



python



PowerShell

FPDF Library  
PDF generator

*Full credit is given to the above companies including the OS that this PDF file was generated!*

### **PowerShell Get-Help on command 'Register-CimIndicationEvent'**

**PS C:\Users\wahid> Get-Help Register-CimIndicationEvent**

#### NAME

Register-CimIndicationEvent

#### SYNOPSIS

Subscribes to indications using a filter expression or a query expression.

#### SYNTAX

```
Register-CimIndicationEvent [-ClassName] <System.String> [[-Action]
<System.Management.Automation.ScriptBlock>] [[-SourceIdentifier]
<System.String>] -CimSession <Microsoft.Management.Infrastructure.CimSession>
[-Forward] [-MaxTriggerCount <System.Int32>] [-MessageData
<System.Management.Automation.PSObject>] [-Namespace <System.String>]
[-OperationTimeoutSec <System.UInt32>] [-SupportEvent] [<CommonParameters>]
```

```
Register-CimIndicationEvent [-Query] <System.String> [[-Action]
<System.Management.Automation.ScriptBlock>] [[-SourceIdentifier]
<System.String>] -CimSession <Microsoft.Management.Infrastructure.CimSession>
[-Forward] [-MaxTriggerCount <System.Int32>] [-MessageData
<System.Management.Automation.PSObject>] [-Namespace <System.String>]
[-OperationTimeoutSec <System.UInt32>] [-QueryDialect <System.String>]
```

[-SupportEvent] [<CommonParameters>]

Register-CimIndicationEvent [-ClassName] <System.String> [[-Action]  
<System.Management.Automation.ScriptBlock>] [[-SourceIdentifier]  
<System.String>] [-ComputerName <System.String>] [-Forward] [-MaxTriggerCount  
<System.Int32>] [-MessageData <System.Management.Automation.PSObject>]  
[-Namespace <System.String>] [-OperationTimeoutSec <System.UInt32>]  
[-SupportEvent] [<CommonParameters>]

Register-CimIndicationEvent [-Query] <System.String> [[-Action]  
<System.Management.Automation.ScriptBlock>] [[-SourceIdentifier]  
<System.String>] [-ComputerName <System.String>] [-Forward] [-MaxTriggerCount  
<System.Int32>] [-MessageData <System.Management.Automation.PSObject>]  
[-Namespace <System.String>] [-OperationTimeoutSec <System.UInt32>]  
[-QueryDialect <System.String>] [-SupportEvent] [<CommonParameters>]

## DESCRIPTION

The `Register-CimIndicationEvent` cmdlet subscribes to indications using an indication class name or a query expression. Use the SourceIdentifier parameter give a name to the subscription.

This cmdlet returns an EventSubscription object. You can use this object to cancel the subscription.

## PARAMETERS

-Action <System.Management.Automation.ScriptBlock>

Specifies the commands that handle the events. The commands specified by this 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 script block specified with Action can include the ``$Event``, ``$EventSubscriber``, ``$Sender``, ``$SourceEventArgs``, and ``$SourceArgs`` 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](#)).

**-CimSession** <Microsoft.Management.Infrastructure.CimSession>

Runs the command using the specified CIM session. Enter a variable that contains the CIM session, or a command that creates or gets the CIM session, such as the ``New-CimSession`` or ``Get-CimSession`` cmdlets. For more information, see [about\\_CimSession](#) ([../Microsoft.PowerShell.Core/About/about\\_CimSession.md](#)).

**-ClassName** <System.String>

Specifies the indication class to which you are subscribing. You can use tab completion to browse the list of classes, because PowerShell gets a list of classes from the local WMI server to provide a list of class names.

**-ComputerName** <System.String>

Specifies the name of the computer on which you want to run the CIM operation. You can specify a fully qualified domain name (FQDN), a NetBIOS name, or an IP address.

If you specify this parameter, the cmdlet creates a temporary session to the specified computer using the WsMan protocol. If you do not specify this parameter, the cmdlet performs operation on the local system using Component Object Model (COM).

If multiple operations are being performed on the same computer, connect using a CIM session for better performance.

**-Forward** <System.Management.Automation.SwitchParameter>

Indicates that events for the subscription are forwarded 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>`

Parameter to indicate that the subscriber should be auto-unregistered after being triggered for specified times. If the value is equal or less than zero, there is no limit on the number of times the event can be triggered without being unregistered.

`-MessageData <System.Management.Automation.PSObject>`

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

`-Namespace <System.String>`

Specifies the namespace for the CIM operation. The default namespace is `root/cimv2`. You can use tab completion to browse the list of namespaces, because PowerShell gets a list of namespaces from the local WMI server to provide the list of namespaces.

`-OperationTimeoutSec <System.UInt32>`

Specifies the amount of time that the cmdlet waits for a response from the computer. By default, the value of this parameter is 0, which means that the cmdlet uses the default timeout value for the server.

If the `OperationTimeoutSec` parameter is set to a value less than the robust connection retry timeout of 3 minutes, network failures that last more than the value of the `OperationTimeoutSec` parameter are not recoverable, because the operation on the server times out before the client can reconnect.

`-Query <System.String>`

Specifies a query to run on the CIM server. If the value specified

contains double quotes `"` , single quotes ` ` ` , or a backslash ` ` ` , you must escape those characters by prefixing them with the backslash character. If the value specified uses the WQL LIKE operator, then you must escape the following characters by enclosing them in square brackets `[ ]` : percent `%` , underscore `\_` , or opening square bracket `[` .

**-QueryDialect <System.String>**

Specifies the query language used for the Query parameter. The acceptable values for this parameter are: WQL or CQL . The default value is WQL .

**-SourceIdentifier <System.String>**

Specifies a name for the subscription. The name that you specify must be unique in the current session. The default value is a GUID that PowerShell assigns. This value 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 the event subscription is hidden. Use this parameter when the current subscription is part of a more complex event registration mechanism and it should not be discovered independently.

**<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: Register the events generated by a class -----

```
Register-CimIndicationEvent -ClassName 'Win32_ProcessStartTrace'
```

```
-SourceIdentifier "ProcessStarted"
```

```
Get-Event -SourceIdentifier "ProcessStarted"
```

The `Get-Event` cmdlet gets the events with ProcessStarted subscription. For more information, see [Get-Event \(../Microsoft.PowerShell.Utility/Get-Event.md\)](#).

> [!NOTE] > For this example, you must run PowerShell as an Administrator.

----- Example 2: Register the events using a query -----

```
$query = "SELECT * FROM CIM_InstModification WHERE TargetInstance ISA  
'Win32_LocalTime'"
```

```
Register-CimIndicationEvent -Query $query -SourceIdentifier "Timer"
```

----- Example 3: Run a script when the event arrives -----

```
$action = {  
    $name = $event.SourceEventArgs.NewEvent.ProcessName  
    $id = $event.SourceEventArgs.NewEvent.ProcessId  
    Write-Host -Object "New Process Started : Name = $name  
    ID = $id"
```

```
}
```

```
Register-CimIndicationEvent -ClassName 'Win32_ProcessStartTrace'  
-SourceIdentifier "ProcessStarted" -Action $action
```

For more information, see [Win32\\_ProcessStartTrace \(/previous-versions/windows/desktop/krnlpov/win32-processstarttrace\)](#).

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

```
Register-CimIndicationEvent -ClassName 'Win32_ProcessStartTrace'  
-SourceIdentifier "ProcessStarted" -ComputerName Server01  
Get-Event -SourceIdentifier "ProcessStarted"
```

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

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

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

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