

ProcessWatcher.NET v. 1.3

Manual

1. Installing the program

The install procedure is very simple. Simply save the executable (ProcessWatcher.NET.exe) in the location where you want it. Also, save the configuration file (Config.xml) in the same location. If you don't save this file, or save it elsewhere, the configuration will not be readable when starting the program and you will have to configure it your-self. Your configuration will be saved when exiting the program.

If wanted, create a shortcut to the program from the Start Menu and/or from the Desktop for easy access to the program. Save this document somewhere for future reference.

2. Starting the program

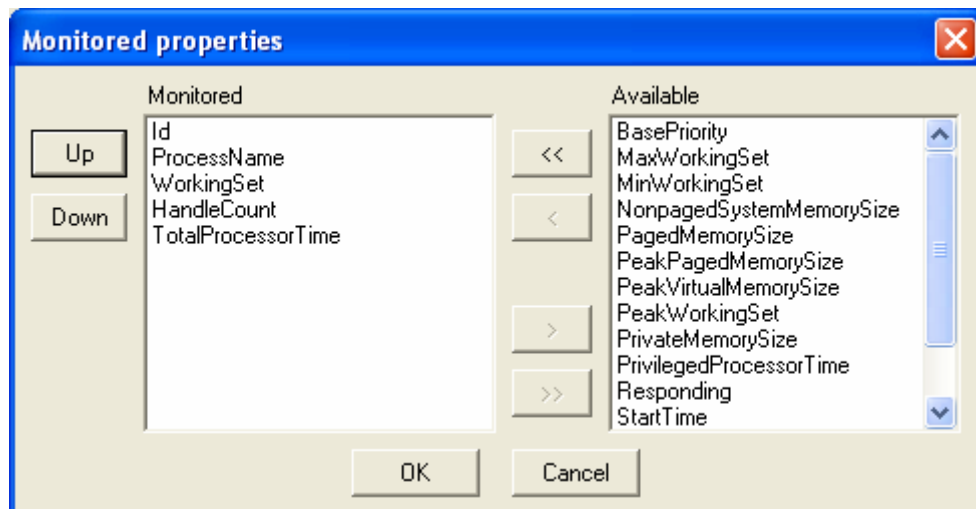
Double click the executable, or one of the created shortcuts, to start ProcessWatcher.NET v. 1.3. If there is a readable configuration available the program will start with the settings specified, otherwise the program will start with default settings.

3. Settings

The settings 'Monitored properties', 'File formats', 'Process filter', 'Log rotate' and 'Alarm settings' can only be changed when ProcessChecker.NET is idle. The color settings can be changed both when idle and when working.

3.1 Monitored properties

Select "Monitored properties..." in the "Settings"-menu to edit what process properties that you want displayed. The following dialog will appear:



Move the properties that you don't want to monitor to the right box with the button labeled ">" and the properties that you want to monitor to the left box with the

button labeled "<". You can also rearrange the monitored properties with the buttons labeled "Up" and "Down".

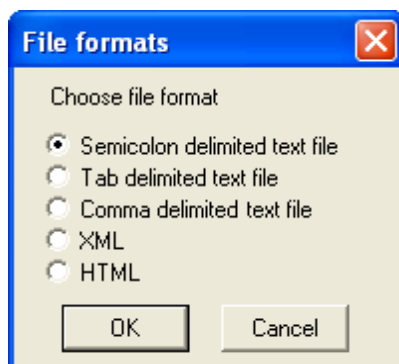
Use the buttons ">>" and "<<" to move all properties to the right box and left box, respectively.

Press OK to save the changed settings. Now, the selected properties will be monitored.

(See Appendix A for a list of properties that can be monitored, complete with descriptions)

3.2 File formats

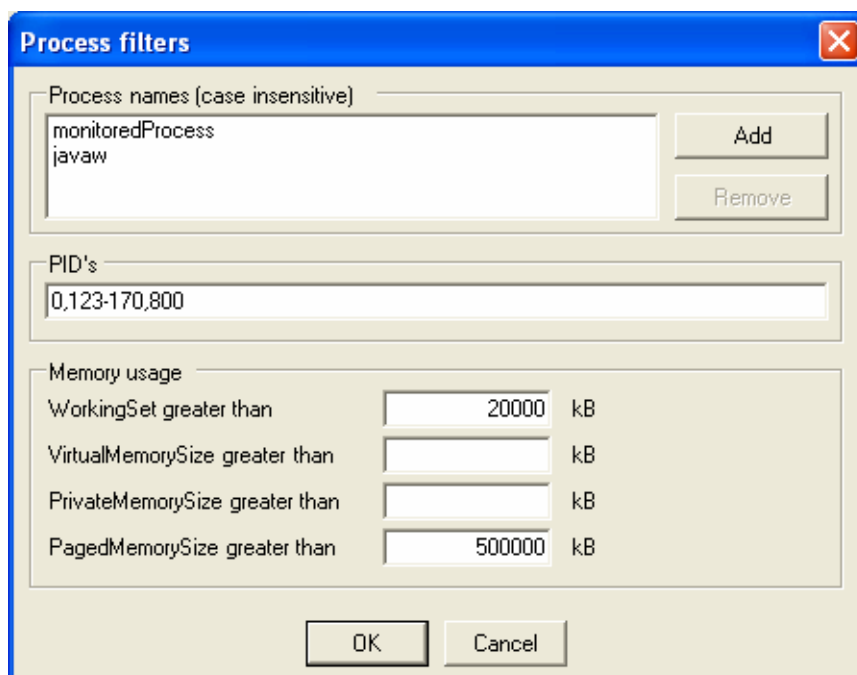
Select "File formats..." in the "Settings"-menu to edit how ProcessWatcher.NET will write files. The following dialog will appear:



Select the file format that you want to use and press OK to save the changed settings. Make sure that the file name extension corresponds to the selected file format.

3.3 Process filters

Select "Process filters..." in the "Settings"-menu to edit which processes should be monitored. The following dialog will appear:

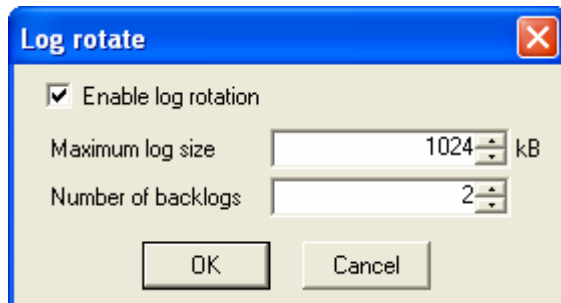


Filters are created in “OR-style”. If there are no filters available, all processes will be monitored, otherwise all processes that pass one or more of the filters will be monitored (i.e. not just the ones that pass all filters). In the example above the processes named “monitoredProcess” or “javaw” will always be monitored, as will processes with id 0, 123-170 or 800. Also, processes whose working set is greater than 20000 kB or paged memory size is greater than 500000 kB will be monitored. You can add as many process names and pid’s as you like, but the memory usage thresholds are global.

Press OK to save the changed settings.

3.4 Log rotation

Select “Log rotation...” in the “Settings”-menu to edit the maximum size of an output file and how many backlogs that should be saved. The following dialog will appear:



Use the up/down-buttons to change the values and press OK to save the changed settings.

3.5 Alarm settings

ProcessWatcher.NET can export alarms to another product from FUA.NU, namely the system monitoring tool Giraffe. Giraffe is a program that can monitor nodes, processes, webs, directories and files. It can also receive alarms from other applications and display them in the alarm monitor. Alarms can then be further exported by Giraffe, either to the Windows EventLog or via e-mail to a specified e-mail address. For more information about Giraffe, please visit <http://www.fua.nu/giraffe>.

NOTE: If you wish to generate an alarm when the process IEXPLORE’s WorkingSet is greater than x kB you need to make sure that WorkingSet is a monitored property and that IEXPLORE is listed in your process filters. An alarm will only be generated if the process can be found. This is however not the case when you check the checkbox “alarm when process isn’t running”. If the process is running but your process filter won’t let it though an alarm will be (erroneously) generated.

Select “Alarm settings...” in the “Settings”-menu to edit when and how alarms should be generated and where the alarm files (because alarms in Giraffe are file based) should be written. The following dialog will appear:

Alarm settings for export to Giraffe

Existing alarm settings

Process\Watcher.NET

Alarm details

Name: Process\Watcher.NET

Alarm text: Process\Watcher.NET not running | not responding

Severity: HIGH

Alarm conditions

Process name: Process\Watcher.NET

☒ alarm when process isn't running

Property: Responding

☒ = ☐ > ☐ < ☐ !=

Value: False

Delete Clear Add

Path to alarm directory: c:\GiraffeAlarms\

Cancel OK

To the left you can see all alarm conditions already defined. You can always view the condition by selecting it. Enter the values for when an alarm should be generated to the right and press Add to add it. When you are done adding alarm conditions, press OK to save the settings and close the window.

3.6 Colors

Select "Colors ..." in the "Settings"-menu to edit the color(s) used in the process list shown on screen (this has nothing to do with the produced documents). The following dialog will appear:

Colors

Odd row color: Choose

Even row color: Choose

Sample

Header1	Header2	Header3
Item 00	Item 01	Item 02
Item 10	Item 11	Item 12
Item 20	Item 21	Item 22
Item 30	Item 31	Item 32
Item 40	Item 41	Item 42
Item 50	Item 51	Item 52
Item 60	Item 61	Item 62
Item 70	Item 71	Item 72

OK Cancel

Press the “Choose” button for the odd row color and the even row color respectively. This will launch a color choosing dialog where you can choose what colors you want to use. The sample in the dialog shown above will automatically be updated and you can see what it would look like.

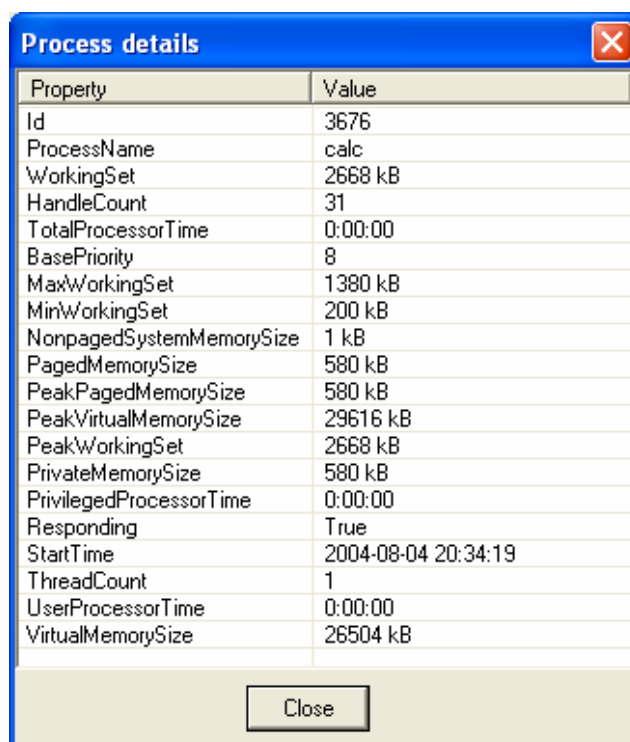
Press OK to save the changed settings or Cancel to keep the old settings.

4. Monitoring

If you want to export the results from ProcessWatcher.NET to a file, make sure that the checkbox is checked and that the Output textbox contains the complete file name (including path) to the file where you want the output saved. If the file doesn’t exist it will be created. If you don’t want to export the results to a file, make sure that the checkbox is cleared. Then edit the update interval to suit your needs and press the button labeled “Start”.

The list will be updated every time that the ProcessWatcher.NET updates the process view and you can always see when the latest update occurred in the status bar at the bottom of the window.

To see the details about the process in a more well-arranged way you can double click the process in the list. The window shown below will appear (here for the MS Windows calculator ‘calc’).



The screenshot shows a dialog box titled "Process details" with a close button in the top right corner. It contains a table with two columns: "Property" and "Value". The table lists various system properties for the 'calc' process, such as Id, ProcessName, WorkingSet, HandleCount, and memory usage. At the bottom of the dialog is a "Close" button.

Property	Value
Id	3676
ProcessName	calc
WorkingSet	2668 kB
HandleCount	31
TotalProcessorTime	0:00:00
BasePriority	8
MaxWorkingSet	1380 kB
MinWorkingSet	200 kB
NonpagedSystemMemorySize	1 kB
PagedMemorySize	580 kB
PeakPagedMemorySize	580 kB
PeakVirtualMemorySize	29616 kB
PeakWorkingSet	2668 kB
PrivateMemorySize	580 kB
PrivilegedProcessorTime	0:00:00
Responding	True
StartTime	2004-08-04 20:34:19
ThreadCount	1
UserProcessorTime	0:00:00
VirtualMemorySize	26504 kB

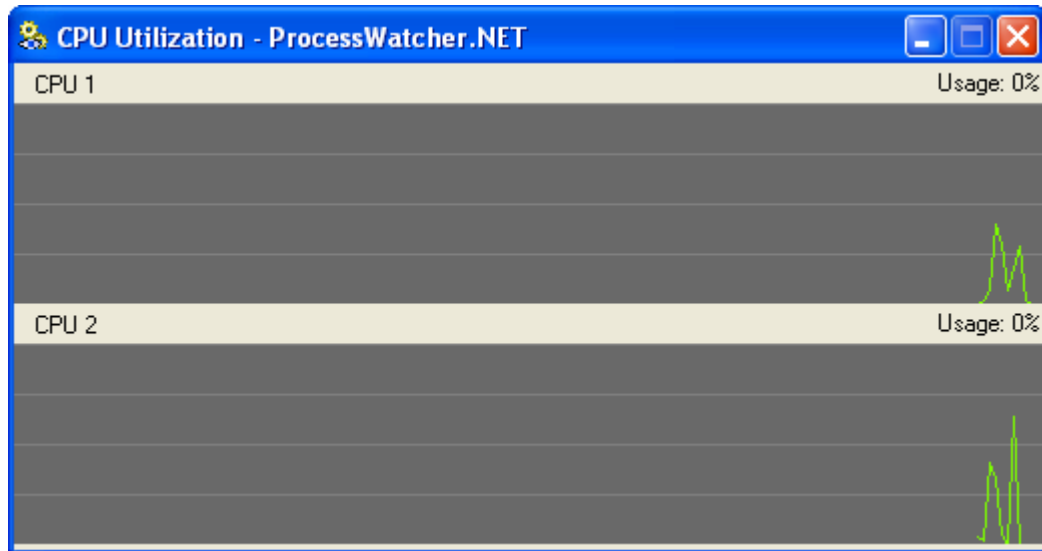
When the monitoring starts the “Stop”-button will be visible and pressing this button will stop the monitoring and making it possible to edit settings again.

6. CPU

ProcessWatcher.NET enables you to monitor the total CPU utilization on up to four processors and also to view all known details about the CPUs.

6.1 CPU utilization

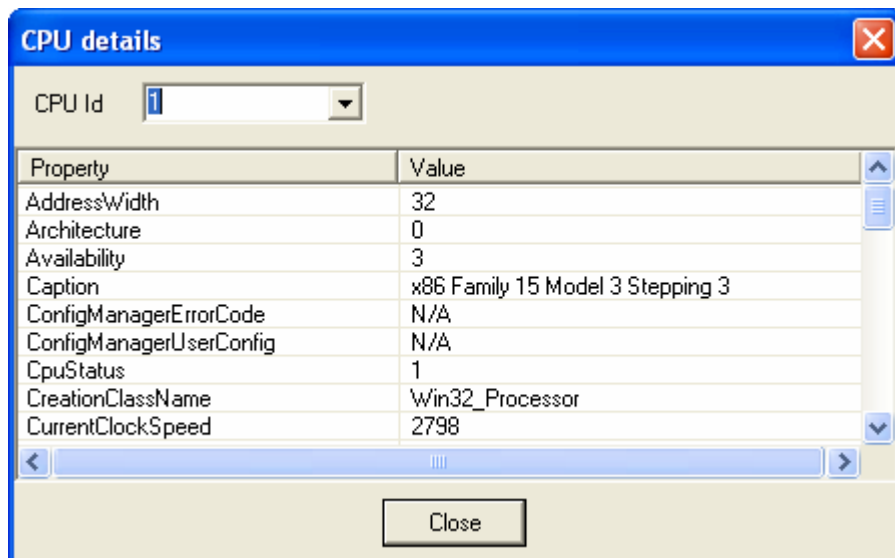
To view the current CPU utilization, select “View CPU utilization...” in the CPU menu. The following dialog will appear (it will look different depending on the number of processors in your computer). The graph will be redrawn approximately once every second. (Here I just started Winamp)



The horizontal lines are drawn at 25%, 50% and 75% so that it is easier to see the CPU utilization in the graph.

6.2 CPU details

To view all known details about the CPU's, select “Details...” in the CPU menu. The following dialog will appear showing details about processor number 1 by default.



To change what processor to view details from, select another CPU Id in the drop down box at the top of the dialog. Press “Close” to close the dialog and return to the main application window.

6. Saving settings

The settings will be saved automatically when you exit the program.

7. Problems?

If you have problems with the program, please send an email to jobro@fua.nu and your questions will be answered as quickly as possible.

APPENDIX A

Properties that can be monitored (with descriptions) in alphabetical order

Property	Description
<i>BasePriority</i>	The base priority of the process
<i>HandleCount</i>	The number of handles opened by the process
<i>Id</i>	The unique identifier for the process
<i>MaxWorkingSet</i>	The maximum allowable working set size for the process
<i>MinWorkingSet</i>	The minimum allowable working set size for the process
<i>NonpagedSystemMemorySize</i>	The nonpaged system memory size allocated to the process
<i>PagedMemorySize</i>	The paged memory size
<i>PeakPagedMemorySize</i>	The peak paged memory size
<i>PeakVirtualMemorySize</i>	The peak virtual memory size
<i>PeakWorkingSet</i>	The peak working set size for the associated process
<i>PrivateMemorySize</i>	The private memory size
<i>PrivilegedProcessorTime</i>	The privileged processor time for the process
<i>ProcessName</i>	The name of the process
<i>Responding</i>	A value indicating whether the user interface of the process is responding
<i>StartTime</i>	The time that the process was started
<i>ThreadCount</i>	The number of threads running in the associated process
<i>TotalProcessorTime</i>	The total processor time for the process
<i>UserProcessorTime</i>	The user processor time for the process
<i>VirtualMemorySize</i>	The size of the process's virtual memory
<i>WorkingSet</i>	The process's physical memory usage