

The Kicker Applets Handbook



The Kicker Applets Handbook

Contents

1 Kicker applets	1
2 Introduction	2
2.1 Thanks and Acknowledgments	1
3 Introduction	2
3.1 Onscreen Fundamentals	1
3.1.1 Display Modes	1
3.1.1.1 Normal Mode	1
3.1.1.2 Extended Mode	2
3.2 Menu Structure	3
3.2.1 Settings...	3
3.2.2 Docked In Panel	3
3.2.3 Help	3
3.2.4 Horizontal Bars	3
3.2.5 Quit	4
3.3 Configuration	4
3.4 Common Questions and Answers	4
3.5 Thanks and Acknowledgments	5
4 Credits and License	6
A Installation	7
B Index	8

Abstract

This is the handbook for the addon applets for Kicker, KDE's panel.

Chapter 1

Kicker applets

Applets are small applications running inside of Kicker. Almost everything besides the application starter buttons are applets. They can be added from the context menu or from the submenu Configure Panel by choosing Add and the Applet.

More information on the use of Kicker and on the base set of applets that come with Kicker, can be found in the Kicker handbook.

The Applets described in this handbook are:

- [KolourPicker](#)
- [KTimemon](#)

Chapter 2

Introduction

KolourPicker

Abstract

To be written

The Kicker Applets Handbook

The Kicker plugin 'Kolourpicker' currently has no documentation. If you are interested in rectifying that situation, please contact the kde-doc-english mailing list, which you can reach at <http://master.kde.org/mailman/listinfo/kde-doc-english>.

2.1 Thanks and Acknowledgments

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

Chapter 3

Introduction

KTimemon Martin Maierhofer 2001-11-29 0.03.01

Abstract

KTimemon is a system monitor for the K Desktop Environment

KTimemon is a small program to keep track of your computer's system usage. It can display bar graphs containing information about CPU, memory, and swap usage as well as disk usage and context switch activity. In keeping with the spirit of KDE, it supports configuration via a graphical user interface. It also supports *docking*, i.e. it can display information in the system panel tray.

NOTE

Currently, KTimemon only supports a limited number of systems: Linux® based installations with the `/proc` file system, Solaris™ based installations with the `ks-tat` library, and Digital UNIX® (formerly DEC/OSF1) based installations with the `table(2)` system call. Help with porting it to other platforms is most welcome.

KTimemon can be started from the command line or from the KDE start menu (in the Utilities submenu). If you choose to start from the command line, KTimemon honors the usual X Window System® program flags such as `--geometry`. KTimemon is *session-aware*, i.e. it keeps track of the current state (colors, etc.) and restores it in the user's next session.

3.1 Onscreen Fundamentals

After starting KTimemon a small window will appear displaying information gathered from the operating system. If you move the mouse pointer over the KTimemon window and let it rest for a small amount of time, a *tool-tip* (i.e. a small transient window) will appear. The tool-tip contains numeric information about the system parameters displayed by the bar graphs. Tool-tips can be disabled (refer to [Configuration](#)).

3.1.1 Display Modes

KTimemon can display two different sets of system information. As explained in the [Configuration](#) chapter, mouse buttons can be bound to various actions. Per default, the left mouse button is bound to the mode switch action, i.e. by clicking the left mouse button anywhere in the KTimemon window, the displayed information switches from Normal Mode (the default) to Extended Mode, and vice versa.

3.1.1.1 Normal Mode

After starting KTimemon for the first time, it will show information about the current CPU activity, as well as memory and swap usage. Three bar graphs are used to show this information; they are updated regularly (the default sample interval is 0.5s, but it can be changed, see [Configuration](#)). The three bar graphs represent (from left to right):

CPU usage. KTimemon shows the bar in three different colors, representing CPU time spent in various modes. From bottom to top they are: kernel mode, user mode, and user mode with lowered priority (*nice*) - since Solaris™ does not seem to support statistics for nice mode, the topmost part of the bar represents time spent in the *wait* state on such systems. The gap from the top of the bar to the top of the window represents the percentage the CPU idle time.

Memory usage. Similar to the CPU usage bar, this bar is composed of three sub fields, representing (from bottom to top): memory allocated by processes, memory used for I/O buffering, and memory used for file caching. For Digital UNIX® based systems, the middle section represents 'inactive' memory (i.e. memory allocated and not used for a certain amount of time), and for Solaris™ based systems, the middle section of the bar is not used, and the topmost section represents the amount of memory used by the kernel. Again, the gap from the top of the bar to the top of the window represents free memory.

Swap usage. This bar consists of a single field representing the current swap usage relative to the system's total amount of swap space.

Clicking the mouse button bound to 'mode switch' in the KTimemon window switches to 'Extended Mode'.

3.1.1.2 Extended Mode

In this mode, the three bar graphs are used to display a different set of system information. Again from left to right, they show:

Paging activity. This bar consists of two parts, the lower half of which shows the number of memory pages written to secondary storage in the last sample interval. Similarly, the upper half indicates the number of pages read from secondary storage.

Swapping activity. The second bar displays the analog information for swap activity.

Context switches. Again, this bar graph consists of a single field which indicates the number of context switches in the last sample interval.

Since there is no 'natural' way of scaling the information shown in 'Extended Mode', by default KTimemon uses *autoscaling* (explained in the [Common Questions Section](#)). There is, however, the possibility of specifying the scaling information, see the [Configuration](#) section.

Note that the two sets of bar graphs share the same colors, i.e. the colors setup for 'Normal Mode' is also used for displaying information in 'Extended Mode' (see also [Configuration](#) on how to change the color scheme).

3.2 Menu Structure

By default, the right mouse button mouse button is bound to the 'menu pop-up' action, i.e. clicking the right mouse button anywhere in the KTimemon window brings up a menu, which is discussed in the following sections.

3.2.1 Settings...

The Settings... menu item is used to pop up the configuration dialog. Configuration options are discussed in section [Configuration](#).

3.2.2 Docked In Panel

By selecting the Docked In Panel menu item, KTimemon switches between its standard display (i.e. a normal window) and the panelized state, where the KTimemon window disappears and a smaller version is displayed in the system panel. Apart from the reduction in size, the 'panelized' KTimemon behaves exactly like its big brother.

3.2.3 Help

Help → **this application Handbook (F1)** Invokes the KDE Help system starting at the this application help pages. (this document).

Help → **What's This? (Shift+F1)** Changes the mouse cursor to a combination arrow and question mark. Clicking on items within this application will open a help window (if one exists for the particular item) explaining the item's function.

Help → **Report Bug...** Opens the Bug report dialog where you can report a bug or request a 'wishlist' feature.

Help → **About this application** This will display version and author information.

Help → **About KDE** This displays the KDE version and other basic information.

3.2.4 Horizontal Bars

By selecting the Horizontal Bars menu entry, KTimemon switches from vertical bars to horizontal bars and vice versa. Not very useful, but it was easy to implement ;-)

3.2.5 Quit

The Quit menu item - surprise, surprise -- is used to terminate KTimemon. It will save the current state (e.g. the color scheme, window size, whether it is displayed in the panel) and restore the state in the next invocation.

The configuration information is saved in the file `$HOME/.kde/share/config/ktimemonrc`, where `$HOME` refers to the user's home folder. If this file is deleted, KTimemon will start in its default state in the next invocation.

3.3 Configuration

KTimemon can be configured via a straight-forward dialog (see also the discussion of the [Configuration Menu](#)). On the General page, the sample interval can be specified as well as scaling information (see also the discussion of the [extended mode](#)). If the Autoscaling check box is ticked (autoscaling is explained in the [FAQ](#) section), the scaling factors cannot be edited, since KTimemon determines them automatically.

The Colors page can be used to tailor the colors of the bar graph to individual preferences. A small sample bar graph gives immediate feedback.

In the Interaction page, mouse bindings can be adapted. Clicking a mouse button on the KTimemon window can be ignored, trigger a mode switch (see also [Modes](#)), invoke the context menu (see also [Menu](#)), or invoke an external process. The command line specified for external processes is interpreted by the standard shell, i.e. shell commands, environment variables, redirection etc. can be used.

The Interaction page also contains a check box which can be used to disable to automatic appearance of tool-tips with numeric information about the bar graphs (compare [Onscreen Fundamentals](#)).

3.4 Common Questions and Answers

1. *Which operating systems does KTimemon support?*

KTimemon supports Linux® based systems with the `/proc` file system, Solaris™ based systems with the `kstat` library, and Digital UNIX® (formerly DEC/OSF1) systems with the `table(2)` system call interface. Only the Linux® version has been thoroughly tested, if you experience any problems with the Solaris™/Digital UNIX® port, please do not hesitate to contact me. Also, contributions to KTimemon to adapt it to other platforms are most welcome. Please contact me at m.maierhofer@tees.ac.uk if you intend to port KTimemon to other flavors of UNIX®.

2. *How does autoscaling work?*

Glad you asked. Since there is no sensible predetermined scaling factor for paging/swapping operations and context switches (unlike e.g. memory utilization, where you can take the total memory size as baseline), KTimemon uses a semi-intelligent (well, ...) autoscaling mechanism. Autoscaling works as follows:

- Each of the three bar graphs as described in the [extended mode section](#) has an associated scaling factor. The initial values of these factors are set to some predetermined value.
 - Each time a new sample is displayed, the respective value is tentatively scaled with the corresponding factor. If the value can be displayed in the scale chosen by the factor, no change occurs (i.e. small changes in the activity are reflected by a changing height of the bar).
 - If the scaled value would be either too large or too small to be displayed with the current scaling factor, the scaling is adjusted so that the new value displayed is roughly halfway up the bar graph. Thus, subsequent changes should have a good chance of getting displayed relative to the current value, without having to change the scale again.
3. *Why does a message box with diagnostic output from child command pop up?*

If you bind a mouse button to an external command as described in the [Configuration](#) chapter, KTimemon does not check for a valid command name. Instead a command shell is invoked to execute the statement, so shell commands, environment variables and more can be used. To allow some feedback to the user, KTimemon monitors the `stderr` output of the command shell, and reports it in this message box. While this scheme can be helpful in case a command is not found, it can be quite annoying if the invoked command prints harmless diagnostic information on `stderr`. A simple and elegant solution to this problem is to add `2>/dev/null` at the end of the command specification. This redirects diagnostic messages to message nirvana, and stops the message box popping up.

3.5 Thanks and Acknowledgments

KTimemon is based on an Xt version by my brother.

Thanks to Tobe Toben, ttoben@artis.uni-oldenburg.de, Cristian Tibirna ctibirna@gch.ulaval.ca, Dirk A. Mueller dmuell@rhrk.uni-kl.de, Mark Krischer krischem@amp.com, and Lubos Lunak l.lunak@sh.cvut.cz for bug reports, patches, comments, suggestions.

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

Chapter 4

Credits and License

Copyrights for each applet is listed in the applicable chapter.

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](#).

Appendix A

Installation

To be written

Appendix B

Index

C
color, 1

K
KDE, 1
ktimemon, 1

S
selector, 1
system monitor, 1

T
timemon, 1