



www.stc.hu

Activating your multimedia keys with ReMoot

PLAYBACK

If your laptop multimedia keys aren't working in Linux, the fix could be easier than you think. We'll show you how to manage multimedia applications from the keyboard using ReMoot. **BY FEDERICO KEREKI**

From this article, you will learn how to make Linux (with KDE) recognize special-function keys and how to make them function with all your favorite multimedia applications.

Many contemporary desktop and laptop keyboards include multimedia keys (for "Play," "Mute," "Volume Up," and so on), but often these special keys don't work well with Linux. Recently, I decided it was high time to get the multimedia keys on my laptop to work.

The Key Problem

I needed to solve two problems. The first was getting Linux to recognize the (potentially non-standard) keys; the second was getting each key to perform the necessary multimedia function. Fortunately, KDE lets you easily associate keyboard actions with commands, and a tool called ReMoot provides the necessary command set for accessing multimedia applications. In this article, I describe

how to activate keyboard and laptop multimedia keys with ReMoot.

The Solution

The first step is getting the keys to work. If you want to go to the effort, you can define your own keyboard map and use a tool such as Lineak [1] to configure special keys, but the `xmodmap` program offers a simpler approach.

For each key you want to use, first identify its scancode (what the key produces when you press it). To identify the scancode, open a command window and key in `xev`.

You'll see lots (I mean lots!) of messages in the command window. Press any of the multimedia keys and watch the output. In my case, when I press the "Volume Up" key I get:

```
KeyRelease event, serial 31, ↵
synthetic NO, window ↵
0x2a00001,
```

```
root 0x4c, subw 0x0, time ↵
1349394, (173,-13), ↵
root:(922,5),
state 0x0, keycode 176 ↵
(keysym 0x0, NoSymbol), ↵
same_screen YES,
XLookupString gives 0 bytes:
XFilterEvent returns: False
```

The important feature of this output is the number after "keycode," which is the desired scancode value; in this case, the "Volume Up" key generates a 176. For each multimedia key on your laptop, you will have to go through this process, and don't forget to use modifiers (Shift, Control, Alt); you will discover that you have many more possibilities available than you imagined.

Table 1: ReMoot Commands

<code>remoot play</code>	start playing
<code>remoot playpause</code>	pause if playing; restart if paused
<code>remoot stop</code>	stop playback
<code>remoot next</code>	change to next track
<code>remoot prev</code>	change to previous track
<code>remoot mute</code>	mutes/unmutes sound
<code>remoot volup</code>	volume up a notch
<code>remoot voldown</code>	volume down a notch

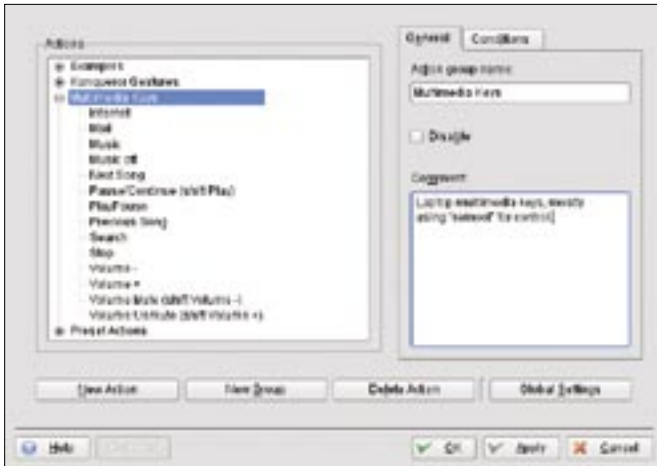


Figure 1: Defining a new group for the multimedia keys.

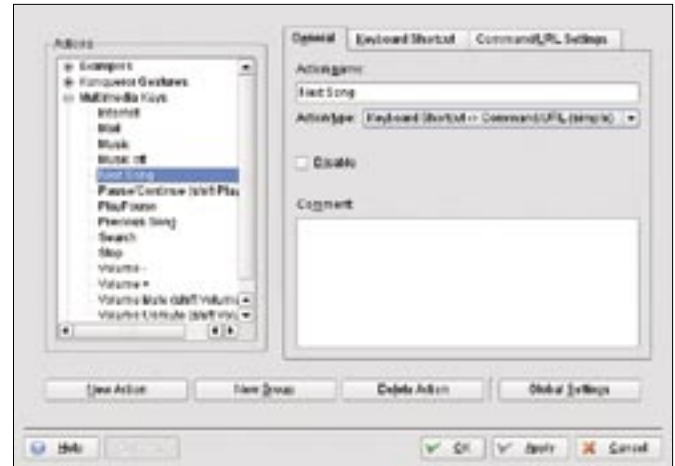


Figure 2: Creating a new action for the "Next Song" key.

Other behaviors might be noted. For example, I found out that the "Brightness Up" and "Brightness Down" keys weren't recognized by *xev*, so I couldn't configure them with this technique.

Now you know your keys, but Linux still doesn't, so you must assign a specific keycode to each key. To get the possible keycodes, examine the `/usr/X11R6/lib/X11/XKeysymDB` file near the bottom: you want the lines starting with `XF86`

that match your keys. (By the way, the path for this file can vary among distributions; if you cannot find it, use the `find / -name XKeysymDB` command to locate it.)

After getting all scancodes and deciding the corresponding keycodes, I created in my home path a file named `.Xmodmap` (note the initial period; the period makes the file hidden) with the following contents; of course, you will

have to modify this file to suit your own machine:

```
keycode 144 = XF86AudioPrev
keycode 153 = XF86AudioNext
keycode 164 = XF86AudioStop
keycode 174 = XF86AudioLowerVolume
keycode 176 = XF86AudioRaiseVolume
```

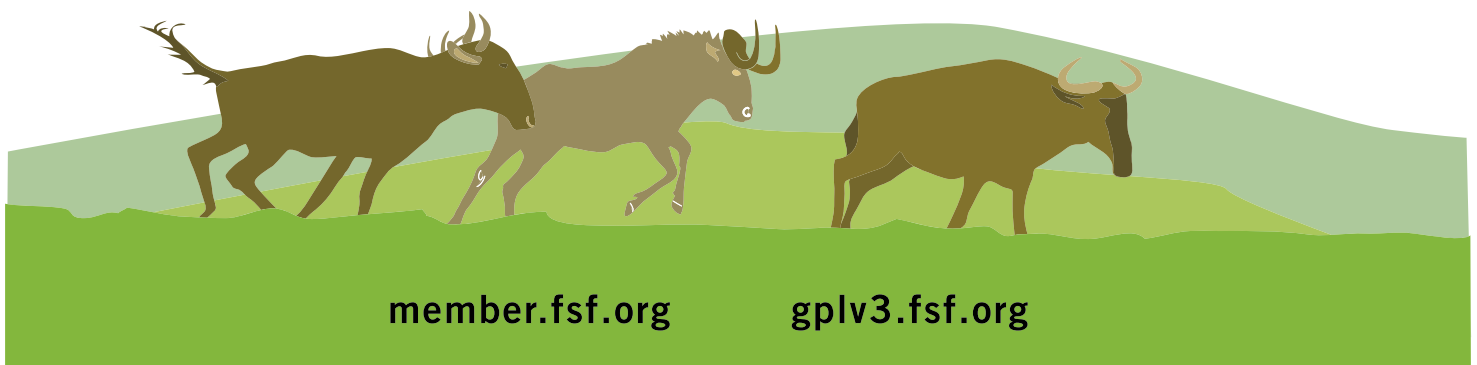
The format of the file is quite simple: Each scancode has a line (those I found

Freedom:

the freedom to run the software as you wish.

Join the FSF as an associate member.

Your donation will support our 20-year-old commitment to community.



member.fsf.org

gplv3.fsf.org

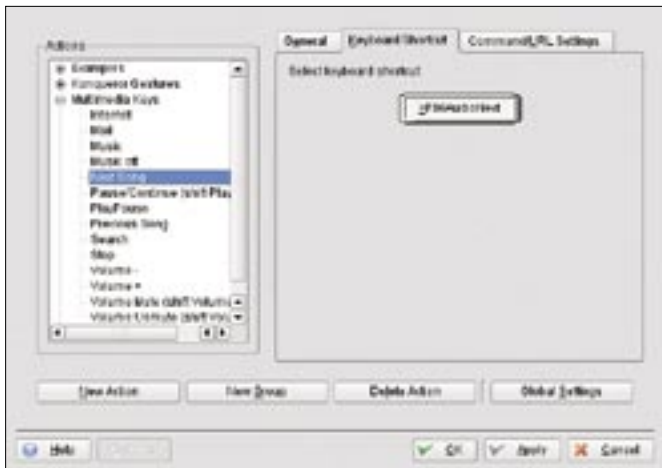


Figure 3: Assigning a keyboard shortcut to "Next Song."

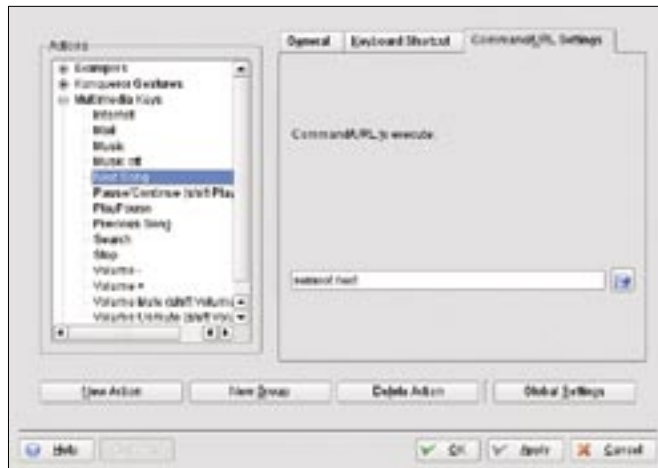


Figure 4: Specifying the reemoot command for "Next Song."

with `xev`) relating it to a keycode. Finally, I added a short script, `keys.sh` (the name doesn't matter), under my home directory at `./kde/Autostart/` with two lines:

```
#!/bin/sh
xmodmap .Xmodmap
```

It runs automatically whenever you start a session; don't forget to make this script executable with `chmod +x keys.sh`.

Installing ReMoot

Now that Linux recognizes the multimedia keys, you have to specify what you want the keys to do. For starters, go to the ReMoot website [2] and click on *Source Download* to get the latest version of ReMoot (v0.9).

ReMoot is a Perl script. After the download (which should be over quickly for the 12KB file), go to the directory from which you downloaded it, `su` to root, and enter:

```
$ tar zxf reemoot-0.9.tar.gz
$ cd reemoot-0.9/
$ chmod +x *moo*
$ cp *moo* /usr/bin
```

ReMoot comprises three related packages: The ReMoot application itself; `daemoot`, which is used by ReMoot; and `reemoot-remote`, a graphical front-end. (A separate package, `Rewww00t`, lets you use a Palm Pilot or Nokia N800 as a remote for ReMoot.) In this case, ReMoot is enough, but feel free to experiment with the other tools; you'll find more information at the ReMoot wiki [3]. If you don't have Perl, use your favorite package manager to install it.

To see whether ReMoot is working, open any music program and start playing something (Amarok and some jazz works for me!), then try raising or lowering the volume with the commands shown in Table 1. `reemoot mute` should immediately mute the sound; key in the same command again, and you will get the sound back.

What's great about ReMoot is that it supports many programs: If you are running Amarok, then `reemoot playpause` will pause the playback, and if you are running Kaffeine, the result is the same. As long as you are running a program known to ReMoot (see the box "Programs Supported by ReMoot"), you will get consistent results; the multimedia keys will work no matter what program you are using to listen or view media.

Once you install ReMoot, all that's left is to associate ReMoot commands with the multimedia keys. If you use KDE, open the Control Center, and in the *Regional & Accessibility* tab, pick the *Input*

Actions option. Then, add a group with the *New Group* command and name it something like *Multimedia Keys* (see Figure 1). Then, for each key:

- add a new action to the group (see Figure 2);
- go to the *General* tab and fill the *Action name* field;
- set the *Action type* to *Keyboard Shortcut -> Command/URL (simple)*;
- go to the *Keyboard Shortcut* tab, click on the button, and press the special key you want to link to this action – remember, you can also use the Shift, Control, and Alt modifiers if they work with the key (see Figure 3); then
- go to the *Command/URL Settings* tab and type a ReMoot command you want to execute whenever you press the key (e.g., for the "Volume Up" key, I specified the `reemoot volup` command) (see Figure 4).

If all goes well, these steps will map the multimedia keys to the necessary ReMoot commands.

Success

Getting the multimedia keys to work wasn't as complicated as I first imagined. KDE lets you associate commands with keyboard actions, and ReMoot makes it easy to control multimedia applications from the command line. ■

INFO

[1] Lineak site:
<http://lineak.sourceforge.net/>

[2] The ReMoot website:
<http://www.kde-apps.org/content/show.php/ReMoot?content=63140>

[3] The ReMoot wiki site:
<http://reemoot.wiki.sourceforge.net/>

Programs Supported by ReMoot	
amarok	mpd
aqualung	mplayer
audacious	noatun
banshee	pytone
bmp	quark
exaile	quodlibet
gmusicbrowser	rhythmbox
juk	smplayer
kaffeine	totem
kmplayer	vlc
kscd	xine
listen	xmms
moc	xmms2