# C-MP-3A : Creating a Health Monitor

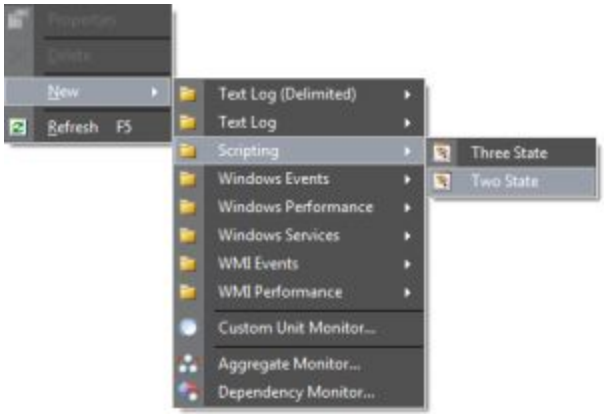
If you remember that from my last post (Testing our custom discovery) it worked and found some instances of the Print Spooler service. The only problem was that all the discovered instances were in a Not Monitored state. The reason for this is that there is no monitor to tell SCOM that the Print Spooler is healthy.

The health state of an object in SCOM is set with a monitor. As you know there are a wide variety of monitors available in SCOM. I am going to be making use of a script based monitor targeted at our Richard.CustomMP.Spooler class. Depending on what our script returns we will set the health of the class to Healthy or Warning. The script I am going to be making use of is fairly simple and checks the state of the service using WMI. It returns the state and StartMode of the service as a property bag (http://msdn.microsoft.com/en-us/library/bb437556.aspx) which is something that SCOM can work with.

You can test the script by calling up a cmd window and calling "cscript SCRIPT_NAME SERVICE_NAME". If the service was found you will get a property bag with the state and StartMode of the service, otherwise a blank property bag will be returned.

To add the health monitor script onto your Management Pack, follow the steps taken by me below.

Open up the Authoring Console and navigate to Health Model -> Monitors. Right click in the monitor's pane and select New -> Scripting -> Two State. We are going to use a 2 state monitor as we will be setting the state of the class to healthy (service is running) or warning (service is stopped).

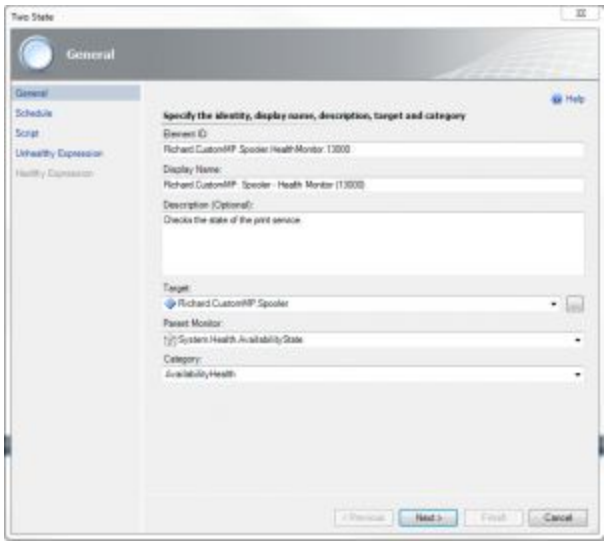



Complete the first screen that appears with the following information:

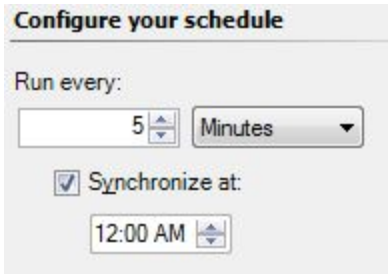
1. Element ID: Richard.CustomMP.Spooler.HealthMonitor.13000
2. Display Name: Richard.CustomMP: Spooler - Health Monitor (13000)
3. Description: Checks the state of the print service.
4. Target: Richard.CustomMP.Spooler
5. Parent Monitor: System.Health.AvalabilityState

6. Category: Avalability Health

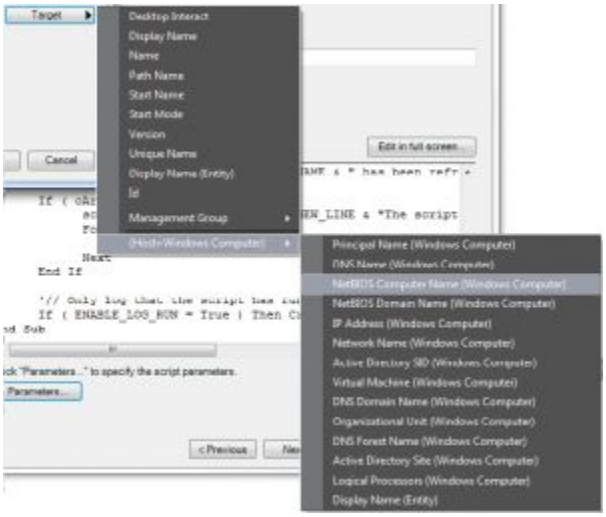
The screen should look like this:



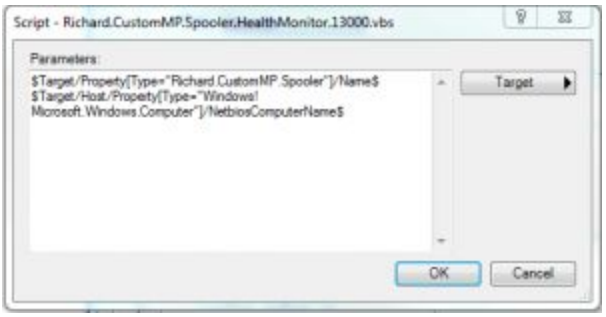

Click Next to continue. If you had read the script you would know that the interval that we want to run this is every 5 min, so set the Schedule for your script to that.




Click Next to go to the Script screen. Complete this screen by giving the script a name (don't forget the .vbs at the end of it). Drop your script into the script pane and hit the Parameters button. The only parameters that this script needs is the service name and computer name (the computer name is used for logging the script events). Use the fly outs to add in the service name and the computer name (as the image shows below):




When finished your parameter screen should look something like this:

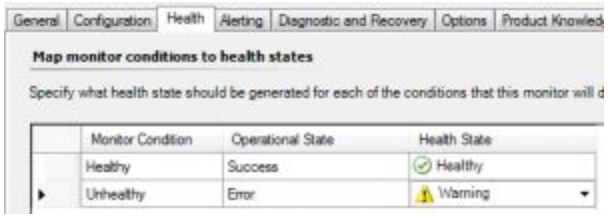



Once you have set your parameter, click OK to close the parameters window, and Next on the Script screen. Now we need to configure the unhealthy expression for our monitor. I have decided that in order for the monitor to be unhealthy the service should not be in a running state. Click the Insert button to add a condition to the screen. For the Parameter Name enter in Property[@Name='State'] (XPath - http://www.w3schools.com/xpath/xpath_syntax.asp) to access the state value inside the property bag, for the operator choose the does not equal qualifier, finally for the value enter in Running. When complete your expression screen should look something like this:


Note: ensure that you use the single quotation marks if you copy the XPath query from this post (replaces them with '). Click Next to continue for the Healthy Expression screen. For the service to be healthy I want it to be in a Running state. Complete the expression as follows and click Finish to complete the wizard.

You should now have a monitor under the Richard.CustomMP.Spooler class:


We now need to configure the states of the monitor. To do this right click on the monitor and select properties, on the screen that appears click the Health tab. Ensure that the health of the monitor is set to Healthy for the healthy state and Warning for the unhealthy state:




If you would like to you can configure the monitor alert when its state changes to warning (this is not covered here to save time). Click OK to close the properties dialog box and save the changes made to your management pack.

The final thing we will need to do is import this into our lab to ensure that our new monitor is working. Import the pack and wait until the script has run (every interval of 5 min from 12:00 am), if everything goes well the state of your Richard.CustomMP.Spooler should change to Healthy:


Try stopping the Print Spooler service on a random computer, wait 5 min and see how the state changes for the discovered instance of the Richard.CustomMP.Spooler class.


Note: I have changed the interval for the discovery (Richard.CustomMP.Spooler.Discovery.10001) to run every 1 hour as 5 min is a bit too much, but in a production environment 1 hour is still too much. That's pretty much all there is to a script based monitor. There is a lot of content to take in with this post so maybe a second read through might make things a bit clearer. I have attached the complete management pack if you would like to see all the components in action.

Grab the current management pack here.