PowerShell Get-Help on command 'Register-WmiEvent'

PS C:\Users\wahid> Get-Help Register-WmiEvent

```
NAME
    Register-WmiEvent


SYNOPSIS
    Subscribes to a Windows Management Instrumentation (WMI) event.



SYNTAX
    Register-WmiEvent [-Class] <System.String> [[-SourceIdentifier]
    <System.String>] [[-Action] <System.Management.Automation.ScriptBlock>]
    [-ComputerName <System.String>] [-Credential
    <System.Management.Automation.PSCredential>] [-Forward] [-MaxTriggerCount
    <System.Int32>] [-MessageData <System.Management.Automation.PSObject>]
    [-Namespace <System.String>] [-SupportEvent] [-Timeout <System.Int64>]
    [<CommonParameters>]

    Register-WmiEvent [-Query] <System.String> [[-SourceIdentifier]
    <System.String>] [[-Action] <System.Management.Automation.ScriptBlock>]
    [-ComputerName <System.String>] [-Credential
    <System.Management.Automation.PSCredential>] [-Forward] [-MaxTriggerCount
    <System.Int32>] [-MessageData <System.Management.Automation.PSObject>]
```

[-Namespace <System.String>] [-SupportEvent] [-Timeout <System.Int64>]

[<CommonParameters>]

DESCRIPTION

The `Register-WmiEvent` cmdlet subscribes to Windows Management Instrumentation (WMI) events on the local computer or on a remote computer.

When the subscribed WMI event is raised, it is added to the event queue in your local session even if the event occurs on a remote computer. To get events in the event queue, use the `Get-Event` cmdlet.

You can use the parameters of `Register-WmiEvent` to subscribe to events on remote computers and to specify the property values of the events that can help you identify the event in the queue. You can also use the Action parameter to specify actions to take when a subscribed event is raised.

When you subscribe to an event, an event subscriber is added to your session. To get the event subscribers in the session, use the `Get-EventSubscriber` cmdlet. To cancel the subscription, use the `Unregister-Event` cmdlet, which deletes the event subscriber from the session.

New Common Information Model (CIM) cmdlets, introduced Windows PowerShell 3.0, perform the same tasks as the WMI cmdlets. The CIM cmdlets comply with WS-Management (WSMan) standards and with the CIM standard, which enables the cmdlets to use the same techniques to manage computers that run the Windows operating system and those that run other operating systems. Instead of using `Register-WmiEvent`, consider using the Register-CimIndicationEvent (../cimcmdlets/register-cimindicationevent.md)cmdlet.

PARAMETERS

-Action <System.Management.Automation.ScriptBlock>

Specifies commands that handle the events. The commands in the Action parameter run when an event is raised instead of sending the event to the event queue. Enclose the commands in braces (`{}`) to create a script block.

The value of Action can include the `$Event`, `$EventSubscriber`, `$Sender`, `$EventArgs`, and `$Args` automatic variables, which provide information about the event to the Action script block. For more information, see about_Automatic_Variables (../Microsoft.PowerShell.Core/About/about_Automatic_Variables.md).

When you specify an action, `Register-WmiEvent` returns an event job object that represents that action. You can use the cmdlets that contain the Job noun (the Job cmdlets) to manage the event job.

-Class <System.String>

Specifies the event to which you are subscribing. Enter the WMI class that generates the events. A Class or Query parameter is required in every command.

-ComputerName <System.String>

Specifies the name of the computer on which the command runs. The default is the local computer.

Type the NetBIOS name, an IP address, or a fully qualified domain name of the computer. To specify the local computer, type the computer name, a dot (`.`), or localhost.

This parameter does not rely on Windows PowerShell remoting. You can use the ComputerName parameter even if your computer is not configured to run remote commands.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to perform this action. The default is the current user.

Type a user name, such as User01 or Domain01\User01, or enter a PSCredential object, such as one generated by the `Get-Credential` cmdlet. If you type a user name, this cmdlet prompts you for a password.

-Forward <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet sends events for this subscription to the session on the local computer. Use this parameter when you are registering for events on a remote computer or in a remote session.

-MaxTriggerCount <System.Int32>

Specifies the maximum trigger count.

-MessageData <System.Management.Automation.PSObject>

Specifies any additional data to be associated with this event subscription. The value of this parameter appears in the MessageData property of all events associated with this subscription.

-Namespace <System.String>

Specifies the namespace of the WMI class.

-Query <System.String>

Specifies a query in WMI Query Language (WQL) that identifies the WMI event class, such as: `select * from __InstanceDeletionEvent`.

-SourceIdentifier <System.String>

Specifies a name that you select for the subscription. The name that you select must be unique in the current session. The default value is the GUID that Windows PowerShell assigns.

The value of this parameter appears in the value of the SourceIdentifier

property of the subscriber object and of all event objects associated with
this subscription.

-SupportEvent <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet hides the event subscription. Use this parameter when the current subscription is part of a more complex event registration mechanism and it should not be discovered independently.

To view or cancel a subscription that was created by using the SupportEvent parameter, specify the Force parameter of the `Get-EventSubscriber` and `Unregister-Event` cmdlets.

-Timeout <System.Int64>

Specifies how long Windows PowerShell waits for this command to finish.

The default value, 0 (zero), means that there is no time-out, and it causes Windows PowerShell to wait indefinitely.

<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: Subscribe to events generated by a class -----

```
Register-WmiEvent -Class 'Win32_ProcessStartTrace' -SourceIdentifier
"ProcessStarted"
```

---- Example 2: Subscribe to creation events for a process ----

```
$wmiParameters = @{
  Query = "select * from __instancecreationevent within 5 where targetinstance
isa 'win32_process'"
  SourceIdentifier = "WMIProcess"
  MessageData = "Test 01"
  TimeOut = 500
}
Register-WmiEvent @wmiParameters
```

------- Example 3: Use an action to respond to an event -------

```
$action = { Get-History | where { $_.commandline -like "*start-process*" } |
export-cliXml "commandHistory.clixml" }
Register-WmiEvent -Class 'Win32_ProcessStartTrace' -SourceIdentifier
"ProcessStarted" -Action $action

Id    Name            State      HasMoreData   Location  Command
--    ----            -----      -----------   --------  -------
1     ProcessStarted  NotStarted False                   get-history | where
{...
```

When you use the Action parameter, `Register-WmiEvent` returns a background job that represents the event action. You can use the Job cmdlets, such as `Get-Job` and `Receive-Job`, to manage the event job.

For more information, see about_Jobs (../Microsoft.PowerShell.Core/About/about_Jobs.md).

----- Example 4: Register for events on a remote computer -----

```
Register-WmiEvent -Class 'Win32_ProcessStartTrace' -SourceIdentifier "Start"
-Computername Server01
Get-Event -SourceIdentifier "Start"
```

WMI returns the events to the local computer and stores them in the event queue in the current session. To retrieve the events, run a local `Get-Event` command.

REMARKS

To see the examples, type: "get-help Register-WmiEvent -examples".

For more information, type: "get-help Register-WmiEvent -detailed".

For technical information, type: "get-help Register-WmiEvent -full".

For online help, type: "get-help Register-WmiEvent -online"