

Unicom Systems Development

open source • open networks • open protocols

Setting up Function Keys for a Linux Media PC

Submitted by [chip \(/user/2\)](#) on Sat, 22-Oct-2005 - 2:57 pm


I thought it would be nice if I could use the multimedia special function keys to control the music player. Sure would be handy to have a *mute* button on my Linux-based music player when a phone call comes in. I thought I might be in for a challenge, winding through Windows-specific crap and missing Linux support. I was surprised to find it wasn't that hard at all. Here is [an article \(http://gentoo-wiki.com/HOWTO_Use_Multimedia_Keys\)](http://gentoo-wiki.com/HOWTO_Use_Multimedia_Keys) that helped me greatly.

There are three parts to the process to setting up the multimedia keys:

- Define the keyboard map for the multimedia keys.
- Associate the multimedia keys with actions.
- Make the mapping persistent.

Define the Keyboard Map

The procedure to define the keyboard map is straightforward and described in that article. Just grab a pencil and paper and follow these steps:

- Run `xev` in a command terminal. Test each of the multimedia keys and record their associated keycodes.
- Match up the keys to symbols in the `/usr/X11R6/lib/X11/XKeysymDB` file. I was pleased to find that every key had a match, even the "My Computer" and  (music note) keys. (I used XF86MyComputer and XF86Music, respectively.)
- Create an `~/.Xmodmap` file that defines the symbol associated with each keycode.
- Run: `xmodmap ~/.Xmodmap`
- Re-run `xev`, and verify that this time when you press a key, it displays the correct symbol name along with the keycode value.

Here is the `.Xmodmap` I used for the Dell keyboard:

```
keycode 234 = XF86Back
keycode 233 = XF86Forward
keycode 232 = XF86Stop
keycode 231 = XF86Refresh
keycode 130 = XF86HomePage

keycode 236 = XF86Mail
keycode 198 = XF86MyComputer
keycode 161 = XF86Calculator

keycode 164 = XF86AudioStop
keycode 144 = XF86AudioPrev
keycode 162 = Setting up Function Keys for a Linux Media PC | Unicom Systems Development
keycode 153 = XF86AudioNext
keycode 160 = XF86AudioMute
keycode 129 = XF86Music
keycode 174 = XF86AudioLowerVolume
keycode 176 = XF86AudioRaiseVolume
```

Associate the Multimedia Keys

Now that the function keys have been identified to the system, the next step is to associate them to actions. Surprisingly, the volume and mute buttons worked right away. Apparently in my distribution (*Kubuntu* Linux) they already are associated with the *kmix* applet. Also, the web browser keys were associated with *konqueror* actions. All I had to do was associate the remaining audio keys with the *amarok* music player.

If you go into the *Configure Global Shortcuts* menu for *amarok*, you'll find all of the player actions you want to control. All you need to do is open up the shortcut you want to set and press the function key that should control it. If you've setup your mapping correctly then the symbol name will appear.

That's simple enough, but there is one significant drawback: the key is going to work only when the *amarok* application has focus. Wouldn't it be nice to be able to hit *pause* even if some other application is running on the desktop?

The KDE desktop has a great facility called [DCOP](http://developer.kde.org/documentation/other/dcop.html) (<http://developer.kde.org/documentation/other/dcop.html>) (Desktop COmmunications Protocol) that allows you to control running applications. For instance, I can use the following Linux command to pause the player:


```
$ dcop amarok player pause
```


That command locates the *amarok* application and sends it a message to pause the player. You can find a complete list of supported DCOP functions in the [Amarok Handbook](http://amarok.kde.org/component/option,com_staticxt/Itemid,49/xt_item,1/staticfile,the-dcop-interface.html) (http://amarok.kde.org/component/option,com_staticxt/Itemid,49/xt_item,1/staticfile,the-dcop-interface.html). (Check your local documentation, it may be newer and support more functions.)

So, rather than associate the multimedia keys with *amarok* commands, I associated them with DCOP instructions that sent player commands. Of course, that's going to work only if you run a KDE desktop. If you use Gnome or something else, your mileage will vary.

Here is the procedure I used to configure the keys to control the audio player:

- Launch KDE control panel.
- Select: Regional & Accessibility → Khotkeys.
- Select *New Group* and give it a name (such as "Amarok Controls").
- Select *New Action*.
- Set the *Action Name* to a descriptive name (such as "Amarok Play/Pause").
- Set the *Action Type* to Keyboard Shortcut → DCOP Call.
- Go to the *Keyboard Shortcut* tab, click into the shortcut field, and press the key to associate with this action.
- Go to the *DCOP Call Settings* tab and configure the action. It should be something like:
 - Remote application: amarok
 - Remote object: player
 - Called function: playPause
 - You can press the *try* button to verify the action is correct.
- Save this action, and repeat for each additional action needed.

I configured all of the media player keys to perform DCOP commands, except one: the  (music note) key. All of the other keys send commands to a running *amarok* application. I wanted this key to launch the *amarok* music player. This can be done with a different kind of *khotkey* action:

- Select *New Action*.
- Set the *Action Name* to a descriptive name (such as "Launch Amarok Music Player").
- Set: *Action Type* to Keyboard Shortcut → Command/URL.
- Go to the *Keyboard Shortcut* tab, click into the shortcut field, and press the  (music note) key. The "XF86Music" symbol appears.
- Go to the *Command/URL Settings* tab and set the command to: `amarok --toggle-playlist-window`
- Save the action.

Make the Mapping Persistent

At this point the music player is fully configured. The only problem is that the keyboard map modifiers will be lost when the session ends. To load the map every time I start KDE, I created a file called `~/.kde/env/xmodmap.sh` that says:

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

I also ran `chmod 755` on the file to make it executable.

Note that a more general solution would load this map for any user and any display manager or desktop. At this time, there aren't any other user accounts (there isn't even a login screen on the console), so this works for me.

The media function keys work great. Initially I thought I'd want a remote control for the media PC. Now, not so much. So long as I keep a keyboard in the living room, I may prefer the multimedia keys instead.

The next challenge is to extend the controls further by network access. That way, for instance, I can control the player when I'm across the room at my workstation.