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

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

NAME

Get-PcsvDeviceLog

SYNOPSIS

Gets System Event Log entries from a PCSV device.

SYNTAX

Get-PcsvDeviceLog [-TargetAddress] <String> [-Credential] <PSCredential> [-ManagementProtocol] {WSMan | IPMI} [[-Port] <UInt16>] [-AsJob] [-Authentication {Default | Basic | Digest}] [-CimSession <CimSession[]>] [-SkipCACheck] [-SkipCNCheck] [-SkipRevocationCheck] [-ThrottleLimit <Int32>] [-TimeoutSec <UInt32>] [-UseSSL] [<CommonParameters>]

Get-PcsvDeviceLog [-AsJob] [-CimSession <CimSession[]>] -InputObject <CimInstance[]> [-ThrottleLimit <Int32>] [<CommonParameters>]

Get-PcsvDeviceLog [-AsJob] [-CimSession <CimSession[]>] [-ThrottleLimit <Int32>] [-TimeoutSec <UInt32>] [<CommonParameters>]

DESCRIPTION

The Get-PcsvDeviceLog cmdlet gets System Event Log entries from a Physical Computer System View (PCSV) device . This cmdlet returns each entry as an

MSFT_PCSVLogRecord object. This cmdlet currently supports devices that use the Intelligent Platform Management Interface (IPMI) protocol. You can use this cmdlet for

both in-band and out-of-band connections. To use this cmdlet with an in-band connection, you must have elevated privileges.

PARAMETERS

-AsJob [<SwitchParameter>]

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

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Authentication <Authentication>

Specifies an authentication method to use for devices managed by WS-Management. Do not specify this parameter for devices managed by using IPMI. The acceptable

values for this parameter are:

- Basic

- Digest

- Default

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? 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.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Credential <PSCredential>

Specifies a PSCredential object based on a user name and password. To obtain a PSCredential object, use the Get-Credential cmdlet. For more information, type

`Get-Help Get-Credential`. For IPMI devices, specify credentials that correspond to a user with Administrator privileges on the device.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-InputObject <CimInstance[]>

Specifies the input object that is used in a pipeline command.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-ManagementProtocol <ManagementProtocol>

Specifies a management protocol used to communicate with a device. The acceptable values for this parameter are:

- WSMAN

- IPMI

This cmdlet currently supports only devices that use the IPMI protocol.

Required? true

Position? 3

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Port <UInt16>

Specifies a port on the remote computer to use for the management connection. If you do not specify a port, the cmdlet uses the following default ports:

- IPMI and WSMAN over HTTP. Port 623. - WSMAN over HTTPS. Port 664.

Required? false

Position? 4

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SkipCACheck [<SwitchParameter>]

Indicates that the client connects by using HTTPS without validating that a trusted certification authority (CA) signed the server certificate. Do not specify

this parameter if you specify a value of IPMI for the ManagementProtocol parameter.

Do not specify this parameter unless you can establish trust in another way, such as if the remote computer is part of a network that is physically secure and

isolated, or if the remote computer is a trusted host in a Windows Remote Management (WinRM) configuration.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SkipCNCheck [<SwitchParameter>]

Indicates that the certificate common name (CN) of the server does not need to match the host name of the server. Do not specify this parameter if you specify a

value of IPMI for the ManagementProtocol parameter.

Specify this parameter only for managing devices by using WSMAN over HTTPS. Be sure to specify this parameter only for trusted computers.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SkipRevocationCheck [<SwitchParameter>]

Indicates that the cmdlet skips the revocation check of server certificates.

Be sure to specify this parameter only for trusted computers.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-TargetAddress <String>

Specifies the name or IP address of the remote hardware device.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-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.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-TimeoutSec <UInt32>

Specifies how long to wait, in seconds, for a response from the remote hardware device. After this period, the cmdlet abandons the connection attempt.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-UseSSL [<SwitchParameter>]

Indicates that the server connects to the target computer by using SSL. WSMAN encrypts all content transmitted over the network. Specify this parameter to use the

additional protection of HTTPS instead of HTTP. If you specify this parameter and SSL is not available on the connection port, the command fails.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

<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

System.String

System.Management.Automation.PSCredential

System.UInt16

Microsoft.PowerShell.Cmdletization.GeneratedTypes.PcsvDevice.Authentication

System.Management.Automation.SwitchParameter

System.UInt32

Microsoft.Management.Infrastructure.CimInstance[]

OUTPUTS

Microsoft.Management.Infrastructure.CimInstance[]

Microsoft.Management.Infrastructure.CimInstance#root/Microsoft/Windows/HardwareManagement/MSFT_PCSVLogRecord

This cmdlet returns instances of MSFT_PCSVLogRecord objects. This object contains the following properties.

- InstanceID . String - Caption . String - Description . String - ElementName . String - RecordFormat . String - RecordData . String - Locale . String -

PerceivedSeverity . String - LogCreationClassName . String - LogName . String - CreationClassName . String - RecordID . String - MessageTimestamp . DateTime -

DataFormat . String - RawData . Byte[]

NOTES

---------- Example 1: Get log entries for a device ----------

PS C:\>$Credential = Get-Credential Admin

PS C:\> Get-PcsvDeviceLog -TargetAddress "10.1.2.3" -Credential $Credential -ManagementProtocol IPMI

Caption : Sensor 10 event with offset 02 (Event Logging Disabled)

Description : (I2C Slave Addr 16, Channel 0, LUN 0): Sensor 10 event with offset 02 (Event Logging Disabled)

ElementName : PCSV Device Log Record

InstanceID : SEL Record Instance 1

Locale :

PerceivedSeverity :

RecordData : *0001*02*5463B8A0*0020*04*10*72*Assertion*6F*02*FF*FF

RecordFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint16 generatorID*uint8 evmRev*uint8 sensorType*uint8 sensorNumber*string

assertionDirection*uint8 eventType*uint8 eventData1*uint8 eventData2*uint8 eventData3

CreationClassName : MSFT_PCSVLogRecord

DataFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint16 generatorID*uint8 evmRev*uint8 sensorType*uint8 sensorNumber*string

assertionDirection*uint8 eventType*uint8 eventData1*uint8 eventData2*uint8 eventData3

LogCreationClassName : MSFT_PCSVLogRecord

LogName : PCSV Device Log

MessageTimestamp : 11/12/2014 7:44:32 PM

RecordID : 1

RawData : {1, 0, 2, 160...}

PSComputerName :

Caption : OS Boot

Description : OS Boot

ElementName : PCSV Device Log Record

InstanceID : SEL Record Instance 13

Locale :

PerceivedSeverity :

RecordData : *000D*DC*546B28B9*37 01 00*00 B5 28 6B 54 00

RecordFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint8[3] manufacturerID*uint8[6] oemData

CreationClassName : MSFT_PCSVLogRecord

DataFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint8[3] manufacturerID*uint8[6] oemData

LogCreationClassName : MSFT_PCSVLogRecord

LogName : PCSV Device Log

MessageTimestamp : 11/18/2014 11:08:41 AM

RecordID : 13

RawData : {13, 0, 220, 185...}

PSComputerName :

The first command uses the Get-Credential cmdlet to create a credential, and then stores it in the $Credential variable. The cmdlet prompts you for a user name and

password. For more information, type `Get-Help Get-Credential`.

The second command gets the System Event Log entries on the PCSV device that has the specified IP address. The command uses the credential stored in $Credential .

Example 2: Get log entries for a device by using the pipeline

PS C:\>$Credential = Get-Credential Admin

PS C:\> Get-PcsvDevice -TargetAddress "10.1.2.3" -Credential $Credential -ManagementProtocol IPMI | Get-PcsvDeviceLog

Caption : Sensor 10 event with offset 02 (Event Logging Disabled)

Description : (I2C Slave Addr 16, Channel 0, LUN 0): Sensor 10 event with offset 02 (Event Logging Disabled)

ElementName : PCSV Device Log Record

InstanceID : SEL Record Instance 1

Locale :

PerceivedSeverity :

RecordData : *0001*02*5463B8A0*0020*04*10*72*Assertion*6F*02*FF*FF

RecordFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint16 generatorID*uint8 evmRev*uint8

sensorType*uint8 sensorNumber*string

assertionDirection*uint8 eventType*uint8 eventData1*uint8 eventData2*uint8 eventData3

CreationClassName : MSFT_PCSVLogRecord

DataFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint16 generatorID*uint8 evmRev*uint8 sensorType*uint8 sensorNumber*string

assertionDirection*uint8 eventType*uint8 eventData1*uint8 eventData2*uint8 eventData3

LogCreationClassName : MSFT_PCSVLogRecord

LogName : PCSV Device Log

MessageTimestamp : 11/12/2014 7:44:32 PM

RecordID : 1

RawData : {1, 0, 2, 160...}

PSComputerName :

Caption : OS Boot

Description : OS Boot

ElementName : PCSV Device Log Record

InstanceID : SEL Record Instance 13

Locale :

PerceivedSeverity :

RecordData : *000D*DC*546B28B9*37 01 00*00 B5 28 6B 54 00

RecordFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint8[3] manufacturerID*uint8[6] oemData

CreationClassName : MSFT_PCSVLogRecord

DataFormat : *uint16 recordID*uint8 recordType*uint32 timestamp*uint8[3] manufacturerID*uint8[6] oemData

LogCreationClassName : MSFT_PCSVLogRecord

LogName : PCSV Device Log

MessageTimestamp : 11/18/2014 11:08:41 AM

RecordID : 13

RawData : {13, 0, 220, 185...}

PSComputerName :

The first command uses Get-Credential to create a credential, and then stores it in the $Credential variable.

The second command uses the Get-PcsvDevice cmdlet to get the device that has the specified IP address. Pass the cmdlet

uses the credential stored in $Credential . The

command passes that device to the current cmdlet by using the pipeline operator. The current cmdlet gets the System Event Log entries on that device.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/pcsvdevice/get-pcsvdevicelog?view=windowsserver2022-ps&wt.mc_id=ps-gethelp

Clear-PcsvDeviceLog

Get-PcsvDevice