

Jupiter

Information Technology

Analysing Computer Performance

01 October 2006

18 Church St
Swansea. SA4 3EA
United Kingdom

+44 (0)1792 875340
info@jup-it-er.co.uk
www.jup-it-er.co.uk

Summary

Computer performance can be affected by many factors. A single software application can sometimes use more than it's fair share of CPU time. In these cases, it is easy to see where the problem lies. In other cases certain combinations of applications and background processes can bring a computer grinding to a halt, without any obvious cause. In these cases it is necessary to delve deeper into the causes of the performance problems. Only when an objective analysis has been made can informed decisions be taken as to how to improve performance. This document describes a method of capturing computer performance data using standard Windows tools, and then analysing the data using a spreadsheet.

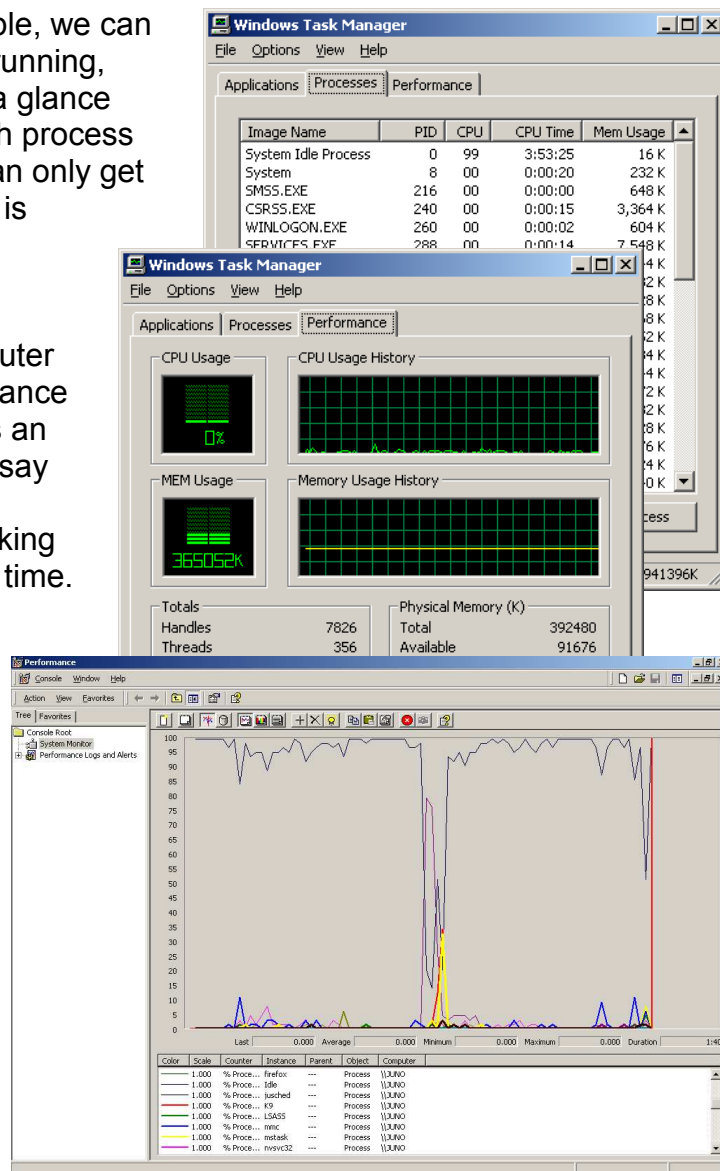
What processes are running?

In order to analyse computer performance, we need to know what is running. We can usually see which applications are running, but what about background processes? Windows offers some tools but they are limited in how we can use them. With the Task Manager, for example, we can see what processes are running, and we can even see at a glance how much CPU time each process is using. However, we can only get a subjective feel for what is happening.

We can check the overall performance of the computer by looking at the Performance tab, but this only gives us an overall view. We cannot say which applications or processes are actually taking a large share of the CPU time.

For more detail we could use the Performance Monitor, which can be found under Administrative Tools.

We can select which processes to monitor and this will give us a graphical display of process activity. However, this is not that useful.

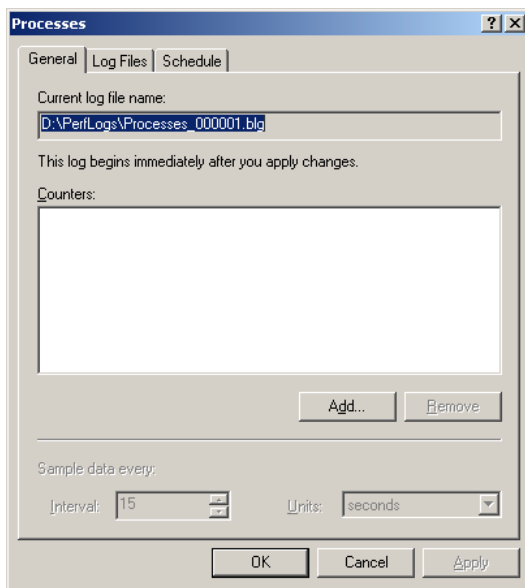
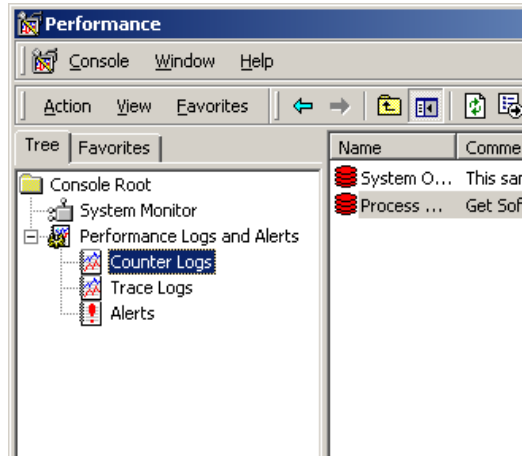


We need to find a way of getting more useful data, and capturing it for objective analysis.

Fortunately, Performance Monitor has all the tools we need to do the job.

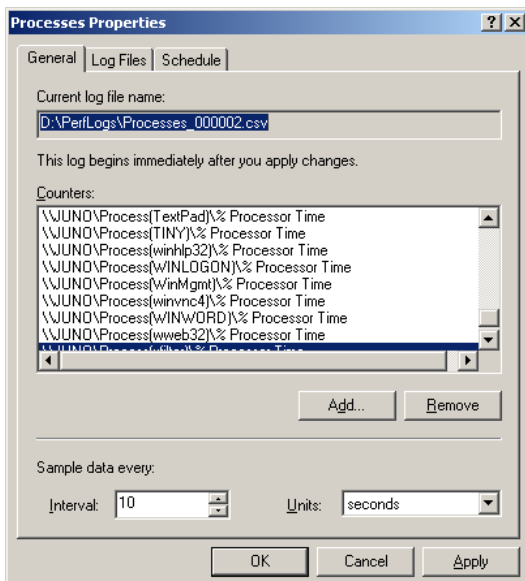
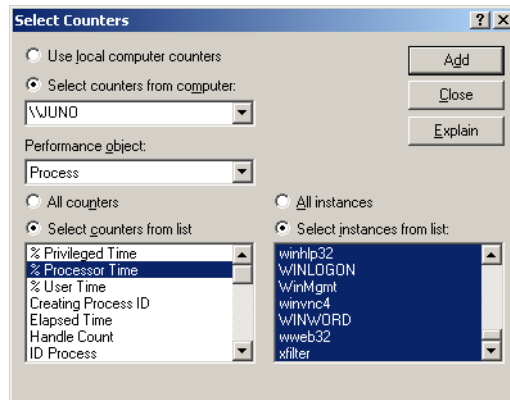
Data Capture

In the left hand pane, expand 'Performance Logs and Alerts', and highlight 'Counter Logs'. Right click and select 'New Log Settings...' and in the textbox type a name for the log and click OK.



Click on the 'Add' button. A dialog will appear. From the 'Performance object' drop-down list, select 'Process'. Now highlight all the processes that you want to monitor. For our purposes, we want to select all except 'Total'.

Now click the 'Add' button and click 'Close'.



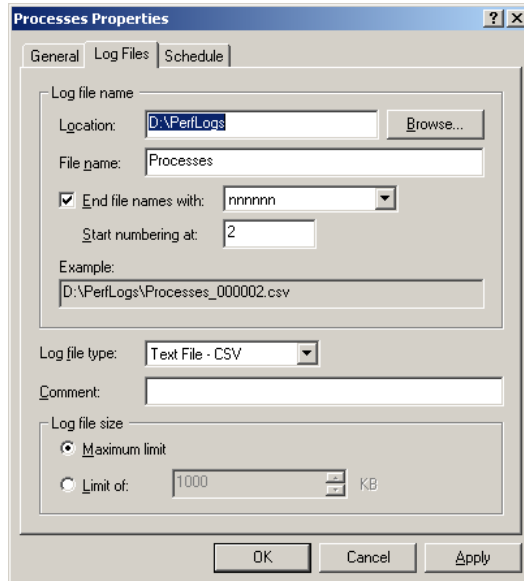
All the processes you selected will be displayed in the window. Adjust the sample rate to suit your needs, then select the 'Log Files' tab.

To be useful, we need the log file in a human readable form, so select 'Text File – CSV'. The log data will be sent to a Comma Separated Value list that can be imported into a spreadsheet for further analysis.

The log will be activated immediately, so you can stop it whenever it's convenient from the Performance Monitor window.

Preparing and analysing the data

Now that the data has been captured we need to do a little work before we can find out what processes are using the most CPU time. Import the CSV file into a spreadsheet. It is preferable to use Excel or Open Office / Star Office for our purposes.



Lotus 123, for example, won't recognise the numeric data as anything except text. Consequently, any calculations we try to do will not work.

	A	B	C	D	E	F
1	(PDH:CSV 4) \JUNOVProc	\JUNOVProc	\JUNOVProc	\JUNOVProc	\JUNOVProc	\JUNOVProc
2	09/30/2006 1▶	2.41E-007	5.92E-008	2.44E-009	7.31E-008	8.56E-007
3	09/30/2006 1▶	0	0	0	0	0.94
4	09/30/2006 1▶	0	0.16	0	0	1.25
5	09/30/2006 1▶	0	0	0	0	2.19
6	09/30/2006 1▶	0	0	0	0	0
7	09/30/2006 1▶	0	0	0	0	0.16
8	09/30/2006 1▶	0	0	0	0	0.16
9	09/30/2006 1▶	0	0	0	0	0
10	09/30/2006 1▶	0	0	0	0	0.78
11	09/30/2006 1▶	0	0	0	0	2.34
12	09/30/2006 1▶	0	0	0	0.78	0.63
13	09/30/2006 1▶	0	0	0	0.16	2.5
14	09/30/2006 1▶	0	0	0	0	1.41
15	09/30/2006 1▶	0	0	0	0	0.47
16	09/30/2006 1▶	0	0	0	0	0.31

The data will have to be converted to numeric type data first. We can do this directly within the CSV file using a text editor, but as the number of data points is going to be very high this will be a tedious and laborious task.

A quicker method would be to import the data into a database and then re-export it in the correct format.

Once we have it in our spreadsheet, we can see that the raw data needs some work. The headings for each column are impossible to distinguish and the default scientific notation format is not one that lends itself well to understanding by mere mortals. So, the first thing to do is to set the number format for each column to numeric to 2 decimal places.

Now, each number in each cell represents percentage of CPU usage at a particular moment in time. We need to know what the average CPU usage is over the duration of monitoring. Therefore, we need to create a formula at the

bottom of each column to sum the data, and then divide it by the number of data points. This is simple enough. In our example, the formula is `AVERAGE(C4:C24)`. Now, we just copy the formula across to each remaining column.

Next, we would like to be able to easily identify each process that is described in each column. A typical heading, so far, reads something like `\\JUNO\Process(123w)\% Processor Time`, which doesn't exactly trip off the tongue. We can see the relevant part of the heading is '123w', and this tells us that the process is the Lotus 123 spreadsheet. So we need to extract just that part of the heading. We could just visit each heading in turn and delete the extraneous portions. However, a quicker and simpler method would be to create a formula. So, first, insert a new row above the current headings. To extract a string from a string we can use the 'MID()' function. But to work out the length of the string to extract we need to use the 'FIND()' function. So that gives us `(MID(C$2;16;FIND(" ");C$2;16)-16)`. A further refinement is to give each heading a leading upper case character. For this we need the 'UPPER()' function. Now we have `UPPER(MID(C$2;16;1)) & (MID(C$2;17;FIND(" ");C$2;17)-17)`.

Now to easily understand which processes are using the most CPU time, it would be useful to sort the data so that the most CPU intensive processes are together. So select the entire data area, including formulae and headings, and sort them from left to right so that the highest CPU time processes are displayed at the left of the worksheet. This is a descending sort from highest to lowest. In most cases, the first process displayed should now be 'Idle'.

It's now easy to see which processes are using the most CPU time. This is very useful when you know something is affecting computer performance, but you don't know where the problem lies.

Once you have this information you can make informed choices about what processes you have running, or even which software you want to use.

It's useful to test different software combinations on a single computer until you find a combination that provides the optimum level of performance and usability. You can then deploy the tested configuration to other computers in your organisation.

	B	C	D	E	F
1	09/30/2006	Idle	PXAgent	Explorer	BigFix
3					
4	14:58:50	0.00	0.00	0.00	0.00
5	14:59:00	98.28	0.00	0.16	0.94
6	14:59:10	95.63	0.00	0.00	1.25
7	14:59:20	97.34	0.00	0.00	2.19
8	14:59:30	99.69	0.00	0.00	0.00
9	14:59:40	99.84	0.00	0.00	0.16
10	14:59:50	98.13	0.00	0.00	0.16
11	15:00:00	94.69	0.00	0.00	0.00
12	15:00:10	95.78	0.16	0.16	0.78
13	15:00:20	96.09	0.00	0.00	2.34
14	15:00:30	84.06	0.00	12.03	0.63
15	15:00:40	90.47	0.00	4.38	2.50
16	15:00:50	91.25	0.00	0.47	1.41
17	15:01:00	98.44	0.00	0.00	0.47
18	15:01:10	60.78	37.03	0.00	0.31
19	15:01:20	93.28	0.00	0.16	0.78
20	15:01:30	97.34	0.16	0.00	0.63
21	15:01:40	99.06	0.00	0.00	0.16
22	15:01:50	98.28	0.47	0.00	0.31
23	15:02:00	96.56	0.00	0.00	0.94
24	15:02:10	97.97	0.16	0.00	0.63
25					
26	Average CPU Time	89.67	1.81	0.83	0.79