

# The Euro symbol on FreeBSD

Aaron Kaplan

aaron@lo-res.org

Copyright © 2002, 2003 The FreeBSD Documentation Project  
\$FreeBSD: doc/en\_US.ISO8859-1/articles/euro/article.sgml,v 1.12 2010/07/31  
11:08:14 pgj Exp \$

FreeBSD is a registered trademark of the FreeBSD Foundation. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the “™” or the “®” symbol.

This document will try to help you in getting started with the new **Euro** Symbol on your new keyboard that you had to buy in early 2002 because of the switch to the new common currency. We will first focus on the more important parts like being able to correctly display the symbol on the console. Later sections will deal with configuring particular programs like **X11**.

Lots of helpful input came from Oliver Fromme, Tom Rhodes and countless others. Thanks! Without you this article would not have been possible!

## 1 The Euro in a nutshell

If you already feel comfortable with localization ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/books/handbook/l10n.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/l10n.html)) as described in the FreeBSD Handbook you might be only interested in the following facts which will get you started quickly:

ISO8859-15

This is a slight modification of the commonly used ISO8859-1 character map. It includes the Euro symbol. Used for the `LANG`, `LC_CTYPE` environment variables.

`iso15-8x16.fnt`

The `vidcontrol(1)` font for the console

`/usr/share/syscons/keymaps/*.iso.kbd`

Appropriate keyboard maps depending on your language. Set your `keymap` entry in `rc.conf` to one of these.

LC\_CTYPE

Used to specify the correct character type in your locale.

```
XkbLayout "lang(euro)"
```

**XFree86** config option.

```
/usr/X11R6/lib/X11/fonts/*/fonts.alias
```

Be sure to adapt your X11 fonts to `--iso8859-15`

## 2 A general remark

In the following sections we will often refer to *ISO8859-15*. This is the standard notation starting with FreeBSD 4.5. In older versions, the standard notation was either *ISO\_8859-15* or *DIS\_8859-15*.

If you are using an older version of FreeBSD, be sure to take a look at `/usr/share/locale/` in order to find out which naming convention is in place.

## 3 The console

### 3.1 Setting up your console font

Depending on your console resolution and size you will need one of the following lines in `rc.conf`:

```
font8x16="iso15-8x16.fnt" # from /usr/share/syscons/fonts/*
font8x14="iso15-8x14.fnt"
font8x8="iso15-8x8.fnt"
```

This will effectively select the ISO8859-15 also known as Latin-9 font. ISO8859-15 is a variation of ISO8859-1. You can tell the difference between the two by looking at the Euro symbol: its decimal value is 164. In ISO8859-1 you will notice a circle with four little strokes at the corners. This is often termed the “universal currency symbol”. In ISO8859-15, instead of the little circle, you have the Euro Symbol. Otherwise the fonts are more or less identical.

**Warning:** As of the time of this writing the only usable font seems to be `iso15-8x16.fnt`. The others seem to only show ISO8859-1 even though the name suggest otherwise.

**Note:** By specifying this font some console applications will look garbled. This is due to the fact that they assume you are using a different font/character set such as ANSI 850. One notable example is **sysinstall**. However most of the time this should not be of much concern.

As the next step you should either reboot your system to let the changes take effect or (manually) take the steps that would have been taken at the system startup:

```
% vidcontrol -f iso15-8x16.fnt
```

To check if the font has been selected execute the following short `awk` script:

```
#!/usr/bin/awk -f
BEGIN {
    for(i=160;i<180;i++)
        printf"%3d %c\n",i,i
}
```

The result should reveal the Euro sign at position 164.

## 3.2 Setting up your keyboard for the Euro

Most keyboard maps should already be set up correctly. I.e: If you have a german keyboard and your Umlaut keys are working, you can safely skip this section since the keyboard already maps whatever key combination is necessary (e.g.: **Alt Gr+e**) to decimal value 164. If running into problems, the best way to check is to take a look at `/usr/share/syscons/keymaps/*.kbd`. The format of the key mapping files is described in `keyboard(4)`. `kbdcontrol(1)` can be used to load a custom keymap.

Once the correct keyboard map is selected, it should be added to `/etc/rc.conf` with the line:

```
keymap="german.iso" # or another map
```

As stated above, this step has most probably already been taken by you at installation time (with `sysinstall`). If not, either reboot or load the new keymap via `kbdcontrol(1)`.

To verify the keyboard mapping, switch to a new console and at the login prompt, *instead of logging in*, try to type the **Euro** key. If it is not working, either file a bug report via `send-pr(1)` or make sure you in fact chose the right keyboard map.

**Note:** At this stage the Euro key will not yet work in `bash` or `tcsh`.

## 3.3 Fixing the environment variables

The shells (`bash`, `tcsh`) revert to the `readline(3)` library which in turn respects the `LC_CTYPE` environment variable. `LC_CTYPE` must be set before the shell is completely running. Luckily it suffices to add the line:

```
export LC_CTYPE=de_DE.ISO8859-15
```

to your `.bash_profile` (`bash`), or:

```
setenv LC_CTYPE de_DE.ISO8859-15
```

to your `.login` (`tcsh`) file. Of course, `de_DE` should be replaced by your language. Next, log out, log back in again, and verify your Euro key is working. By now most console applications should respond to the Euro key. Extra configuration steps for special programs like `pine` might still be necessary however.

**Note:** An alternative to modifying `.login` and `.bash_profile` is to set the environment variables through the `login.conf(5)` mechanism. This approach has the advantage of assigning login classes to certain users (e.g. French users, Italian users, etc) *in one place*.

## 4 Modifying X11

Modify `/etc/X11/xorg.conf` (`/etc/X11/XF86Config` if you're running XFree86™) in the following manner:

```
Option "XkbLayout" "de(euro)"
```

Again, replace `de` with your language. By now, the keyboard should be set up correctly. As in the console section, the correct font must be chosen. For **KDE**, go to the **KDE control center** -> Personalization -> Country & Language -> Charset and change it to ISO8859-15. Similar steps apply to **kmail** and other applications.

Another good idea is to modify your `fonts.alias` files. Notably the `fixed` font should be changed to the right character set: The author's `/usr/X11R6/lib/X11/fonts/misc/fonts.alias` looks like this:

```
! $Xorg: fonts.alias,v 1.3 2000/08/21 16:42:31 coskrey Exp $
fixed -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-15
variable --helvetica-bold-r-normal-*--120-*--*--iso8859-15
(...)
```

As in the console sections, special applications still have ISO8859-1 fonts configured in their respective `xrdb(1)` databases. One notable example is **xterm**. As a general rule of thumb it suffices to change the corresponding configuration file in `/usr/X11R6/lib/X11/app-defaults` and add the correct font. Let us demonstrate this with **xterm**.

```
# cd /usr/X11R6/lib/X11/app-defaults/
# vi XTerm
```

Add the following line to the beginning of the file:

```
*font: -misc-fixed-medium-r-normal-*--120-*--c--iso8859-15
```

Finally, restart X and make sure, fonts can be displayed by executing the above `awk` script. All major applications should respect the keyboard mapping and the font settings.

## 5 Open problems

Of course, the author would like to receive feedback. In addition, at least let me know if you have fixes for one of these open problems:

- Describe alternative way of setting up **XFree86**: `x11/xkeycaps`
- Settings in **GNOME**
- Settings in **XFCE**
- Settings for **(X)Emacs**
- Describe UTF-8
- Describe **libiconv** as a effective way to convert between ISO8859-15 and UTF-{8,16} from within applications