

## KDE Tricks Revealed by KDE Developers

# Hidden but sooo useful!

Even KDE developers get a bad conscience when it comes to documentation: No one would deny it's there but its quality and quantity varies. And to be honest: Even if it's there – who reads a manual as long as he or she can get away without? Unfortunately a lot of interesting features remains hidden this way. A good reason to question KDE developers about their favourite not-so-well-known features.

## General Desktop-usage

Konqueror and KWord maintainer *David Faure* and *Sarang Lakare* who claims fame for the desktop “reading-glass” *KMagnifier*, and currently works on *Kile*, a LaTeX editor for KDE, share ideas for clever mouse-usage on the desktop. Asks David: “Did you ever curse a window because its titlebar went off-screen, and you did not know how to move it back? Simply press the [Alt] key, and use the left mouse-button to click anywhere in the window. Now you can move the window with the mouse. Similarly, one can resize a window by pressing [Alt] and the right mouse-button.”

Sarang has a useful hint for all wheel-mouse users who make extensive use of virtual desktops, and draws our attention to the small pager-applet embedded into Kicker, the KDE panel: “Certainly a mouse-click on the field representing the virtual desktop of your choice takes you there. But why not switch by simply moving the mouse-cursor over the pager, and scrolling the wheel of the mouse?”

This is simpler than clicking because to click, you need to position the mouse over the exact virtual desktop entry in the pager, whereas using the mouse-wheel the cursor can be anywhere on the pager. In addition, by simply scrolling the mouse-wheel, you can take a quick look at the applications running in each of the virtual desktops. The downside: You need KDE version 3.1 or higher.”

To those preferring the keyboard to the mouse, *Lubos Lundák*, the maintainer of KDE's window manager *KWin*, suggests

As long as you don't know them, you probably won't miss them – but once you learn about them, it's like an eye-opener: The KDE desktop has many a useful feature that remains secret to the eyes of the users.

BY PATRICIA JUNG



defining keyboard-shortcuts which execute often used commands. “These shortcuts are assigned in *KMenuEdit*, which can be launched e. g. by right-clicking the *K-Menu* button, and selecting *Menu Editor* from the popup menu.”

## Shortcuts

“As soon as the menu editor appears, it's only a matter of selecting the menu entry of the application to start by shortcut or creating a new item. Now change the shortcut key using the key-icon in the down-right corner of the dialog (highlighted in Figure 1), and hit the *Apply* button.”

Thanks to Aaron J. Seigo, in KDE 3.2 users will find this functionality where they may actually look for it: in the *Keyboard Shortcuts* module of the Control Center. If you can't find the relevant

*kcontrol* dialog, use the Control Center's *Search* tab to search after *shortcut*.

A totally different kind of shortcut KDE e.V. board member *Eva Bruchers-eifer* has in mind when she suggests: “Did you know the KDE file dialog may be configured to contain shortcuts for everything that can be written as an URL? This way, a simple mouse-click leads you to an often used local directory. But that's not all: Using the powerful *ioslaves* architecture, the shortcuts can point to Windows or Samba shares, *sftp*-connections or audio CDs.”

These shortcuts are nothing but icons in the file dialog's sidebar as shown in Figure 2a and 2b. “To configure them simply choose the *Add entry...* item in the menu that pops up when you right-click the sidebar with your mouse. Add a description and the relevant URL, and for new directories, don't forget to

uncheck the *Only show when using this application* checkbox, in order to make your new URL available in all KDE file dialogs.”

You wonder how URLs of remote directories might look like? For directories on machines that you might access via *ssh* KWinTV-maintainer and KDE backend-allrounder *George Staikos* has the answer: “Try opening a file and typing in:

```
fish://user@somehost/some/path/
```

You will be prompted for a password and then you can browse the remote site as though it were local (Figure 3).” Adds *Antonio Larrosa Jiménez*, well-known for lots of bug-fixes, feature implementations and tutorials all around KDE: “Of course, this also works in Konqueror’s location bar. And if you add your local public SSH key to the `~/.ssh/authorized_keys` file of the remote site (see [1]), Konqueror won’t even ask for the password, and you get a completely transparent file management.”

## Scripting KDE

Even more advanced desktop usage we learn from George: “My favorite ‘power user’ feature is *kdcop*, an application that lets users graphically browse the DCOP bus (see

Figure 4). It can be used to send messages (using the “Desktop Communication Protocol”) to applications, thus triggering all kinds of interesting things, like manipulate window geometry, save files, etc.

For instance, if you want to restart the KDE-panel Kicker for some reason (say you’re developing a Kicker applet or extension), you just click on the plus-sign in front of the *kicker* entry and do the same with the *Panel* sub-entry. Then click *void restart()*, and it will restart Kicker!

Want to disable the screensaver temporarily? Click *kdesktop / KScreensaver Iface / bool enable(bool e)* and leave the value unselected (see Figure 4)!”

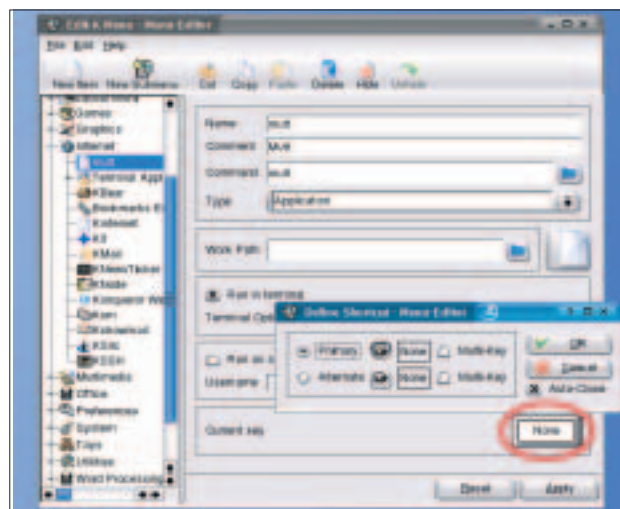


Figure 1: To assign the [Alt-M] shortcut to the e-mail program *mutt*, simply press [Alt-M] in this dialog. Note the activated *Run in terminal* option

“Want to manage your cookies? Browse through *kded / kcookiejar* and you’ll see a great interface with which you can write a shell script such as”:

```
#!/bin/sh
echo "Deleting cookies from $1:"
dcopt kded kcookiejar &
"findCookies(QString)" &
"http://$1"
dcopt kded kcookiejar &
deleteCookiesFromDomain "$1"
```

When started with:

```
sh cookie_script pravda.ru
```

it finds and deletes all cookies set by the *pravda.ru* server.

## Power Users

Typing *gg:* in the location bar of the Konqueror and getting the appropriate Google results has probably become one of the most famous KDE features.

KDE core developer *Daniel Molkentin* explains how to define your own web-shortcuts: “Simply choose *Settings / Configure Konqueror...* and select *Web Shortcuts* (in older versions *Enhanced Browsing*). Using the *Add...* button you can add new entries. If unsure, clicking the questionmark icon in the titlebar and using the ?-shaped cursor to

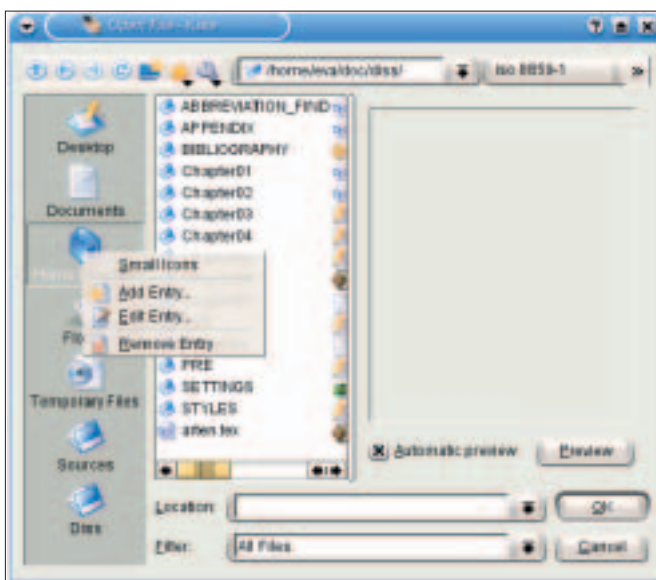


Figure 2a: Expand your file dialog with a new shortcut for the sidebar (to the left)!

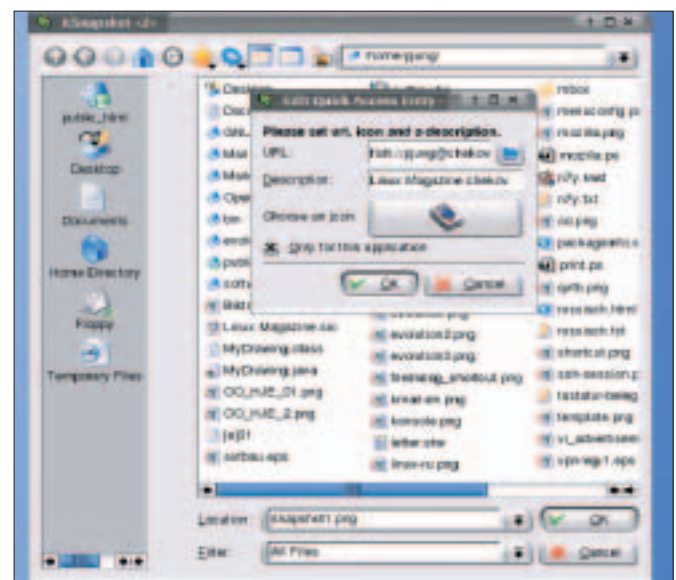


Figure 2b: The tickbox decides whether the new shortcut will be available in all KDE file dialogs or just in the ones belonging to this application

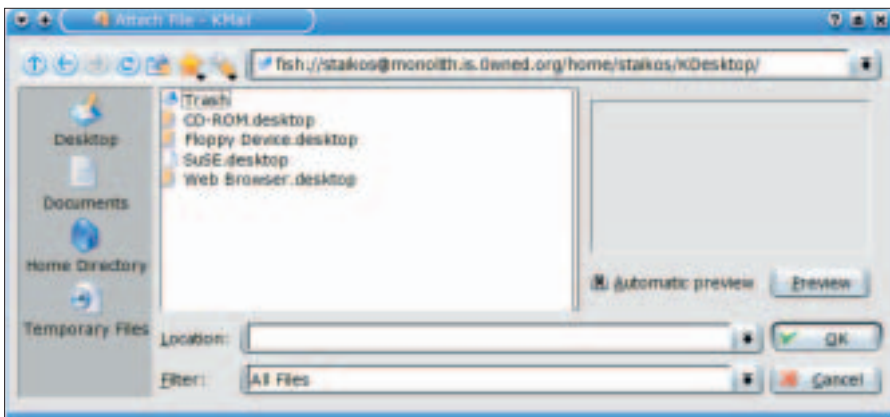


Figure 3: Handling remote files as if they were local



Figure 4: kdcop lets you manipulate running KDE programs

click the field description in question provides information on how to proceed. You can add shortcuts for virtually every webpage that provides search functionality using the HTTP *GET* command. Recently I added one (*wp:keyword*) for Wikipedia which will ship with KDE 3.2.”

Starting with the search results one often wishes to browse presented links without loosing the result page. *Charles Samuels*, the developer of KDE’s Media Player, Noatun, has a solution for this problem: “First right click on Konqueror’s status bar (or click the *Window* menu) and select *Split View Left/Right* from this menu. The window now has split in half. By default, each of the two frames is independent, however, one can select a “controlling frame” (by clicking on it, or just on the status bar) and then use the same context menu to lock this side to its current location. Then use the mysterious square checkbox on the statusbar to link the two frames (see the highlighted parts in Figure 5).

Now, when you select a link on the first frame, only the second frame’s location changes. It’s of course not limited to the web, you can also use this approach to split views of Konqueror when browsing local directories, and even have one side showing any embedded KPart like Kate, or KWord.”

### Enhance Your Calendar!

Ever wished your calendar showed not only local holidays, but also space shuttle launches, celebrity birthdays and world championships? KOrganizer-maintainer *Cornelius Schumacher* suggests visiting websites like <http://www.icalshare.com/> using Konqueror and subscribe to the calendar data you wish. You will be asked whether you would like to open the relevant *webcal* URL using KOrganizer (see Figure 7).

“Although these calendar data are intended for use with Apple’s calendar program iCal, KOrganizer, since KDE version 3.1, can make use of them, too. Once KOrganizer has opened the chosen calendar, one can save these data locally and merge them with one’s personal schedule using *File / Merge Calendar*. Unlike iCal, KOrganizer at present still does not keep these data up-to-date automatically, but this will change in KDE 3.2.”

*Lisiane Sztoltz Teixeira*, who coordinates the Brazilian Portuguese translation effort, shares the solution of a totally different problem with us: “I always have problems with cryptography plugins (namely *openpgp.so*) in KMail. If KMail gives you the error-message shown in Figure 8 when you try to sign an e-mail, it in fact wants to tell you the *gpg-agent* is not running. To solve the problem, extend your X start-up file (usually *~/.xinitrc* or *~/.xsession*) the following way:

```
eval $(gpg-agent --daemon)
startkde
killall gpg-agent
```

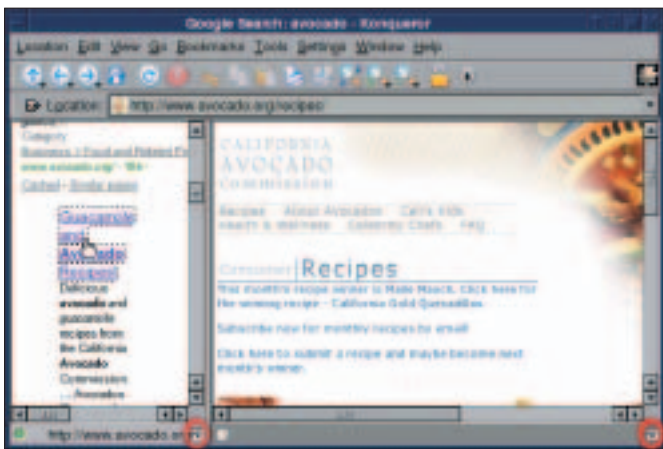


Figure 5: Browsing Google search results is easy when the right frame shows the page belonging to the followed link on the left hand side

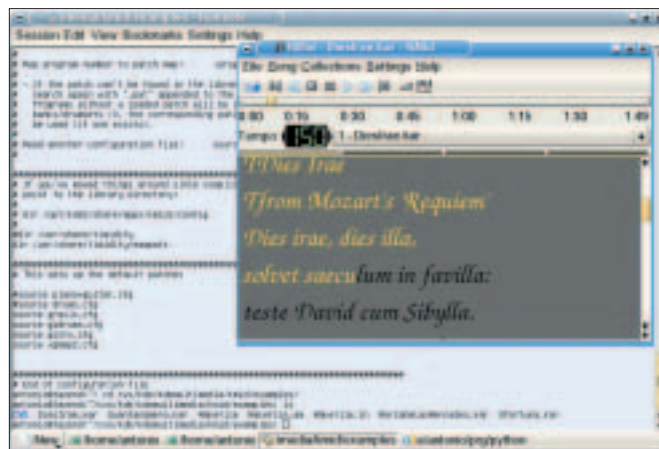


Figure 6: KMID playing “Dies Irae” from Mozart’s Requiem which should be installed on your system in *\$KDEDIR/share/apps/kmid/*

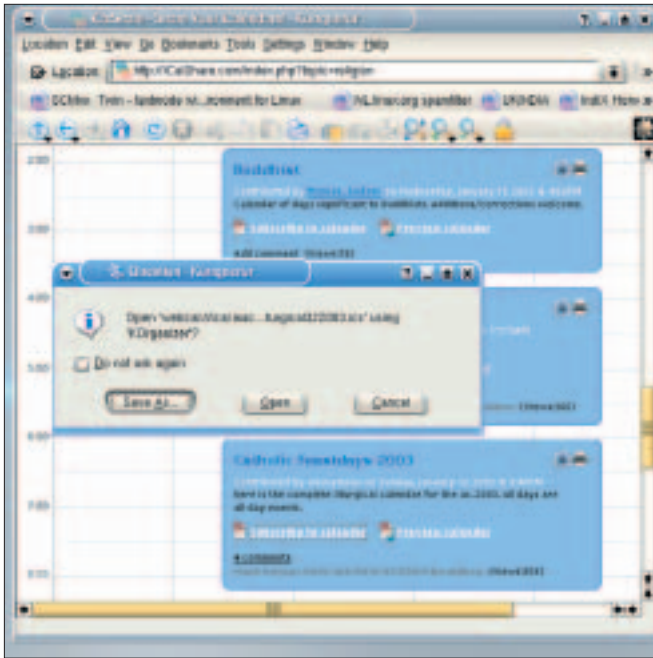


Figure 7: Apple users are not the only ones who can make use of .ics calendar files

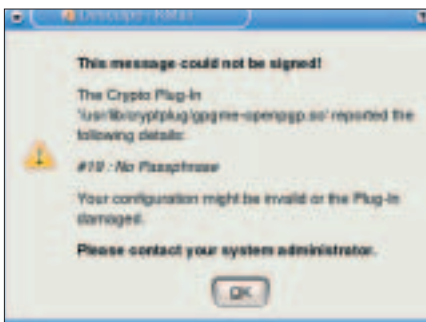


Figure 8: Not actually a descriptive error-message but Lisiane has a solution for the problem

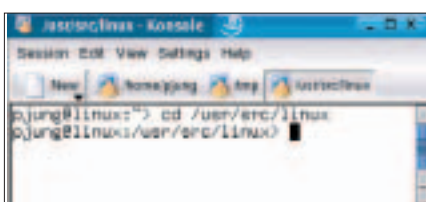


Figure 9: When PROMPT\_COMMAND is set accordingly, the Konsole session name changes with the working directory

The first line configures *gpg-agent* to be run in the background. After this, *startkde* initializes KDE, and when KDE is finished, *killall* kills all GnuPG agent processes.” In addition, make sure, the GnuPG configuration-file `~/.gnupg/gpg.conf` does not have a comment-sign # in front of the *use-agent* line.

If you belong to the crowd pestering Antonio (who also maintains the KDE midi player *KMid*) why his program

can't be used to hear karaoke music, here is his answer: “At present *KMid* basically works only with an external synthesizer connected to the midi port of the sound card, or with cards that have an internal midi synth (unfortunately, fewer and fewer cards support this, I suppose in order to decrease prices). But even with other cards, not all hope is lost: The trick is to make *Timidity++* work as an also midi

synthesizer. Start *Timidity++* using the *timidity -iA* command. With some sound cards – in case you hear some pops and beeps – you need to add the *-s 48000* flag and set the sampling rate to 48000 in *konsole's aRTs* configuration dialog.

When you run *kmid* later, you'll find a new *ALSA device* in the *MIDI Setup...* dialog. Then you can play a midi file, and *timidity* will emulate the synthesizer, while you can read the lyrics with *KMid* (Figure 6).

Note also that you need a good set of *Timidity* patches to make this work nicely. My preferred one is *EAWPats*, which you can install following the instructions at <http://www.onicos.com/staff/iz/timidity/doc/eawpat.html>.”

## The Benefits of the KDE Commandline

Let's finish this kaleidoscope of useful KDE hints with Antonio's *Konsole* tricks. It's probably known that it is possible to change the button name of the active *Konsole* session (by default reading *Shell*, *Shell No. 2* etc.) in the *Konsole* window using [Ctrl-Alt-S]. But did you ever think of how nice it would be if the *Konsole* tabs showed something useful like the current directory automatically?

In fact you can tell *Konsole* what to display in the tab button by setting the shell variable *PROMPT\_COMMAND*. It

then shows the output of this command in the tab button. To achieve this add the following to `~/.bashrc`:

```
if test "$DISPLAY" ; then
  export PROMPT_COMMAND='echo `
  -ne "\033]30;`echo $PWD | sed `
  -e "s/^\.*\(\.\{20\}\)\$/\1/"`
  ` \007\033]31;$PWD\007``'
fi
```

Each session started after this addition will show the working directory in the tab button (see Figure 9), stripped to the last 20 characters (otherwise the tab bar suffers ultra-long buttons).

If you're working on remote machines a lot it's a good idea to not lose control over which *Konsole* session belongs to which machine. To help this, *Konsole* allows to configure new session types using *Settings / Configure Konsole... / Session*.

In this dialog you can add a session type that, instead of running *bash*, runs *ssh -C user@anotherhost* (*-C* compresses the connection's data stream), change the associated icon to something you recognize easily, and – if you wish to – the background schema (I find it helpful to tint the shell background for different hosts distinguishable). ■

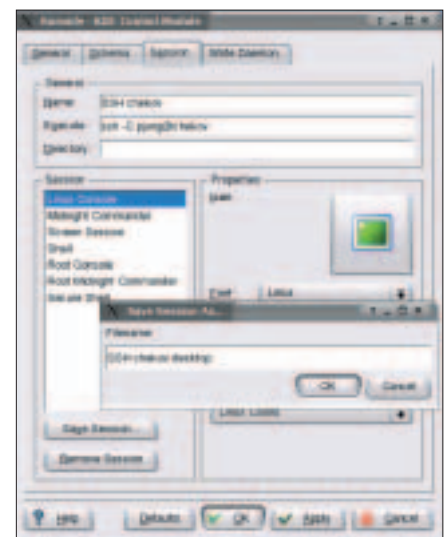


Figure 10: A session of this new type leads you directly to the SSH password-prompt

## INFO

- [1] Heike Jurzik: “Secure Transmissions”, *Linux Magazine* Issue 33, August 2003, p82