Visual Bell mini-Howto

Alessandro Rubini

rubini@linux.it

This document explains how to use termcap to configure a visual bell on one's system and describes how to disable audible bells on demand.

1. Copyright and Licensing Information

2. Introduction

The Linux console driver beeps the audible bell whenever a BEL char is output (ASCII code 7). Though this is a right choice for the default behaviour, many users don't like their computer to beep. This mini-Howto is meant to explain how to tell applications not to output the BEL code. It also explain how to instruct the kernel and the X Window System to avoid beeping when a BEL is output.

Note that most of this document refers to the text console, as configuring the X server is an easy catch-all for any user who works in a graphic environment.

In my opinion the best way to face a fussy computer is fixing the hardware, and my own computer doesn't even carry a loudspeaker.

3. Speakerectomy

Speakerectomy is by far the most brilliant solution to the audible bell problem. As its name implies, it consists in removing the beeps by removing the beeper. The operation is straightforward and you don't even need any anesthetic, but if you want there's room for refinement.

PC's are usually equipped with a silly switch to lower CPU clock. The switch is never used when you work in a multitasking environment, as you don't even need to slow the computer down to run games based on software loops. Unfortunately we can't use the switch to increase processor speed, but we can use it to enable/disable the loudspeaker. Sometimes the speaker is useful even if you enjoy a silent number cruncher, for example to signal the end of a lengthy compilation. To modify the switch functionality, just detach it from the main board and connect its wires in series with the loudspeaker.

Owners of laptop boxes, unfortunately, don't have easy access to the loudspeaker, and neither they have a spare switch to turn to a different task. The preferred solution for such users is configuring their software to avoid beeping, as described below.

4. Per-console Beep Configuration

As of Linux 1.3.43, Martin Mares added the ability to configure the pitch and duration of the beep, by modifying console.c. Each console can be configured to feature a different duration and/or pitch of the bell sound; the task is accomplished by using escape sequences to the console device. You can configure your own ~/.profile or ~/.login file to select a different beep sound associated to each console (or no beep at all, if needed).

The escape sequences work as follow:

> ESC-[10;xx] selects the bell frequency in Hertz. The value should be in the range 21-32766, otherwise the result is undefined. If the xx argument is missing, the default value (750Hz) will apply, as in 'ESC-[10].
>
> ESC-[11;xx] selects the bell duration, in milli-seconds. If you specify more than 2 seconds, the default applies (125ms). Once again, if the 'xx' argument is missing (ESC-[11]) the default value will be used.

To select, for example, a 50Hz pitch for one-second duration, you can echo -e "\\33[10;50]\\33[11;1000]" with bash (where -e means 'understand Escape sequences'. If you use tcsh the same command spells echo "\\033[10;50]\\033[11;1000]".

Although I don't know of any version of the setterm command that supports such configuration, a future version of the command might well support a command-line option to configure the bell sound.

If you run Linux-1.3.43 or newer, you may be satisfied with the escape sequences and avoid reading further. If you run an older kernel, or if you want the visual bell, you'll enjoy the rest of this document.

5. Basic Concepts About Termcap and Terminfo

The file /etc/termcap is a text file that lists the terminal capabilities. Several applications use the termcap information to move the cursor around the screen and do other screen-oriented tasks. tcsh, bash, vi and all the curses-based applications use the termcap database.

The database describes several terminal types. The TERM environment variable selects the right behaviour at run-time, by naming a termcap entry to be used by applications.

Within the database, each capability of the terminal appears as a two-letter code and a representation of the actual escape sequence used to get the desired effect. The separator character between different capabilities is the colon (":"). As an example, the audible bell, with code "bl", usually appears as bl=^G. This sequence tells that the bell sound is obtained by printing the CTRL-G character, the ASCII BEL.

In addition to the bl capability, the vb capability is recognized. It is used to represent the "visible bell". vb is usually missing in the linux entry of the termcap file.

Most modern applications and libraries use the terminfo database instead of termcap. This database uses one file per terminal-type and lives in /usr/lib/terminfo; to avoid using huge directories, the description of each terminal type is stored in a directory named after its first letter; the linux entry, therefore, is /usr/lib/terminfo/l/linux. To build a terminfo entry you'll "compile" the termcap description; refer to the tic program and its manual page.

6. Defining a Visible Bell

You can add the entry for the vb capability in your own termcap file, if it doesn't already define one. Dennis Henriksen (duke@diku.dk) suggested to insert the following line in the termcap entry for linux (note that the entry is called console in old distributions):

```
:vb=\E7\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E8:\
```

The trailing backslash is used to escape the newline in the database. Dennis' code does the following (his own words):

Save the cursor position (uust a safety precaution).

Change the background color several times between normal and reverse.

Restore the cursor position.

7. Disabling the Audible Bell on the Text Console

If you want to force the visible bell on your console you can use the bl entry in termcap and define it with the same string suggested for vb above. This approach is handy if you don't want to customize each application (which is described below, anyway). I use this option on all the machines where I can run Linux and I can't detach the speaker.

8. Telling Applications to Avoid Beeping

This is an incomplete list of applications that can be instructed to use the vb entry for the current terminal type (using either the termcap information or the terminfo one):

The X server: use the xset b command to select the bell's behaviour. The command takes three numeric arguments: volume, pitch and duration. xset -b disables the bell altogether. Configuring the X server affects all the applications running on the display.

xterm: xterm can convert each bell to either a visible or audible signal. If you use the audible bell, the settings of xset will apply. The bell in xterm defaults to be audible, but you can use the -vb command line option and the xterm*visualBell: true resource to turn it to a visible flash. You can toggle visible/audible signaling at run-time by using the menu invoked by control–left-mouse-button. If you run X you most likely won't need the following information.

tcsh (6.04 and later): set visiblebell. The instruction can be placed in .cshrc or can be issued interactively. To reset the audible bell just unset visiblebell. To disable any notification issue use set nobeep instead.

bash (any bash, as fas as I know): put set bell-style visible in your ~/.bashrc. Possible bell-style's are also none or audible.

bash (with readline, as well as other readline-based applications): put set prefer-visible-bell in ~/.inputrc.

nvi and elvis: put set flash in ~/.exrc or tell :set flash interactively (note the colon). To disable the visible bell use noflash in place of flash.

emacs: put (setq visible-bell t) in your ~/.emacs. It is disabled by (setq visible-bell nil).

less: use -q on command line to use the visual bell, use -Q to disable any reporting. Default options can be put in your environment variable LESS.

screen: issue the CtrlA-CtrlG command. It changes the behaviour of all the virtual screens. Refer to the man page under "CUSTOMIZATION" for setting the default.

9. The Dark Side of the Problem

The bad news is that not every application uses termcap or terminfo. Most small programs feature 'backslash-a' (alarm) characters in the C source code. The "alarm" code becomes a literal ASCII BEL in the strings as stored in the executable binary. Real application don't usually fall in this category, but be careful of C newcomers who give you their own programs. Students of computer science are the worst of all, granted.

The only way to make these programs silent applications is speakerectomy, or using the escape sequences by Martin Mares.