

# The System Monitor Handbook

Chris Schlaeger  
John Tapsell  
Chris Schlaeger  
Tobias Koenig



# The System Monitor Handbook

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Using System Monitor</b>	<b>7</b>
2.1	Getting started . . . . .	7
2.2	Process Table . . . . .	7
2.2.1	End Processes . . . . .	9
2.2.2	Filter Bar . . . . .	9
2.2.3	Process Filter . . . . .	9
2.3	Work Space . . . . .	9
2.3.1	Sensor Browser . . . . .	10
2.3.2	Line Graph . . . . .	10
2.3.3	Digital Display . . . . .	10
2.3.4	Bar Graph . . . . .	10
2.3.5	Log to a File . . . . .	10
2.3.6	Partition Table . . . . .	10
2.3.7	Connecting to other hosts . . . . .	11
<b>3</b>	<b>Configuring ksysguardd</b>	<b>12</b>
<b>4</b>	<b>Credits and License</b>	<b>13</b>

# List of Tables

2.1	Default Columns in the Process Table . . . . .	8
2.2	Additional Columns in the Process Table . . . . .	9

### **Abstract**

System Monitor is a network enabled task and system monitor application.

# Chapter 1

## Introduction

System Monitor is the Plasma Task and Performance Monitor.

It features a client/server architecture that allows monitoring of local as well as remote hosts. The graphical front end uses so-called sensors to retrieve the information it displays. A sensor can return simple values or more complex information like tables. For each type of information, one or more displays are provided. Displays are organized in worksheets that can be saved and loaded independently from each other. So, System Monitor is not only a simple task manager but also a very powerful tool to control large server farms.

## Chapter 2

# Using System Monitor

### 2.1 Getting started

System Monitor can be started from the application launcher menu, using the entry **System Monitor** in the **Applications** → **System** menu. Alternatively, you can start it by typing **ksysguard** in a terminal.

The System Monitor main window consists of a menu bar, an optional tool bar and status bar, and the work space. Custom worksheets will also show the sensor browser.

By default System Monitor shows two worksheets: **Process Table** and **System Load**. The **Process Table** lists the running processes and lets the user control them. Multiple processes can be selected and controlled at once. The **System Load** worksheet shows graphs of system utilization: **CPU History**, **Memory and Swap History**, and the **Network History**.

This default setup is sufficient enough for an inexperienced user to do some system management. An experienced user or even a system administrator of a large computer lab has different needs. To address a wide range of users, System Monitor is highly flexible.

#### NOTE

You can launch the **Process Table** using the global shortcut **Ctrl+Esc** at any time. The process table is displayed in a window titled **System Activities**.

### 2.2 Process Table

The Process Table gives you a list of processes on your system. The list can be sorted by each column. Just press the left mouse button at the head of the column.

Use the **What's This** help for the columns titles to get additional information about the value displayed here.

In the context menu of a process in the list view you find additional actions like changing the priority, sending signals to the process, switching to the application window, showing detailed memory information and killing the process.

The list shows the following information about each process. Please note that not all properties are available on every operating system.

## The System Monitor Handbook

<b>Name</b>	The name of the executable that started the process
<b>Username</b>	The user who owns this process
<b>CPU %</b>	The current total CPU usage of the process, divided by the number of processor cores in the machine
<b>Memory</b>	This is the amount of real physical memory that this process is using by itself, and approximates the Private memory usage of the process. It does not include any swapped out memory, nor the code size of its shared libraries. This is often the most useful figure to judge the memory use of a program.
<b>Shared Mem</b>	This is approximately the amount of real physical memory that this process's shared libraries are using. This memory is shared among all processes that use this library
<b>Window Title</b>	The title of any window that this process is showing

Table 2.1: Default Columns in the Process Table

<b>PID</b>	The unique Process ID that identifies this process
<b>TTY</b>	The controlling terminal on which this process is running
<b>Niceness</b>	The priority with which this process is being run. For the normal scheduler, this ranges from 19 (very nice, least priority) to -19 (top priority)
<b>CPU Time</b>	The total user and system time that this process has been running for, displayed as minutes:seconds
<b>IO Read</b>	The number of bytes read. The <b>Display Units</b> and the <b>Displayed Information</b> can be changed using the context menu of this column header
<b>IO Write</b>	The number of bytes written. The <b>Display Units</b> and the <b>Displayed Information</b> can be changed using the context menu of this column header
<b>Virtual Size</b>	This is the amount of virtual memory space that the process is using, included shared libraries, graphics memory, files on disk, and so on. This number is almost meaningless. Use the context menu to select the <b>Display Units</b>
<b>Command</b>	The command with which this process was launched



---

Table 2.2: Additional Columns in the Process Table

At the top of the table you find three controls which will be described now from left to right.

### 2.2.1 End Processes

If you have selected one or more processes you can press the **End Process** button to kill them. A so called SIGKILL is sent to the processes which causes them to terminate immediately. If these applications still have unsaved data this data will be lost. So use this button with care.

### 2.2.2 Filter Bar

Filter which processes are shown by the text given here. The text can be a partial string match of the **Name**, **Command** or **Window Title** of the process. It can also be a **Username** or a Process ID number.

### 2.2.3 Process Filter

The Process Filter can be used to reduce the number of processes displayed in the table. You can filter out processes you are not interested in. Currently you can display **All Processes** in a flat or tree view, **System Processes** only, **User Processes** only, your **Own Processes** only or **Programs Only**.

The tree view has been designed to show the relationships between the running processes. A process that is started by another process is called the child of that process. A tree is an elegant way to show this parent-child relationship. The *init* process is the ancestor of all processes.

If you are not interested in the children of a particular process you can click on the little box to the left of the parent and the subtree will collapse. Another click on that box will unfold the subtree again.

## 2.3 Work Space

The work space is organized as worksheets. Select **New Tab...** from the **File** menu to create a new worksheet. A dialog will appear where you can set the name, the dimension and the update interval of the worksheet. To remove a worksheet again, select **Close Tab** from the **File** menu. Any modifications will be saved to the worksheet file. If a worksheet has never been saved, you will be asked for a file name. Worksheets consist of cells organized as a grid.

Each cell can be filled with a display for one or more sensors. You can fill a cell by dragging a sensor from the sensor browser and dropping it over the cell. If there is more than one type of display available for that type of sensor, a popup menu will appear. You can then select which display you prefer to use. Certain types of displays can display more than one sensor. Add more sensors to a display by dragging them over from the sensor browser and dropping them over the already existing display.

Worksheets can be configured by clicking **Tab Properties** at the **View** menu. In the appearing dialog you can set the dimension and the update interval.

Displays can be configured by clicking with the right mouse button on them. A popup menu appear where you can select whether you want to change the properties of that display or remove it from the worksheet.

### 2.3.1 Sensor Browser

The sensor browser exposes System Monitor's advanced functionality. To use it, you must first go to the **File** menu and create a new worksheet. It is shown whenever a custom worksheet is selected.

The sensor browser displays the registered hosts and their sensors in a tree form. Click on the tree handles to open or close a branch. Each sensor monitors a certain system value.

**TIP**

After you have configured your custom worksheet use the splitter and move it to the right edge of the window to hide the sensor browser.

If the sensor browser does not appear on a custom worksheet, it is probably hidden. To unhide it, select the right edge of the window and drag it to the left.

### 2.3.2 Line Graph

The line graph prints samples of one or more sensors over time. If several sensors are displayed, the values are piled in different colors. If the display is large enough a grid will be displayed to show the range of the plotted samples. By default, the automatic range mode is active so the minimum and maximum values will be set automatically. Sometimes you want fixed minimum and maximum values. In that case, you can deactivate automatic range mode and set the values in the properties dialog.

### 2.3.3 Digital Display

The multimeter displays the sensor values as a digital meter. In the properties dialog you can specify a lower and upper limit. If the range is exceeded, the display is colored in the alarm color.

### 2.3.4 Bar Graph

The bar graph displays the sensor values as dancing bars. In the properties dialog you can specify minimum and maximum values of range and a lower and upper limit. If the range is exceeded, the display is colored in the alarm color.

### 2.3.5 Log to a File

The sensor logger does not display any values, but logs them in a file with additional date and time information. For each sensor you can specify a lower and upper limit in the properties dialog. If the range is exceeded, the entry of the sensor table is colored in the alarm color.

### 2.3.6 Partition Table

The **Partition Usage** has a special table sensor showing information about all mounted partitions

### 2.3.7 Connecting to other hosts

Note that the **Monitor Remote Machine** menu item is only available on custom tabs.

To connect to a new host use **Monitor Remote Machine...** from the **File** menu. A dialog box will appear and allows you to enter the name of the host you want to connect to. Below the name you can choose the connection method. The default is ssh, the secure shell. Alternatively the rsh, the remote shell, the daemon mode or a custom command can be used. Click **OK** to establish the connection. Shortly afterwards the new host will appear in the sensor browser and you can browse the list of sensors.

To disconnect from a host, you have to shut down it and restart System Monitor again.

To establish a connection, a program called `ksysguardd`, that can be started in the following two modes, must be installed on the new host.

#### daemon mode

You can start `ksysguardd` at boot time in **Daemon** mode by adding `-d` as the argument. In this case, you have to select daemon mode at the connection dialog of `ksysguard`. A disadvantage of this connection type is that you won't be able to kill or renice a process in the **Process Table** and the data exchange over network won't be encrypted. As a result, daemon mode is not recommended.

#### shell mode

In this mode `ksysguardd` is started at connecting time by `ksysguard`. To make that possible, its location needs to be included in your `PATH`. Unfortunately the ssh does not source your `.profile` file, so your regular `PATH` setting will not be available. Instead it uses a default `PATH` like `/bin:/usr/bin`. Since it is very likely that Plasma is not installed in these folders you need to create or update a file in your home folder. The file is called `environment` and needs to be in a hidden folder called `.ssh`. See the manual page for ssh for more details. The file needs to contain a line similar to:

```
PATH=/bin:/usr/bin:/opt/kde/bin
```

assuming that `ksysguardd` can be found under `/opt/kde/bin/ksysguardd`.

#### TIP

When using ssh you should make sure that you have your `identity.pub` installed on the remote machine and the host key of the remote machine is already registered on your machine. If you don't set up `identity.pub` correctly, you will be asked for your password every time you start `ksysguard`. The easiest way to make sure that everything is working is to run **ssh remotehost ksysguardd** in a shell. If you are greeted by `ksysguardd`, then everything is working correctly and you can type **quit** to exit `ksysguardd`.

#### NOTE

For experts: `ksysguardd` is a very small program that is only linked against the `libc`. So it can also be used on machines that do not have a full blown Plasma installed, such as servers. Many major distributions provide a separate `ksysguardd` package for your convenience. If you choose the custom command option in the host connector you need to specify the complete command to start `ksysguardd`.

## Chapter 3

# Configuring ksysguardd

The graphical front-end is available on any platform that Plasma runs on. The back-end is at the moment available on the following flavors of UNIX<sup>®</sup>:

### Linux<sup>®</sup>

For ksysguardd to work it is necessary to compile the Linux<sup>®</sup> Kernel with the `/proc` File system enabled. This is the default setting and most Linux<sup>®</sup> Distributions have it already.

### FreeBSD

The ksysguardd program needs to be owned by the `kmem` group and needs to have the `setgid` bit set.

### Solaris<sup>™</sup>

To be written

Support for other platforms is in progress. Your help is greatly appreciated.

## Chapter 4

# Credits and License

System Monitor is currently developed and maintained by John Tapsell [john.tapsell@kde.org](mailto:john.tapsell@kde.org). System Monitor is a rewrite of KTop, the KDE 1.x task manager. Several other people have worked on KTop:

- A. Sanda [alex@darkstar.ping.at](mailto:alex@darkstar.ping.at)
- Ralf Mueller [ralf@bj-ig.de](mailto:ralf@bj-ig.de)
- Bernd Johannes Wuebben [wuebben@math.cornell.edu](mailto:wuebben@math.cornell.edu)
- Nicolas Leclercq [nicknet@planete.net](mailto:nicknet@planete.net)

The porting to other platforms than Linux<sup>®</sup> was done by:

- FreeBSD: Hans Petter Bieker [zerium@traad.lavvu.no](mailto:zerium@traad.lavvu.no)

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).