C-MP 4B : Collecting Memory Usage – WMI

My last post detailed how to use our custom class to collect performance data for our Richard.CustomMP.Spooler class using the Windows Performance Collection template and ultimately Perfmon on the target box. The Performance Collection template is useful for collecting standard performance data, but what if you want to collect something non-standard like the number of files in a folder?

The solution would be to create a custom performance collection script. The concept is basically the same for a custom script and a performance collection rule, the only catch here is that you will need to be able to create the script to do the collection. I would recommend using google to get you started with the scripting if you are not 100% sure where to start.

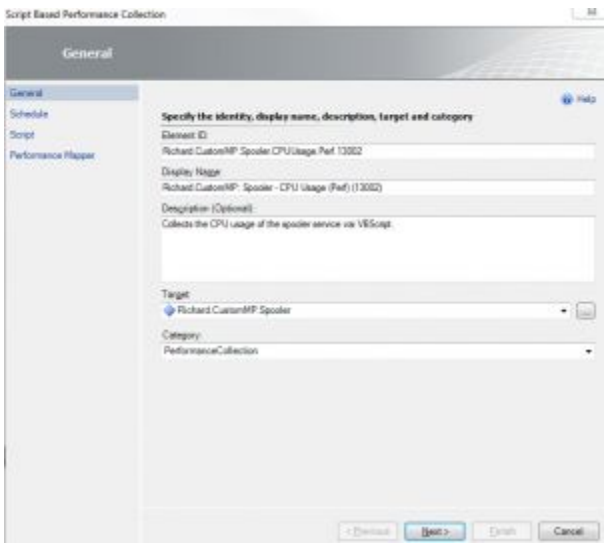
For the example used in this management pack I will be collecting the CPU usage of the spooler service. I will be using VBScript as this is the simplest way of collecting the information (until PowerShell is supported with wizards in the authoring console).

Adding our collection script to the Management Pack

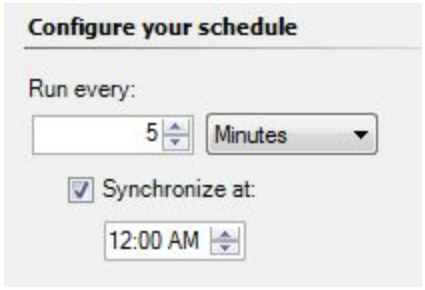
The first thing we will need to do is open up the Authoring Console, open our management pack and navigate over to Health Model -> Rules. Right click in the Rules pane and select New -> Collection -> Performance Based -> Script Based Performance Collection, this will launch the required wizard.

Complete the first page as follows:

ElementID: Richard.CustomMP.Spooler.CPUUsage.Perf.13002
Display Name: Richard.CustomMP: Spooler - CPU Usage (Perf) (13002)
Description: Collects the CPU usage of the spooler service vai VBScript.
Target: Richard.CustomMP.Spooler
Category: PerformanceCollection

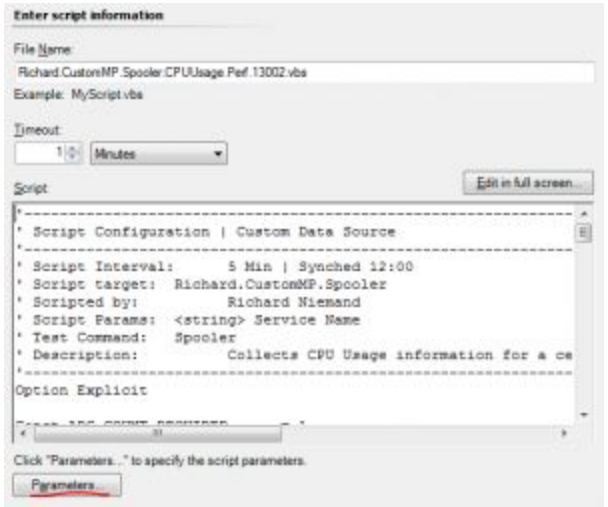


Click Next to continue to the schedule. For the schedule I am going to run this collection every 5 min.

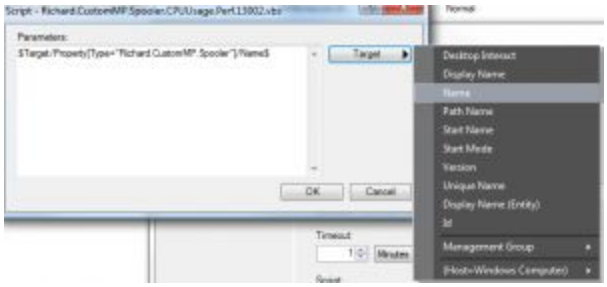


Click next to continue over to the script screen. Complete the following information and click the Parameters button when complete:

File Name: Richard.CustomMP.Spooler.CPUUsage.Perf.13002.vbs
Timeout: 1 Min
Script: Paste your script here



On the parameters screen, from the fly-out select the Name of the service, click OK when done.



Click Next to continue to the Performance Mapper Screen.

Our script works by connecting to WMI and querying for the relevant information regarding our service (in this case the CPU Usage), and submitting that information back to SCOM in the form of a property bag. The performance mapper section of the wizard allows us to tell SCOM which bits of the information returned needs to be collected as performance data. As you may have noticed (if you tested the script first) is that the data returned is in the form of XML which is an easy format to query. SCOM stores all the information back from the script in the $data/...$ variable, and to get the values we need to map we make use of Xpath queries against the returned XML. So basically $Data/Property[@Name='bob']$ is referring to an element name in the returned XML called bob.

Armed with my quick description of how information is given back from SCOM, lets complete the Performance Mapper page, the following information will need to be entered in:

Object: Spooler CPU
Counter: % CPU
Instance: Total
Value: $Data/Property[@Name='PercentProcessorTime']$

After importing the updated management pack and waiting 15 min I finally see the following information coming into SCOM, the only problem is that the spooler service is nothing special to look at :)

Click here to grab the latest build of the management pack.