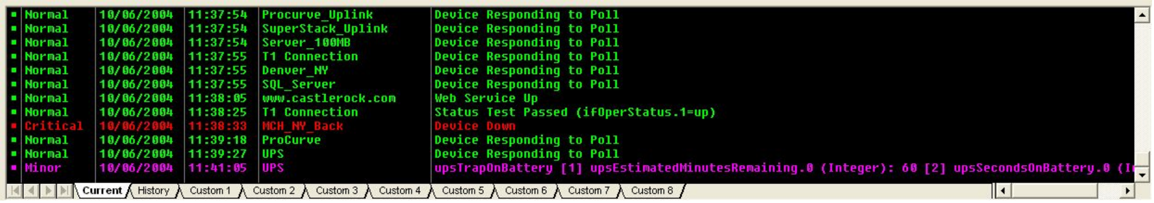
How to create Event Filters directly from the Event Viewer

Event Filters determine the action that SNMPc takes when a trap is received or an event is triggered. SNMPc 7.0 supports the ability to create an event filter directly from a trap or event displayed in the log view.

SNMPc can create an event filter without requiring the correct MIB to be compiled. It is good practice though to add the relevant MIB's where possible. If the correct MIB is compiled SNMPc will be able to decode the variables contained within the SNMP Trap. There are separate How-To guides which cover MIB compiling.

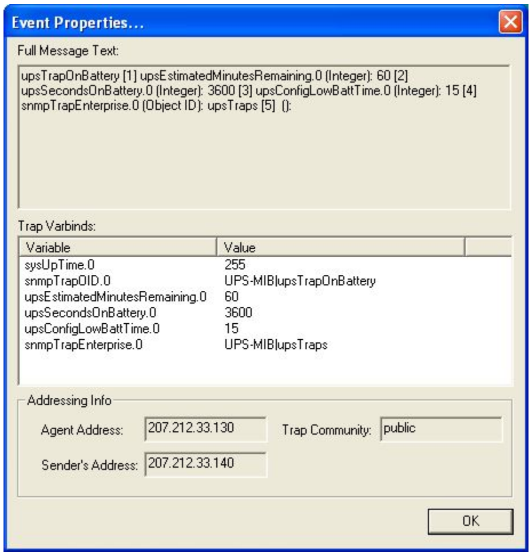
In the following example we will create an Event Filter to match on a trap received from a UPS device. When the power is interrupted the UPS sends a trap to SNMPc. The log view is displayed below



```
■ Normal    10/06/2004  11:37:54  Procurve_Uplink      Device Responding to Poll
■ Normal    10/06/2004  11:37:54  SuperStack_Uplink    Device Responding to Poll
■ Normal    10/06/2004  11:37:54  Server_100MB         Device Responding to Poll
■ Normal    10/06/2004  11:37:55  T1 Connection        Device Responding to Poll
■ Normal    10/06/2004  11:37:55  Denver_NY            Device Responding to Poll
■ Normal    10/06/2004  11:37:55  SQL_Server           Device Responding to Poll
■ Normal    10/06/2004  11:38:05  www.castlerock.com   Web Service Up
■ Normal    10/06/2004  11:38:25  T1 Connection        Status Test Passed (ifOperStatus.1=up)
■ Critical  10/06/2004  11:38:33  MCH_NY_Back          Device Down
■ Normal    10/06/2004  11:39:18  ProCurve             Device Responding to Poll
■ Normal    10/06/2004  11:39:27  UPS                  Device Responding to Poll
■ Minor     10/06/2004  11:41:05  UPS                  upsTrapOnBattery [1] upsEstimatedMinutesRemaining.0 (Integer): 60 [2] upsSecondsOnBattery.0 (I
```

This is a pretty typical display for a Trap received without an event filter. Using an event filter we can customize SNMPc so that it displays the trap in a more readable format.

The first stage in creating the event filter is to decide which variables are of use in the trap message. To view the trap variables, right-click on the event message and choose *Event Properties*. You will see a display similar to the following:



Event Properties...

Full Message Text:

upsTrapOnBattery [1] upsEstimatedMinutesRemaining.0 (Integer): 60 [2] upsSecondsOnBattery.0 (Integer): 3600 [3] upsConfigLowBattTime.0 (Integer): 15 [4] snmpTrapEnterprise.0 (Object ID): upsTraps [5] ():

Trap Varbinds:

Variable	Value
sysUpTime.0	255
snmpTrapOID.0	UPS-MIB|upsTrapOnBattery
upsEstimatedMinutesRemaining.0	60
upsSecondsOnBattery.0	3600
upsConfigLowBattTime.0	15
snmpTrapEnterprise.0	UPS-MIB|upsTraps

Addressing Info

Agent Address: 207.212.33.130 Trap Community: public

Sender's Address: 207.212.33.140

The variable number is displayed in [brackets]. Therefore in this example

Variable 1 is upsEstimatedMinutesRemaining.0 (Integer): 60;
Variable 2 is upsSecondsOnBattery.0 (Integer): 3600;
Variable 3 upsConfigLowBattTime.0 (Integer) 15; etc.

Within SNMPc to include a trap variable as part of an event message you use the Event Parameter ‘$(variable number)’. Therefore from the example:

$1	=	60	(upsEstimatedMinutesRemaining)
$2	=	3600	(upsSecondsOnBattery)
$3	=	15	(upsConfigLowBattTime)

In this guide we will create an event message that displays:

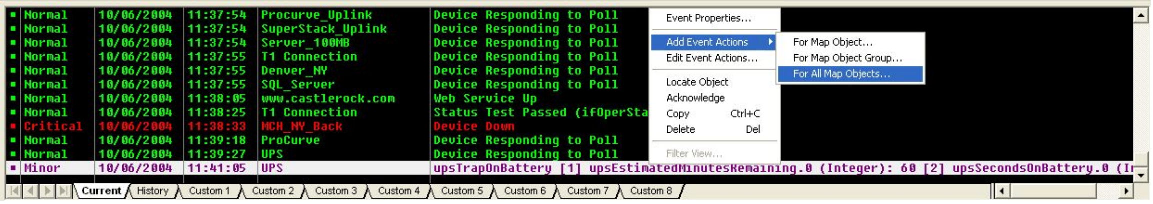
> UPS on Battery: Estimated battery life: XX minutes, time on battery YY seconds.

Therefore the event ‘message string’ would be

> UPS on Battery: Estimated battery life: $1 minutes, time on battery $2 seconds.

SNMPc has a comprehensive range of ‘Event Parameters’ that can be used in the event message. A full list is included in Appendix A of the *Getting Started* guide.

To create an event filter simply right click on the event and choose *Add Event Actions*.



```
Normal    10/06/2004  11:37:54  Procurve_Uplink      Device Responding to Poll
Normal    10/06/2004  11:37:54  SuperStack_Uplink    Device Responding to Poll
Normal    10/06/2004  11:37:54  Server_100MB         Device Responding to Poll
Normal    10/06/2004  11:37:55  T1 Connection        Device Responding to Poll
Normal    10/06/2004  11:37:55  Denver_NY            Device Responding to Poll
Normal    10/06/2004  11:37:55  SQL_Server           Device Responding to Poll
Normal    10/06/2004  11:38:05  www.castlerock.com   Web Service Up
Normal    10/06/2004  11:38:25  T1 Connection        Status Test Passed (ifOperSta
Critical  10/06/2004  11:38:33  MCH_NY_Back          Device Down
Normal    10/06/2004  11:39:18  ProCurve             Device Responding to Poll
Normal    10/06/2004  11:39:27  UPS                  Device Responding to Poll
Minor     10/06/2004  11:41:05  UPS                  upsTrapOnBattery [1] upsEstimatedMinutesRemaining.0 (Integer): 60 [2] upsSecondsOnBattery.0 (In
```

There are three options provided in the menu

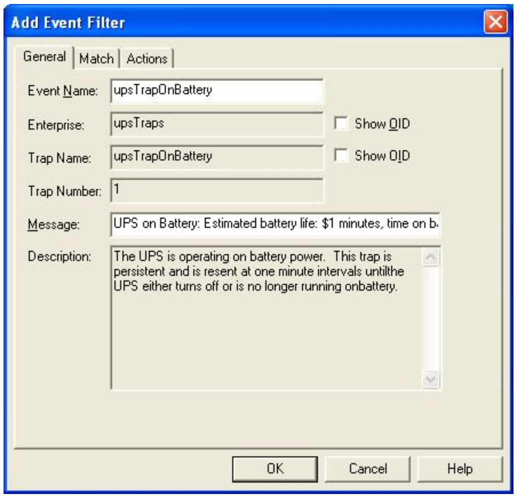
For Map Object.. The event filter will only be matched if the trap is received from this device

For Map Object Group… The Event Filter will be matched by any device in the same node ‘Group’. You can view or configure a node group by right-clicking on an icon and choosing *Properties*.

For All Map Objects... The event Filter will match on the trap irrespective of which device generated the alert.

In this example we will create an event filter that will match on the trap irrespective of which device on the network generated it. Therefore we choose *For All Map Objects.*

You will now be presented with the *Add Event Filter* window.



Event Name: upsTrapOnBattery

Enterprise: upsTraps ☐ Show OID

Trap Name: upsTrapOnBattery ☐ Show OID

Trap Number: 1

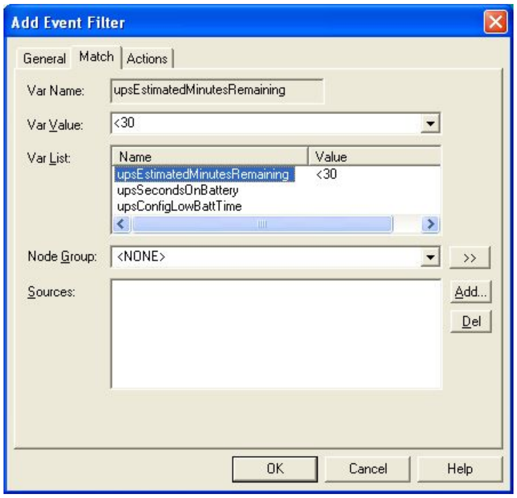
Message: UPS on Battery: Estimated battery life: $1 minutes, time on b

Description: The UPS is operating on battery power. This trap is persistent and is resent at one minute intervals untilthe UPS either turns off or is no longer running onbattery.

In the *Message* area enter *UPS on Battery: Estimated battery life: $1 minutes, time on battery $2 seconds.*

Under the *Match* Tab you can specify to match on variables within the trap or on the device that generated the trap.

The Match tab allows a great deal of flexibility with event response. The example shows an event filter that would ‘match’ only if the value of Estimated Battery life was less than 30 minutes. This could be useful for example if you wanted to create several courses of action. One filter could generate a general email to be sent to the support team when the UPS went on battery. The second filter generates a pager notification to be sent to the support manager if the battery time remaining was under 30 minutes.



Var Name: upsEstimatedMinutesRemaining

Var Value: <30

Var List:

Name	Value
upsEstimatedMinutesRemaining	<30
upsSecondsOnBattery	
upsConfigLowBattTime	

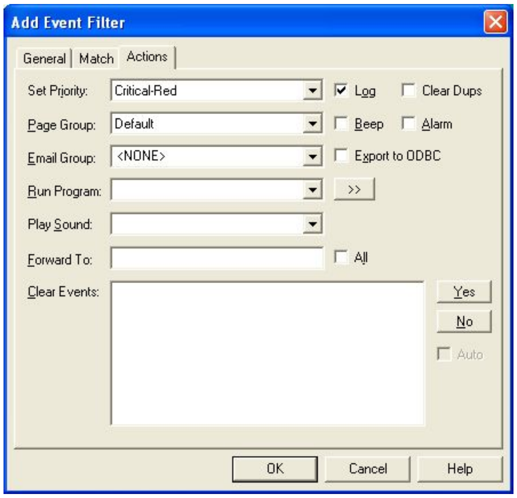
Node Group: <NONE>

Sources:

The Actions tab allows you to specify the actions for SNMPc to take when the event filter conditions are met. The range of options include the color of the icon and event message; paging and email messaging; ability to run a program or batch file; play WAV sound or forward events to another management system. A full description of the options is included in the online help.

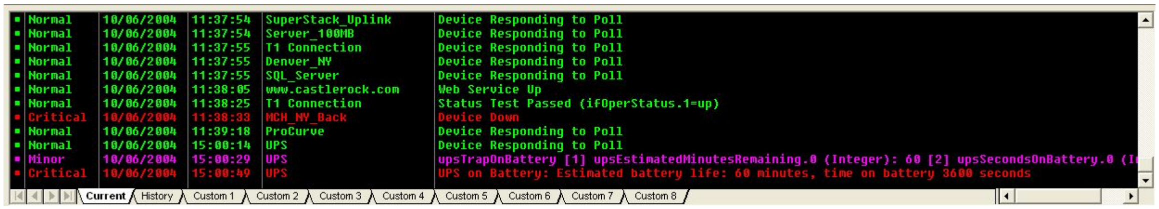
In this example we are going to specify that this is a critical alarm and should be displayed in red. Also the users in the *Default* group will be paged.

Select *OK* to add the Event Filter.



Set Priority: Critical-Red ☑ Log ☐ Clear Dups
Page Group: Default ☐ Beep ☐ Alarm
Email Group: <NONE> ☐ Export to ODBC
Run Program:
Play Sound:
Forward To: ☐ All
Clear Events: ☐ Auto

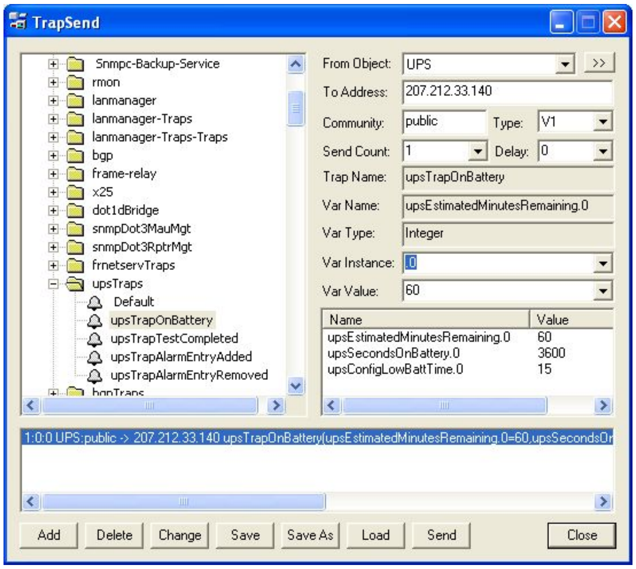
When the Trap is received you should now see a customized event message.



```
■ Normal    10/06/2004  11:37:54  SuperStack_Uplink     Device Responding to Poll
■ Normal    10/06/2004  11:37:54  Server_100MB          Device Responding to Poll
■ Normal    10/06/2004  11:37:55  T1 Connection         Device Responding to Poll
■ Normal    10/06/2004  11:37:55  Denver_NY             Device Responding to Poll
■ Normal    10/06/2004  11:37:55  SQL_Server            Device Responding to Poll
■ Normal    10/06/2004  11:38:05  www.castlerock.com    Web Service Up
■ Normal    10/06/2004  11:38:25  T1 Connection         Status Test Passed (ifOperStatus.1=up)
■ Critical  10/06/2004  11:38:33  MCH_NY_Back           Device Down
■ Normal    10/06/2004  11:39:18  ProCurve              Device Responding to Poll
■ Normal    10/06/2004  15:00:14  UPS                   Device Responding to Poll
■ Minor     10/06/2004  15:00:29  UPS                   upsTrapOnBattery [1] upsEstimatedMinutesRemaining.0 (Integer): 60 [2] upsSecondsOnBattery.0 (I
■ Critical  10/06/2004  15:00:49  UPS                   UPS on Battery: Estimated battery life: 60 minutes, time on battery 3600 seconds
```

Tips and tricks:

SNMPc includes a *Trap Sender* Tool which allows you to spoof traps from any MIB that has been compiled into SNMPc. It is available from the *Tools* menu. The following screenshot shows the configuration of the Trap Sender to generate the alert used in this example.



From Object: UPS
To Address: 207.212.33.140
Community: public Type: V1
Send Count: 1 Delay: 0
Trap Name: upsTrapOnBattery
Var Name: upsEstimatedMinutesRemaining.0
Var Type: Integer
Var Instance: .0
Var Value: 60

Name	Value
upsEstimatedMinutesRemaining.0	60
upsSecondsOnBattery.0	3600
upsConfigLowBattTime.0	15

```
1:0:0 UPS:public -> 207.212.33.140 upsTrapOnBattery(upsEstimatedMinutesRemaining.0=60,upsSecondsOr
```