

# OpsMgr Custom Alert: Alert on process using too much memory (without a monitor)

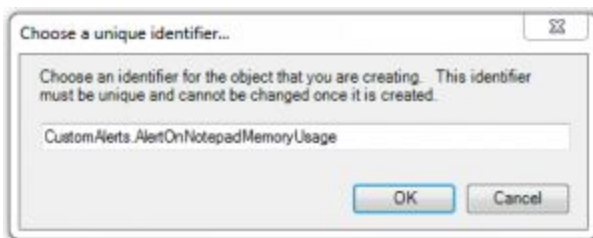
I needed to create a rule in SCOM that could alert if a process consumed more than x memory. To my surprise there is nothing out of the box in the Authoring Console that allows for this – I find this a bit strange as this is something that would be used quite often. There are ways to create the rule I am looking for through a custom application monitor, or a VBScript, but those are time consuming and I feel are more effort than reward. Right clicking in the Authoring Console under the rules section proves my point.



For this example I am going to create an alert that will fire if notepad.exe is using more than 1.2 Mb of memory. Please Note: if you are testing this pack you will need to open notepad before importing this rule – if OpsMgr cannot resolve the performance counter at the time of the workflow initialization the workflow will be unloaded. The reason for choosing notepad is simple, it ships with every version of windows and we can easily change the amount of memory that it uses by opening a big text file.

## Let's Begin

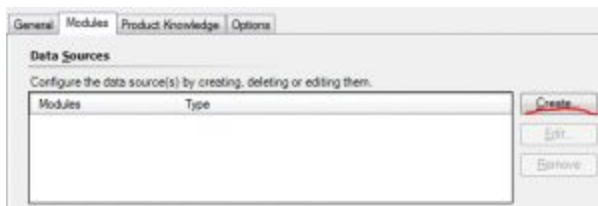
Open up the Authoring Console and create \ open a management pack. Navigate to Health Model, Rules, right click, and select New -> Custom Rule. Give the rule an ID and click OK. For this example my ID will be "CustomAlerts.AlertOnNotepadMemoryUsage".



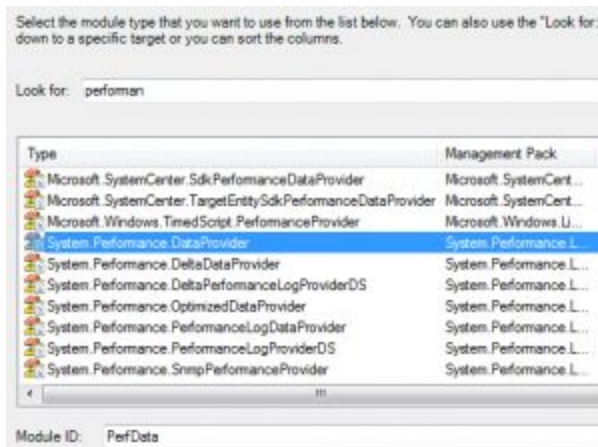
Under general give the rule a name, description and target it at the Microsoft.Windows.Server.Computer class (or any other class that you want).



Click on the Modules Tab and create a new data source.



Select the System.Performance.DataProvider module, give it an ID and click OK.



Under the Data Source Module section select the module you just added and click Edit, on the screen that appears click Configure in the bottom left corner. The performance counter selection wizard appears. Select your counter (in this case its Process \ Working Set \ notepad). Use the picker below to choose the frequency to run this alert, finally click OK twice to return to the Modules Tab.



**Performance Counter**

**Specify the performance counter**  
Enter performance counter details or click Select to browse computers and select details.

Object:  
Process

Counter:  
Working Set

Instance:  
notepad

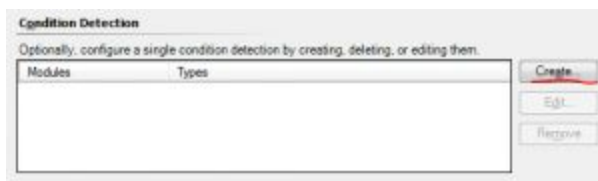
☐ Include all instances for the selected counter

Select...

**Specify the interval**

Interval:  
1 Minutes

Now that we have our counter, the next thing we need to do is decide if an alert should be generated. Since the counter could potentially be below our threshold we will need to compare the value to decide if an alert should be created. In order to do this we will need to add a Condition Detection module to our alert, under the Condition Detection section click Create.



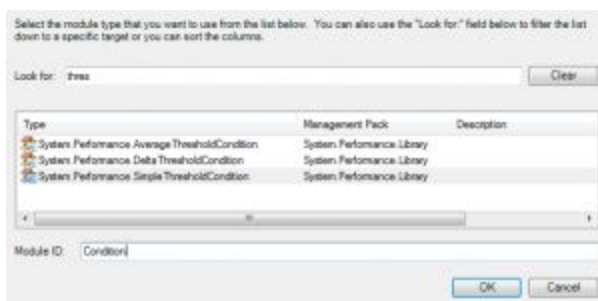
**Condition Detection**  
Optionally, configure a single condition detection by creating, deleting, or editing them.

Modules	Types

Create  
Edit  
Remove

Select the System.Performance.SimpleThresholdCondition module and click OK.

Note: you can use the any other module you like for the condition detection in the list that appears (i.e. average threshold), but for now we are going to keep it simple by using theSystem.Performance.SimpleThresholdCondition module.



Select the module type that you want to use from the list below. You can also use the "Look for:" field below to filter the list down to a specific target or you can sort the columns.

Look for:  Clear

Type	Management Pack	Description
System Performance Average ThresholdCondition	System Performance Library	
System Performance Delta ThresholdCondition	System Performance Library	
System Performance Simple ThresholdCondition	System Performance Library	

Module ID:

OK Cancel

Once added, click the Edit button for the new module. There is no configuring wizard so you will have to either edit the module through notepad or use the dialog box on screen. As you can see there are 2 options that can be changed, they are Threshold and Operator these are pretty self-explanatory so I am going to go ahead and enter in 1258291 and Greater respectively.

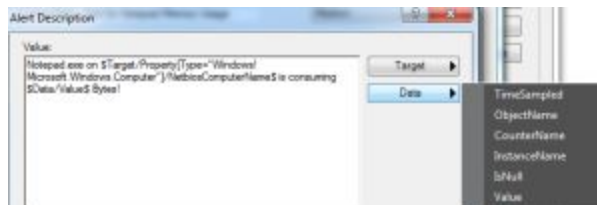
**Configuration**

System.Performance.SimpleThresholdCondition

Name	Value
Threshold	1258291
Operator	Greater

Click OK twice to close the current window and return to the Modules Tab.

The last thing we will need to do is create an alert if the condition detection returns true. To do this, simply add the “System.Health.GenerateAlert” module to the Actions section. Click Edit and then Configure to bring up the configuration wizard for the alert. Enter in a name for the alert and any message that you want to appear, you can make use of the \$Data\$ variables to pull information back about the process. You can make use of the fly-outs on the message editor to get all values available.



Once complete your alerts screen should look something like this.

**Alerting**

Specify the information that will be generated by the alert

Alert name: CustomAlerts: Alert On Notepad Memory Usage

Priority: Medium

Alert description: Notepad.exe on \$Target/Property[Type=Windows/Microsoft.Windows.Computer]/%WebResourceName\$ is consuming \$Data/Value\$ Bytes!

Severity: Critical

Custom alert fields: Alert suppression...

OK Cancel Apply

Click OK twice to get back to the Modules Tab.

At this point you are pretty much complete with the creating of your alert, I would recommend disabling the rule by default and then targeting it at the servers (or class) that you want to monitor. Additionally take the time to add some KB entry into the alert so support staff knows what to do with it appears in the console. I normally like to change the category of custom rules to its correct type (Alert in this case); you can do this on the Options Tab. You should now have a rule looking something like this.



Right, let's test this rule.

Fire up your lab and import the pack, add any overrides to get the rule targeted at the correct computer.

I made a bit of a stuff up with the amount of memory to alert on so straight away I get an alert :/.



This means that the rule is working. I close notepad and wait for 5 min to see if I get any more alerts coming through (to ensure that the logic is working correctly). As mentioned earlier if I the performance counter cannot be found the workflow will be unloaded from the OpsMgr agent. Like clockwork 1 min later I get the following alert in the agents event log.

```

Log Name:      Operations Manager
Source:        Health Service Modules
Date:          12/15/2010 11:24:38 AM
Event ID:      10103
Task Category: None
Level:         Warning
Keywords:      Classic
User:          N/A
Computer:      SQL.home.net
Description:
In PerfDataSource, could not find counter Process, Working Set, notepad in Snapshot.
Unable to submit Performance value. Module will not be unloaded.
One or more workflows were affected by this.
Workflow name: CustomAlerts.AlertOnNotepadMemoryUsage
Instance name: SQL.home.net
Instance ID: {1A5EB665-A107-EEC5-7394-60D9B5EF8882}
Management group: HOME
Event Xml:
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider />
    <EventID Qualifiers="49152">10103</EventID>
    <Level>3</Level>
    <Task>0</Task>
  </System>
  <Text>
    In PerfDataSource, could not find counter Process, Working Set, notepad in Snapshot.
    Unable to submit Performance value. Module will not be unloaded.
    One or more workflows were affected by this.
    Workflow name: CustomAlerts.AlertOnNotepadMemoryUsage
    Instance name: SQL.home.net
    Instance ID: {1A5EB665-A107-EEC5-7394-60D9B5EF8882}
    Management group: HOME
  </Text>
</Event>

```

```

<Keywords>0x80000000000000</Keywords>
<TimeCreated SystemTime="2010-12-15T19:19:38.000000000Z" />
<EventRecordID>9440</EventRecordID>
<Channel>Operations Manager</Channel>
<Computer>SQL.home.net</Computer>
<Security />
</System>
<EventData>
<Data>HOME</Data>
<Data>CustomAlerts.AlertOnNotepadMemoryUsage</Data>
<Data>SQL.home.net</Data>
<Data>{1A5EB665-A107-EEC5-7394-60D9B5EF8882}</Data>
<Data>Process</Data>
<Data>Working Set</Data>
<Data>notepad</Data>
</EventData>
</Event>

```

I expected that. Something interesting to note here – although the performance counter cannot be resolved the workflow is not uninitialized, I suspect this has something to do with the fact that we have already submitted data back to SCOM from this workflow. After leaving notepad closed on my target computer for a while I decide to re-open it and open a 500kb text file (“hello world” +- 5000 times), and after a minuet a new alert appears in the SCOM console.

Severity: Critical (2)					
SQL.home.net	CustomAlerts.AlertOnNotepadMemoryUsage	New	12/15/2010 11:21:08 AM	< 1 Minute	
SQL.home.net	CustomAlerts.AlertOnNotepadMemoryUsage	New	12/15/2010 11:22:08 AM	19 Minutes	

Comparing the 2 alerts side by side I see that the memory usage reflects that the 500kb file has been opened.

#### Alert Description

Notepad.exe on SQL is consuming 4456448 Bytes!

#### Alert Description

Notepad.exe on SQL is consuming 7131136 Bytes!

That’s all there is to creating a custom alert in OpsMgr based on the memory usage of an application. Although this particular example will not be useful in production, this example should enable you to create an alert of any performance data easily.

Feel free to leave questions / comments / requests.

Grab the example management pack [here](#).