GNOME and XKB: looking at the issues

Introduction

Since GNOME 2.6, released in the spring 2004, GNOME has got new keyboard configuration/indication architecture, based on GSwitchIt project. This code brought several components into the GNOME desktop and its dependencies:

- Two tabs of the Keyboard Preferences in the GNOME Control Center (gnome-keyboard-properties): "Layouts" and "Layout options". The UI stores user preferences into GConf.
- Keyboard Indicator applet in the GNOME Applets. The applet indicates the current state of the keyboard.
- Keyboard component of the gnome-settings-daemon. The code listens to the changes in GConf settings and actualizes them in X server configuration (using XKB or xmodmap).
- libxklavier library – common keyboard management library (not part of the GNOME desktop, external dependency). The library is used directly or indirectly by all other components.
- libgswitchit virtual module, statically linked to the Keyboard Preferences and Keyboard Indicator. The library provides utility functions for the integration of libxklavier into GNOME Desktop (like saving the keyboard configuration into GConf).

All the functionality relies on some important components of the X11 distribution:

- XKB code in the X server
- XKB code in Xlib
- xkbcomp utility
- libxkbfile library
- XKB configuration repository

Since the overall architecture is relatively complex, it may fail at various points. Every component may fail as a result of an internal bug. In some circumstances, the components may fail to communicate – if they make different assumptions regarding their interfaces.

This document presents most general ideas regarding the troubleshooting of the keyboard-related problems in the GNOME Desktop (version 2.10). It is intended for users experiencing issues with the layout management. It is strongly recommended to read this document before applying for help in bugzillas.

Statistics analysis

Querying bugzillas (GNOME, Red Hat), there was found 208 keyboard-related bugs in the gnome-applets and gnome-control-center (statistics on 26.03.2005). Taking the usability, i18n, keybinding etc. bugs aside (non-critical and/or irrelevant), there are 40 bugs which are really related to the layout management, switching, indication. etc functionality. Of them, 24 are caused by the problems in XKB configuration repository, 6 – by the libxklavier library (2 are reported against historical xmodmap

incompatibility), 4 – by the libgswitchit library. The rest of the problems is caused by the xmodmap support in the gnome-settings-daemon – and various issues with X server, xlib library, xkbcomp utility.

This statistics is not 100% accurate (some bugs may have been overlooked by the simple queries used) – but it demonstrates several important points.

The keyboard-related GNOME user experience heavily depends on the X server – and, most of all, XKB configuration repository. GNOME development process has no direct control over it – but users can report the problems to the teams maintaining their X implementations. X servers without XKB support are still in use, users still tweak the keyboard with .Xmodmap files – so GNOME should provide decent support for xmodmap.

General considerations

This chapter describes most common troubleshooting methods user can apply to the keyboard-related problems in GNOME. These techniques are recommended for everyone experiencing problems with the keyboard configuration in GNOME.

XKB extension

In case of troubles, first of all, user should determine whether his X server provides XKB extension or not. The simplest and most natural way to check it is to run the command

```
$ xdpyinfo | grep KEYBOARD
```

If XKB extension is present, one word is printed on output:

```
    XKEYBOARD
```

Otherwise, nothing is printed.

On XFree86 and X.Org implementations, it is strongly recommended to have XKB enabled, if possible (in XF86Config and xorg.conf respectively). It is enabled by default, but if it not, the line

```
          Option  "XkbDisable"    "true"
```

Should be commented out (of course, X server should be restarted in order to pick up the modification).

It may help to analyse X server log files:

```
grep -i xkb /var/log/Xorg.0.log
```

or /var/log/XFree86.0.log (filenames for X display :0.0). The output may look like:

```
(**) Option "XkbRules" "base"
(**) Keyboard0: XkbRules: "base"
(**) Option "XkbModel" "pc105"
(**) Keyboard0: XkbModel: "pc105"
(**) Option "XkbLayout" "us,ru(winkeys)"
(**) Keyboard0: XkbLayout: "us,ru(winkeys)"
```

In case of problems, the log file may give some insights on the matter (failed xkbcomp utility, invalid options etc).

Libxkbfile functionality

There is controversy regarding this library and its usage in GNOME (indirectly, through libxklavier). This library appeared in the reference implementation of X11R6 and exists in two major free implementations (XFree86 and X.Org) – but some commercial distributions lack it which causes problems for people trying to use GNOME in these environments (see discussion at http://bugzilla.gnome.org/show_bug.cgi?id=152105).

Nevertheless, this library is required for XKB support in GNOME – and its proper functioning should be checked in the proper troubleshooting process. The library uses so called "window properties" in order to store the XKB configuration, they can be checked using the command:

```
$ xprop -root | grep XKB
```

If libxkbfile is working correctly (with XKB extension enabled), at least one line is printed, displaying the property "_XKB_RULES_NAMES":

```
_XKB_RULES_NAMES(STRING) = "base", "acer_tm_800", "us,ru",
",winkeys", "eurosign:e,lv3:ralt_switch,grp:rctrl_toggle"
```

This property shows the current XKB configuration.

In GNOME environment, there should be second line printed:

```
_XKB_RULES_NAMES_BACKUP(STRING) = "base", "pc105", "us,ru",
",winkeys", ""
```

This property contains the XKB configuration at X server startup (taken from the configuration file).

The values of these properties are displayed here as example, they depend on the current keyboard configuration – so in your system, they are different most probably.

There are five elements in each of these properties:

- The ruleset (in the examples, it is "base", in XFree86 implementation it is "xfree86" by default, in X.Org implementation – "xorg").
- The keyboard model chosen (in the examples, "acer_tm_800" and "pc105")
- The keyboard layouts configured (in the examples, "us,ru" - two layouts)
- The keyboard layout variants (in the examples, ",winkeys" - which means first layout uses some "default" variant and second uses "winkeys" variant)
- The keyboard configuration options (in the examples, "eurosign:e,lv3:ralt_switch,grp:rctrl_toggle" - three options)

In the properly functioning GNOME environment, the property _XKB_RULES_NAMES contains the configuration set in the gnome-keyboard-properties utility. It represents some internal identifiers which can be easily mapped to the user-friendly descriptions in the GNOME interface.

The ruleset name is important – in bug reports, some problems were reported in the situations where it was not set correctly or pointed to the broken rulesets. In XFree86 and X.Org, the ruleset is defined in the configuration file by the line:

```
        Option  "XkbRules" "base"
```

If this line is missing, the default ruleset is used ("xfree86" and "xorg" correspondingly). For the GNOME Desktop to function properly, this name should define 3 existing valid files: the ruleset itself, the ruleset listing file and the ruleset

registry file. These files are located in the X server lib/X11/xkb/rules subdirectory. In the example above, the names are “base”, “base.lst” and “base.xml” correspondingly. If any of these files is missing or invalid, it may cause problems (sometimes broken packages do not install important files – it was the case for original Fedora Core 2 installation).

GNOME keyboard configuration

GNOME uses its own configuration repository GConf to keep the keyboard-related information. In order for GNOME (namely, gnome-settings-daemon) to configure the keyboard properly, GConf registry should work properly and contain some set of values. These values can be analyzed querying the /desktop/gnome/peripherals/keyboard/kbd subtree with the gconftool-2:

```
$ gconftool-2 -R /desktop/gnome/peripherals/keyboard/kbd
```

In correctly functioning system, the output looks like:

```
 layouts = [us,ru       winkeys]
 model = acer_tm_800
 overrideSettings = false
 options = [eurosign    eurosign:e,lv3  lv3:ralt_switch,grp
grp:rctrl_toggle]
 update_handlers = []
```

The most important variables “layouts”, “options”, “model” are self-explanatory – they comprise the configuration chosen in the Keyboard Preferences utility (gnome-keyboard-preferences). These parameters are actualized by the gnome-settings-daemon into proper XKB configuration – which makes them appear in the libxkbfile-controlled window property _XKB_RULES_NAMES (see above).

It is not recommended to modify these variables manually using Configuration Editor (gconf-editor). The delimiter used in the “layouts” and “options” values is not reproducible in this GUI tool.

Specific per-server and per-version issues

GNOME keyboard layout management, tightly integrated with XKB extension and corresponding libraries and configuration files since release 2.6, introduced hard functional dependency on the quality of the keyboard code in X server.
The first most critical point in interaction with X server historically was the support for the “multiple layouts”, introduced in XFree 4.3.0. GNOME keyboard configuration (earlier – GSwitchIt) saluted this feature because it allows to combine up to 4 layouts in one setup – easily and simply. So, the first point of divide for X servers is the support of this feature.

XFree86 4.2

The first X server supported by GNOME 2.6+ keyboard infrastructure (since it was used in the Debian stable). The XKB implementation does not support “multiple layouts” - so only one layout can be chosen in the configuration UI. Some layouts are multi-group ones, so users still can work with several languages - though they can never tell, choosing the layouts, which languages (groups) are going to be actually available.

XFree86 4.3

The very first X implementation with “multiple layouts” functionality. At the time of GNOME 2.6 release, it was the recommended X server from the keyboard management point of view.

The new feature caused major redesign of the XKB configuration repository, performed by Ivan Pascal. In particular, most of the layouts were converted into single-group format. Unfortunately, some layouts were left in the old format – which caused some problems for the GNOME users, trying to combine them with other, “new” layouts. The list of “old format” layouts was declared in the /usr/X11R6/lib/X11/xkb/rules/xfree86 file as:

```
// Layouts that still need be composed by old rules
! $oldlayouts = az bs ca ca_enhanced de_CH fr_CH hu hu_qwerty lt_std
mn vn
```

So, many users reported problems related to these layouts – which GNOME could not resolve because the issue was not in the GNOME code.

Also, the release code had a bug which prevented users from combining some valid layouts (the fix was committed to XFree86 CVS just after the release and some distributions included it as a post-release patch). This bug caused some confusion for GNOME users.

XFree86 4.4+/X.Org 6.7.0+

These two releases were very similar (taking the fact that X.Org was based on some XFree86 prerelease) – and they did not have any difference in the keyboard layout management area. The multiple layouts code is pretty solid at this point – but the list of “old format” layouts was not reduced (even increased by one):

```
// Layouts that still need be composed by old rules
! $oldlayouts = az bs ca ca_enhanced de_CH fr_CH hu hu_qwerty lt_std
mn vn hr_US
```

From this point, nothing seriously changed in the mainstream open sources X server – neither in XFree86 nor in monolithic X.Org implementation. The “old format” layouts are still around.

XKeyboardConfig (X.Org 7.x.x?)

This project was created as a part of the effort put an order into X.Org server XKB configuration repository – and later joined the movement aiming to modularize the X.Org server. The major achievement of the project is eliminating “old format” layouts – converting them to the new format. Unfortunately, the compatibility is broken – so users may have to reconfigure the keyboard in GNOME.

Users have to update their X server configuration file (XF86Config or xorg.conf) with the line

```
Option  "XkbRules" "base"
```

and keep it there (some distributions, like Fedora Core, perform maintenance of this file during the upgrade procedures). Though, it is possible to build xkeyboard-config in the “compatility” mode (–with-xkb-rules-symlink option) to workaround this issue.

Conclusion

Users of GNOME still cannot be absolutely happy with the keyboard management. GNOME development process is trying to help and improve things – but sometimes it fails because the stakeholders are outside of the community. GNOME is obviously trying to communicate with X server developers (there is very productive dialogue with the X.Org team).

GNOME encourages people to report bugs – and do it smartly. If user knows (feels) that some keyboard-related problem is specific to the X server – he can report it to the GNOME bugzilla – but it would be more efficient to report it to the respective X server support service (or both). But the golden rule here is: the bug is better be reported to the wrong service than not reported at all.