# The XKB Configuration Guide

Kamil Toman, Ivan U. Pascal

25 November 2002

**Abstract**

This document describes how to configure XFree86 XKB from a user's point a few. It converts basic configuration syntax and gives also a few examples.

## 1. Overview

The XKB configuration is decomposed into a number of components. Selecting proper parts and combining them back you can achieve most of configurations you might need. Unless you have a completely atypical keyboard you really don't need to touch any of xkb configuration files.

## 2. Selecting XKB Configuration

The easiest and the most natural way how to specify a keyboard mapping is to use `rules` component. As its name suggests it describes a number of general rules how to combine all bits and pieces into a valid and useful keyboard mapping. All you need to do is to select a suitable rules file and then to feed it with a few parameters that will adjust the keyboard behaviour to fulfill your needs.

The parameters are:

- `XkbRules` - files of rules to be used for keyboard mapping composition
- `XkbModel` - name of model of your keyboard type
- `XkbLayout` - layout(s) you intend to use
- `XkbVariant` - variant(s) of layout you intend to use
- `XkbOptions` - extra xkb configuration options

The proper rules file depends on your vendor. In reality, the commonest file of rules is `xfree86`. For each rules file there is a description file named `<vendor-rules>.lst`, for instance `xfree86.lst` which is located in xkb configuration subdirectory `rules` (for example `/etc/X11/xkb/rules`).

### 2.1 Basic Configuration

Let's say you want to configure a PC style America keyboard with 104 keys as described in `xfree86.lst`. It can be done by simply writing several lines from below to you XFree86 configuration file (often found as `/etc/X11/XF86Config-4` or `/etc/X11/XF86Config`):

```
Section "InputDevice"
    Identifier "Keyboard1"
    Driver "Keyboard"

    Option "XkbModel" "pc104"
    Option "XkbLayout" "us"
    Option "XKbOptions" ""
EndSection
```

The values of parameters `XkbModel` and `XkbLayout` are really not surprising. The parameters `XkbOptions` has been explicitly set to empty set of parameters. The parameter `XkbVariant` has been left out. That means the default variant named `basic` is loaded.

Of course, this can be also done at runtime using utility setxkbmap. Shell command loading the same keyboard mapping would look like:

```
setxkbmap -rules xfree86 -model pc104 -layout us -option ""
```

The configuration and the shell command would be very analogical for most other layouts (internationalized mappings).

## 2.2 Advanced Configuration

Since XFree86 4.3.x you can use multi-layouts xkb configuration. What does it mean? Basically it allows to load up to four different keyboard layouts at a time. Each such layout would reside in its own group. The groups (unlike complete keyboard remapping) can be switched very fast from one to another by a combination of keys.

Let's say you want to configure your new Logitech cordless desktop keyboard, you intend to use three different layouts at the same time - us, czech and german (in this order), and that you are used to `Alt-Shift` combination for switching among them.

Then the configuration snippet could look like this:

```
Section "InputDevice"
    Identifier "Keyboard1"
    Driver "Keyboard"

    Option "XkbModel" "logicordless"
    Option "XkbLayout" "us,cz,de"
    Option "XKbOptions" "grp:alt_shift_toggle"
EndSection
```

Of course, this can be also done at runtime using utility setxkbmap. Shell command loading the same keyboard mapping would look like:

```
setxkbmap -rules xfree86 -model logicordless -layout "us,cz,de" \
         -option "grp:alt_shift_toggle"
```

## 2.3 Even More Advanced Configuration

Okay, let's say you are more demanding. You do like the example above but you want it to change a bit. Let's imagine you want the czech keyboard mapping to use another variant but basic. The configuration snippet then changes into:

```
Section "InputDevice"
    Identifier "Keyboard1"
    Driver "Keyboard"

    Option "XkbModel" "logicordless"
    Option "XkbLayout" "us,cz,de"
    Option "XkbVariant" ",bksl,"
    Option "XKbOptions" "grp:alt_shift_toggle"
EndSection
```

That's seems tricky but it is not. The logic for settings of variants is the same as for layouts, that means the first and the third variant settings are left out (set to `basic`), the second is set to `bksl` (a special variant with an enhanced definition of the backslash key).

Analogically, the loading runtime will change to:

```
setxkmap -rules xfree86 -model logicordless -layout "us,cz,de" \
         -variant ",bksl," -option "grp:alt_shift_toggle"
```

## 2.4 Basic Global Options

See rules/*.lst files.

# 3. Direct XKB Configuration

Generally, you can directly prescribe what configuration of each of basic xkb components should be used to form the resulting keyboard mapping. This method is rather "brute force". You precisely need to know the structure and the meaning of all of used configuration components.

This method also exposes all xkb configuration details directly into XFree86 configuration file which is a not very fortunate fact. In rare occasions it may be needed, though. So how does it work?

## 3.1 Basic Components

There are five basic components used to form a keyboard mapping:

- *key codes* - a translation of the scan codes produced by the keyboard into a suitable symbolic form
- *types* - a specification of what various combinations of modifiers produce
- *key symbols* - a translation of symbolic key codes into actual symbols
- *geometry* - a description of physical keyboard geometry
- *compatibility maps* - a specification of what action should each key produce in order to preserve compatibility with XKB-unware clients

## 3.2 Example Configuration

Look at the following example:

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver "Keyboard"

    Option "XkbKeycodes" "xfree86"
    Option "XkbTypes"    "default"
    Option "XkbSymbols"  "en_US(pc104)+de+swapcaps"
    Option "XkbGeometry" "pc(pc104)"
    Option "XkbCompat"   "basic+pc+iso9995"
EndSection
```

This configuration sets the standard XFree86 default interpretation of keyboard keycodes, sets the default modificator types. The symbol table is composed of extended US keyboard layout in its variant for pc keyboards with 104 keys plus all keys for german layout are redefined respectively. Also the logical meaning of `Caps-lock` and `Control` keys is swapped. The standard keyboard geometry (physical look) is set to pc style keyboard with 104 keys. The compatibility map is set to allow basic shifting, to allow Alt keys to be interpreted and also to allow iso9995 group shifting.

# 4. Keymap XKB Configuration

It is the formerly used way to configure xkb. The user included a special keymap file which specified the direct xkb configuration. This method has been obsoleted by previously described rules files which are far more flexible and allow simpler and more intuitive syntax. It is preserved merely for compatibility reasons. Avoid using it if it is possible.

CONTENTS

1. Overview ..... 1

2. Selecting XKB Configuration ..... 1
   2.1 Basic Configuration ..... 1
   2.2 Advanced Configuration ..... 2
   2.3 Even More Advanced Configuration ..... 2
   2.4 Basic Global Options ..... 3

3. Direct XKB Configuration ..... 3
   3.1 Basic Components ..... 3
   3.2 Example Configuration ..... 3

4. Keymap XKB Configuration ..... 4

$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/XKB-Config.sgml,v 1.4 2003/09/21 01:55:03 dawes Exp