[]>

Specifies the maximum number of times the computers sends SYN packets without receiving a response. The cmdlet gets settings that have the maximum values that you specify.

-MemoryPressureProtection <MemoryPressureProtection[]>

Specifies an array of values that determines whether to use memory pressure protection. The cmdlet gets settings that have the memory pressure protection values that you specify. TCP memory pressure protection helps ensure that a computer continues normal operation when low on memory due to denial of service attacks. The acceptable values for this parameter are:

- Enabled. When low on memory, during an attack, close existing TCP connections and drop incoming SYN requests. - Disabled. Do not use memory pressure protection. - Default. Use the computer default value for memory pressure protection.

-MinRtoMs <UInt32[]>

Specifies an array of values, in milliseconds, for the TCP retransmission to time out. The cmdlet gets settings that have the timeout values that you specify. The acceptable values for this parameter are: increments of 10, from 20 ms through 300 ms.

-NonSackRttResiliency <NonSackRttResiliency[]>

Specifies an array of values that determine whether to enable round trip time resiliency for clients that do not support selective acknowledgment. The cmdlet gets settings that have the enable round trip time resiliency values that you specify. The acceptable values for this parameter are:

- Enabled.

- Disabled.

-ScalingHeuristics <ScalingHeuristics[]>

    Specifies an array of values that determine whether to enable scaling

    heuristics. The cmdlet gets settings that have scaling heuristics values

    that you specify. The acceptable values for this parameter are:

    - Enabled

    - Disabled

-SettingName <String[]>

    Specifies an array of setting names. The acceptable values for this

    parameter are:

    - Internet. Optimized for networks with higher latency and lower

    throughput. - Datacenter. Optimized for networks with lower latency and

    higher throughput. - Compat. Optimized for compatibility with legacy

    network equipment. - Custom. Custom settings. - Automatic. The computer

    uses latency to select either Internet or Datacenter.

-ThrottleLimit <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 PowerShellr 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.

-Timestamps <Timestamps[]>

    Specifies an array of values that determine whether to enable timestamps.

    The cmdlet gets settings that have the enable timestamps values that you

    specify. Timestamps facilitate round trip measurement, and can help

    protect against wrapped sequence numbers on high throughput links. For

    more information about TCP timestamps, see RFC 1323

(http://www.ietf.org/rfc/rfc1323.txt). The acceptable values for this

parameter are:

- Enabled

- Disabled

<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: Get all TCP settings ---------------

PS C:\>Get-NetTCPSetting

This command gets TCP settings. The command specifies no parameters, so it

gets all TCP settings.
------------- Example 2: Get Internet TCP settings -------------

PS C:\>Get-NetTCPSetting -Setting Internet

This command gets TCP settings that are optimized for connectivity across the

Internet.
------------ Example 3: View formatted TCP settings ------------

PS C:\>Get-NetTcpSetting | Format-Table

This command gets TCP settings, and then passes them to the Format-Table

cmdlet by using the pipeline operator. That cmdlet displays the results as a

summary in a table view. For more information, type `Get-Help Format-Table`.

REMARKS

To see the examples, type: "get-help Get-NetTCPSetting -examples".

For more information, type: "get-help Get-NetTCPSetting -detailed".

For technical information, type: "get-help Get-NetTCPSetting -full".

For online help, type: "get-help Get-NetTCPSetting -online"