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 'New-Event'

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

NAME

New-Event

SYNOPSIS

Creates a new event.

SYNTAX

New-Event [-SourceIdentifier] <System.String> [[-Sender] <System.Management.Automation.PSObject>] [[-EventArguments] <System.Management.Automation.PSObject[]>]

[[-MessageData] <System.Management.Automation.PSObject>] [<CommonParameters>]

DESCRIPTION

The `New-Event` cmdlet creates a new custom event.

You can use custom events to notify users about state changes in your program and any change that your program can detect, including hardware or system conditions,

application status, disk status, network status, or the completion of a background job.

Custom events are automatically added to the event queue in your session whenever they are raised; you do not need to subscribe to them. However, if you want to forward an event to the local session or specify an action to respond to the event, use the `Register-EngineEvent` cmdlet to subscribe to the custom event.

When you subscribe to a custom event, the event subscriber is added to your session. If you cancel the event subscription by using the `Unregister-Event` cmdlet, the event subscriber and custom event are deleted from the session. If you do not subscribe to the custom event, to delete the event, you must change the program conditions or close the PowerShell session.

PARAMETERS

-EventArguments <System.Management.Automation.PSObject[]>

Specifies an object that contains options for the event.

Required? false

Position? 2

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-MessageData <System.Management.Automation.PSObject>

Specifies additional data associated with the event. The value of this parameter appears in the MessageData property of the event object.

Required? false

Position? 3

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Sender <System.Management.Automation.PSObject>

Specifies the object that raises the event. The default is the PowerShell engine.

Required?   false

Position?   1

Default value   None

Accept pipeline input?   False

Accept wildcard characters? false

-SourceIdentifier <System.String>

Specifies a name for the new event. This parameter is required, and it must be unique in the session.

The value of this parameter appears in the SourceIdentifier property of the events.

Required?   true

Position?   0

Default value   None

Accept pipeline input?   False

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

NOTES

The new custom event, the event subscription, 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.

------- Example 1: Create a new event in the event queue -------

```
PS C:\> New-Event -SourceIdentifier Timer -Sender windows.timer -MessageData "Test"
```

This command creates a new event in the PowerShell event queue. It uses a Windows.Timer object to send the event.

---- Example 2: Raise an event in response to another event ----

```
PS C:\> function Enable-ProcessCreationEvent
{
   $Query = New-Object System.Management.WqlEventQuery "__InstanceCreationEvent", (New-Object TimeSpan 0,0,1), "TargetInstance isa 'Win32_Process'"
   $ProcessWatcher = New-Object System.Management.ManagementEventWatcher $Query
   $Identifier = "WMI.ProcessCreated"
   Register-ObjectEvent $ProcessWatcher "EventArrived" -SupportEvent $Identifier -Action
   {
       [void] (New-Event -SourceID "PowerShell.ProcessCreated" -Sender $Args[0] -EventArguments $Args[1].SourceEventArgs.NewEvent.TargetInstance)
   }
}
```

This sample function uses the `New-Event` cmdlet to raise an event in response to another event. The command uses the `Register-ObjectEvent` cmdlet to subscribe to the Windows Management Instrumentation (WMI) event that is raised when a new process is created. The command uses the Action parameter of the cmdlet to call the

`New-Event` cmdlet, which creates the new event.

Because the events that `New-Event` raises are automatically added to the PowerShell event queue, you do not need to register for that event.

RELATED LINKS

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

Get-Event

Register-EngineEvent

Register-ObjectEvent

Remove-Event

Unregister-Event

Wait-Event