Debian Euro HOWTO

Javier Fernández-Sanguino Peña <jfs@computer.org>

version 0.91, 25 February, 2002.

Copyright Notice

Contents

1 Introduction **1**

1.1 Why euro support? 1

1.2 What is the euro symbol? 1

1.3 Why all this fuss for just one character? 2

1.4 Standards 2

1.5 Is Debian euro-ready? 3

2 Automatic configuration **5**

2.1 The language-env package 5

2.2 The euro-support package 5

2.2.1 The euro-test program 6

2.3 The user-euro-XXX packages 6

3 Configuring euro support **7**

3.1 Initial considerations 7

3.2 Localisation issues 7

3.2.1 Locales in Debian woody 9

3.2.2 Locales in Debian 2.2 9

3.3 Configuring the Console 10

3.3.1 Configuring the console keyboard 10

3.3.2 How the keyboard is loaded in Debian 11

3.3.3 Configuring the console fonts 11

3.4 Configuring the X environment 12

3.4.1 Keyboard configuration 12

3.4.2 Font configuration 14

4 Euro support in applications **17**
4.1 Why talk about applications? . . . 17
4.2 Applications with known euro support . . . 17
4.2.1 XTerm and its derivatives . . . 18
4.2.2 GNOME Terminal . . . 18
4.2.3 RXVT and its derivatives . . . 18
4.2.4 Eterm . . . 19
4.2.5 gVim . . . 19
4.2.6 Emacs, XEmacs . . . 19
4.2.7 GNOME and GTK+ . . . 20
4.2.8 KDE . . . 20
4.2.9 Apache . . . 21
4.2.10 Mutt . . . 21
4.2.11 LaTeX . . . 21
4.2.12 Kword . . . 22
4.2.13 LyX . . . 22
4.2.14 groff (nroff, troff, grotty) . . . 22
4.3 Applications that do not support the euro character . . . 22

5 Frequently Asked Questions **25**
5.1 I see a strange character instead of the euro . . . 25
5.2 The euro character gets lost when switching from X to console . . . 25
5.3 How do I see if my keyboard is properly configured? . . . 25
5.4 How do I see if I can represent euros properly? . . . 26
5.5 I'm using framebuffer, can I represent euros on console? . . . 26
5.6 What is the longterm solution for this issue? . . . 26

6 About this document **27**
6.1 Why this document? . . . 27
6.2 References . . . 27
6.3 Changelog/History . . . 28
6.4 Acknowledgements . . . 29

A File definitions for LaTeX **31**

A.1 Latin9.def 31

A.2 latin10.def 34

Chapter 1

Introduction

1.1 Why euro support?

As of January 1st 2002, twelve European Union countries, and several others, are starting to use the **euro** as the only official currency. Thus, all the prices will be shown in euros and all the monetary transactions will use it. Euro is expected to become a common currency throughout Europe and even some other continents.

Computers, of course, need to correctly represent the euro in order for users to make their own documents (invoices, spreadsheets or whatever) using this currency, and read others' documents that use it. As the European Comission guidelines (`http://europa.eu.int/euro/`) state, operating systems need to be ready to represent this character.

The euro is a currency but also the name of a symbol. Since the Unicode Standard 2.1 version (dated 1998) the EURO SIGN is added, so it's also an Unicode character that can be represented (interpreted) with different glyphs (different fonts can change height or width).

1.2 What is the euro symbol?

The euro symbol resembles the letter "e" – it looks like a 'c' with two parallel horizontal lines that go through the middle of it. Some of the people will understand it better if we say it looks like the Quake symbol rotated 90 degrees clockwise :-)

The symbol is inspired by the Greek letter epsilon and also denotes the first letter of the word "Europe". The two parallel lines are meant to refer to the stability inside the euro area.

The official (ISO-compliant) abbreviature for the euro is EUR and can be used as representation for the currency.

1.3 Why all this fuss for just one character?

The problem is that changing the character involve a change in the font map used by the user. The font map is the list of character representations used by the system. Currently, most users in the euro-zone use the latin1 font map. The font map, however, is limited to 256 characters. The euro character is introduced by removing another character from the font map and calling this replacement a new font map. Latin9 (ISO-8859-15 or codepage 924 for IBM, usually shortened to latin0) replaces Latin1 (ISO-8859-1), and Latin10 (ISO-8859-16) replaces Latin2 (however Czech is not fully covered in Latin10 so it's not a full replacement, and it does cover Romanian which Latin2 didn't). Keep in mind that the font map is limited to 256 characters (see `charsets(7)`)

Latin9 differs from Latin1 in eight positions: A4, A6, A8, B4, B8, BC, BD, BE where Euro, Scaron, scaron, Zcaron, zcaron, OEligature, oeligature and Ydiaeresis, are defined.

Of course, users already using UTF-8 fonts are not affected by this problem since unicode is a superset of all ISO 8859 sets plus the characters required to represent practically all known languages (see `unicode(7)`). However, Unicode support is currently far from complete. For more information read Unicode in X-Windows (http://www.cs.usyd.edu.au/~matty/).

1.4 Standards

The euro definition is part of some ISO (http://www.isoc.org/) standards:

- The EUR currency code (numeric code 978) is a part of the ISO-4217 standard.
- For 8-bit systems, the ISO/IEC defines the new charset 8859-15 (also known as latin 9) and 8859-16 (also known as latin 10) where the euro symbol replaces the international currency symbol in the 164 position (hex A4)
- The Unicode (http://www.unicode.org/) system uses a 16 bit symbol, the euro has been assigned 8364 (hex 20AC) as part of ISO/IEC-10646-1
- The screen representation (font) for the euro is part of the ISO/IEC-10036 standard. The glyph is the real representation (bit pattern) of the character. The euro is assigned the 8059 glyph.
- The World Wide Web Consortium uses € as the representation for the symbol in HTML 4.0 (See http://www.w3.org/TR/html4/sgml/entities.html#h-24.4.1)

The European Comission has published in its official recommendations (http://europa.eu.int/euro/) both short term and long term fixes for the euro character. The short term solution is fixed by having keyboards input the euro character through the AltGr+E combination (AltGr is the 'Alt' key to the right of the spacebar), the long term solution considers using a new key for the euro character. Most keyboard manufacturers have only implemented the short-term solution by including a euro representation under the 'E' key. The operating system must take this input and convert it into the euro symbol internally. However:

- The international US keyboard and the greek latin use AltGr+5 and Ctrl+Alt+5 for euro representation.
- The greek keyboard uses AltGr+? for euro representation.
- The english and irish keyboards use the combinations AltGr+4 and Ctrl+Alt+4
- The hungarian and polish keyboards use the AltGr+U combination.

In this document, however, most examples will assume that the euro symbol will be generated by typing AltGr+E (the `euro-test` program assumes this too).

1.5 Is Debian euro-ready?

The Debian operating system can be properly configured to show the euro character, both in console and in the X windowing system since the 2000 release Debian 2.2 (aka 'potato'). However, many users have not properly configured that support since then, and there are some caveats in configuration that have not been completely fixed until Debian woody (future 3.0).

In any case, some problems might arise from programs defaulting to ISO-8859-1 or even ASCII-US, and some even can't be configured properly to use ISO-8859-15 (bug reports through the Bug Tracking System (http://bugs.debian.org/) should/will be filed against them).

Chapter 2

Automatic configuration

2.1 The language-env package

This package is an attempt to setup the user's environment locale properly by changing the user's `~/.bash_profile`, `~/.Xresources` and other user's configuration files. The included program guides the user in the definition of the locale by asking for his country and wether he desires (or not) euro support. Those tasks that need to be taken aside (since the program runs as an ordinary user it cannot change system-wide settings) are presented to the user so he can (as superuser) make the appropiate changes.

2.2 The euro-support package

The euro-support package is an attempt to provide appropiate *system-wide* configuration to represent the euro character easily in the Debian GNU/Linux operating system. This package provides this configuration in two ways:

- Depending on the appropiate packages needed in order to have euro support.
- Configuring the system upon installation in order to provide euro support.

NOTE: The second part is not yet built-in `euro-support`.

There are currently three packages:

euro-support includes this document as well as the `euro-test` program

euro-support-console includes the dependencies for providing euro icon in console environment

euro-support-X includes the dependencies for the X windows environment

2.2.1 The euro-test program

There is a test program called `euro-test` in the `euro-support` package program provided in order for the user to see if the euro symbol is properly represented in his system. This program has been written in order to test if the system configuration is appropiate and works both for the text console and the X graphic environment. Although developed for the Debian GNU/Linux system it can be used in other GNU/Linux operating systems in order to test for the euro support.

The goal of the program is to determine if/why the user's system provides the euro symbol. The information gathered could be used to manually configure the system properly together with this document.

2.3 The user-euro-XXX packages

There are also some packages (currently there is only a spanish version: the `user-euro-es` package) that will fully customize the system's (not the user's) enviroment in order to provide euro support. This packages will (through the execution of a given command) modify the system's configuration files (as explained below) and take the necessary steps to configure the environment.

Administrators in a haste are encouraged to use this packages and run the aforementioned scripts (`eurocastellanizar` for `user-euro-es`). However, users are encouraged to read through this document in order to be forewarned of all the issues (some of which cannot be tackled automatically).

This package is an attempt to setup the user's environment locale

Chapter 3

Configuring euro support

3.1 Initial considerations

Configuring euro support in a system involves two steps, which should be pretty simple:

- tell the system you want to use euros (aka. localisation)
- have a font available that represents euros.
- provide a mechanism for the keyboard to send the euro character.

The recommended mechanism (http://europa.eu.int/euro/html/dossiers/00124/00124-en.pdf) in the euro zone for the euro symbol generation on modern keyboards is the AltGr+e combination for the euro currency and AltGr+c for the cent. Note: Laptop users might not have AltGr.

This document will approach the euro configuration in Debian taking a look, first, towards localisation issues, and how localisation can be properly configured in Debian. Afterwards, it will go into how to configure the console (virtual terminals) available in all Debian GNU/Linux systems. Many users might want to skip this section and go directly to the next section, which discusses the proper configuration of the graphic environment (X windows) for euro support.

Even if it might sometimes confuse the reader, the different configuration issues are discussed both for the latest Debian release (as of this writing Debian 2.2 codename 'potato') and the next Debian release (Debian 3.0 codename 'woody'). Future versions of this document might separate the information regarding different Debian versions (when it really gets messy :)

3.2 Localisation issues

Programs use the localisation environment in order to know both the language and the charset being used. Currently there is no separation, unless you are using UTF-8 from locale and representation. Environment locales use both the language for example:

```
es_ES.ISO-8859-1
en_US.utf
....
```

Locale definitions are stored in Debian at `/etc/locale.alias` for the libc library and `/usr/X11R6/lib/X11/locale/locale.alias`. In order to indicate which charset you are using, you need to set your LANG environment variable. One of the ways of doing this is changing `/etc/enviroment` or `/etc/profile`.

However, there is a problem due to the different representation of locales in XFree86 (Xlib) and glibc (one uses ISO8859 and the other ISO-8859, note the dash). Thus, setting the locale to XX_XX.ISO-8859-15 in `/etc/environment` is doomed to cause problems in the X environment (might not be recognised or generate errors in applications).

Since Debian woody, the XX_XX@euro alias is provided in the `/usr/X11R6/lib/X11/locale/locale.alias` and in the `/etc/locale.alias`, users that wish to setup their locale environment should use this abbreviation instead of the previous XX_XX@ISO-...

You can see your current environment running `locale`, and your current map character using `locale charmap`. In order to change your locale edit `/etc/environment` or `/etc/profile` and add (the example is for Spanish, change as needed):

```
export LANG=es_ES@euro
export LC_ALL=es_ES@euro
```

Note: You *must* use the @euro part. Otherwise, if you just use `es_ES` the locale definition will be that of the ISO-8859-1 charset.

Please, note that the LANG and LC_ALL definitions should be similar (or even better the same). Otherwise libraries might warn against incompatible locale. You will see warnings if, for example, you set:

```
export LANG=spanish
export LC_ALL=es_ES@euro
```

since 'spanish' is aliased to es_ES.ISO-8859-1 and es_ES@euro is alised to es_ES.ISO-8859-15. Be careful with this issue since some programs do not check the `locale.aliases` file and use hardcoded value for them. It is not recommended to change the `locale.alias` to have, for example, 'spanish' aliased to es_ES.ISO-8859-15 since programs that use these method will not properly work. If your favorite software does not work with the 'XX_XX@euro' please file abug against it.

You can check all the available locales by running `locale -a`

For programs that give too many warnings much because of localisation/representation issues with stuff similar to

```
Gdk-WARNING **: locale not supported by C library
```

or

```
Warning: locale not supported by C library, locale unchanged
```

use `LANG=XX_XX.ISO-8859-1 program&` to run them and avoid the errors/warnings. In any case, check that you have properly set your locale and, if so, file a bug against the offending package.

In order for your X environment to work there should be a definition of the iso8859-15 definition at `/usr/X11R6/lib/X11/locale`. If you do not have properly configured locales, X might warn about this (before it runs any program). Please check your `~/.xsession-errors` file since this might be happening without the user noticing in modern desktop enviroments (all errors are directed there and not to the screen). A sample warning would be:

```
Warning: locale not supported by Xlib, locale set to C
```

3.2.1 Locales in Debian woody

In order for your locales to work in woody (candidate for Debian 3.0) the administrator has to configure properly the system's localization. The easiest way to make this modifications is to call `dpkg-reconfigure -plow locales`, and choose your locale in the version with an @euro appended.

You can also edit `/etc/locale.gen` directly, inserting there the euro locales definitions and run `locale-gen`. If your locales are already defined there but appended with a '#' sign, uncomment (remove that sign) the lines needed for the users' locale. Mainly the XX_XX.ISO-8859-15 lines and the XX_XX@euro lines and run `locale-gen`. This should create `/usr/lib/locale/XX_XX@euro`.

3.2.2 Locales in Debian 2.2

You cannot edit `/etc/locale.gen` if you are using Debian 2.2 since it does not exist. But, in any case, you do not need to do it since all the locales are included in Debian as default. However, Debian 2.2 `locales` (version 2.1.3-13) does not provide the euro definition. If an upgrade to woody locale is not possible (this upgrade will most probably change the libc version), you will need to make the changes manually.

In Debian 2.2, users need to change some files (these changes are unnecessary in woody since they already have been made). Supposing an Spanish environment (es_ES locale), the change for `/etc/locale.alias` is:

```
es_ES@euro es_ES.ISO-8859-15
```

and for `/usr/X11R6/lib/X11/locale/locale.alias` is:

```
es_ES@euro es_ES.ISO8859-15
```

After doing this you have to *generate* your locales You can do so running `localedef` manually. Charmapfiles are found in `/usr/share/i18n/charmaps`, inputfiles can be found in `/usr/share/i18n/locales`. In order to create, for instance, a en_US.ISO-8859-15 locale, you'd do this:

```
$ localedef --force -i /usr/share/i18n/locales/en_US \
           -f /usr/share/i18n/charmaps/ISO-8859-15 \
            en_US.ISO-8859-15
```

You can usually also use a simplified version since `localedef` should be able to find the charpmaps and locales on its own:

```
$ localedef --force -v -i en_US -f ISO-8859-15 en_US.ISO-8859-15
```

The directory created is named `/usr/share/locale/en_US.iso885915`. You also have to create the en_US.ISO-8859-15 directory, do so by symlinking it to the previous one.

```
$ cd /usr/share/locale
$ ln -s en_US.ISO-8859-15 en_US.iso885915
```

3.3 Configuring the Console

Configuring the text console in order to have the euro symbol ready is simple. The following will work in a Debian GNU/Linux system:

```
$ loadkeys euro.inc
$ consolechars -f lat0-16.psf
```

However, this changes are lost after reboot. In order to make them permanent, some changes need to be done to the configuration files of Debian packages.

3.3.1 Configuring the console keyboard

Changing the key mappings involves changing `/etc/console-tools/default.map.gz` or `/etc/kbd/default.map.gz`. Usually, this keymap is changed by using `kbdconfig` (if you are not using console-tools see below for the different console keyboard schemes in Debian), since it is loaded before any network filesystem is mounted, you cannot change it directly (for example by adding `include euro` there) as it might not be able to load extensions (since they are located at `/usr/share/keymaps`). In order to have euro support, if your keymap currently does not have one you will have to modify an existing one. Here is an example on how to change the Spanish keymap to support euro characters (kbdconfig will ask some questions, only the answers are shown below):

```
$ cd /usr/share/keymaps/i386/qwerty
$ cp es.kmap.gz es-euro.kmap.gz
$ gzip -d es-euro.kmap.gz
$ echo "include \"euro\"" >>es-euro.kmap
$ echo "altgr keycode  46 = cent" >>es-euro.kmap
$ gzip es-euro.kmap
$ kbdconfig
-----ANSWERS:
n
i386
qwerty
es-euro
y
y
$ zgrep "keycode  18" /etc/console-tools/default.kmap.gz
keycode  18 = +e                +E                 currency          Control_e
    Control_e        Meta_e           Meta_E            Meta_Control_e
```

FIXME: A bug should be opened against console-data in order to include this by default in all the euro-zone language maps.

3.3.2 How the keyboard is loaded in Debian

The `console-common` in Debian woody installs `/etc/init.d/keymap.sh` which runs `loadkeys` of console-tools (using `/etc/console-tools/default.map.gz`) or kbd (using `/etc/console/boottime.kmap.gz`. This file has been generated using `/usr/sbin/install-keymap`.

When a system boots up it runs `/etc/rcS.d/S05keymap.sh` (for kbd) or `/etc/rcS.d/S05keymap-lct.sh` (for console-tools) which installs the appropiate keymap. Both scripts will make either `/etc/init.d/keymap.sh` or `/etc/init.d/S05keymap-lct.sh` being called with the 'start' option.

3.3.3 Configuring the console fonts

The easy way to configure a console in a Debian system is to install the `fonty` provided since Debian 2.2 and tell it to use iso15 fonts (fonty currently does not support iso-8859-16 yet, however). These fonts will include the euro character and the package will configure properly the configuration files needed. You will need, however, to configure the keyboard mappings, as describe previously.

If you do not want (or like) the `fonty` font you can configure the console fonts yourself. In order for the fonts to be loaded automatically when the system is started, you (as superuser) have to change some of the configuration files.

If the system is using the `console-tools` package, then the `/etc/console-tools/config` file will be present. In order to load the appropiate font you need to include the line:

```
SCREEN_FONT=lat0-16
```

```
APP_CHARSET_MAP=iso15
```

If you are using the `kbd` package you will need to edit the `/etc/kbd/config` adding, for example:

```
CONSOLE_FONT=lat0-16.psfu.gz
# CONSOLE_MAP=iso15
CONSOLE_MAP=8859-15_to_uni
```

Of course, the `lat0-16` font is not the only one available, since the console-data package version 1999.08.29-11.1, there are quite a number of latin9 (aka. latin0) fonts available at `/usr/share/consolefonts`. Some other people prefer the `lat0-sun16` font, it's just a matter of taste.

If you are using the `svgatextmode` package (obsoleted by the framebuffer) you might configure the console font also by changing the `/etc/TextConfig` file.

NOTE: If you are using the framebuffer device for console text, read the note regarding console-tools configuration in the `/etc/console-tools/config` file. Basically, the global definition will only work for the first virtual terminal, you need to explicitly state the font for the other terminals like this:

```
SCREEN_FONT_vc1=lat0-16
SCREEN_FONT_vc2=lat0-16
SCREEN_FONT_vc3=lat0-16
....
```

3.4 Configuring the X environment

For the graphic environment to represent euros you also need to have:

- fonts which include the euro symbol
- proper configuration of the keyboard

XFree86 includes euro-ready fonts since version 3.3.3.1, like, for example 7x13euro.bdf or 7x13euroB.bdf. However, the user might not have them installed since ISO-8859-15 fonts are not part of the default (basic) XFree86 installation.

3.4.1 Keyboard configuration

The keyboard configuration here is easier that in console. If you are using the Xkb extensions in the X server (the default since 3.3.5) you can change it using the `XkbLayout` option in the `InputDevice` section of the X configuration file. That file is located in `/etc/X11/XF86Config` for XFree86 3.3.6 and `/etc/X11/XF86Config-4` for XFree86 4.1.

There are several ways you can add support for the euro definition:

- by changing the xkb extensions included in the X server.
- by loading a new modmap (using `xmodmap`)

If you change the modmap with `xmodmap` and it works, it's a temporary fix, after restarting the server the changes will get lost. In order to have this definition loaded on startup you have to add the following definition to your `Xmodmap` by doing:

```
$ xmodmap -pke > ~/.xmodmap
```

NOTE: Xmodmap is obsolete, please use the Xkb extensions. Future releases of XFree86 might not even support user's defined xmodmaps.

NOTE: the locales setting must be done before Xsession starts, so make sure that your configuration is loaded properly before executing `startx` (if you are using an XDM application such as `gdm` you can choose the settings from it).

NOTE: The symbol used for the Euro sign varies from Xfree 3.3.6 (standard X server in Debian 2.2) to Xfree 4.1 (standard X server in Debian 3.0). In Xfree 3.3.6 you will have to use "currency" instead of "EuroSign" as the symbol definition. This has changed in Xfree 4.1 which has built-in support for the EuroSign character. If you use currency that same symbol in Xfree 4.1, due to UTF support, it might not represent properly the Euro character.

Xfree in Debian 3.0

If you are using Debian woody (3.0), which includes XFree86 4.1, you only need to have it properly configured to a language in the euro zone, this is done in the configuration file `/etc/X11/XF86Config-4`. Inputting the euro symbol should work correctly (if not, please file a bug).

You can also make some changes to provide euro support if your system does not work properly. You can change the Xkb extensions (at the files in `/usr/X11R6/lib/X11/xkb/symbols`) to include Euro support with something along the lines of:

```
    key <AD03> {         [                  e,     E                ],
                        [         EuroSign,       cent         ]        };
```

This, however, should already be provided in all the files of languages from the Euro zone (es, de...)

Or, you can change your Xmodmap definition with:

```
$ echo "keycode 0x1A = e  E EuroSign cent" | xmodmap -
```

If you are not using Xkb extensions you will have to change the definitions available at the `/usr/share/xmodmap/` directory (the file name is xmodmap.XXX with XXX the one appropiate to your language).

Since Xkb extensions is the default behavior the Xmodmap files provided by Xfree86 4.1 might not be updated properly. Check that the line related to the `keycode 26` (E character) looks like the following line:

```
keycode  26 = e E EuroSign cent
```

Xfree in Debian 2.2

If you are using versions equal to, or prior to Debian potato (2.2), the language definition might not be properly defined. If an upgrade to a newer version of XFree86 is not possible, you can use any of the previous approaches to add support for the euro definition:

You can change the Xkb extensions (`/usr/X11R6/lib/X11/xkb/symbols`) to include currency support with something along the lines of

```
key <AD03> {        [                 e,    E               ],
                   [        currency,      cent        ]       };
```

Or, you can change your Xmodmap definition with:

```
$ echo "keycode 0x1A = e  E currency cent" | xmodmap -
```

You can also use the definitions available at the `/usr/share/xmodmap/` directory (change the file appropiate to your language) if you are not using Xkb extensions. Check that the line related to the `keycode 26` (E character) looks like the following line:

```
keycode  26 = e E currency cent
```

Debian 2.2 Xmodmap definitions of the languages in the euro zone contain the euro character (using the currency symbol), so you can use the following `xmodmap /usr/share/xmodmap/xmodmap.XX` (XX is the language you want to use).

3.4.2 Font configuration

In order to represent the euro symbol in X you need to have appropiate fonts to represent the ISO-8859-15 encoding (if not using UTF). You can see which fonts could be used for this by doing:

```
$ xlsfonts | grep `iso8859-15$`
```

In a default Debian installation, the only font that provides this encoding is the fixed (misc) font, which might be proper for an xterm but might not show up nice in many other applications. However, Debian woody provides the transcoded package fonts which include other ISO-8859-15 fonts. The packages are `xfonts-base-transcoded`, `xfonts-75dpi-transcoded`, `xfonts-100dpi-transcoded`, and you can install them by running `apt-get install 'xfonts.*transcoded'`.

Some people might not want to install these packages because they are quite large, around 15 Mb. There is a hacked version available, with all non-15 fonts removed and it is less than 3Mb in size. You can find it at http://sites.inka.de/W1752/reduced-xfonts/. Note that this is not part of the standard Debian distribution.

(FIXME if users really need this the transcoded packages should be broken in two.)

Please note that if you are using Debian 2.2 you cannot install these packages without a system upgrade since they depend on XFree86 4.

The following packages also provided ISO-8859-15 fonts: xfonts-jmk, xfonts-arphic-bsmi00lp, xfonts-arphic-gbsn00lp, xfonts-arphic-bkai00mp and xfonts-arphic-gkai00mp.

The `xfonts-cyrillic` package provides also (in the XFree86 3.3.6) some ISO-8859-15 fonts.

FIXME: Verify this last statement (seems to due to the data sent to me from users of the euro-test program)

NOTE: Be sure you do not have a default definition for another font in your `.Xdefaults` file. For example with

```
*VT100.font: -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-16
```

If you cannot find any suitable fonts, you can use the ISO10646-1 fonts which are a superset of the different ISO8859-X fonts and are also provided in Debian. Markus Kuhn provides a script called `ucs2any.pl` which states:

```
This Perl script allows you to generate from an ISO10646-1 encoded
BDF font other BDF fonts in any possible encoding. This way, you can
derive from a single ISO10646-1 master font a whole set of 8-bit
fonts in all ISO 8859 and various other encodings.
```

There are Unicode fonts for X-windows available at http://www.cl.cam.ac.uk/~mgk25/ucs-fonts.html. In ucs-fonts.tar.gz and ucs-fonts-75dpi100dpi.tar.gz there are Times, Helvetica, Lucida, Utopia, New Century, Schoolbook, Courier. . . fonts that can be used. You can also use the `yudit` program to change these fonts.

There are Debian packages available to install the Kuhn's unicode fonts. FIXME: What are the names for them?

There are other useful XFree86 3.3.6 ISO-8859-15 fonts at http://linux.ee/~ville/vh-fonts-0.99.2.tar.gz, even if provided mostly for Estonian users they are correct fonts and include modified helvetica, times, courier and one font for terminal, and at http://httpd.chello.nl/fonturi/index-en.html (FIXME: not tested).

Chapter 4

Euro support in applications

FIXME: Text needed

4.1 Why talk about applications?

Even if you have the euro symbol working correctly (you can input it from the keyboard and see it on your screen) you still need to see if your applications work properly.

Some problems here arise in graphic applications which might use their own fonts and might, therefore, be unable to represent the euro symbol (even if you input it correctly) because they do not have an internal representation for it.

Hint: you could make your life easier if you run a font selector program like `gtkfontsel` (`gtkfontsel` package) and you set the mask of visible fonts to ISO-8859-15.

However, the encoding made by the program for texts and data that it uses is also an important issue. If it's unable to represent internally the charset used (be it ISO-8859-15 or Unicode) support for euro might not fully work. So, one thing is using ISO-8859-15 for menubars, program messages et al, and a different one is using ISO-8859-15 for data used by the program (text, information on databases…).

4.2 Applications with known euro support

The following applications are known to have support for the euro character:

- Terminals: XTerm, Rxvt and their derivatives, GNOME Terminal, Eterm.
- Editors: gVim, Emacs, XEmacs, Kword, Mcedit, kedit, kwite, kwrite. Note: Emacs21 (in woody) does support latin9 documents.
- Programs using GTK+/GLib
- Desktop environments: GNOME and KDE.

- Konqueror, Mozilla
- Mutt
- Apache
- LaTeX
- groff (nroff, troff, grotty)
- a2ps
- Staroffice 5.0 (not provided in Debian but a FAQ) it seems to use it own fonts, so you cannot use the locally installed fonts, however it seems the 'Conga' font does include the euro-character.
- LyX (1.1.6fix4 and above)

4.2.1 XTerm and its derivatives

If the euro character is not represented in your X terminal emulator, you can change the default font by changing either the users' configuration files (`.Xdefaults` or `.Xresources`) or the system-wide configuration at `/etc/X11/app-defaults/XTerm`:

```
*font:     -misc-fixed-medium-r-normal-*-*-120-*-*-c-*-iso8859-15
*VT100*font2:          -misc-fixed-medium-r-normal-*-*-70-*-*-c-*-iso8859-15
*VT100*font3:          -misc-fixed-medium-r-normal-*-*-100-*-*-c-*-iso8859-15
*VT100*font4:          -misc-fixed-medium-r-normal-*-*-120-*-*-c-*-iso8859-15
*VT100*font5:          -misc-fixed-medium-r-normal-*-*-140-*-*-c-*-iso8859-15
*VT100*font6:          -misc-fixed-medium-r-normal-*-*-200-*-*-c-*-iso8859-15
```

In `/etc/X11/app-defaults/XTerm`, make sure you replace old lines with these options.

After editing a `.Xdefaults` file, reload it with `xrdb -merge ~/.Xdefaults`. (Similarly for `.Xresources`.)

Note that the derivative programs also use the `font` resource to set the default font, so the procedure is analogous.

4.2.2 GNOME Terminal

You can configure the Gnome terminal to use a euro-ready font by changing the font in the Configuration->Preferences menu.

4.2.3 RXVT and its derivatives

Rxvt and the programs derived from it (e.g. Aterm, Wterm) also use the `font` resource from `~/.Xresources` or `~/.Xdefaults`, see above for how it's done in XTerm.

4.2.4 Eterm

Change the user configuration (`~/.Eterm/user.cfg`) with:

```
<Eterm-0.9.1>
begin attributes
    scrollbar_type motif
    scrollbar_width 10
    font default 2
    font proportional 0
    font 0 -b&h-lucidatypewriter-medium-r-normal-*-*-80-*-*-m-*-iso8859-15
    font 1 -b&h-lucidatypewriter-medium-r-normal-*-*-100-*-*-m-*-iso8859-15
    font 2 -b&h-lucidatypewriter-medium-r-normal-*-*-120-*-*-m-*-iso8859-15
    font 3 -b&h-lucidatypewriter-medium-r-normal-*-*-140-*-*-m-*-iso8859-15
    font 4 -b&h-lucidatypewriter-medium-r-normal-*-*-180-*-*-m-*-iso8859-15
end attributes
```

4.2.5 gVim

`~/.vimrc` or (systemwide) `/etc/vim/vimrc`:

```
set guifont=-b&h-lucidatypewriter-medium-r-normal-*-*-140-*-*-m-*-iso8859-15
set encoding=iso-8859-15
```

4.2.6 Emacs, XEmacs

GNU Emacs 21 and XEmacs 21 provide support for latin9. However, in versions previous to Emacs21, (Mule) does not show an option to save documents using latin9 (latin0) or ISO-8859-15.

You might need, however, to change the font that Emacs runs with in order to present the Euro character in X windows. To do so, run emacs with a euro font with the -fn switch or configure it to always use a given font by editing `~/.Xresources`:

```
Emacs.default.attributeFont: -*-Lucidatypewriter-Medium-R-*-*-*-110-*-*-*-*-iso
8859-15
```

You can also try adding the following lines in `.emacs`, or `.xemacs/init.el` for XEmacs:

```
(set-face-font
 'default '"-*-courier-medium-r-*-*-*-120-*-*-*-*-iso8859-15")
```

4.2.7 GNOME and GTK+

Gnome applications do mostly support another charset without problems. Depending on your local configuration, you probably would have to change the default font. Please start (in Gnome) the Control Center and choose a font with `iso8859-15` encoding. If you don't have gnomecc installed, you could make this setting manually, creating an customised gtkrc file in your home directory (`~/.gtkrc`) and addind the lines show below.

Better yet, change the systemwide GTK+ settings in `/etc/gtk/gtkrc`. You can do this in two different ways:

- Linking (or copying) `/etc/gtk/gtkrc.iso-8859-15` to `/etc/gtk/gtkrc` (recommended). In Debian this file contains:

  ```
  style "gtk-default-iso-8859-15" {
         fontset = "-*-helvetica-medium-r-normal--12-*-*-*-*-*-iso8859-1,\
                    -*-arial-medium-r-normal--12-*-*-*-*-*-iso8859-1,\
                    -*-helvetica-medium-r-normal--12-*-*-*-*-*-iso8859-15,\
                    -*-arial-medium-r-normal--12-*-*-*-*-*-iso8859-15,*-r-*
  }
  class "GtkWidget" style "gtk-default-iso-8859-15"
  ```

- Adding the needed lines to `/etc/gtk/gtkrc` directly (discouraged but might be necessary sometimes)

Here are some sample lines you can add to the configuration file:

```
style "user-font"
{
  font="-monotype-arial-medium-r-normal-*-12-*-*-*-p-*-iso8859-15"
}
widget_class "*" style "user-font"
```

4.2.8 KDE

KDE euro support works as described at http://lists.debian.org/debian-kde/2001/debian-kde-200110/msg00423.html. You have to set up yor Xfree environment as described above. Users have reported even to have KDE's euro support working in Potato using custom XFree86 3.3.6 fonts (as described in 'Font configuration' on page 14.

Be careful when setting the locale and use the aliases defined in the X library since, as described at http://bugs.kde.org/db/32/32919-b.html, setting the charset as 'ISO-8859-15' will not work, it needs to be 'ISO8859-15'. This issue is further discussed at 'Localisation issues' on page 7.

Once this is done, you have to go to KDE's Control Center::Personalization::Country & Language. And set your Country name and "Charset: iso8859-15".

I first though, reading http://users.pandora.be/sim/euro/112/ that KDE didn't work with Euro characters. But you only have to configure it properly. You can see it for yourself (http://m3d.uib.es/~gallir/ext/tmp/euro.png). If it does not work for you check your charset and the fonts available.

However, there are know bugs due to the `localesconf` which does not set the KDE environment properly. You should take your time and read Bug 122533 (http://bugs.debian.org/122533). and Bug 130259 (http://bugs.debian.org/130259).

4.2.9 Apache

Apache

You should modify your webserver settings if you want to present some sites with a non-ISO8859-1 charset, unless you want your users to change their charset manually each time. Following settings for Apache (eg. put into an `.htaccess` file) tells the browsers the charset they have to use:

```
AddType         text/html;charset=ISO-8859-15   html
```

You can use the euro character directly in the documents, this information could be provided also in the HTML documents DTD. In any case you can use, the HTML 4.0 euro representation and not configure Apache.

4.2.10 Mutt

Works flawlessly adding to the muttrc file (user's or global):

```
set charset=iso-8859-15
set send_charset="us-ascii:iso-8859-15:iso-8859-1:utf-8"
```

4.2.11 LaTeX

There are several ways to introduce the euro character in LaTeX:

- With textcomp package and the \texteuro macro (TS1 fonts)
- With the marvosym package, using type1 fonts.
- With the eurosym package using metafont fonts.

Thus, you can use the marvosym package that is included in `tetex-base` (`/usr/share/texmf/tex/latex/misc/marvosym.sty`). This package includes some symbols, including the euro symbol, in different fonts (Times, Helvetica and Courier). Of course, you do not need to be able to input the euro character (or see it in X) since the LaTeX files will be translated into postscript files (no font needed for their viewing with `xpdf` or other postscript viewers). The include it in your documents with

```
\EUR
```

Debian woody has also the `tetex-eurosym` package which allows the euro representation too. You can use this package even if on a pure stable system to reproduce Euro symbols.

```
\texteuro
```

In order to represent the cent you need to use `textcomp.sty` which is provided in `tetex-base`.

A common problem is, however, not having an input encoding in order to include this characters directly. You can use, however the files provided at 'File definitions for LaTeX' on page 31, and place them under `/usr/share/texmf/tex/latex/base/` in order to do so.

FIXME: Wishlist bug against tetex-base so they get included.

4.2.12 Kword

Kword includes a document in the demos directory called `eurosign.kwd` which can be used to determine if fonts are properly installed. It is available at `/usr/share/doc/kword/examples/eurosign.kwd.gz`

Note: This file was available since Kword 1.1.1-5, see #132627 (`http://bugs.debian.org/132627`).

4.2.13 LyX

As of version 1.1.6fix4-2 LyX adds support for latin3, latin4 and latin9 encodings.

4.2.14 groff (nroff, troff, grotty)

It provides latin1, ascii8 and utf8 as devices. In order to generate manpages in latin0 it seems the `ascii8` device needs to be used.

FIXME Needs to be checked.

4.3 Applications that do not support the euro character

The following applications (and associated versions) have been reported *not* to work with the euro character:

- LyX 1.1.6fix3. ISO-8859-15 is not included in `/usr/share/lyx/encodings` and `/usr/share/lyx/languages` shows ISO8859-1 for euro-zone languages (for example, for Spanish). Problems with LyX are similar to LaTeX, there is a need for a new `inputenc`. Check, however 'File definitions for LaTeX' on page 31, you will need, in any case type1 fonts for LaTeX to be able to print the character properly (currently not provided).

- Xfig 3.2.3
- GnuPG, supports only ISO-8859-1, ISO-8859-2, koi8-r and utf-8 (see the –charset option in `gpg(1)`)
- SGML tools (`nsgml`, `sgml-tools` and `debiandoc-sgml`. Most tools will currently warn if you are using any @euro locale, the `nsgmls` has currently no support for the iso-8859-15 encoding.
- Perl is not euro friendly, Perl is used by quite a number of administrative scripts including Debconf.

Chapter 5

Frequently Asked Questions

5.1 I see a strange character instead of the euro

If you are seeing a character that seems to be a circle with four lines streching out of it (the international symbol for currency) and not the euro symbol then the font you are using does not properly represent euros but your keyboard is sending it properly. Please check your environment/applications in order to see that you are using ISO-8859-15 fonts and not ISO-8859-1.

5.2 The euro character gets lost when switching from X to console

FIX: Run (as superuser) `/etc/init.d/console-screen reload` (if the `console-tools` is installed), or run `setfont -u` (if the `kbd` is installed.

REASON: There are fonts with an unicode map in the .psf file and others that do not include it. If these last ones are used the Linux kernel unicode map resets and when you return from an X virtual terminal the map is garbled. The Keyboard and Console HOWTO (available at `/usr/share/doc/HOWTO/en-txt/Keyboard-and-Console-HOWTO.txt.gz` if you have the `doc-linux` package) elaborates a little bit on this.

5.3 How do I see if my keyboard is properly configured?

(Console terminal) You should see 'currency' when doing:

```
$ dumpkeys |grep -i currency
```

(X graphic environment) You should see 'currency' when doing:

```
$ xmodmap -pke | grep -i EuroSign
```

5.4 How do I see if I can represent euros properly?

If using ISO-8859-15:

```
$ printf "\xa4\n"
```

If using UTF-8:

```
$ printf "\xe2\x82\xac\n"
```

Of course, you can also see if the characters euro and cent are represented correctly by taking a look at a document that includes them. `euro-support` includes a representation of these in `/usr/share/doc/euro-support/examples/characters`, just `cat` the file and see if they get printed to the screen correctly.

5.5 I'm using framebuffer, can I represent euros on console?

Yes you can, (from the FrameBuffer-HOWTO (http://www.linuxdoc.org/HOWTO/Framebuffer-HOWTO.html#toc15)) you just need to use the kbd package version 0.99 or later.

5.6 What is the longterm solution for this issue?

Move towards UTF-8 encoding and separation of localisation and representation (no more XX_XX.ISO-8859-X).

Chapter 6

About this document

6.1 Why this document?

I (Javier) have been thinking for a time on how to provide automatic configuration of the euro issue on the lines of the automatic stuff done by `castellanizar` in `user-es`. After going through some information on the problem and reading some threads in debian related mailing lists like debian-i18n (http://lists.debian.org/debian-i18n/2001/debian-i18n-200109/msg00018.html), debian-devel (http://lists.debian.org/debian-devel/2001/debian-devel-200105/msg00696.html) and debian-user-spanish (http://lists.debian.org/debian-user-spanish/2001/debian-user-spanish-200105/msg00488.html), I decided to post a poll to test how euro support was amongst Spanish speakers (in debian-user-spanish (http://lists.debian.org/debian-user-spanish/2001/debian-user-spanish-200110/msg01265.html)). The results indicated that many people have not properly configured their systems even though Debian 2.2 (released more than a year ago) was ready for the euro problem.

This document is the first step towards writing an automated tool to configure the user system for full euro support. Even if this could be done when Debian users move to UTF-8 there is a need of a short term solution before that move comes about.

6.2 References

The following documents complement this one and are useful for the reader to learn more information regarding the euro, Internationalisation and Unicode:

- Euro-mini-HOWTO (http://www.linuxdoc.org/HOWTO/mini/Euro-Char-Support/).
- Introduction to I18n (http://www.debian.org/doc/devel-manuals#i18n) by Tomohiro Kubota, which introduces basic concepts of internationalization and is centered on displaying and inputting characters with different encodings (ASCII, ISO-8859, multybyte characters…)

- UTF-8 in Debian (http://melkor.dnp.fmph.uniba.sk/~garabik/debian-utf8/HOWTO/)
- Unicode Mutt (http://www.rano.org/mutt.html)
- Unicode in X-Windows (http://www.cs.usyd.edu.au/~matty/).

Some other (official) references:

- European Comission guidelines (http://europa.eu.int/euro/html/rubrique-defaut5.html?rubrique=177\delimiter"026B30D[lang]\delimiter"026B30D=5).
- Migration to Euro (http://europa.eu.int/euro/html/rubrique-defaut5.html?rubrique=177\delimiter"026B30D[lang]\delimiter"026B30D=5) document produced by the European Comission.

6.3 Changelog/History

List of changes done to this document.

- Changes in 0.91
  - Updated KDE information including pointers to two bug reports
  - Updated LyX info (it is not euro-compliant)
  - Added Perl to the list of software not euro-compliant (but?)
- Changes in 0.9
  - Added emacs 21 note
  - Removed repeated chapter (Why all this fuss. . . ? and Why this document?)
  - Added information on user-euro-es
  - Changed from currency to EuroSign (under discussion)
  - Major rewritting of Xfree section in order to distinguish xfree versions properly (currency and EuroSign symbol)
  - Rewriting of Locale section to separate woody and potato information
- Changes in 0.8
  - added note regarding tetex-eurosym on stable sent by Martin Schulze
  - fixed HTML representation, suggested by Bernd Eckenfels
  - added a note regarding the use of kbdconfig, suggested by Bernd Eckenfels
  - added an acknowledgment to Guylhem Aznar

  - fixed missing quote, sent by Cyrille Artho

- Changes in 0.7

  - proofreading and slight reorganization by Josip Rodin

- Changes in 0.6

  - fixed grammar errors with patch sent by Matt Kraai

- Changes in 0.5

  - Added contributions from Juan Rafael Fernández, Miguel Sanjuan, Aurelien Jano, Phillip Siegert, Tomohiro Kubota, Ionel Mugurel and Alexander Steinert.
  - Moved the location of the explanations regarding fonty.
  - Rewrote the presentacion of supported applications to improve readability.
  - Rewrote acknowledgments (I'm getting kind of sentimental :)

- Changes in 0.4

  - Added contributions submitted by Juan Rafael Fernández.
  - Fixed a few typos spotted by Carlos Valdivia.
  - Added more information on the euro-support package.

- Changes in 0.3

  - Added information regarding LaTeX provided by Juan Rafael Fernández
  - Added more information and fixes contributed by Eduard Bloch.

- Changes in 0.2

  - added more information regarding euro standards
  - added links to mailing list threads
  - fixed some typos

6.4 Acknowledgements

I would like to take the opportunity to thanks all the people have contributed (knowingly or not) to the information contained in this HOWTO, specifically:

- Juan Rafael Fernández, who wrote a first draft of a euro-howto in Spanish.
- Hue-Bond, who answered himself some FAQs in the debian-user-spanish mailing list.

- Jose Carlos Garcia Sogo, who showed up some very good insights on May 9th on the debian-users-spanish mailing list regarding this issue.
- Ionel Mugurel who did an extensive explanation on the euro issues on 14th September 2000 on the debian-i18n mailing list and provided me with the so-much-needed LaTeX input encodings.
- Guylhem Aznar the writer of the Euro Pack and the Euro FAQ (http://www.ibiblio.org/guylhem/programmes/EURO-2.tgz). Presented in a Linux Journal (http://www.linuxjournal.com/article.php?sid=3200) article in 1998.
- Eduard Bloch, the writer of the DE-Debian-HOWTO (http://channel.debian.de/faq/).
- Tomohiro Kubota who enlightened me on some of the problems of playing with the locale and its risks.
- Pablo de Vicente, from the Spanish KDE translation team who made efforts to prove me wrong when I said that KDE did not support the euro.
- Thomas Roessler, who contributed useful input regarding the locale section and made me separate it in two.

And of course, all other people that contributed bits of typos/corrections/suggestions of which these HOWTO is made of. If you, reader, have to thank someone is theirs for their effort and knowledge, I only put it together here.

Appendix A

File definitions for LaTeX

Ionel Mugurel has provided the following input encoding definition files for latin9 and latin10 (not yet included in Debian).

A.1 Latin9.def

```
%%
%% This is file `latin9.def',
%%
%% This is a new file.
%%
%% Copyright 2001
%%    Ionel Mugurel Ciobica
%%
%% Permision granted to copy, distribute and redistribute this file.
%%
%% Because of the euro symbol the tetex-eurosym package has to be
%% installed. Otherwise an alternative is made to "draw" the character
%% on place. Uncomment that line and comment the next one.
%%
%% Because of \textdegree and many other \text... commands,
%% you might want to use \usepackage{textcomp} in your document.
%%
%% \CharacterTable
%%  {Upper-case    \A\B\C\D\E\F\G\H\I\J\K\L\M\N\O\P\Q\R\S\T\U\V\W\X\Y\Z
%%   Lower-case    \a\b\c\d\e\f\g\h\i\j\k\l\m\n\o\p\q\r\s\t\u\v\w\x\y\z
%%   Digits        \0\1\2\3\4\5\6\7\8\9
%%   Exclamation   \!     Double quote  \"     Hash (number) \#
%%   Dollar        \$     Percent       \%     Ampersand     \&
%%   Acute accent  \'     Left paren    \(     Right paren   \)
%%   Asterisk      \*     Plus          \+     Comma         \,
```

```
%%   Minus          \-     Point          \.     Solidus        \/
%%   Colon          \:     Semicolon      \;     Less than      \<
%%   Equals         \=     Greater than   \>     Question mark  \?
%%   Commercial at  \@     Left bracket   \[     Backslash      \\
%%   Right bracket  \]     Circumflex     \^     Underscore     \_
%%   Grave accent   \`     Left brace     \{     Vertical bar   \|
%%   Right brace    \}     Tilde          \~}
 \ProvidesFile{latin9.def}
   [2001/10/07 v0.01 Input encoding file
                         (test version: still liable to change)]
\makeatletter
\DeclareInputText{160}{\nobreakspace}
\DeclareInputText{161}{\textexclamdown}
\DeclareInputText{162}{\textcent}
\DeclareInputText{163}{\pounds}
%\DeclareInputText{164}{{\sffamily C\makebox[0pt][l]{\kern-.70em\mbox{--}}\makebo
\DeclareInputText{164}{\euro}
\DeclareInputText{165}{\textyen}
\DeclareInputText{166}{\v S}
\DeclareInputText{167}{\S}
\DeclareInputText{168}{\v s}
\DeclareInputText{169}{\copyright}
\DeclareInputText{170}{\textordfeminine}
\DeclareInputText{171}{\guillemotleft}
\DeclareInputMath{172}{\lnot}
\DeclareInputText{173}{\-}
\DeclareInputText{174}{\textregistered}
\DeclareInputText{175}{\@tabacckludge={}}
\DeclareInputText{176}{\textdegree}
\DeclareInputMath{177}{\pm}
\DeclareInputMath{178}{^2}
\DeclareInputMath{179}{^3}
\DeclareInputText{180}{\v Z}
\DeclareInputMath{181}{\mu}
\DeclareInputText{182}{\P}
\DeclareInputText{183}{\textperiodcentered}
\DeclareInputText{184}{\v z}
\DeclareInputMath{185}{^1}
\DeclareInputText{186}{\textordmasculine}
\DeclareInputText{187}{\guillemotright}
\DeclareInputText{188}{\OE}
\DeclareInputText{189}{\oe}
\DeclareInputText{190}{\" Y}
\DeclareInputText{191}{\textquestiondown}
\DeclareInputText{192}{\@tabacckludge`A}
\DeclareInputText{193}{\@tabacckludge'A}
```

```
\DeclareInputText{194}{\^A}
\DeclareInputText{195}{\~A}
\DeclareInputText{196}{\"A}
\DeclareInputText{197}{\r A}
\DeclareInputText{198}{\AE}
\DeclareInputText{199}{\c C}
\DeclareInputText{200}{\@tabacckludge`E}
\DeclareInputText{201}{\@tabacckludge'E}
\DeclareInputText{202}{\^E}
\DeclareInputText{203}{\"E}
\DeclareInputText{204}{\@tabacckludge`I}
\DeclareInputText{205}{\@tabacckludge'I}
\DeclareInputText{206}{\^I}
\DeclareInputText{207}{\"I}
\DeclareInputText{208}{\DH}
\DeclareInputText{209}{\~N}
\DeclareInputText{210}{\@tabacckludge`O}
\DeclareInputText{211}{\@tabacckludge'O}
\DeclareInputText{212}{\^O}
\DeclareInputText{213}{\~O}
\DeclareInputText{214}{\"O}
\DeclareInputMath{215}{\times}
\DeclareInputText{216}{\O}
\DeclareInputText{217}{\@tabacckludge`U}
\DeclareInputText{218}{\@tabacckludge'U}
\DeclareInputText{219}{\^U}
\DeclareInputText{220}{\"U}
\DeclareInputText{221}{\@tabacckludge'Y}
\DeclareInputText{222}{\TH}
\DeclareInputText{223}{\ss}
\DeclareInputText{224}{\@tabacckludge`a}
\DeclareInputText{225}{\@tabacckludge'a}
\DeclareInputText{226}{\^a}
\DeclareInputText{227}{\~a}
\DeclareInputText{228}{\"a}
\DeclareInputText{229}{\r a}
\DeclareInputText{230}{\ae}
\DeclareInputText{231}{\c c}
\DeclareInputText{232}{\@tabacckludge`e}
\DeclareInputText{233}{\@tabacckludge'e}
\DeclareInputText{234}{\^e}
\DeclareInputText{235}{\"e}
\DeclareInputText{236}{\@tabacckludge`\i}
\DeclareInputText{237}{\@tabacckludge'\i}
\DeclareInputText{238}{\^\i}
\DeclareInputText{239}{\"\i}
```

```
\DeclareInputText{240}{\dh}
\DeclareInputText{241}{\~n}
\DeclareInputText{242}{\@tabacckludge`o}
\DeclareInputText{243}{\@tabacckludge'o}
\DeclareInputText{244}{\^o}
\DeclareInputText{245}{\~o}
\DeclareInputText{246}{\"o}
\DeclareInputMath{247}{\div}
\DeclareInputText{248}{\o}
\DeclareInputText{249}{\@tabacckludge`u}
\DeclareInputText{250}{\@tabacckludge'u}
\DeclareInputText{251}{\^u}
\DeclareInputText{252}{\"u}
\DeclareInputText{253}{\@tabacckludge'y}
\DeclareInputText{254}{\th}
\DeclareInputText{255}{\"y}
\makeatother
\endinput
%%
%% End of file `latin9.def'.
```

A.2 latin10.def

```
%%
%% This is file `latin10.def',
%%
%% This is a new file.
%%
%% Copyright 2001
%%    Ionel Mugurel Ciobîcã
%%
%% Permision granted to copy, distribute and redistribute this file.
%%
%% The comma below accent for S, s, T and t doesn't look good
%% for large characters. A solution would be to include internal
%% support for comma below in the same way like for the dot below,
%% so \C{t} to create the t comma below, etc.
%%
%% Because of the euro symbol the tetex-eurosym package has to be
%% installed. Otherwise an alternative is made to "draw" the character
%% on place. Uncomment that line and comment the next one.
%%
%% Latin10 is also comming with support for the German double quotations.
%% You have to use babel with a language which support those quotations,
```

```
%% German and Romanian come now in my mind...
%%
%% Because of \textdegree you might want to use \usepackage{textcomp} in
%% your document.
%%
%% \CharacterTable
%%  {Upper-case    \A\B\C\D\E\F\G\H\I\J\K\L\M\N\O\P\Q\R\S\T\U\V\W\X\Y\Z
%%   Lower-case    \a\b\c\d\e\f\g\h\i\j\k\l\m\n\o\p\q\r\s\t\u\v\w\x\y\z
%%   Digits        \0\1\2\3\4\5\6\7\8\9
%%   Exclamation   \!     Double quote  \"     Hash (number) \#
%%   Dollar        \$     Percent       \%     Ampersand     \&
%%   Acute accent  \'     Left paren    \(     Right paren   \)
%%   Asterisk      \*     Plus          \+     Comma         \,
%%   Minus         \-     Point         \.     Solidus       \/
%%   Colon         \:     Semicolon     \;     Less than     \<
%%   Equals        \=     Greater than  \>     Question mark \?
%%   Commercial at \@     Left bracket  \[     Backslash     \\
%%   Right bracket \]     Circumflex    \^     Underscore    \_
%%   Grave accent  \`     Left brace    \{     Vertical bar  \|
%%   Right brace   \}     Tilde         \~}
 \ProvidesFile{latin10.def}
   [2001/10/07 v0.01 Input encoding file
                        (test version: still liable to change)]
\makeatletter
\DeclareInputText{160}{\nobreakspace}
\DeclareInputText{161}{\k A}
\DeclareInputText{162}{\k a}
\DeclareInputText{163}{\L}
%\DeclareInputText{164}{{\sffamily C\makebox[0pt][l]{\kern-.70em\mbox{--}}\makebo
\DeclareInputText{164}{\euro}
\DeclareInputText{165}{\guillemotleft}
\DeclareInputText{166}{\v S}
\DeclareInputText{167}{\S}
\DeclareInputText{168}{\v s}
\DeclareInputText{169}{\copyright}
\DeclareInputText{170}{\ooalign{S\crcr\hidewidth\raise-.31ex\hbox{\scriptsize,}\h
\DeclareInputText{171}{\quotedblbase}
\DeclareInputText{172}{\@tabacckludge'Z}
\DeclareInputText{173}{\-}
\DeclareInputText{174}{\@tabacckludge' Z}
\DeclareInputText{175}{\.Z}
\DeclareInputText{176}{\textdegree}
\DeclareInputMath{177}{\pm}
\DeclareInputText{178}{\v C}
\DeclareInputText{179}{\l}
\DeclareInputText{180}{\v Z}
```

```
\DeclareInputText{181}{\textquotedblleft}
\DeclareInputText{182}{\P}
\DeclareInputText{183}{\textperiodcentered}
\DeclareInputText{184}{\v z}
\DeclareInputText{185}{\v c}
\DeclareInputText{186}{\ooalign{s\crcr\hidewidth\raise-.31ex\hbox{\scriptsize,}\h
\DeclareInputText{187}{\guillemotright}
\DeclareInputText{188}{\OE}
\DeclareInputText{189}{\oe}
\DeclareInputText{190}{\" Y}
\DeclareInputText{191}{\.z}
\DeclareInputText{192}{\@tabacckludge`A}
\DeclareInputText{193}{\@tabacckludge'A}
\DeclareInputText{194}{\^A}
\DeclareInputText{195}{\u A}
\DeclareInputText{196}{\"A}
\DeclareInputText{197}{\@tabacckludge'C}
\DeclareInputText{198}{\AE}
\DeclareInputText{199}{\c C}
\DeclareInputText{200}{\@tabacckludge`E}
\DeclareInputText{201}{\@tabacckludge'E}
\DeclareInputText{202}{\^ E}
\DeclareInputText{203}{\" E}
\DeclareInputText{204}{\@tabacckludge`I}
\DeclareInputText{205}{\@tabacckludge'I}
\DeclareInputText{206}{\^I}
\DeclareInputText{207}{\" I}
\DeclareInputText{208}{\D}
\DeclareInputText{209}{\@tabacckludge'N}
\DeclareInputText{210}{\@tabacckludge`O}
\DeclareInputText{211}{\@tabacckludge'O}
\DeclareInputText{212}{\^O}
\DeclareInputText{213}{\H O}
\DeclareInputText{214}{\"O}
\DeclareInputText{215}{\@tabacckludge'S}
\DeclareInputText{216}{\H U}
\DeclareInputText{217}{\@tabacckludge`U}
\DeclareInputText{218}{\@tabacckludge'U}
\DeclareInputText{219}{\^ U}
\DeclareInputText{220}{\"U}
\DeclareInputText{221}{\k E}
\DeclareInputText{222}{\ooalign{T\crcr\hidewidth\raise-.31ex\hbox{\scriptsize,}\h
\DeclareInputText{223}{\ss}
\DeclareInputText{224}{\@tabacckludge`a}
\DeclareInputText{225}{\@tabacckludge'a}
\DeclareInputText{226}{\^a}
```

```
\DeclareInputText{227}{\u a}
\DeclareInputText{228}{\"a}
\DeclareInputText{229}{\@tabacckludge’c}
\DeclareInputText{230}{\ae}
\DeclareInputText{231}{\c c}
\DeclareInputText{232}{\@tabacckludge‘e}
\DeclareInputText{233}{\@tabacckludge’e}
\DeclareInputText{234}{\^e}
\DeclareInputText{235}{\"e}
\DeclareInputText{236}{\@tabacckludge‘\i}
\DeclareInputText{237}{\@tabacckludge’\i}
\DeclareInputText{238}{\^\i}
\DeclareInputText{239}{\"\i}
\DeclareInputText{240}{\d}
\DeclareInputText{241}{\@tabacckludge’n}
\DeclareInputText{242}{\@tabacckludge‘o}
\DeclareInputText{243}{\@tabacckludge’o}
\DeclareInputText{244}{\^o}
\DeclareInputText{245}{\H o}
\DeclareInputText{246}{\"o}
\DeclareInputText{247}{\@tabacckludge’s}
\DeclareInputText{248}{\H u}
\DeclareInputText{249}{\@tabacckludge‘u}
\DeclareInputText{250}{\@tabacckludge’u}
\DeclareInputText{251}{\^u}
\DeclareInputText{252}{\"u}
\DeclareInputText{253}{\k e}
\DeclareInputText{254}{\ooalign{t\crcr\hidewidth\raise-.31ex\hbox{\scriptsize,}\h
\DeclareInputText{255}{\"y}
\makeatother
\endinput
%%
%% End of file ‘latin10.def’.
```