

Keyboard shortcuts for KDE, Mozilla and Opera

Murophobia (Fear of Mice)

Imagine how quickly you would be able to operate your computer without those pesky mouse clicks. There is no need to revert to the command line to avoid using your mouse. With the right programming, you can quickly access dialog boxes and menus using keyboard shortcuts.

BY HAGEN HÖPFNER

Although this may come as a surprise to some people, running a GUI does not automatically rule out keyboard commands. On the contrary, assuming that the developers have done their homework, you can completely forget your mouse most of the time. If you find that hard to believe, try looking at a professional designer, or any other creative wiz, working with a Mac; you would be surprised how much you can do without touching the mouse.

The magic words are keyboard shortcuts. Unfortunately, you will need to learn them first – and don't expect the shortcuts you use at home to work on your office desktop. This is an issue that developers need to address by consistent implementation of cross-project style guidelines.

Having said that, it is well worth a Linux user's while to put the mouse



Sabine Simon

aside, and learn how to use KDE, Mozilla, Opera and other GUI-based programs. With some practice, you will definitely note the extra speed you can eke out of typical operations.

KDE Mouseless

There are two good reasons for KDE being so supportive to murophobics: for one thing, the underlying Qt library has always supported keyboard shortcuts. For another, there is definite peer pressure towards integrating core applications, and this becomes obvious when you look at the keyboard shortcuts. Learn once, use many is a good thing that applies to all core KDE applications.

Distributors often throw a spanner in the works. Radical changes to the GUI desktop cause keyboard incompatibilities, among other problems. This article assumes Suse 8.2 with KDE version 3.1.3, and describes the original KDE

keyboard assignments. You can check the KDE Control Center to find out which keyboard mapping your system uses. To launch the Control Center, press [Alt-F2] to pop the KDE Run Command window up on your screen (see Figure 1), type `kcontrol` and then hit [Enter] to confirm. Incidentally, you can use this method to launch non-KDE-based GUI applications.

After you type the command, the original icon is replaced by the icon for the application. If this does not happen, you may have you mistyped the program name, the software may not be installed, or may not be in your search **path**.

The Control Center item we need to look at is called *Regional and Accessibility*, and you can use the keyboard to get

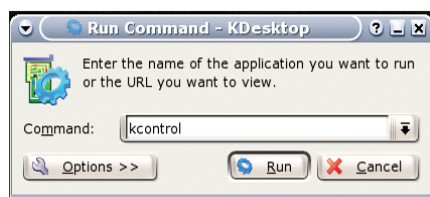


Figure 1: You can use KDE's Run Command window to launch commands that do not generate output.

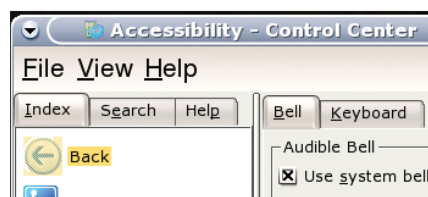


Figure 2: Use the [Tab] key to navigate the input elements.

THE AUTHOR

Hagen Höpfner is a member of technical staff at the Institute of Technical and Entrepreneurial Information Systems of the Faculty of Computer Science, at the University of Magdeburg, Germany. In his leisure time he is a loving father and plays guitar in a rock band called Gute Frage (Engl. "Good Question", <http://www.gutefrage.de>).



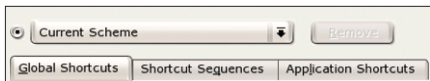


Figure 3: You can use this dialog box to select a previously defined keyboard scheme.

there. Inside a dialog box, the [Tab] key moves to the next input element. You can toggle between the *Index* tab and the tree view by pressing the [Tab] key. The box that surrounds the first point in the list of modules indicates that the focus has shifted to the tree view (see Figure 2). Pressing [Alt] and the underlined letter will take you to the other tabs, *Search* and *Help*, as well as to the menus.

You can now use the arrow keys to navigate the list of modules. Move down to the *Regional and Accessibility* item and press [Enter] to expand the item. Now move to *Keyboard shortcuts*, and press [Enter] again, to open a Control Center module that allows you to edit shortcut schemes.

The [Tab] key now allows you to navigate the elements in the right panel. You can access the tabs in this panel, *Global Shortcuts*, *Shortcut Sequences*, and *Application Shortcuts* (see Figure 3) using the underlined letters, [Alt-G], [Alt-Q], and [Alt-L] respectively.

When the [Tab] key reaches the *Current scheme* scroll list, the list is highlighted (see Figure 3), and the arrow keys allow you to scroll through the list of keyboard schemes altering the contents of all three keyboard shortcut tabs! The various schemes will emulate the keyboard assignments of Windows or MacOS, and even provide a classical Unix variant where [Ctrl + Alt + Del] will terminate KDE.

If you restrict yourself to KDE core applications, there is no reason why you should not use one

of the pre-defined keyboard schemes. However, non-KDE applications like XEmacs, Mozilla, Opera, or Vi will not respect the keyboard shortcuts you define, which in turn can lead to cross-application conflicts. For example, the *KDE Default for three modifier keys* uses [Alt-F5] to display the window list, whereas Opera uses this to reload the current frame.

To remove ambiguous shortcuts from KDE, recall that not many applications will map the Windows key. Thus, it makes sense to use the Windows key as a **Modifier** to avoid conflicting mappings between KDE and your applications.

One Key, One Program

Just like you can quit KDE using a single keyboard shortcut, it should be possible to bind applications to a shortcut; one could use [Win-T], for example, to launch the KDE *Konsole* program, which opens up a Shell. To do so, first launch the KDE Menu Editor, *kmenuedit*, by pressing [Alt-F2] to pop up the Run command window. Use [Tab] and the down arrow to navigate to the point where you

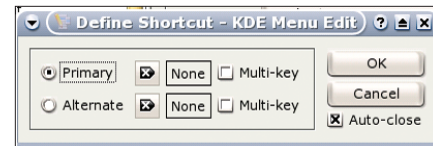


Figure 5: Defining a shortcut key in the Menu Editor.

can edit the *System* item in the *K* menu. Now press the right arrow key to expand the node, and look for the *Konsole* entry below it (see Figure 4).

Press [Tab] to navigate to the stylized key where the caption currently says *None* at the bottom right in this window, and press the [Space] key to assign a shortcut key. This pops up the window shown in Figure 5. Now hold down the left Windows key and press [T], and then press [Enter] to confirm this change in the Menu Editor. Pressing [Win-T] will now launch *Konsole*.

But you should be aware – and I am sure that most users of KDE 3.1.3 are – that totally consistent keyboard assignments are often hard to achieve.

To define shortcut keys for applications that do not currently exist in the *K* menu, first select the position where you want to insert an entry in the left panel. Then use the *File* menu ([Alt-F]) to add a new item. Incidentally, you can use the [Esc] key to close a menu that you opened by mistake (and to close message boxes).

Ten Fingers

After assigning a keyboard shortcut to launch Mozilla, you might like to add a few keyboard shortcuts for Internet functions. We will use Mozilla Version 1.4 in this example, and sincerely hope that the developers will support the functions discussed here in later versions, and particularly in the new Firebird browser.

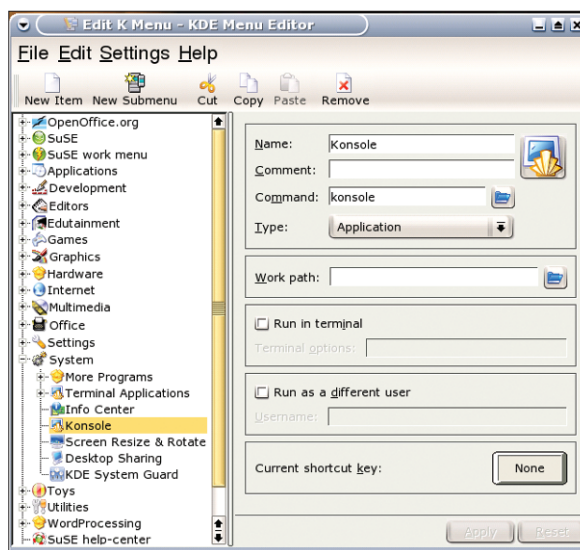


Figure 4: Use the KDE Menu Editor to define keyboard shortcuts that call individual programs.

GLOSSARY

Qt: is a toolkit that provides developers with the functions required to draw buttons, scroll-bars, menus, and other GUI elements. Besides KDE, many programs including Opera, for example, use Qt.

Path: The PATH environment variable specifies the subdirectories the Shell should search to find executables. You can enter "echo \$PATH"

in the command line to output the current value of the variable; individual directories are separated by a colon.

[Ctrl+Alt+Del] This keyboard shortcut was originally programmed to solve the problem of speeding up the reboot sequence of the IBM System/23, but is better known to most people as the "three fingered salute" used to

reboot PCs back in the bad old days of DOS. The developer actually chose this sequence on the lines that it is highly unlikely that a user would press these three keys by mistake. Latching on to this tradition, Linux can use Ctrl+Alt+Del to reboot.

Modifier: A key that modifies the affect of another key.

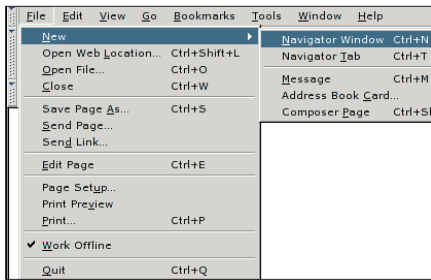


Figure 6: Menus often indicate the available accelerators.

Just like on KDE, underlined letters indicate the keyboard shortcut that will launch a specific function in the Mozilla menus, for example, [Alt-F] opens the *File* menu (see Figure 6). Having opened up a menu, you can then either use the arrow keys, and press [Enter], or type the underlined letter (without holding down the [Alt] key this time) to select a menu item.

But there are a number of operations that have individual keyboard shortcuts, so-called accelerators, assigned to them. For example, [Ctrl-Shift-L] pops up the Open Web Location window.

You can type the required URL in this dialog box (e.g. <http://www.linux-magazine.com>), and confirm by pressing [Enter]. You can then search the Web site that you accessed in this way just as if you were using a command line program such as *less* or *vi* (this is a special feature of newer Mozilla versions). The slash / (press [Shift-7] to type it) toggles to type ahead find mode, allowing you to simply type your search string.

If your search takes you to a link for another Web site, you can follow the link by pressing [Enter]. To restrict the search to links only, simply type the search string without toggling to type ahead find mode.

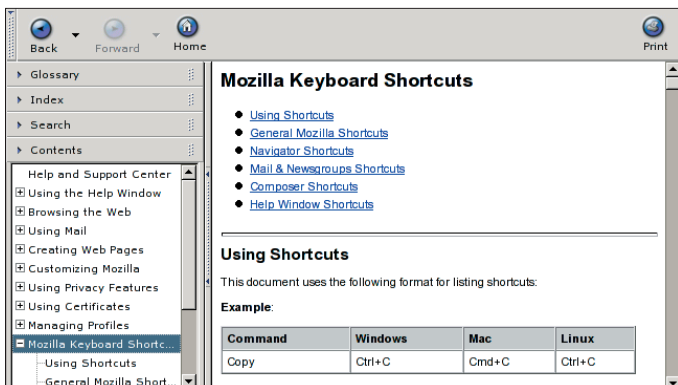


Figure 7: Mozilla provides neatly organized tables of keyboard shortcuts.

When you press [Enter] to open a link, the effect is the same as left-clicking the link. The target address is opened up as defined by the author of the Web page: within the current frame or window, or in a new window. [Shift-Enter] pops up the Save as dialog box allowing you to save the document the link points to. [Ctrl-Enter] opens the page in a new window.

Alternatively, you can press [Ins] and [Shift-Ins] to open a linked page in a new tab. [Ins] additionally moves the tab to the foreground, whereas [Shift-Ins] retains the current page. You can then press [Ctrl-PgDn] [Ctrl-PgUp] to toggle between tabs.

Pressing [Alt-Left Arrow] takes you back to the last site you viewed, and [Alt-Right Arrow] takes you forward. You can press [Ctrl-W] to close a single tab, or a single window, whereas [Ctrl-Q] will quit Mozilla and send all your Mozilla windows packing.

KDE allows you to toggle between multiple browser windows. To navigate the windows on your current virtual desktop, hold down the [Alt] key, and press [Tab]. You can hold down the [Ctrl] key and press [Tab] to switch desktops.

The Online help ([Alt-H][Enter]) provides a description of the *Mozilla Keyboard Shortcuts* (see Figure 7).

The Phantom of the Opera

Users who are prepared to pay for Opera's surfing speed, will definitely not want the mouse slowing them down.

Of course, the following keyboard shortcuts work just as

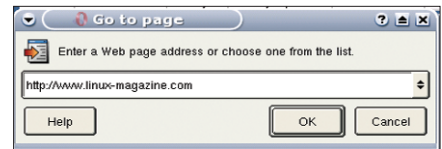


Figure 8: Opera's URL entry dialog box.

well in the banner ad sponsored, free version of the browser. [F2] opens a dialog box where you can type a URL (see Figure 8); again, pressing [Enter] will open the required page in your browser.

[Ctrl-N] opens a new tab, [Ctrl-W] closes the active tab, and [Ctrl-Q] quits the browser. Opera unfortunately restricts the type ahead find function, which is again activated by pressing /, to the links on the current page. [Enter] opens the link in the current tab; [Shift-Enter] opens a new tab with the link, and moves the tab to the foreground. And you can press [Shift-Ctrl-Enter] to keep the new tab in the background.

To search the whole text, rather than just the links, press [Ctrl-F] to open up a search window. Incidentally, the same shortcut will open up a similar dialog box in Mozilla.

Opera allows you to define your own keyboard. As might be expected, you need to open the Preferences to do so. Press [Alt-P] and use the arrow keys to navigate the list in the left panel; Opera automatically opens the nodes that you navigate to. The keyboard mappings are located below *General* in the *Accessibility* section (see Figure 9).

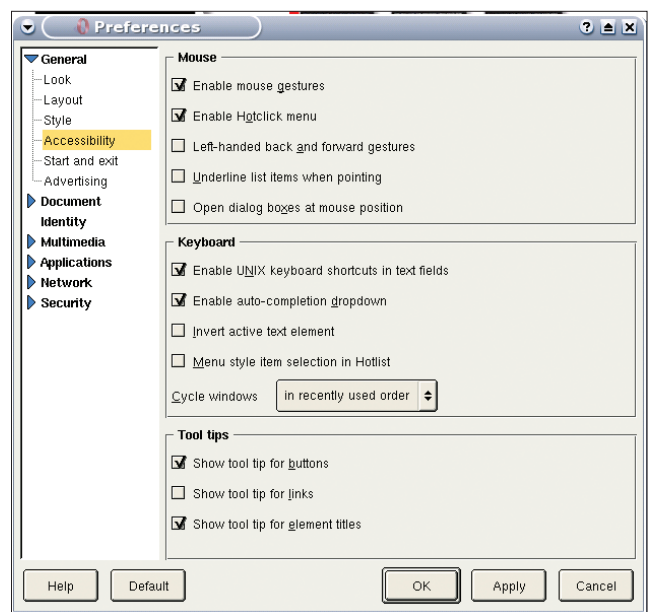


Figure 9: Use this dialog box to define own keyboard shortcuts for Opera.