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 'Get-Event'*

***PS:\>Get-HELP Get-Event -Full***

NAME

Get-Event

SYNOPSIS

Gets the events in the event queue.

SYNTAX

```
    Get-Event [-EventIdentifier] <System.Int32> [<CommonParameters>]


    Get-Event [[-SourceIdentifier] <System.String>] [<CommonParameters>]
```

DESCRIPTION

The `Get-Event` cmdlet gets events in the PowerShell event queue for the current session. You can get all events or use the EventIdentifier or SourceIdentifier

parameter to specify the events.

When an event occurs, it is added to the event queue. The event queue includes events for which you have registered, events created by using the `New-Event` cmdlet, and the event that is raised when PowerShell exits. You can use `Get-Event` or `Wait-Event` to get the events.

This cmdlet does not get events from the Event Viewer logs. To get those events, use `Get-WinEvent` or `Get-EventLog`.

## PARAMETERS

-EventIdentifier <System.Int32>

Specifies the event identifiers for which this cmdlet gets events.

```
Required?                    true
Position?                    0
Default value                None
Accept pipeline input?       True (ByPropertyName)
Accept wildcard characters?  false
```

-SourceIdentifier <System.String>

Specifies source identifiers for which this cmdlet gets events. The default is all events in the event queue. Wildcards are not permitted.

```
Required?                    false
Position?                    0
Default value                None
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

None

You can't pipe objects to this cmdlet.

## OUTPUTS

System.Management.Automation.PSEventArgs

This cmdlet returns a PSEventArgs object for each event. To see a description of this object, type `Get-Help Get-Event -Full` and see the Notes section of the

help topic.

## NOTES

Events, event subscriptions, and the event queue exist only in the current session. If you close the current session, the event queue is discarded and the event

subscription is canceled.

The `Get-Event` cmdlet returns a PSEventArgs object ( System.Management.Automation.PSEventArgs ) with the following properties:

- ComputerName. The name of the computer on which the event occurred. This property value is populated only when the event is forwarded from a remote computer.

- RunspaceId. A GUID that uniquely identifies the session in which the event occurred. This property value is populated only when the event is forwarded from a

remote computer.

- EventIdentifier. An integer (Int32) that uniquely identifies the event notification in the current session.

- Sender. The object that generated the event. In the value of the Action parameter, the `$Sender` automatic variable contains the sender object.

- SourceEventArgs. The first parameter that derives from EventArgs, if it exists. For example, in a timer elapsed event in which the signature has the form

Object sender, Timers.ElapsedEventArgs e, the SourceEventArgs property would contain the Timers.ElapsedEventArgs . In the value of the Action parameter, the

`$EventArgs` automatic variable contains this value.

- SourceArgs. All parameters of the original event signature. For a standard event signature, `$Args[0]` represents the sender, and `$Args[1]` represents the

SourceEventArgs . In the value of the Action parameter, the `$Args` automatic variable contains this value.

- SourceIdentifier. A string that identifies the event subscription. In the value of the Action parameter, the SourceIdentifier property of the `$Event` automatic

variable contains this value.

- TimeGenerated. A DateTime object that represents the time at which the event was generated. In the value of the Action parameter, the TimeGenerated property

of the `$Event` automatic variable contains this value.

- MessageData. Data associated with the event subscription. Users specify this data when they register an event. In the value of the Action parameter, the

MessageData property of the `$Event` automatic variable contains this value.

------------------ Example 1: Get all events ------------------

```
PS C:\> Get-Event
```

This command gets all events in the event queue.

---------- Example 2: Get events by source identifier ----------

```
PS C:\> Get-Event -SourceIdentifier "PowerShell.ProcessCreated"
```

This command gets events in which the value of the SourceIdentifier property is PowerShell.ProcessCreated.

-- Example 3: Get an event based on the time it was generated --

```
PS C:\> $Events = Get-Event

PS C:\> $Events[0] | Format-List -Property *

ComputerName     :

RunspaceId       : c2153740-256d-46c0-a57c-b805917d1b7b

EventIdentifier  : 1

Sender           : System.Management.ManagementEventWatcher

SourceEventArgs  : System.Management.EventArrivedEventArgs

SourceArgs       : {System.Management.ManagementEventWatcher, System.Management.EventArrivedEventArgs}

SourceIdentifier : ProcessStarted

TimeGenerated    : 11/13/2008 12:09:32 PM

MessageData      : PS C:\> Get-Event | Where {$_.TimeGenerated -ge "11/13/2008 12:15:00 PM"}

ComputerName     :

RunspaceId       : c2153740-256d-46c0-a57c-b8059325d1a0

EventIdentifier  : 1

Sender           : System.Management.ManagementEventWatcher

SourceEventArgs  : System.Management.EventArrivedEventArgs

SourceArgs       : {System.Management.ManagementEventWatcher, System.Management.EventArrivedEventArgs}

SourceIdentifier : ProcessStarted

TimeGenerated    : 11/13/2008 12:15:00 PM

MessageData      :
```

This example shows how to get events by using properties other than SourceIdentifier.

The first command gets all events in the event queue and saves them in the `$Events` variable.

The second command uses array notation to get the first (0-index) event in the array in the `$Events` variable. The command uses a pipeline operator (`|`) to send the

event to the `Format-List` command, which displays all properties of the event in a list. This allows you to examine the properties of the event object.

The third command shows how to use the `Where-Object` cmdlet to get an event based on the time that it was generated.

---------- Example 4: Get an event by its identifier ----------

```
PS C:\> Get-Event -EventIdentifier 2
```

This command gets the event with an event identifier of 2.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/get-event?view=powershell-5.1&WT.mc_id=ps-gethelp

New-Event

Register-EngineEvent

Register-ObjectEvent

Remove-Event

Unregister-Event

Wait-Event