

# Who Says Penguins Can't Fly?

## Mouse shortcuts with xbindkeys

October 13, 2009 — Hans

**Note:** This isn't directly related to KDE. However, since there isn't such a tool for KDE at the moment, I figured some KDE users might find it interesting as well.

I recently bought a new mouse for my desktop computer, more specifically a Logitech Corded Mouse M500. Up until then I had been using an old mouse with two buttons and a scroll wheel. Now I suddenly have two extra mouse buttons, and a scroll wheel that you can tilt to the left and right. Let's see how to take advantage of this.



The freewheel scroll feature is pretty nice as well

## Bind keys with xbindkeys

As the name suggests, xbindkeys is a small program that lets you bind hotkeys to actions under X. Although it doesn't include a GUI, it's very easy to use. **There is a graphical configuration tool called xbindkeys-config** (GTK frontend), **but in this guide I'll edit the config file directly.**

1. First of all you need to install xbindkeys, preferably through your package manager.
2. You can create a default configuration file with the command

```
xbindkeys --defaults > $HOME/.xbindkeysrc
```

3. Open ~/.xbindkeysrc in your favorite text editor. (If you can't find it, make

sure you have “Show hidden files” enabled).

[This guide aims to be quite comprehensive – if you don’t want to read everything, you can look at the default configuration file and try to figure out the rest yourself. It’s not difficult if you read the included comments]

4. Let’s check the syntax. Everything after # are comments, which means that the program will ignore them.

You can find the following text in the default file:

```
# The format of a command line is:
#   "command to start"
#       associated key
```

Below are also some examples – great. It’s probably a good idea to include a short comment (start the line with #) before each command, for example

```
# short comment
#   "command to start"
#       associated key
```

5. "command to start" is simply a shell command (that you can run from a terminal). I’ll talk more about it under *Actions*.
6. associated key is the hotkey. Let’s take a closer look at how to find what we should write here. But before that, don’t forget the last step.
7. Start xbindkeys by running the command in a terminal.

## Keyboard shortcuts

This section deals with keyboard shortcuts. If you’re only interested in mouse shortcuts, please see the next section *Mouse shortcuts*.

The first example in the generated .xbindkeysrc shows how you can define a hotkey:

```
control+shift + q
```

A list of modifiers is included in the comments.

```
# List of modifier:
#   Release, Control, Shift, Mod1 (Alt), Mod2 (NumLock),
#   Mod3 (CapsLock), Mod4, Mod5 (Scroll).
#
# The release modifier is not a standard X modifier, but you can
# use it if you want to catch release events instead of press events
#
# By defaults, xbindkeys does not pay attention with the modifiers
# NumLock, CapsLock and ScrollLock.
```

Pretty straightforward, don’t you think? You can also use raw keycodes, as shown in the next example:

Now this might look cryptic, but it's really no problem – you can easily find the keycodes with the `--key` option. Run

```
xbindkeys -k
```

in a terminal and a window pops up. Press the desired hotkey and you should get an output similar to this:

```
Press combination of keys or/and click under the window.  
You can use one of the two lines after "NoCommand"  
in $HOME/.xbindkeysrc to bind a key.  
"(Scheme function)"  
m:0x15 + c:34  
Control+Shift+Mod2 + q
```

In this case I pressed `Ctrl+Shift+Q`. `Mod2` corresponds to Numlock which is ignored by default, as explained in the earlier text about modifiers. Apparently the keycodes for this hotkey are `m:0x15 + c:34`.

## Mouse shortcuts

Now that we know how to deal with keyboard hotkeys, mouse shortcuts should be a piece of cake. The first mouse button (left) is called `b:1`, the second (right) `b:2` and so forth. Take a look at the third example and you'll see how a mouse shortcut can be defined:

```
control + b:2
```

That's right, you can combine mouse and keyboard shortcuts – great for those who don't have a fancy mouse with extra buttons. I used to bind `Super+` mouse buttons to various actions with my old mouse (`Super` is the Windows key on most keyboards).

To find out what a mouse button is called, you can use `xev`. Similar to `xbindkeys -k` that you used earlier, it pops up a window. However, this one isn't closed automatically, and it also grabs mouse movement and clicks. Try to click/scroll in the window (without too much mouse movement), and you should see something like

```
ButtonPress event, serial 31, synthetic NO, window 0x3a00001,  
root 0x189, subw 0x0, time 55432271, (4,0), root:(1102,842),  
state 0x10, button 9, same_screen YES
```

in the terminal output. According to the last line, I just pressed mouse button 9, which translates to `b:9` in `.xbindkeysrc`.

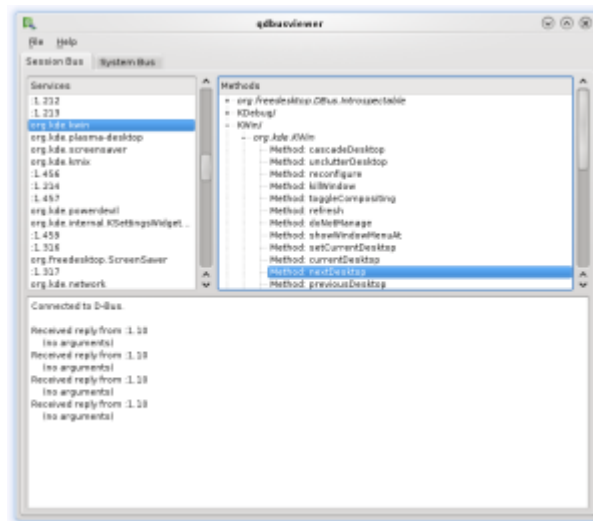
## Actions

command to start is a command that you can run from a terminal, for example an application (dolphin) or a shell script (out of scope for this article).

Let's take a concrete example. I want to bind wheel tilt -> switch to previous/next desktop. To do this I have to communicate with KWin, the default KDE window manager. KDE applications use D-Bus to communicate with each other, and you can "talk" to them with D-Bus as well. (This is for KDE SC4, in the KDE SC3 series DCOP is used instead).

I'm not very familiar with D-Bus myself, but with the help of the qdbusviewer tool and some Googling I found that I can use the following command to switch desktop:

```
qdbus org.kde.kwin /KWin org.kde.KWin.nextDesktop
```



Find the D-Bus call with qdbusviewer

According to xev (explained in *Mouse shortcuts*) the tilt to the left and right corresponds to b:6 and b:7. Therefore, a part of my .xbindkeysrc looks like this:

```
# Previous desktop
"qdbus org.kde.kwin /KWin org.kde.KWin.previousDesktop"
b:6

# Next desktop
"qdbus org.kde.kwin /KWin org.kde.KWin.nextDesktop"
b:7
```

Don't forget the quotation marks (") around the commands!

Another tool you can use to find D-Bus calls is dbus-monitor, as suggested by Balden in the comments. Run it in a terminal and trigger a command the normal way, for example by hitting a keyboard combination, and dbus-monitor will show the call(s).

## Emulate key presses

and “Desktop Grid”. I couldn’t find any D-Bus calls for this, so let’s try to achieve it in another way. The effects are already mapped to some keyboard shortcuts – maybe we can find a command to “emulate” these key presses?

The tool for doing this is called `xte`, and is included in the `xautomation` package. Again, install it with your package manager.

The default shortcut for “Show Desktop Grid” is `Ctrl+F8`. The following command should then activate the effect:

```
xte 'keydown Control_L' 'key F8' 'keyup Control_L'
```

Whether we choose `Control_L` or `Control_R` doesn’t really matter. After confirming that it works, I added the following lines to `.xbindkeysrc`:

```
# Present Windows
"xte 'keydown Control_L' 'key F10' 'keyup Control_L'"
b:8

# Desktop Grid
"xte 'keydown Control_L' 'key F8' 'keyup Control_L'"
b:9
```

If nothing happens for you, make sure you emulate the right key presses. Have you changed any shortcuts?

This trick can also be used to map the back/forward buttons on the mouse. Here we can’t use D-Bus, since we want to “talk” to the active window and not a specific application. Fortunately for us, most applications use the key shortcuts `Alt+Left` and `Alt+Right` for Back and Forward, respectively. In other words, we can do something like this:

```
# Back
"xte 'keydown Alt_L' 'key Left' 'keyup Alt_L'"
b:6

# Forward
"xte 'keydown Alt_L' 'key Right' 'keyup Alt_L'"
b:7
```

This’ll work in Dolphin, Konqueror, Firefox, and basically any application that use the same shortcut for Back/Forward.

`xte` can also be used to simulate mouse clicks, as pointed out by Rahul (see comment for an example).

Want to know more about `xte`? `xte --help` displays some helpful text that you might find useful.

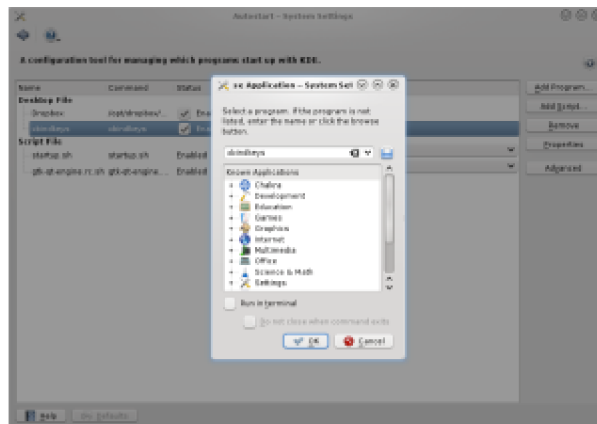
# Apply settings and make xbindkeys autostart

Now we're almost done! Remember that you have to restart xbindkeys to apply the changes:

```
killall xbindkeys && xbindkeys
```

As a final touch, let's make xbindkeys autostart with KDE. It's very easy to do with the new Autostart module.

1. Open System Settings.
2. Go to Advanced tab -> Autostart.
3. Click on "Add Program...".
4. Write xbindkeys and press OK. A new dialog pops up. Press OK again.
5. Done!



Make xbindkeys autostart

## Closing words

In this tutorial we've learned how to use the xbindkeys program to bind keyboard and mouse shortcuts to different commands. We also took a quick look at how to communicate with an application using qdbus and emulate key presses with xte. Finally, we made xbindkeys autostart.

I think it would be great if the "Input Actions" module in System Settings also supported mouse shortcuts. A more simple module for assigning mouse buttons to actions would be nice as well.

Hope that you found this guide helpful, don't hesitate to leave a comment below if there's anything on your mind.

## 113 Responses to “Mouse shortcuts with xbindkeys”

**Ryan** Says:

October 13, 2009 at 19:52

You shouldn't have to edit your `.xbindkeysrc` file at all. There's a GUI program called `xbindkey-config` to do it for you.

**Hans** Says:

October 13, 2009 at 20:02

If you read the blog post, I already mentioned `xbindkeys-config`. However, I'm more of a KISS person, and wanted to show how simple it is to edit `.xbindkeys`.

**atomopawn** Says:

October 13, 2009 at 20:21

Absolutely excellent. Well done, Hans!

**Gianvacca** Says:

October 13, 2009 at 20:32

Cool guide, very useful. Thanks a lot.

**Mickaël Sibelle** Says:

October 13, 2009 at 20:43

Hi!

Thank you, it is very clear.

My girlfriend and I oftenly use the same PC with a x-session opened for each...

One question: what about multi-user/multi-X-session mode?

Will we have to have the same `xbindkey` configuration?

Will we be able to have two configurations (one for each) running at the same time?

Thank you for your KISS approach on text files: GUIs are oftenly a source of problem in configuration processes ; moreover for beginners, while they know how to modify a text file 😊

**bbandi** Says:

October 13, 2009 at 20:47

And what if I want to map a keyboard button to a mouse button (Like `XF86AudioRaiseVolume`)?

**bbandi** Says:

October 13, 2009 at 20:49

Ehh, sorry, I see now, you wrote about it.

**David** Says:

October 13, 2009 at 21:18

Editing a text file isn't easier than running a GUI program. You have the wrong mentality and are making in fact things worse for new users.

**Hans Chen (mogger) 's status on Tuesday, 13-Oct-09 19:23:31 UTC -Identi.ca**

October 13, 2009 at 21:23

[...] to use mouse shortcuts: <https://hanswchen.wordpress.com/2009/10/13/mouse-shortcuts-with-xbindkeys/> [...]

**Hans** Says:

October 13, 2009 at 21:40

@Mickaël Sibelle:

Unfortunately I don't know about multiple X session, but I think it should be fine to have two different configurations.

@David:

Have I claimed that editing a text file is easier? I do think it's easier for *some* users (myself included), but not for everyone.

I also fail to see how this makes it worse for new users. Apparently there are people who appreciate it. For those who prefer GUI, I already pointed out that I would work with text files, and that there is a tool called xbindkeys-config. Maybe I should make this part bold to make it harder to miss?

Edit:

OK changed the text to bold, and also edited the "aimed at beginners" part.

**dave** Says:

October 13, 2009 at 23:00

Well, maybe its simple but not for me.

It's not plug and play and this is too advanced for me. This is was keeping me from Linux (I'm leaning) Now everything is working for me, KDE 4 is like a dream (only renaming is not good, i did that and after that action all my extensions are gone so Windows can't read them anymore 😞 ) but I'm learning... Why is this not plug and play or is there not a resend Gui, delivered whit KDE (all distributions). a lot of people have a mouse whit a back and forward button.

Ps. completely off topic i know, but rotation a image inside dolphin is not possible and if you opened inside the viewer it's ok but can't save it in that state, you have to rotate it to the left and back again than save the file and it's ok, is this a bug?

**Mickaël Sibelle** Says:

October 13, 2009 at 23:32

David, I think you are too "sure" of what you say.

Modifying a file to modify a parameter is easier than using a GUI made to configure the entire application.

It is a fact.

File:

- 1) open the text file;
- 2) modify the line;
- 3) save the text file.

Configuration GUI:

- 1) Find the GUI (it may change whether you use this WM KDE or this one Gnome, etc).
- 2) Find the right tab, pannel, place, etc. as there are often MANY parameters, so they are "organized" :fear:.
- 3) Find the right parameter name (may be translated), IF the GUI guy thought about



creating this parameter GUI element.

4) Set the value.

Ok, it would be easier if the GUI were better, but they are NOT.

Let me also give you an example about GUI and command line: VLC. I tried to use the GUI to diffuse music. It is HORRIBLE. In command line it is Easy.

I know this is a “VLC” problem, BUT it is an real example about “GUI creates limits and complexity”.

I wish GUI were more easy to use because “end-user” people like them. But “end-user” are often lost, GUI and commandlines/text files, so text files remains more efficient (I meant: for “complex” configuration – keys in this case).

**Hans** Says:

October 14, 2009 at 0:05

@dave:

There’s a simple reason why no KCM module for this exists in System Settings – no one has written it yet. But as I wrote, I hope to see it in the near future (may even give it a try myself if I find the time).

“Well, maybe its simple but not for me.”

It is even for you, but not necessary easier. The ‘simple’ I speak of is not the same thing as ‘easy’ (see the KISS principle).

For example, it’s often more simple to install an application from the command line (a) than using a graphical tool (b). Just look at the different descriptions:

(a) Run ‘pacman -Sy xbindkeys’ in a terminal.

(b) Open “Shaman” from the application launcher, click on the search field, enter ‘xbindkeys’, click on xbindkeys in the list, click on “Mark for installation”, click on “Process Queue”.

See what I mean?

“This is was keeping me from Linux”

One important thing in the FOSS world is ‘freedom’. There are distributions and applications aimed at beginners, and also those intended for more “advanced” users. Saying that one way is wrong is, in my opinion, wrong.

I don’t blame people who prefer GUIs, but this is the way I like to do it – and the way I’ll teach others. If you don’t like it, I already pointed out an alternative way.

“Now everything is working for me, KDE 4 is like a dream (only renaming is not good, i did that and after that action all my extensions are gone so Windows can’t read them anymore 😞 “

Glad to hear that you like it. File extensions shouldn’t be a problem, you can just add them again if you can’t open the files in Windows. By default renaming (F2) in e.g. Dolphin only selects the file name.

“Ps. completely off topic i know, but rotation a image inside dolphin is not possible and if you opened inside the viewer it’s ok but can’t save it in that state, you have to rotate it to the left and back again than save the file and it’s ok, is this a bug?”

I don’t quite understand what you mean, but I’m sure you’ll get better support at the KDE Community Forums.

**TheBlackCat** Says:

October 14, 2009 at 1:08

“But as I wrote, I hope to see it in the near future (may even give it a try myself if I find the time).”

I wouldn’t hold your breath, there are wishlist items in bugs.kde.org asking for this going back 8 years.

**Jonas** Says:

October 14, 2009 at 1:09

Like you, I too just switched from a simple 2 buttons + wheel and managed to get a setup like the one described work to my advantage. Back and forth to switch desktop only really but still...

However, I ran into one obstacle I haven’t found a way around yet. According to xev, two of my mouse buttons are considered one and the same (namely clicking on the wheel and one button below it designed by Logitech to flip between windows – at least that’s what the logo on the button suggests...). Both give me the same button id.

Do you know if there’s a way around that?

**Hans** Says:

October 14, 2009 at 1:22

@TheBlackCat:

I won’t hold my thumbs either – they are better used on the space bar. 😊

You never know though, many old wishlists I’ve voted for have been implemented in KDE4.

@Jonas:

Sorry, I have no idea. Which mouse model do you use?

**Andi** Says:

October 14, 2009 at 8:32

Hi Hans.

Thank you very much for this description. I allways missed to use the back and froth buttons of my mouse during browsing. Your really made my day!

Thanks!

**dave** Says:

October 14, 2009 at 8:57

Thanks for all the reply, I know that Linux stand for freedom, thats the main reason I try to switch to Linux. (and I’m getting there 😊 ). I Agree that editing a file can be

easier than using a GUI, But I prefer a still a gui. That's freedom you're talking about I think.

Ps. renaming WITH the original extension works, now Windows accepts the files again, but I know now what went wrong. I did select all my images (200) and rename it to foto1 foto2 etc. But Dolphin removes the extension. If you do this on Vista it selects only the name and leaves the extension alone, so you rename it to "foto1.jpg" and not "foto1" And if you change it to "Foto1" without an extension the explorer gives a warning if you want to do that action.

Well, I will make a bug report on the KDE site for this and the other problem. Again... Thanks for all the replies that's why I want to use Linux p-)

**dave** Says:

October 14, 2009 at 9:05

One small thing, you say that you have to install a gui etc.

I would like to see an entry inside the KDE settings,... a mouse icon and there you can select the type of mouse you have, or you can select an action for all the buttons. That doesn't have to be that advanced.

**Splash** Says:

October 14, 2009 at 9:15

Thanks for this great introduction. I wanted to have something like this for quite some time. I've never recognized xbindkeys.

**Hans** Says:

October 14, 2009 at 10:28

@Everyone:

Thanks for your comments, it makes me happy to hear that you appreciated the guide!

It would be nice to hear how you've mapped your mouse buttons. Personally I find "Present Windows" for the thumb button incredibly useful when I use the mouse (I prefer the keyboard for most tasks).

@dave:

"I would like to see an entry inside the KDE settings"

That's what a KCM module is. 😊

**Jonas** Says:

October 17, 2009 at 2:59

I was afraid so...that you wouldn't know I mean. I'm using a Logitech mx518, and it works great except for that button (well, and two buttons marked with + and - respectively that are supposed to increase/decrease the dpi - I haven't bothered with those since they're located in such awkward positions I wouldn't use them even if they worked).

I bought the mouse for two reasons: it fits my hand great which is a rare thing when it comes to my hands, and its resolution makes precise work much easier. The extra buttons are a bonus really, so not a big deal it's proving impossible (so far at least) to get that button to work. It's just one of those "I'm gonna get it to work just"

Anyhow, it seems as if I either need to provide a custom xorg.conf file or a new HAL-rules file for all buttons to work, and as yet I haven't had time to look into how that works except that it probably has to do with x.org's ButtonMapping option.

**Hans** Says:

October 17, 2009 at 12:13

@Jonas:

Do you use evdev? Which distro?

Here everything worked by just plugging it in, which made me pretty impressed. I still remember the time when you had to add "ZAxisMapping" "4 5" to your xorg.conf to get the scroll wheel to work.

**Jonas** Says:

October 18, 2009 at 0:23

Evdev is installed by the core x-packages, according to the package manager (openSuse 11.1). I assume it's in use since I can't see a reason for it not to be used, but I can't be sure since I don't really know how it's supposed to look in xorg.conf. Right now it's like this:

```
Section "InputDevice"
Driver "mouse"
Identifier "Mouse[1]"
Option "CorePointer"
Option "Buttons" "10"
Option "Device" "/dev/input/mice"
Option "Name" "ImPS/2 Logitech Wheel Mouse"
Option "Protocol" "explorerps/2"
Option "Vendor" "Sysp"
Option "ZAxisMapping" "4 5"
Option "Resolution" "1800"
Option "ButtonMapping" "1 2 3 6 7"
EndSection
```

Is that how it's supposed to look when evdev is used? My knowledge of how a xorg.conf file is supposed to look is VERY rusty....

That being said, MOST things worked out of the box - including the thumb buttons. They worked as back and forth in firefox without me having to do anything. All I needed to do was to make sure the buttons did what I wanted them to do. There's just this one button...oh, and that "out of the box" experience applies to the "special" keys on my new keyboard as well. Start system-settings, and applying them to my preferred actions was all that was required.

**Hans** Says:

October 18, 2009 at 1:54

As far as I know, these sections in xorg.conf will be ignored if you have hotplugging enabled and use a newer version of Xorg. I don't know which version or if it applies to openSUSE. (I'm an Arch user myself).

**How do you use activites? « Who Says Penguins Can't Fly? Says:**

November 17, 2009 at 0:41

[...] Mouse shortcuts with xbindkeys [...]

**ioannis** Says:

December 5, 2009 at 12:33

Thanks for this howto.

I have a question. I've tried to map the side buttons of my microsoft inteli mouse, to perform back and forward in dolphin. I've used this command for backwards action:  
qdbus org.kde.dolphin /dolphin/Dolphin\_1/actions/go\_back  
com.trolltech.Qt.QAction.trigger

and a similar for forward, but then realised that this triggers irrespectively if Dolphin is the application in focus or not. Also, the back/forward in the internet browser stopped working.

Is there a way to have a conditional expression I can use, that checks if dolphin is the application in focus and if not forwards through the event to whatever app is active (so it can handle it as normal) ?

thanks

**Hans** Says:

December 5, 2009 at 12:45

@ioannis:

I would map the mouse buttons to Alt+Left and Alt+Right instead. That way it only affects the focused window, and it works for most applications with Back/Forward actions.

You can use xte to do this (as explained in the text), for example:

```
"xte 'keydown Alt_L' 'key Left' 'keyup Alt_L'"
```

**ioannis** Says:

December 5, 2009 at 15:57

@Hans

yeah that works. Thanks !

**Tony** Says:

January 27, 2010 at 19:05

Thanks, it works great for the forward and back mouse buttons in dolphin on KDE 4.4.

**Hans** Says:

January 27, 2010 at 19:35

That's very nice to hear. I think I'll add the back/forward example to the article, since many persons seem to want to remap the buttons like this.

**Update Jan 27, 2010:** Added a small section about back/forward mouse buttons.

**Isaac** Says:

February 2, 2010 at 17:58

Thanks! Have been looking to do something exactly like this. Works great.

**Hans** Says:

February 2, 2010 at 18:25

Glad that you found it useful, and thanks for the feedback.

**Clod** Says:

March 9, 2010 at 12:02

Thank you so much!

I wasted two days before arriving here and learn about "xte"! 😊

Thanks again,

Clod

**Michael** Says:

August 15, 2010 at 23:06

Hello.

I can't use the Meta key in xbindkeys.

I'm writng this in .xbindkeysrc

```
1 | "xte 'keydown Super_L' 'key z' 'keyup Super_L' "  
2 |     Mod4 + Super_L + b:5
```

Can you try to use meta key and give me the code?

Best regards...

**Hans** Says:

August 16, 2010 at 0:43

Hey Michael,

I mistook your first comment as spam and deleted it, sorry about that. Anyway it read:

Thanks for the good stuff. Those are really helpful for me...

Regarding your question, I'll give it a try when I get hold of a computer that runs Linux (not at home at the moment).

**Michael** Says:

August 21, 2010 at 21:48

I'm still waiting 😊

**Hans** Says:

August 24, 2010 at 12:51

Not sure which is your Meta key, but the problem with your configuration seems to be Super\_L (xbindkeys doesn't list it as a supported modifier).

On this computer Super\_L corresponds to Mod2+Mod4, so you can try somethings like this:

```
1 | "xte 'keydown Super_L' 'key z' 'keyup Super_L' "  
2 |     Mod2 + Mod4 + b:5
```

(I get Mod2+Mod4 + Super\_L from xbindkeys -k when I press my left Win key, which is where I guess you got the Super\_L from. However when I hit Super\_L + letter, xbindkeys suggests Mod2+Mod4 + letter.)

**Michael** Says:

September 7, 2010 at 16:44

Thanks.

This worked!

**Balden** Says:

October 9, 2010 at 19:28

Thanks this is a really clear and useful guide.

It's amazing that KDE 4 is still lacking this functionality.

Just for the record, with the latest KDE 4.5.2, I was able to trigger several KWin actions directly with the dbus trick, without using "xte". For instance adding those lines :

```
"qdbus org.kde.kglobalaccel /component/kwin  
org.kde.kglobalaccel.Component.invokeShortcut view_zoom_in"  
Mod4 + b:4
```

```
"qdbus org.kde.kglobalaccel /component/kwin  
org.kde.kglobalaccel.Component.invokeShortcut view_zoom_out"  
Mod4 + b:5
```

in my ".xbindkeysrc" allowed me to zoom in and out using the usual "Compiz" shortcut (Left Windows key + wheel up/down).

For the record : I was able to figure out the names of many other KWin actions shortcuts by running dbus-monitor in the background and triggering the effects with their "official" keyboard shortcuts (from the central KDE config). Note that I'm not at all dbus-literate (nor kde4-literate) myself so there may well be a much easier way to find this information.

**Hans** Says:

October 12, 2010 at 18:54

Thank you for your kind words Balden, and for the tip about dbus-monitor. I've added a small comment about it in the main text.

**Update Oct 12, 2010:** Added comment about using dbus-monitor to find D-Bus calls.

**Rahul** Says:

January 9, 2011 at 23:57

Really useful information and well written post. Thanks.

**Rahul** Says:

January 11, 2011 at 8:16

xte can also be used to simulate mouse clicks. Below is an example.

```
"xte 'mouseclick 1' 'keydown Control_L' 'key v' 'keyup Control_L' 'key F2'"  
b:2 + Release
```

I use this configuration when I use Openoffice calc. What it does is that when you do a middle click, it will evaluate to a left click (for selecting the cell under the cursor) followed by Ctrl+V (for pasting) followed by F2 (for editing the cell).

Hans you could perhaps consider adding this example to your main post.

**Hans** Says:

January 18, 2011 at 2:09

Thank you Rahul. I don't want to add too much to the guide, but I've included a link to your comment in the main post.

**Update Jan 18, 2011:** Added link to comment about using xte to simulate mouse clicks.

**7 ways to switch activities « Who Says Penguins Can't Fly?** Says:

May 15, 2011 at 17:00

[...] calls to achieve this, so instead I'll use a tool called xte. You can read more about xte in this post under "Emulate key presses". All you really need to do now is to install xte, which is [...]

**drman** Says:

June 7, 2011 at 15:39

Many Thanks!!!

Hopeful and clear!!!

**Dmitry** Says:

June 23, 2011 at 12:45

A very useful article. I was using xvkdb instead of xte for quite a while and experienced occasional glitches. Sometimes the assigned key combination was generated multiple times on a single mouse button press: a single wheel tilt switched through several desktops (instead of one), mouse button with Ctrl+W assigned to it closed multiple tabs in Firefox etc. Mostly occurred when the overall CPU load was heavy. Switching to xte has solved the problem so far. Using KDE 4.5.2 with desktop effects enabled.

**alex** Says:

July 3, 2011 at 16:25

It works perfectly, thank you 😊

**Hans** Says:

July 7, 2011 at 10:29

@drman, Dmitry and alex:

Thank you for your comments!

**33kg** Says:

August 19, 2011 at 17:34

@David: I quite like the fact I didn't have to install xbindkeys-config and I'm pretty used to editing configuration files (I'm using arch linux). GUI configuration tools are sometimes limiting.

Also, there is no such thing as a wrong mentality.

**Zoffix Znet** Says:

April 21, 2012 at 21:18

I seems to be the only unlucky one here. Trying to bind a Alt+Mouse button 2 to bring the desktop cube...



I even went and created a script and placed xte command in it; if I run the script the command works, but not when I trigger it through xbindkeys.

Any idea on what's wrong?

**Zoffix Znet** Says:

April 21, 2012 at 21:27

It gets even weird. If I modify my script to call xterm first, then xte works after I close xterm... :S :S :S what's going on ? 😞

**Zoffix Znet** Says:

April 21, 2012 at 21:37

Sorry for comment spam, but I made it work with dbus after finding the dbus string with dbus-monitor.

To active the cube I used this string: "qdbus org.kde.kglobalaccel /component/kwin org.kde.kglobalaccel.Component.invokeShortcut Cube"

Awesome guide! Thanks so much!

**Hans** Says:

April 22, 2012 at 13:41

@Zoffix Znet:

No worries; I'm glad that you solved the issue by yourself. If you're still curious about the xte issue, you can post the part from .xbindkeysrc here so I can look at it. (My initial guess would be problems with ' and " characters.)

**Mapping KDE actions to extra mouse buttons | Marian Schedenig's Website**  
Says:

June 7, 2012 at 0:46

[...] KDE as well, but the specific details may differ in those cases. For more details, please refer to this article, which has been my primary source and describes the used tools in much more [...]

**Thomas** Says:

August 18, 2012 at 20:06

Thanks for the nice input here. I had the problem that my Mac trackpad created multiple button events when swiped. So I hacked a script which sends out an event only once a second. The config file has to be modified like that then:

```
-----  
# Back  
"path_to_script 6"  
b:6  
-----
```

The script looks like that:

```
-----  
#!/bin/sh  
if [ -e "/tmp/xte.lock" ]; then
```

```

exit 0
fi
touch /tmp/xte.lock
if [ "$1" = "6" ]; then
/usr/bin/xte 'keydown Alt_L' 'key Left' 'keyup Alt_L'
touch /tmp/xte.6
sleep 1
fi
if [ "$1" = "7" ]; then
/usr/bin/xte 'keydown Alt_L' 'key Right' 'keyup Alt_L'
touch /tmp/xte.7
sleep 1
fi
rm /tmp/xte.lock
-----

```

**Hans** Says:

August 23, 2012 at 14:40

@Thomas:

Cool, thanks for sharing your useful script here!

**prisms** Says:

September 1, 2012 at 8:49

if you're interested, a dbus call for present windows is "qdbus org.kde.kglobalaccel /component/kwin org.kde.kglobalaccel.Component.invokeShortcut ExposeAll"

**Lucas** Says:

September 19, 2012 at 18:42

Thank you very much for this article. It helped me a lot changing the default file browser in Unity to Dolphin!

Dolphin really rocks, and its feature looks better then the one of Nautilus 😊

**FixeR** Says:

December 18, 2012 at 0:31

THANKS A LOT!! Exactly what I was looking for!! Especially the 'Emulate Key Presses' section!!! You are the best Hans!! Thanks again!! 😊 Now I feel even more proud of myself for being a linux user! 😊

**Hans** Says:

December 18, 2012 at 0:44

Nice to see that people still find this article useful (wow, was it really more than three years ago when I wrote it?). Thanks everyone for your encouraging comments!

**Gerardus Calcitrosus** Says:

May 26, 2013 at 10:11

I've noticed some problems when using combination of keyboard and mouse.

For example, ctrl+alt+down is the shortcut for 'switch to workspace down', and I want to bind this action to 'shift + scrollwheel rotate down (i.e. button 5):

```
# move workspace down
```

```
"xte 'keydown Control_L' 'keydown Alt_L' 'key Down' 'keyup Control_L' 'keyup Alt_L'"
```

```
Shift + b:5
```

This code, however, executes the shortcut ctrl+alt+SHIFT+down, as the shift button is obviously pressed when the action is executed. Anyone an idea how to get this working?

Being on Mint, I don't think there is a 'terminal' command one could execute instead of the shortcut keys to accomplish the same action.

**Hans** Says:

May 26, 2013 at 10:27

@Gerardus Calcitrosus:

For me it seems to work if you add 'keyup Shift\_L', i.e.,  
"xte 'keyup Shift\_L' 'keydown Control\_L' 'keydown Alt\_L' 'key Down'  
'keyup Control\_L' 'keyup Alt\_L'"

There should be a DBus command for that action as well, assuming that you use KWin (I can't look it up since I don't use KWin anymore).

**Building My Linux Box...The Implementation « Jeff's Skinner Box** Says:

August 1, 2013 at 23:00

[...] in these posts: How to get all those extra mouse buttons to work, How do I remap certain keys, Mouse shortcuts with xbindkeys, and Guide for setup Performance MX mouse on Linux (with [...])

**Rodrigo** Says:

August 3, 2013 at 1:41

Hey thank you very much for this guide I think It is great, I'm using fedora 18 and have a Logitech mouse T400. I'm having problems trying to config any command to one of the The buttons of the mouse, It is one of the top scroll buttons. When I use xev with that mouse button this is what I get:

KeymapNotify event, serial 31, synthetic NO, window 0x0,  
keys: 4294967199 0 0 0 0 1 0 0 0 0 0 0 0 0 0  
32 0 0 0 0 0 0 0 0 0 0 0 0 0 0

KeyPress event, serial 31, synthetic NO, window 0x2200001,  
root 0x9f, subw 0x0, time 2827093, (136,69), root:(137,160),  
state 0x50, keycode 40 (keysym 0x64, d), same\_screen YES,  
XLookupString gives 1 bytes: (64) "d"  
XmbLookupString gives 1 bytes: (64) "d"  
XFilterEvent returns: False

KeyRelease event, serial 31, synthetic NO, window 0x2200001,  
root 0x9f, subw 0x0, time 2827136, (136,69), root:(137,160),  
state 0x50, keycode 40 (keysym 0x64, d), same\_screen YES,  
XLookupString gives 1 bytes: (64) "d"  
XFilterEvent returns: False

KeyRelease event, serial 31, synthetic NO, window 0x2200001,  
root 0x9f, subw 0x0, time 2827159, (136,69), root:(137,160),  
state 0x50, keycode 133 (keysym 0xffeb, Super\_L), same\_screen YES,  
XLookupString gives 0 bytes:  
XFilterEvent returns: False

And this is what I get with xbindkeys -k when I press the mouse button:

```
"(Scheme function)"
```

```
m:0x50 + c:40
```

```
Mod2+Mod4 + d
```

I already try to assign what I got from xbindkeys -k to a command in .xbindkeysrc like this:

```
#Mozilla
```

```
"firefox"
```

```
m:0x50 + c:40
```

```
Mod2+Mod4 + d
```

And it doesn't work. I assigned another mouse button to this command for example b:3 and it works. Maybe you could help me a little with this, Thank you very much in advance.

**Hans** Says:

August 3, 2013 at 14:40

@Rodrigo:

Thanks for your kind comment! As for your issue, try to remove one of the "trigger" line, e.g.:

```
#Mozilla
```

```
"firefox"
```

```
Mod2+Mod4 + d
```

Restart xbindkeys and see if that works. You can also try to omit Mod2 (numlock).

**Rodrigo** Says:

August 20, 2013 at 1:25

@Hans:

First apologies for the late reply Hans, but I have been very busy lately. Thanks for the suggestion. I've tried what you posted with and without Mod2 and it doesn't work. I just keep having a "d" everytime I press the mouse button. I have also tried:

```
#Mozilla
```

```
"firefox"
```

```
m:0x50 + c:40
```

```
#Mozilla
```

```
"firefox"
```

```
m:0x50 + d
```

```
#Mozilla
```

```
"firefox"
```

```
Mod4 + d
```

```
#Mozilla
```

```
"firefox"
```

```
Mod2+Mod4 + d
```

And no luck. If you have any other suggestions, I'd really appreciate it. Cheers!!!

**Hans** Says:

August 21, 2013 at 7:39

@Rodrigo:

No worries! The question is, is it xbindkeys that doesn't work, or does the mouse button not send Super+D properly? To test this you can try the following:

1. Does the xbindkeys binding work if you press the shortcut on the keyboard?
2. If you go to e.g. System Settings → Shortcuts and Gestures → Global Keyboard Shortcuts, click on an unset shortcut, click the "None" button, and then press your mouse button. Does it register the binding the mouse sends?

Do you know what the intention of the mouse button is? From the output you get from xev, it seems like it's bounded to "show desktop" in Windows.

**Rodrigo** Says:

August 21, 2013 at 9:49

@Hans:

Thanks again for your reply.

1. xbindkeys is binding the keyboard shortcut well, nevertheless I noticed that pressing my right Super key + d did the shortcut immediately, but if I press the left Super key + d it didn't do anything, for it to do it with the left Super I have to keep it press and then press d repeatedly two times in a row. It'd be something like Super+d+d I guess.

2. I checked, and yes it detects the shortcut from the mouse as Super+D. It was supposed to "show desktop" originally as you said, but I'd like to assign it a regular middle button press (When you highlight text and then it pastes the highlighted text or pile i guess) . I assigned the key on the Global Keyboard Shortcuts to the "hide all normal window" command and now the button hides the windows effectively. I left xbindkeys running, and now if I press the left Super + d it hides all windows and shows the desktop as if the mouse button was pressed, but if I press the right Super + d then firefox is started as set in xbindkeys.

Hope now it's more clear, I still don't know how to set it right on xbindkeys, maybe you could help me, I've been trying but still haven't hit the jackpot. Thanks for everything!

**Hans** Says:

August 21, 2013 at 17:39

@Rodrigo:

That certainly sounds strange. Do you do anything special with the left Super key, e.g. remap it somewhere? Some places to check:

1. System Settings → Input Devices -> (Keyboard →) Advanced, do you have any options checked here?
2. Have you used xmodmap in the past? Maybe you have an old .Xmodmap file in your home directory, or you have some startup script running xmodmap?
3. Do you perhaps run ksuperkey? It shouldn't interfere, but it wouldn't hurt to kill it and see if it helps.

You could run `xmodmap` in a terminal and see if the modifiers look correctly configured. This is how my output looks like when I haven't done anything weird:

`xmodmap`: up to 4 keys per modifier, (keycodes in parentheses):

```
>shift      Shift_L (0x32),  Shift_R (0x3e)
lock        Caps_Lock (0x42)
control     Control_L (0x25), Control_R (0x69)
mod1        Alt_L (0x40),   Meta_L (0xcd)
mod2        Num_Lock (0x4d)
mod3
mod4        Super_L (0x85), Super_R (0x86), Super_L (0xce), Hyper_L (
mod5        ISO_Level3_Shift (0x5c), Mode_switch (0xcb)
```

**Rodrigo** Says:

August 22, 2013 at 1:44

@Hans:

I haven't set anything to the left Super key by myself.

1. I don't have that Advanced option on my keyboard settings, and I don't have anything weird set.

I did try the keyboard layout distribution map, to see which keys highlight when I press one on my physical keyboard. I noticed that when I press the right Super key, control, alt, etc, the appropriate key highlights on the map. Nonetheless when I press the left Super key it doesn't highlight any keys on the layout map, and what happens is that it shows me the menu and zooms out all my windows, just as if I go with the mouse to the upper left corner of my desktop (activities on fedora 18), this function was also assigned to one of my mouse buttons by default (I like it so don't need to change it).

2. I did find `xmodmap` but I don't have the `.Xmodmap` file on my home directory. This is the output I get when running `xmodmap`.

`xmodmap`: up to 4 keys per modifier, (keycodes in parentheses):

```
shift Shift_L (0x32), Shift_R (0x3e)
lock Caps_Lock (0x42)
control Control_L (0x25), Control_R (0x69)
mod1 Alt_L (0x40), Alt_R (0x6c), Meta_L (0xcd)
mod2 Num_Lock (0x4d)
mod3
mod4 Super_L (0x85), Super_R (0x86), Super_L (0xce), Hyper_L (0xcf)
mod5 ISO_Level3_Shift (0x5c), Mode_switch (0xcb)
```

3. I don't have any other thing running, I checked on my processes and killed any suspicious thing.

When I press the left Super key + d this is what I get (It is exactly the same as if I press the mouse button I'd like to assign)

```
KeyPress event, serial 31, synthetic NO, window 0x2200001,
root 0x9f, subw 0x0, time 2949203, (59,-1), root:(60,90),
state 0x40, keycode 40 (keysym 0x64, d), same_screen YES,
XLookupString gives 1 bytes: (64) "d"
```

XmbLookupString gives 1 bytes: (64) "d"

XFilterEvent returns: False

KeyRelease event, serial 31, synthetic NO, window 0x2200001,

root 0x9f, subw 0x0, time 2949323, (59,-1), root:(60,90),

state 0x40, keycode 40 (keysym 0x64, d), same\_screen YES,

XLookupString gives 1 bytes: (64) "d"

XFilterEvent returns: False

KeyRelease event, serial 31, synthetic NO, window 0x2200001,

root 0x9f, subw 0x0, time 2949521, (59,-1), root:(60,90),

state 0x40, keycode 133 (keysym 0xffeb, Super\_L), same\_screen YES,

XLookupString gives 0 bytes:

XFilterEvent returns: False

If I press the right Super key + d I get:

KeyPress event, serial 28, synthetic NO, window 0x2200001,

root 0x9f, subw 0x0, time 3097505, (380,151), root:(381,242),

state 0x0, keycode 134 (keysym 0xffec, Super\_R), same\_screen YES,

XLookupString gives 0 bytes:

XmbLookupString gives 0 bytes:

XFilterEvent returns: False

KeyPress event, serial 31, synthetic NO, window 0x2200001,

root 0x9f, subw 0x0, time 3097663, (380,151), root:(381,242),

state 0x40, keycode 40 (keysym 0x64, d), same\_screen YES,

XLookupString gives 1 bytes: (64) "d"

XmbLookupString gives 1 bytes: (64) "d"

XFilterEvent returns: False

KeyRelease event, serial 31, synthetic NO, window 0x2200001,

root 0x9f, subw 0x0, time 3097783, (380,151), root:(381,242),

state 0x40, keycode 40 (keysym 0x64, d), same\_screen YES,

XLookupString gives 1 bytes: (64) "d"

XFilterEvent returns: False

KeyRelease event, serial 31, synthetic NO, window 0x2200001,

root 0x9f, subw 0x0, time 3097901, (380,151), root:(381,242),

state 0x40, keycode 134 (keysym 0xffec, Super\_R), same\_screen YES,

XLookupString gives 0 bytes:

XFilterEvent returns: False

This is what I get when I press the left Super key alone (I pressed it twice, one press it shows the activities and second press returns to normal):

FocusOut event, serial 31, synthetic NO, window 0x2200001,

mode NotifyGrab, detail NotifyAncestor

FocusIn event, serial 31, synthetic NO, window 0x2200001,

mode NotifyUngrab, detail NotifyAncestor

KeymapNotify event, serial 31, synthetic NO, window 0x0,

23 of 24 keys: 4294967199 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
FocusOut event, serial 31, synthetic NO, window 0x2200001,
mode NotifyGrab, detail NotifyNonlinear
```

```
FocusIn event, serial 31, synthetic NO, window 0x2200001,
mode NotifyUngrab, detail NotifyNonlinear
```

```
KeymapNotify event, serial 31, synthetic NO, window 0x0,
keys: 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

I'm sorry for such a complicated thing, and I hope somehow it is fixable. If it isn't I'll just stick to the show the desktop shortcut anyways. Thank you so much again for all the help.

**Hans** Says:

August 22, 2013 at 9:51

@Rodrigo:

Unfortunately I'm running out of ideas. Since the mouse button seems to be picked up by the KDE shortcut manager, you can try to map it there instead. System Settings → Shortcuts and Gestures → (Custom Shortcuts →) Edit → New → Global Shortcut → Command/URL → Trigger tab, set shortcut by clicking on the mouse button → Action tab, Command/URL: xte "mouseclick 2"

Hope that works!

**Rodrigo** Says:

August 23, 2013 at 20:38

@Hans:

I tried that last option, and it seems that: xte "mouseclick 2" is not recognized as the middle mouse button click. I also tried to set it to another mouse button with xbindkeys and nothing happened. Do you happen to know the correct c xte command for the middle mouse button?

Thank you very much for everything again!

**Hans** Says:

August 24, 2013 at 23:45

@Rodrigo:

It's just getting stranger and stranger...

I tried the xte command before posting it and it works fine here - when I run the command in a terminal and has the mouse over the terminal window, it'll paste the clipboard content into the terminal.

I guess you don't have a middle mouse button on your mouse? If you do have one, you can run xev in a terminal, press the mouse button over the white window and look at the output to see which mouse button you pressed. You could also try to press all other buttons and see which one you're missing. Finally, you can try to run the xte command in a terminal with different numbers (not sure what you did with xbindkeys). Just make sure you don't have anything harmful in your clipboard, or



place the mouse cursor over another textbox.

**Rodrigo** Says:

August 25, 2013 at 4:05

@Hans:

Thank you for all the help Hans. The command works as it is supposed to, sorry that I didn't try it in the terminal before. Silly mistake!

And effectively, as you said I don't have a middle mouse button, I have two buttons at each end of a digital scroll. One is the one I want to assign the middle button function, and the other is automatically assigned to the show the activities overview, as I commented before.

Well I've created the shortcut, and if I assign it to the mouse button itself, it just doesn't work. What I did is that I assigned it to Ctrl + Mouse button (Ctrl+Super+D) and it works. This is very weird because if I assign the mouse button to an already given action or command on the shortcuts (not a custom one) like for example Show the desktop, it actually works.

Well I'm kind of happy now, I have a middle mouse button and a show the desktop button, hahaha!

Thank you so much for all of your help, without it I would have never even tried to figure anything, and sorry for all the time taken from you!!

Cheers!!!

**Hans** Says:

August 26, 2013 at 22:05

@Rodrigo:

Glad to hear that you're happy with the solution!

***Use mouse wheel to middle click mouse 3 instead of switching scroll modes in Ubuntu 13.04 | WyldePlayground.net***WyldePlayground.net - Says:

September 22, 2013 at 18:05

[...] <https://blog.hanschen.org/2009/10/13/mouse-shortcuts-with-xbindkeys/> [...]

**Lady Aleena** Says:

December 17, 2013 at 2:22

I am having a problem figuring out the back and forward buttons of the mouse since I altered the key presses. I use the 'Back' (and ctrl+PgUp) and 'Forward' (and ctrl+PgDn) buttons on my keyboard to back and forward. xev returned button 8 for back on my mouse and button 9 for forward on my mouse. Do I need to use the key # for Back and Forward keyboard keys to assign the mouse buttons? I tried using the GUI to set the mouse buttons, but it appears I can't.

I'm totally lost.

Keyboard: Microsoft Natural Ergonomic Keyboard 4000 v 1.0

Mouse: Microsoft Wireless Laser Mouse 5000

**Hans** Says:

December 17, 2013 at 2:34

@Lady Aleena:

**xbindkeys** lets you map a key combination (e.g. button 8 on your mouse) to a command. It sounds like you want to bind b:8 and b:9 to some actions. If you want to use the standard back/forward shortcuts, you can use the snippet from the post, just replace the mouse buttons:

```
# Back
"xte 'keydown Alt_L' 'key Left' 'keyup Alt_L' "
  b:8

# Forward
"xte 'keydown Alt_L' 'key Right' 'keyup Alt_L' "
  b:9
```

If your 'Back' ('Forward') button on your keyboard is the same as Ctrl+PgUp (Ctrl+PgDn), you can bind the mouse button to the same key combination like so:

```
# Back
"xte 'keydown Control_L' 'key Page_Up' 'keyup Control_L' "
  b:8

# Forward
"xte 'keydown Control_L' 'key Page_Down' 'keyup Control_L' "
  b:9
```

In this case, pressing button 8 on your mouse should have the same effect as the 'Back' button on your keyboard.

Hope that helps.

**RobertS** Says:

January 1, 2014 at 17:54

@Hans, maybe you could help me out...

I used this method a long time ago to map right wheel tilt > Control l to close tabs and left wheel tilt > control t to open new tabs in my browser. I had it working this way in xbindkeys:

```
#right Center mouse wheel tilt close browser tabs (ctrl+w)
"/usr/bin/xvkbd -sendevent -text "[Control_L][w]"
m:0x10 + b:7
```

```
#Left Center mouse wheel tilt to open firefox tabs (ctrl+t):
"/usr/bin/xvkbd -sendevent -text "[Control_L][t]"
m:0x10 + b:6
```

Then, my mouse broke down and the logitech model I had was discontinued. I now have a new logitech mouse but can't get it to work. I gave up on trying some versions of ubuntu ago but would like to get this working again.

I tried xev, but it's not giving me a key number for the mouse tilt. Here is what I am getting:

Right wheel tilt:

```
LeaveNotify event, serial 36, synthetic NO, window 0x5c00001,  
root 0x290, subw 0x5c00002, time 7376275, (43,22), root:(109,1043),  
mode NotifyGrab, detail NotifyVirtual, same_screen YES,  
focus YES, state 16
```

```
EnterNotify event, serial 36, synthetic NO, window 0x5c00001,  
root 0x290, subw 0x5c00002, time 7376275, (43,22), root:(109,1043),  
mode NotifyUngrab, detail NotifyVirtual, same_screen YES,  
focus YES, state 16
```

```
KeymapNotify event, serial 36, synthetic NO, window 0x0,  
keys: 4294967184 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

-----  
Left wheel tilt:

```
LeaveNotify event, serial 36, synthetic NO, window 0x5c00001,  
root 0x290, subw 0x5c00002, time 7378809, (43,22), root:(109,1043),  
mode NotifyGrab, detail NotifyVirtual, same_screen YES,  
focus YES, state 16
```

```
EnterNotify event, serial 36, synthetic NO, window 0x5c00001,  
root 0x290, subw 0x5c00002, time 7378809, (43,22), root:(109,1043),  
mode NotifyUngrab, detail NotifyVirtual, same_screen YES,  
focus YES, state 16
```

```
KeymapNotify event, serial 36, synthetic NO, window 0x0,  
keys: 4294967184 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Thanks for any help you can give.

**Hans** Says:

January 11, 2014 at 23:54

@RobertS:

Sorry for the late response, I've been away on a trip over the holidays. What is the model of your Logitech mouse? Do you get the same output from xev if you quit xbindkeys?

**killabee44** Says:

January 12, 2014 at 20:33

Thanks for your reply. It is a Logitech MX 1100. It looks exactly like the one you have pictured up top.

**killabee44** Says:

January 12, 2014 at 20:34

Sorry, this is RobertS. I guess my wordpress login kicked in.

**killabee44** Says:

January 12, 2014 at 20:51

When I shut down xbindkeys, the output is slightly different, but xev still does not

Right tilt gives me:

```
MotionNotify event, serial 36, synthetic NO, window 0x5000001,  
root 0x290, subw 0x5000002, time 160870021, (65,46), root:(131,1067),  
state 0x10, is_hint 0, same_screen YES
```

```
EnterNotify event, serial 36, synthetic NO, window 0x5000001,  
root 0x290, subw 0x0, time 160870029, (73,46), root:(139,1067),  
mode NotifyNormal, detail NotifyInferior, same_screen YES,  
focus YES, state 16
```

```
KeymapNotify event, serial 36, synthetic NO, window 0x0,  
keys: 4294967184 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Left tilt gives me:

```
MotionNotify event, serial 36, synthetic NO, window 0x5000001,  
root 0x290, subw 0x5000002, time 161037728, (60,39), root:(126,1060),  
state 0x10, is_hint 0, same_screen YES
```

```
EnterNotify event, serial 36, synthetic NO, window 0x5000001,  
root 0x290, subw 0x0, time 161037737, (77,39), root:(143,1060),  
mode NotifyNormal, detail NotifyInferior, same_screen YES,  
focus NO, state 16
```

```
KeymapNotify event, serial 36, synthetic NO, window 0x0,  
keys: 4294967184 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

**Hans** Says:

January 13, 2014 at 18:56

Are you sure you haven't tried to remap the mouse button using another application? You could try to make a new user, log in, and run xev again.

If it still doesn't generate a ButtonPress event, I'm pretty much out of ideas. You could try to run

```
xinput test <id>
```

and see if you can get the mouse button that way. The <id> is found by just running xinput.

**Chuck** Says:

January 15, 2014 at 18:43

I have a Logitech Performance MX. All mapped buttons work fine with xbindkeys and xte, but when I map the wheel side buttons to back and forward in history, it reacts too fast, jumping forward/back multiple times. Is there a way to assign only one event instead of multiple per click. I tried b:(number) + Release and the bash script above. Perhaps only delay the event before releasing the side button.

**Hans** Says:

28 of 34  
January 17, 2014 at 19:48

11/4/18, 10:40 PM

@Chuck:

That sounds strange. If you run xev and hold down one of the side buttons, does it generate multiple events? Does the same thing happen if you map it to something else? (E.g., if you map it to launch an application, does the application get launched several times?)

Maybe you can try to map it to release instead, e.g. Release+b:(number).

**Chuck** Says:

January 18, 2014 at 4:05

Thanks for the response, Hans. Yes, xev produces continuous events. It will also launch a program continuously if I hold the side button. I have it now remapped to b:6 + Release (or Release + b:6) and no luck. I'm curious if I could add a .5 or 1 second delay before releasing the button to compensate. Right now, if I flick it fast enough I get only one event....sometimes. Using Ubuntu 13.10 and 14.04.

**Hans** Says:

January 18, 2014 at 10:36

Unfortunately I don't know how to disable repeat for mouse buttons. I found this question (maybe it's yours?), but no answers to far.

The only workaround I can think of right now is pretty ugly; it involves creating a script that checks if an instance of the script is running, if not run the command you want and then sleep for x seconds. In other words, you want to make sure you only get one instance of the script running every x second. Let me know if you need any help with this.

**Chuck** Says:

January 18, 2014 at 17:33

Thanks for the link, Hans. Similar, but it was not my posting. Most times I perservere, because there is always a solution floating somewhere on the Linux 'net. But this time I gave in and bought a Logitech M705. The problem is solved. Thanks for your help.

**Jeff R.** Says:

June 19, 2014 at 8:01

I just wanted to thank you for this tutorial.

As a power user of AutoHotKey on Windows, I was at a loss on how to bring over the basics to Linux (ex. I bind LControl + Primary Mouse for copy and LControl + Secondary Mouse for paste).

This had made the transition to Linux that much more awesome. So thanks again!

**Hans** Says:

June 20, 2014 at 18:58

@Jeff R.:

That makes me happy to hear, thanks for your comment Jeff! Hope you enjoy Linux!

(By the way, did you know that any text you select in Linux (or to be more correct, the X Window System) is copied by default, and you can paste it by clicking on the middle mouse button? There are also clipboard managers like Klipper that allow you

***Disable CTRL + Scroll Zooming In Ubuntu*** Says:

August 14, 2014 at 17:06

[...] found this article about setting up new key bindings using xbindkeys, and it turned out that I already had some [...]

***Pingvin på GXT152 | FOSSWorld.dk*** Says:

March 21, 2015 at 14:20

[...] Who Says Penguins Can't Fly? – Mouse shortcuts with xbindkeys, Guide til opsætning af xbindkeys. [...]

***Chris*** Says:

September 7, 2015 at 8:14

Hey thanks for your article. Helped me.

I like to bind my wheel tilt left and right to prev/next browser tab as such:

```
"xte 'keydown Control_L' 'key Prior' 'keyup Control_L'"  
b:6 + release
```

```
"xte 'keydown Control_L' 'key Next' 'keyup Control_L'"  
b:7 + release
```

***Using xbindkeys for Mouse Button Awesomeness | Random Musings*** Says:

October 4, 2015 at 12:32

[...] some quick Googling about how to map mouse buttons in Fedora 21, I came across a few sites like this one and this one. It looked like it was possible using xte from xautomation, xev from xorg-x11-utils, [...]

***Using xbindkeys for Trackball Button Awesomeness | Random Musings*** Says:

October 4, 2015 at 12:51

[...] some quick Googling about how to map mouse buttons in Fedora 21, I came across a few sites like this one and this one. It looked like it was possible using xte from xautomation, xev from xorg-x11-utils, [...]

***Ubuntu: Use mouse wheel to middle click mouse 3 instead of switching scroll modes in Ubuntu 13.04 - Ubuntu Linux Questions*** Says:

November 19, 2015 at 2:31

[...] <https://blog.hanschen.org/2009/10/13/mouse-shortcuts-with-xbindkeys/> [...]

***Jamie Bird Portfolio*** Says:

February 2, 2016 at 10:18

[...] <https://blog.hanschen.org/2009/10/13/mouse-shortcuts-with-xbindkeys/> [...]

***Using xbindkeys for Trackball Button Awesomeness - Random Musings*** Says:

June 17, 2016 at 10:10

[...] some quick Googling about how to map mouse buttons in Fedora 21, I came across a few sites like this one and this one. It looked like it was possible using xte from xautomation, xev from xorg-x11-utils, [...]

***Logitech Performance MX on Linux - Wired in IT*** Says:

June 27, 2016 at 0:04

**Using the Mouse to Switch Workspaces in Linux Mint - Schlameel** Says:

September 5, 2016 at 13:33

[...] used this page as a basis, I came up with the following [...]

**Binding mouse keys to keyboard keys on Linux | cubethethird's corner** Says:

September 8, 2016 at 0:19

[...] my mouse to Page Up and Page Down. I used for reference several different blogs found here, here and here, along with the information on the Arch Wiki. The resulting method utilizes the xbindkeys and xte [...]

**Abhineet** Says:

September 12, 2016 at 17:29

Reblogged this on kaamkichiz.

**How to map mouse side buttons to scroll? | Ubuntutextbook** Says:

September 15, 2016 at 21:08

[...] bought a mouse with side buttons and I want to map them to scroll up and down. As Mouse shortcuts with xbindkeys explains, I created a ~/.xbindkeysrc [...]

**Filip Vidak** Says:

November 9, 2016 at 10:57

Sorry for bothering you on a 6 years old post. I'm for 2 things:

1st, thank you for this tutorial, I really missed some functions that SetPoint did with my Performance MX.

2nd, do you have any idea why mouseclick events in Action don't work? For example, if I assign any keyboard action to any of mouse button, it'll work. But if action is mouseclick smth, it doesn't work. I want to assign button 13 to be middle button, which should be button 2 (xev) or CTRL+left click (button 1). Tried both and neither works, nor does trying mouseclick with other numerations.

These 2 below are what I tried. Changing 2 to any other does the same, nothing.

`"xte 'mouseclick 2'"`

`b:13`

`"xte 'keydown Control_L' 'mouseclick 1' 'keyup Control_L'"`

`b:13`

**Filip Vidak** Says:

November 9, 2016 at 11:13

Can't edit my replay from above.

For some reason I had to add + Release to b:13 in order to make it work. Other buttons didn't need it.

Again, thanks for tutorial.

**Hans** Says:

November 10, 2016 at 18:46

@Filip Vidak:

Happy to hear that you solved it, and thanks for sharing the solution!

**xseektruthx** Says:

December 15, 2016 at 13:18

I've tried quite a few variations to get this to work. Basically, I just want to hold Shift\_L and use the mouse wheel up b:4 and down b:5 to act as Page Up c:112 or Page Down c:117

No matter what actions I tell it, or swapping of xte, xterm, buttons and so on, I just can't seem to get it to work.

When I use bind keys config GUI to "Get Key", it returns the following information:  
Using

Page Up = Prior | m:0x0 + c:112

Page Down = Next | m:0x0 + c:117

Left Shift = Shift + Shift\_L | m:0x1 + c:50

Can't use "Get Key" for mouse wheel

If I ask xev it shows the following:

Page Up

KeyPress event, keycode 112 (keysym 0xff55, Prior)

KeyRelease event, keycode 112 (keysym 0xff55, Prior)

Page Down

KeyPress event, keycode 117 (keysym 0xff56, Next)

KeyRelease event, keycode 117 (keysym 0xff56, Next)

Left Shift

KeyPress event, keycode 50 (keysym 0xffe1, Shift\_L)

KeyRelease event, keycode 50 (keysym 0xffe1, Shift\_L)

xinput for the mouse shows:

Wheel Up

motion a[3]=15

button press 4

button release 4

Wheel Down

motion a[3]=14

button press 5

button release 5

Tried a few of the following...

#Page Up

"Prior | m:0x0 + c:112"

c:50 + b:4 + Release

#Page Down

"Next | m:0x0 + c:117"

c:50 + b:5 + Release

-----  
#Page Up

"Prior | m:0x0 + c:112"

Shift\_L + b:4 + Release

#Page Down

"Next | m:0x0 + c:117"

Shift\_L + b:5 + Release



#Page Up

"c:112"

c:50 + b:4 + Release

#Page Down

"c:117"

c:50 + b:5 + Release

Also tried without Release. Any help would be greatly appreciated!

**xseektruthx** Says:

December 15, 2016 at 14:31

Realized xte was not installed. did so, and now it works with the following changes...

#Page Up

"xte 'key Prior'"

m:0x40 + c:37 + b:4 + Release

#Page Down

"xte 'key Next'"

m:0x40 + c:37 + b:5 + Release

Ok so using the Shift key was a Bad Idea lol. It was selecting upon Page movement, however worked with no problems when viewing online email.

Now I switched to the Super button as shown above, Online email and local PC windows no longer work, but all other online scroll windows do. I believe it has to do with my own Super button bindings preventing it from working on ALL windows.

Thanks for listening! 😊

**Hans** Says:

December 16, 2016 at 23:55

@xseektruthx:

Glad that you found a solution, thanks for sharing!

**Yakup** Says:

January 28, 2017 at 14:46

Hi, Thanks for this good guide. Really helped me. 😊 But I have a little question. I want to "zoom in" and "zoom out" on browser with my mouse. I used to do this, when I turned scroll and I pressed button that I assigned as "ctrl" simultaneously. Now I am assigned ctrl key to Button10. It's ok, running perfectly but if I use simultaneously both scroll and button 10 nothing happens. What can I do to resolve this situation? Thanks again.

**Hans** Says:

February 1, 2017 at 22:46

@Yakup:

Glad to hear that you found the guide helpful. Unfortunately I haven't tried to assign a modifier key to a mouse button. I could give it a try if I find some free time, but it might take a while since I'm quite busy right now.

« KDE Community Forums and Feeds

Interests »

Create a free website or blog at WordPress.com.