

# The KSysguard Handbook

Chris Schlaeger, Chris Schlaeger, and Tobias Koenig



# The KSysguard Handbook

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Using KSysguard</b>	<b>2</b>
2.1	Getting started . . . . .	2
2.2	The Sensor Browser . . . . .	2
2.2.1	Connecting to other hosts . . . . .	2
2.2.2	Disconnecting hosts . . . . .	3
2.3	The Work Space . . . . .	4
2.3.1	Signal Plotter . . . . .	4
2.3.2	Multimeter . . . . .	4
2.3.3	Process Controller . . . . .	5
2.3.3.1	The Tree View . . . . .	5
2.3.3.2	The Process Filter . . . . .	5
2.3.3.3	The Refresh Button . . . . .	6
2.3.3.4	The Kill Button . . . . .	6
2.3.4	BarGraph . . . . .	6
2.3.5	Sensor Logger . . . . .	6
2.3.6	Log File . . . . .	6
2.3.7	List View . . . . .	6
<b>3</b>	<b>Configuring ksysguardd</b>	<b>7</b>
<b>4</b>	<b>Credits and Licenses</b>	<b>8</b>

### **Abstract**

KSysguard is a network enabled task manager and system monitor application, with the additional functionality of top.

## Chapter 1

# Introduction

KSysguard is the KDE Task Manager 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 work sheets that can be saved and loaded independently from each other. So, KSysguard is not only a simple task manager but also a very powerful tool to control large server farms.

## Chapter 2

# Using KSysguard

### 2.1 Getting started

KSysguard can be started from the start menu, using the entry KDE System Guard in the Systems menu. Alternatively, you can start it by typing `ksysguard` in a terminal.

The KSysguard main window consists of a menu bar, an optional tool bar and status bar, the sensor browser and the work space. When first started you see your local machine listed as localhost in the sensor browser and 2 pages in the work space area. This is the default setup.

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, KSysguard is highly flexible.

### 2.2 The Sensor Browser

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.

#### 2.2.1 Connecting to other hosts

To connect to a new host use Connect Hosts 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, or the daemon mode

## The KSysguard Handbook

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 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 with the Process Controller and the data exchange over network won't be encrypted.

**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 KDE 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. The easiest way to check this is to type `ssh remotehost ksysguardd` in a shell. If you are greeted by `ksysguardd` you can type `quit` and everything is in order.

### 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 KDE installed, such as servers. If you choose the custom command option in the host connector you need to specify the complete command to start `ksysguardd`.

## 2.2.2 Disconnecting hosts

To disconnect from a host, select the host in the sensor browser and choose Disconnect Host from the File menu. If you still have sensors in use, the display frames will be grayed and the displays won't update any longer.

## 2.3 The Work Space

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

Work sheets can be configured by clicking Configure Worksheet at the Edit menu. In the appearing dialog you can set the dimension and the update interval. This update interval is used by all displays of the worksheet, which has the use update interval of worksheet set in its timer configuration dialog.

The entry Configure Style of the Settings menu gives you the possibility to configure the global style attributes and apply them to the current active worksheet.

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, remove it from the work sheet, change its update interval type and value or pause and restart its updating.

### 2.3.1 Signal Plotter

The signal plotter 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.

### 2.3.2 Multimeter

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.

### 2.3.3 Process Controller

The Process Controller 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.

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

**Name** The name of the executable that started the process.

**PID** The Process ID. A unique number for each process.

**PPID** The Process ID of the process parent.

**UID** The ID of the user that started the process.

**GID** The ID of the group the process belongs to.

**Status** The process status.

**User%** The processor load of the process in user space (in percent).

**System%** The processor load of the process in system space (in percent).

**Nice** The scheduling priority.

**VmSize** The total amount of virtual memory used by the process (in kBytes).

**VmRss** The total amount of physical memory used by the process (in kBytes).

**Login** The login name of the user that started the process.

**Command** The complete start command of the process.

Underneath the table you find four buttons which will be described now from left to right.

#### 2.3.3.1 The Tree View

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.

#### 2.3.3.2 The 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, system processes only, user processes only or your processes only.

### 2.3.3.3 The Refresh Button

This button can be used to force an immediate update of the process list.

### 2.3.3.4 The Kill Button

If you have selected one or more processes you can press the kill button to kill them. A so called SIGKIL 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.

## 2.3.4 BarGraph

The bargraph 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 Sensor Logger

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 and a notify event is sent.

## 2.3.6 Log File

The log file monitor displays the content of a file e.g. `/var/log/messages`. In the properties dialog, you can compose a list of regular expressions that will be compared with the content of the file. If one of the expressions match, a notify event will be sent.

## 2.3.7 List View

The listview displays the data of some sensors in the form of a table.

## Chapter 3

# Configuring ksysguardd

The graphical front-end is available on any platform that KDE runs on. The back-end is at the moment available on the following flavors of UNIX®:

**Linux® 2.x** For ksysguardd to work it is necessary to compile the Linux® Kernel with the `/proc` Filesystem enabled. This is the default setting and most Linux® 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™** To be written

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

## Chapter 4

# Credits and Licenses

KSysguard is currently developed and maintained by Chris Schläger [cs@kde.org](mailto:cs@kde.org). KSysguard 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® 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](#).