Learn Discover Product documentation Development languages Topics Sign in

PowerShell Overview DSC PowerShellGet Utility modules Module Browser API Browser Resources Download PowerShell

Version

PowerShell 5.1

Search

Get-ControlPanelItem
Get-EventLog
Get-HotFix
Get-Item
Get-ItemProperty
Get-ItemPropertyValue
Get-Location
Get-Process
Get-PSDrive
Get-PSProvider
Get-Service
Get-TimeZone
Get-Transaction
Get-WmiObject
Invoke-Item
Invoke-WmiMethod
Join-Path
Limit-EventLog
Move-Item
Move-ItemProperty
New-EventLog
New-Item
New-ItemProperty
New-PSDrive
New-Service
New-WebServiceProxy
Pop-Location
Push-Location
Register-WmiEvent
Remove-Computer
Remove-EventLog
Remove-Item
Remove-ItemProperty

Download PDF

Learn / PowerShell / Microsoft.PowerShell.Management /

Get-EventLog

Reference

Feedback

Module: Microsoft.PowerShell.Management

In this article

Syntax
Description
Examples
Parameters
Show 4 more

Gets the events in an event log, or a list of the event logs, on the local computer or remote computers.

Syntax

PowerShell Copy

```
Get-EventLog
   [-LogName] <String>
   [-ComputerName <String[]>]
   [-Newest <Int32>]
   [-After <DateTime>]
   [-Before <DateTime>]
   [-UserName <String[]>]
   [[-InstanceId] <Int64[]>]
   [-Index <Int32[]>]
   [-EntryType <String[]>]
   [-Source <String[]>]
   [-Message <String>]
   [-AsBaseObject]
   [<CommonParameters>]
```

PowerShell Copy

```
Get-EventLog
   [-ComputerName <String[]>]
   [-List]
   [-AsString]
   [<CommonParameters>]
```

Description

The `Get-EventLog` cmdlet gets events and event logs from local and remote computers. By default, `Get-EventLog` gets logs from the local computer. To get logs from remote computers, use the **ComputerName** parameter.

You can use the `Get-EventLog` parameters and property values to search for events. The cmdlet gets events that match the specified property values.

PowerShell cmdlets that contain the `EventLog` noun work only on Windows classic event logs such as Application, System, or Security. To get logs that use the Windows Event Log technology in Windows Vista and later Windows versions, use `Get-WinEvent`.

> ⓘ Note
>
> `Get-EventLog` uses a Win32 API that is deprecated. The results may not be accurate. Use the `Get-WinEvent` cmdlet instead.

Examples

Example 1: Get event logs on the local computer

This example displays the list of event logs that are available on the local computer. The names in the Log column are used with the **LogName** parameter to specify which log is searched for events.

PowerShell Copy

```
Get-EventLog -List

Max(K)   Retain   OverflowAction      Entries  Log
------   ------   --------------      -------  ---
15,168        0   OverwriteAsNeeded   20,792   Application
15,168        0   OverwriteAsNeeded   12,559   System
15,360        0   OverwriteAsNeeded   11,173   Windows PowerShell
```

The `Get-EventLog` cmdlet uses the **List** parameter to display the available logs.

Example 2: Get recent entries from an event log on the local computer

This example gets recent entries from the System event log.

PowerShell Copy

```
Get-EventLog -LogName System -Newest 5

Index   Time           EntryType     Source              InstanceID   Message
-----   ----           ---------     ------              ----------   -------
13820   Jan 17 19:16   Error         DCOM                     10016   The descript
13819   Jan 17 19:08   Error         DCOM                     10016   The descript
13818   Jan 17 19:06   Information   Service Control...  1073748864   The start ty
13817   Jan 17 19:05   Error         DCOM                     10016   The descript
13815   Jan 17 19:03   Information   Microsoft-Windows...        35   The time ser
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System event log. The **Newest** parameter returns the five most recent events.

Example 3: Find all sources for a specific number of entries in an event log

This example shows how to find all of the sources that are included in the 1000 most recent entries in the System event log.

PowerShell Copy

```
$Events = Get-EventLog -LogName System -Newest 1000
$Events | Group-Object -Property Source -NoElement | Sort-Object -Property Count

Count   Name
-----   ----
  110   DCOM
```

```
   65   Service Control Manager
   51   Microsoft-Windows-Kern...
   14   EventLog
   14   BTHUSB
   13   Win32k
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System log. The **Newest** parameter selects the 1000 most recent events. The event objects are stored in the `$Events` variable. The `$Events` objects are sent down the pipeline to the `Group-Object` cmdlet. `Group-Object` uses the **Property** parameter to group the objects by source and counts the number of objects for each source. The **NoElement** parameter removes the group members from the output. The `Sort-Object` cmdlet uses the **Property** parameter to sort by the count of each source name. The **Descending** parameter sorts the list in order by count from highest to lowest.

Example 4: Get error events from a specific event log

This example gets error events from the System event log.

PowerShell Copy

```
Get-EventLog -LogName System -EntryType Error

Index Time          EntryType   Source  InstanceID Message
----- ----          ---------   ------  ---------- -------
13296 Jan 16 13:53  Error       DCOM    10016 The description for Event ID '1001
13291 Jan 16 13:51  Error       DCOM    10016 The description for Event ID '1001
13245 Jan 16 11:45  Error       DCOM    10016 The description for Event ID '1001
13230 Jan 16 11:07  Error       DCOM    10016 The description for Event ID '1001
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System log. The **EntryType** parameter filters the events to show only Error events.

Example 5: Get events from an event log with an InstanceId and Source value

This example gets events from the System log for a specific InstanceId and Source.

PowerShell Copy

```
Get-EventLog -LogName System -InstanceId 10016 -Source DCOM

Index Time          EntryType  Source  InstanceID  Message
----- ----          ---------  ------  ----------  -------
13245 Jan 16 11:45  Error      DCOM         10016  The description for Event ID
13230 Jan 16 11:07  Error      DCOM         10016  The description for Event ID
13219 Jan 16 10:00  Error      DCOM         10016  The description for Event ID
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System log. The **InstanceID** parameter selects the events with the specified Instance ID. The **Source** parameter specifies the event property.

Example 6: Get events from multiple computers

This command gets the events from the System event log on three computers: Server01, Server02, and Server03.

PowerShell Copy

```
Get-EventLog -LogName System -ComputerName Server01, Server02, Server03
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System log. The **ComputerName** parameter uses a comma-separated string to list the computers from which you want to get the event logs.

Example 7: Get all events that include a specific word in the message

This command gets all the events in the System event log that contain a specific word in the event's message. It's possible that your specified **Message** parameter's value is included in the message's content but isn't displayed on the PowerShell console.

PowerShell

Copy

```
Get-EventLog -LogName System -Message *description*

Index Time          EntryType   Source       InstanceID   Message
----- ----          ---------   ------       ----------   -------
13821 Jan 17 19:17  Error       DCOM              10016   The description for Ev
13820 Jan 17 19:16  Error       DCOM              10016   The description for Ev
13819 Jan 17 19:08  Error       DCOM              10016   The description for Ev
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System event log. The **Message** parameter specifies a word to search for in the message field of each event.

Example 8: Display the property values of an event

This example shows how to display all of an event's properties and values.

PowerShell

Copy

```
$A = Get-EventLog -LogName System -Newest 1
$A | Select-Object -Property *

EventID            : 10016
MachineName        : localhost
Data               : {}
Index              : 13821
Category           : (0)
CategoryNumber     : 0
EntryType          : Error
Message            : The description for Event ID '10016' in Source 'DCOM'...
Source             : DCOM
ReplacementStrings : {Local,...}
InstanceId         : 10016
TimeGenerated      : 1/17/2019 19:17:23
TimeWritten        : 1/17/2019 19:17:23
UserName           : username
Site               :
Container          :
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System event log. The **Newest** parameter selects the most recent event object. The object is stored in the `$A` variable. The object in the `$A` variable is sent down the pipeline to the `Select-Object` cmdlet. `Select-Object` uses the **Property** parameter with an asterisk (`*`) to select all of the object's properties.

Example 9: Get events from an event log using a source and event ID

This example gets events for a specified Source and Event ID.

PowerShell

Copy

```
Get-EventLog -LogName Application -Source Outlook | Where-Object {$_.EventID -eq
            Select-Object -Property Source, EventID, InstanceId, Message
```

```
Source   EventID   InstanceId   Message
------   -------   ----------   -------
Outlook       63   1073741887   The Exchange web service request succeeded.
Outlook       63   1073741887   Outlook detected a change notification.
Outlook       63   1073741887   The Exchange web service request succeeded.
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the Application event log. The **Source** parameter specifies the application name, Outlook. The objects are sent down the pipeline to the `Where-Object` cmdlet. For each object in the pipeline, the `Where-Object` cmdlet uses the variable `$_.EventID` to compare the Event ID property to the specified value. The objects are sent down the pipeline to the `Select-Object` cmdlet. `Select-Object` uses the **Property** parameter to select the properties to display in the PowerShell console.

Example 10: Get events and group by a property

PowerShell Copy

```
Get-EventLog -LogName System -UserName NT* | Group-Object -Property UserName -Nc
              Select-Object -Property Count, Name

Count  Name
-----  ----
6031   NT AUTHORITY\SYSTEM
  42   NT AUTHORITY\LOCAL SERVICE
   4   NT AUTHORITY\NETWORK SERVICE
```

The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System log. The **UserName** parameter includes the asterisk (`*`) wildcard to specify a portion of the user name. The event objects are sent down the pipeline to the `Group-Object` cmdlet. `Group-Object` uses the **Property** parameter to specify that the **UserName** property is used to group the objects and count the number of objects for each user name. The **NoElement** parameter removes the group members from the output. The objects are sent down the pipeline to the `Select-Object` cmdlet. `Select-Object` uses the **Property** parameter to select the properties to display in the PowerShell console.

Example 11: Get events that occurred during a specific date and time range

This example gets Error events from the System event log for a specified date and time range. The **Before** and **After** parameters set the date and time range but are excluded from the output.

PowerShell Copy

```
$Begin = Get-Date -Date '1/17/2019 08:00:00'
$End = Get-Date -Date '1/17/2019 17:00:00'
Get-EventLog -LogName System -EntryType Error -After $Begin -Before $End

Index Time          EntryType   Source   InstanceID  Message
----- ----          ---------   ------   ----------  -------
13821 Jan 17 13:40  Error       DCOM          10016  The description for Event I
13820 Jan 17 13:11  Error       DCOM          10016  The description for Event I
...
12372 Jan 17 10:08  Error       DCOM          10016  The description for Event I
12371 Jan 17 09:04  Error       DCOM          10016  The description for Event I
```

The `Get-Date` cmdlet uses the **Date** parameter to specify a date and time. The **DateTime** objects are stored in the `$Begin` and `$End` variables. The `Get-EventLog` cmdlet uses the **LogName** parameter to specify the System log. The **EntryType** parameter specifies the Error event type. The date and time range is set by the **After** parameter and `$Begin` variable and the **Before** parameter and `$End` variable.

Parameters

`-After`

Gets events that occurred after a specified date and time. The **After** parameter date and time are excluded from the output. Enter a **DateTime** object, such as the value returned by the `Get-Date` cmdlet.

Expand table

Type:	DateTime
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-AsBaseObject`

Indicates that this cmdlet returns a standard **System.Diagnostics.EventLogEntry** object for each event. Without this parameter, `Get-EventLog` returns an extended **PSObject** object with additional **EventLogName**, **Source**, and **InstanceId** properties.

To see the effect of this parameter, pipe the events to the `Get-Member` cmdlet and examine the **TypeName** value in the result.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-AsString`

Indicates that this cmdlet returns the output as strings, instead of objects.

Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-Before`

Gets events that occurred before a specified date and time. The **Before** parameter date and time are excluded from the output. Enter a **DateTime** object, such as the value returned by the `Get-Date` cmdlet.

⛶ Expand table

Type:	DateTime
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-ComputerName`

This parameter specifies a remote computer's NetBIOS name, Internet Protocol (IP) address, or a fully qualified domain name (FQDN).

If the **ComputerName** parameter isn't specified, `Get-EventLog` defaults to the local computer. The parameter also accepts a dot (`.`) to specify the local computer.

The **ComputerName** parameter doesn't rely on Windows PowerShell remoting. You can use `Get-EventLog` with the **ComputerName** parameter even if your computer is not configured to run remote commands.

⛶ Expand table

Type:	String[]
Aliases:	Cn
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-EntryType`

Specifies, as a string array, the entry type of the events that this cmdlet gets.

The acceptable values for this parameter are:

- Error
- Information
- FailureAudit
- SuccessAudit
- Warning

⛶ Expand table

Type:	String[]
Aliases:	ET
Accepted values:	Error, Information, FailureAudit, SuccessAudit, Warning
Position:	Named

Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-Index

Specifies the index values to get from the event log. The parameter accepts a comma-separated string of values.

⛶ Expand table

Type:	Int32[]
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-InstanceId

Specifies the Instance IDs to get from the event log. The parameter accepts a comma-separated string of values.

⛶ Expand table

Type:	Int64[]
Position:	1
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-List

Displays the list of event logs on the computer.

⛶ Expand table

Type:	SwitchParameter
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

-LogName

Specifies the name of one event log. To find the log names use `Get-EventLog -List`. Wildcard characters are permitted. This parameter is required.

Expand table

Type:	String
Aliases:	LN
Position:	0
Default value:	None
Required:	True
Accept pipeline input:	False
Accept wildcard characters:	True

`-Message`

Specifies a string in the event message. You can use this parameter to search for messages that contain certain words or phrases. Wildcards are permitted.

Expand table

Type:	String
Aliases:	MSG
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	True

`-Newest`

Begins with the newest events and gets the specified number of events. The number of events is required, for example `-Newest 100`. Specifies the maximum number of events that are returned.

Expand table

Type:	Int32
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	False

`-Source`

Specifies, as a string array, sources that were written to the log that this cmdlet gets. Wildcards are permitted.

Expand table

Type:	String[]
Aliases:	ABO

Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	True

`-UserName`

Specifies, as a string array, user names that are associated with events. Enter names or name patterns, such as `User01`, `User*`, or `Domain01\User*`. Wildcards are permitted.

Expand table

Type:	String[]
Position:	Named
Default value:	None
Required:	False
Accept pipeline input:	False
Accept wildcard characters:	True

Inputs

None

You cannot pipe input to `Get-EventLog`.

Outputs

System.Diagnostics.EventLogEntry. System.Diagnostics.EventLog. System.String

If the **LogName** parameter is specified, the output is a collection of **System.Diagnostics.EventLogEntry** objects.

If only the **List** parameter is specified, the output is a collection of **System.Diagnostics.EventLog** objects.

If both the **List** and **AsString** parameters are specified, the output is a collection of **System.String** objects.

Notes

The cmdlets `Get-EventLog` and `Get-WinEvent` are not supported in the Windows Preinstallation Environment (Windows PE).

Related Links

- Clear-EventLog
- Get-WinEvent
- Group-Object
- Limit-EventLog
- New-EventLog
- Remove-EventLog

- Select-Object
- Show-EventLog
- Write-EventLog

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see our contributor guide.

PowerShell feedback

PowerShell is an open source project. Select a link to provide feedback:

Open a documentation issue

Provide product feedback