



How To Bind Mouse Buttons To Keyboard Keys Or Commands (Linux Using X11)

by Logix Updated on November 22, 2019 in [how-to](#)

If you have a mouse with extra buttons, you can bind those buttons to perform a key combination, run a program, command or script. This will only work if you use X11 (it does not work on Wayland) because the tools we'll use for this only support X11.

To bind the mouse buttons to key combinations we'll use the following tools:

- > [xbindkeys](#): a program that grab keys and mouse button events in X and starts associated shell command
- > [xev](#): a tool that prints contents of X events
- > [xdotool](#): command-line X11 automation tool that lets you programmatically (or manually) simulate keyboard input and mouse activity (and more)

In case you don't need to bind mouse buttons to keyboard keys, and all you want to bind to

RECENT POSTS

[KDiskMark Is A GUI HDD / SSD Benchmark Tool For Linux \(Similar To CrystalDiskMark\)](#)

[How To Enable Echo / Noise Cancellation Of Microphone Input On Your Linux Desktop \(PulseAudio\)](#)

[How To Find Files Modified In The Last N Days Or Minutes Using find](#)

[Qt5 Screenshot Tool Flameshot Sees New Release](#)



[How To Install Oracle Java 15 On Ubuntu, Debian Or Linux Mint Using APT PPA Repository](#)

[HOME](#)[NEWS](#)[APPLICATIONS](#)[TUTORIALS](#)[TWEAKS](#)[THEMES & MORE](#)[CONSOLE](#)[🔍](#)

xdotool.

1. Install xbindkeys, xev and xdotool.

> Debian, Ubuntu, Linux Mint, Pop!_OS, Elementary OS and other Debian or Ubuntu based Linux distributions:

Terminal

```
sudo apt install xbindkeys x11-utils xdotool
```

> Fedora:

Terminal

```
sudo dnf install xbindkeys xorg-x11-utils xdotool
```

> openSUSE:

Terminal

```
sudo zypper install xbindkeys xev xdotool
```

> Arch Linux or Manjaro:

Terminal

```
sudo pacman -S xbindkeys xorg-xev xdotool
```



EDITOR'S PICKS

[Top Things To Do After Installing Ubuntu 20.04 Focal Fossa To Make The Most Of It](#)

[OpenSnitch Linux Application Firewall With Improvements And Bug Fixes](#)

[4 Tools To Record Your Linux Desktop \(Screencast\) In 2020](#)

[How To Boot To Console \(Text\) Mode Using Debian / Ubuntu, Fedora, Arch Linux / Manjaro And More](#)

[FFmpeg: Extract Audio From Video In Original Format Or Converting It To MP3 Or Ogg Vorbis](#)

[How To Install Oracle Java 14 \(JDK 14\) On](#)

and `xbindkeys` and install them.

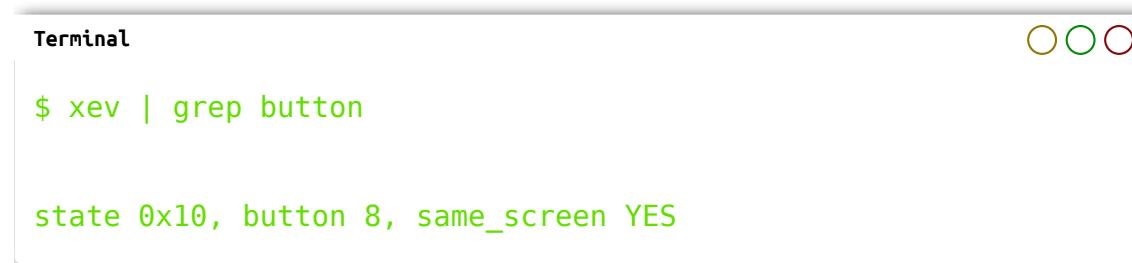
2. Grab the mouse button codes.

To get the mouse button code(s), run this command:



```
Terminal
xev | grep button
```

Next, focus the small window that pops up and watch the terminal output. Now press the mouse button for which you want to grab the code. After pressing the button you should see its code in the terminal where you ran `xev`, e.g.:



```
Terminal
$ xev | grep button

state 0x10, button 8, same_screen YES
```

In this example, the button code we'll need later is `8`.

3. Grab the keystrokes that we'll later send using a mouse button (skip if you want to bind a command / script / program to a mouse button)



Open a terminal and run this command (via [AskUbuntu](#)):

Terminal



```
xev | sed -ne '/^KeyPress/,/^$/p'
```

Once again, a small window will pop up that you need to focus. Next, press the keys on your keyboard that you want to remap to your mouse, each at a time, and watch the xev command output in the terminal.

For example if you want to bind `Ctrl + Alt + Up` to a mouse button, press `Ctrl`, then `Alt` and finally `Up`. In my case, this is the output of `xev` for these 3 key presses (the first one is for `Control_L` or left Control key, the second for `Alt_L` or left Alt, and the third is for `Up` :

```
KeyPress event, serial 33, synthetic NO, window 0x2200001,
  root 0x1eb, subw 0x0, time 6741696, (46,-37), root:(728,598),
  state 0x10, keycode 37 (keysym 0xffe3, Control_L), same_screen YES,
  XLookupString gives 0 bytes:
  XmbLookupString gives 0 bytes:
  XFilterEvent returns: False
```

```
KeyPress event, serial 36, synthetic NO, window 0x2200001,
  root 0x1eb, subw 0x0, time 6743289, (46,-37), root:(728,598),
  state 0x10, keycode 64 (keysym 0xffe9, Alt_L), same_screen YES,
  XLookupString gives 0 bytes:
  XmbLookupString gives 0 bytes:
```



```
KeyPress event, serial 36, synthetic NO, window 0x2200001,
root 0x1eb, subw 0x0, time 6744780, (46,-37), root:(728,598),
state 0x10, keycode 111 (keysym 0xff52, Up), same_screen YES,
XLookupString gives 0 bytes:
XmbLookupString gives 0 bytes:
XFilterEvent returns: False
```

Note down the keycode, keysym or name of the key. You can use any of the 3, e.g. for the first key above you can use either `37` , `0xffe3` , or `Control_L` . In this article I'll use the key name (e.g. `Control_L` for left control key) since they are easier to read.

It's worth noting that `xbindkeys` can be used to grab the keycodes too (`xbindkeys -k`), but it reads all active keys so it can be confusing in some cases, e.g. if you have `Caps_Lock` or `Num_Lock` on, etc., so I personally prefer to use `xev` for this task.

Neither `xed` nor `xbindkeys -k` will show the keycodes / keyboard symbols for multimedia keys. In case you want to remap multimedia keys to mouse buttons, you can get a list of such keyboard symbols by [visiting this link](#).

Mouse-related article: [How To Disable Mouse Acceleration In Ubuntu 18.04 Or 19.10 \(Gnome\)](#).

4. Create the `xbindkeys` configuration (in `~/.xbindkeysrc`).



You may either create an empty `~/.xbindkeysrc` file, or generate a sample configuration file using this command:

HOME

NEWS

APPLICATIONS

TUTORIALS

TWEAKS

THEMES & MORE

CONSOLE



Terminal



```
xbindkeys -d > ~/.xbindkeysrc
```

Next, open `~/.xbindkeysrc` with your favorite text editor. Note that `.xbindkeysrc` is a hidden file in your home directory so you'll need to press `Ctrl + h` (to show hidden files and folders) to see it in your file manager.

To bind a mouse button to a key combination, paste the following at the end of the `~/.xbindkeysrc` file (it may already contain some key binds, it may contain the sample configuration or it can be empty - it doesn't matter):

```
"xdotool key 'KEY-COMBINATION'  
b:MOUSE-BUTTON-CODE
```

Where:

- `KEY-COMBINATION` are the keystrokes you got under step 3 (separate the keys with a `+` sign)
- `MOUSE-BUTTON-CODE` is the mouse button code that you got under step 2

Example:



b:8

This remaps the mouse button 8 (which for my mouse is the button that can be used as a back button in a web browser for example) to `Control_L+Alt_L+Down` which in GNOME Shell is used to switch to workspace down.

To also remap the mouse forward button (mouse button 9 in my case) to `Control_L+Alt_L+Up` (switch to workspace up in GNOME Shell), I'd have to add this to the `~/.xbindkeysrc` file:

```
"xdotool key 'Control_L+Alt_L+Up'  
b:9
```

Another example. To bind the keyboard media keys for raising and lowering the volume to mouse buttons 8 and 9:

```
"xdotool key 'XF86AudioRaiseVolume'  
b:9  
  
"xdotool key 'XF86AudioLowerVolume'  
b:8
```



In the same way, add as many key binds as you wish, but make sure you're not using some already existing key combinations.

[HOME](#)[NEWS](#)[APPLICATIONS](#)[TUTORIALS](#)[TWEAKS](#)[THEMES & MORE](#)[CONSOLE](#)

To bind a command, script or program to a mouse button, use this in the

`~/ .xbindkeysrc` file:

```
"COMMAND"  
b:MOUSE-BUTTON-CODE
```

Here:

- › `COMMAND` is a command, script or program
- › `MOUSE-BUTTON-CODE` is the mouse button code that you got under step 2

You may also like: [Run Or Raise Application Windows On Linux \(X11\) Using A Single Keyboard Shortcut With jumpapp](#).

Example for running a program using a mouse button:

```
"firefox"  
b:8
```

This starts Firefox when pressing mouse button 9 (which for my mouse is the button that can be used as a back button in a web browser for example).



5. Start xbindkeys.

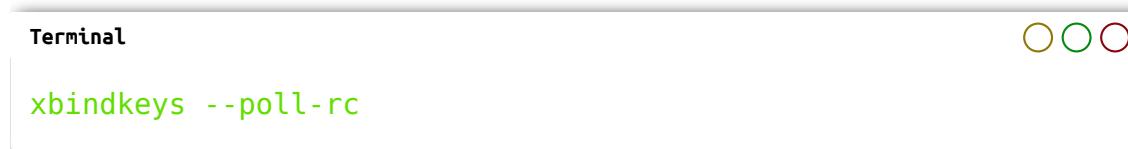
[HOME](#)[NEWS](#)[APPLICATIONS](#)[TUTORIALS](#)[TWEAKS](#)[THEMES & MORE](#)[CONSOLE](#)

Now you can start xbindkeys using a terminal and typing:



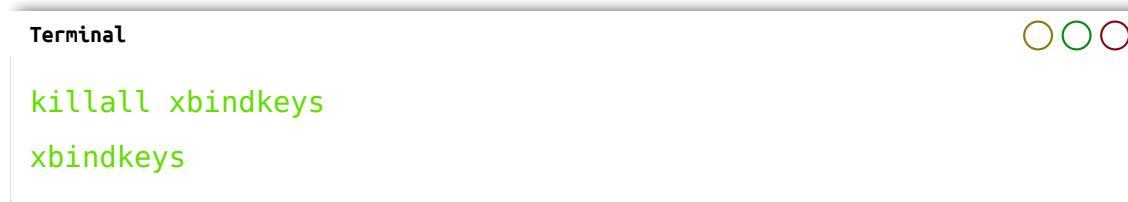
```
Terminal
xbindkeys
```

In case xbindkeys was running, you can get it to use the new configuration by issuing:



```
Terminal
xbindkeys --poll-rc
```

This command failed in a few cases for me when I changed the xbindkeys configuration, I'm not sure why. In such a case you can restart it by killing the xbindkeys process and running it again:



```
Terminal
killall xbindkeys
xbindkeys
```

On Ubuntu, xbindkeys is automatically started on system startup if it finds a non-empty (it needs to have lines that are not commented out) `~/.xbindkeysrc` configuration file. If the tool doesn't automatically start for the Linux distribution you're using, add `xbindkeys` to



SHARE THIS:

[!\[\]\(ac13c516668a3b529e385da83084b241_img.jpg\) Facebook](#) [!\[\]\(9785750a6edead5e3cf5f752d010e69d_img.jpg\) Twitter](#) [!\[\]\(28cd5bfbee0e4025d8ccb0980993dd56_img.jpg\) Diaspora](#) [!\[\]\(c8a8ebd1e75c82612c036d05f998f646_img.jpg\) Reddit](#) [!\[\]\(9786747911728a475082c821e96863ce_img.jpg\) Telegram](#) [!\[\]\(c0bb0082dc9271e9d2dd2bbf14296766_img.jpg\) WhatsApp](#)

[How To Enable Echo / Noise Cancellation Of Microphone Input On Your Linux Desktop \(PulseAudio\)](#)

[How To Find Files Modified In The Last N Days Or Minutes Using find](#)

[How To Install Oracle Java 15 On Ubuntu, Debian Or Linux Mint Using APT PPA Repository](#)

Please post civil and on-topic comments. [Commenting guidelines.](#)



HOME

NEWS

APPLICATIONS

TUTORIALS

TWEAKS

THEMES & MORE

CONSOLE

[Recommend](#)[Tweet](#)[Share](#)[Sort by Newest](#)

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS [?](#)

Name

**cRoot** • a month ago

Yuda man!

Thanks for your time.

[^](#) | [v](#) • Reply • Share [›](#)**Theja Voora** • 4 months ago

Thank you so much! It was very easy to follow and worked like a charm!

[^](#) | [v](#) • Reply • Share [›](#)**user359** • 10 months ago

So I mapped button 2 "middle button" to page down "Next" and of course is worked in x11.

The strange thing is in terminal the output is "-"

I was using easystroke and that works in terminal too but sometimes sends wrong key. I still prefer xbindkeys for that reason.

[2](#) [^](#) | [v](#) • Reply • Share [›](#)**user359** • 10 months ago • edited

thanks, needed this.

Another excellent thing is emulate scroll with hold right click and drag.

xinput --set-prop pointer:"USB OPTICAL MOUSE " "libinput Middle Emulation Enabled" 1

xinput --set-prop pointer:"USB OPTICAL MOUSE " "libinput Button Scrolling Button" 3

xinput --set-prop pointer:"USB OPTICAL MOUSE " "libinput Scroll Method Enabled" 0 0 1

xinput --set-prop pointer:"USB OPTICAL MOUSE " "libinput Natural Scrolling Enabled" 1



Use this to get mouse name

xinput list --name-only

Some mice have spaces at end of name like above "USB OPTICAL MOUSE " so use this too

HOME

NEWS

APPLICATIONS

TUTORIALS

TWEAKS

THEMES & MORE

CONSOLE



LINUX UPRISING

FOLLOW BY EMAIL

BLOG ARCHIVE

[About](#)

[Privacy Policy](#)

[Terms of Service](#)

[Cookie Information](#)

[Privacy Policy](#)

[Cookie Settings](#)

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Email address...

SUBMIT

[Blog Archive](#)

License: Creative Commons Attribution 4.0 International License Linux Uprising.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

