The Linux Danish/International HOWTO

Niels Kristian Bech Jensen, nkbj@image.dk v1.7, 3 October 1997

This document describes how to configure Linux and various Linux applications for Danish locale standards such as keyboard, font, paper-size etc. It is hoped that Linux users from other places in Western Europe will find this document useful too.

Contents

1 Introduction **2**

2 Keyboard setup **2**

2.1 Loading a keytable . . . 2

2.2 Getting the AltGr key to work under X11 . . . 3

2.2.1 Making {, [,] and }work under Metro-X . . . 4

2.3 Dead keys and accented characters . . . 4

2.3.1 Removing dead key functionality . . . 4

2.3.2 Invoking dead key functionality . . . 4

2.4 Making $(the dollar sign), ø (oslash) and Ø (Oslash) work . . . 6

2.4.1 $(the dollar sign) . . . 6

2.4.2 ø (oslash) and Ø (Oslash) . . . 6

3 Display and application setup **7**

3.1 Loading the Latin-1 character set on the console . . . 7

3.2 Characters you can display under Linux . . . 7

3.3 International character sets in specific applications . . . 7

4 Miscellaneous problems **12**

4.1 Time zone . . . 12

4.2 A4 papersize . . . 13

4.3 Text file formats for other platforms . . . 13

5 Locale support in libc 5.4.x **13**

6 Programming tips for X11 **14**

7 Getting X11 applications to speak Danish 14

8 References and FTP sites 15

8.1 Other documents of relevance 15

8.2 FTP and Web sites 15

9 Post-amble: Acknowledgments and Copyright 15

9.1 Disclaimer 16

1 Introduction

All European users of almost any operating system have two problems: The first is to tell the computer that you have a non-American keyboard, and the second is to get the computer to display the special characters. To make matters worse some applications will also consider you an exception if you are not an American and require special options or the setting of environment variables.

Under Linux you change the way your computer interprets the keyboard with the commands `loadkeys` and `xmodmap`. `loadkeys` will modify the keyboard for plain Linux while `xmodmap` makes the modifications necessary when the handshaking between X11 and Linux is imperfect.

To display the characters you need to tell your applications that you use the ISO-8859-1 (a.k.a. Latin-1) international set of glyphs. This is not always necessary, but a number of key applications need special attention.

This HOWTO is intended to tell Danish users how to do this. If you continue to have problems after reading this you can try the German HOWTO, the Linux Keyboard and Console HOWTO or the ISO 8859-1 National Character Set FAQ. Many of the hints contained herein are cribbed from there. See section 8.1 for pointers to these documents. You should also send me a mail describing your problems.

A final problem is that error-messages, menus and documentation of the applications are mostly in English. There is a GNU project under way to address this problem. You can see what it is all about by downloading the file `ABOUT-NLS` or the package `gettext-0.10.tar.gz` (or any later version) from your favourite mirror of `prep.ai.mit.edu`. This project needs volunteers for the translations. Send a mail to `da-request@li.org` with the body "subscribe" if you want to contribute to the Danish part of the project. The documentation in the `gettext` package describes how to use such translations in your own programs.

2 Keyboard setup

2.1 Loading a keytable

You have two tools for configuring your keyboard. Under plain Linux you have `loadkeys` and under X11 you have `xmodmap`.

To try out `loadkeys` type one of these two commands:

```
loadkeys /usr/lib/kbd/keytables/dk.map
```

or

```
loadkeys /usr/lib/kbd/keytables/dk-latin1.map
```

The difference between the two keymaps is that `dk-latin1.map` enables 'dead' keys while `dk.map` does not. Dead keys are explained in section 2.3. The program `loadkeys` and the keymaps are part of the package `kbd-0.??.tar.gz` which (with differing version numbers ??) is available with all Linux distributions.

Usually `loadkeys` is executed at boot-time from one of the scripts under the directory `/etc/rc.d/`. Details vary between distributions.

(Note for non-Danish readers: Support for other languages is enabled in a similar manner. Use `es.map` for Spanish keyboards etc.)

Versions of XFree86 up to and including v3.1.2 will normally follow the keymap used by plain Linux, but you can modify keyboard behavior under X11 with `xmodmap`. Usually the X11 initialization process will run this command automatically if you have a file called `.Xmodmap` in your home directory.

In XFree86 v3.2 and higher you should have the following `Keyboard` section in your `/etc/XF86Config` file (it is made automatically by the program `XF86Setup` if you choose a Danish keytable):

```
Section "Keyboard"
   Protocol        "Standard"
   XkbRules        "xfree86"
   XkbModel        "pc101"
   XkbLayout       "dk"
   XkbVariant      "nodeadkeys"
EndSection
```

The only keyboard variant available at the moment is `"nodeadkeys"`, but dead keys can still be made to work. See section 2.3 for more information on this.

2.2 Getting the AltGr key to work under X11

For versions of XFree86 up to and including v3.1.2 you should edit the file `/etc/X11/XF86Config` (or possibly `/etc/XF86Config`) and make sure the line

```
RightAlt    ModeShift
```

appears in the `Keyboard` section. Usually you can do this by uncommenting the appropriate line. In XFree86 v3.1.2 you can use `AltGr` as an alias for `RightAlt`.

The `AltGr` key should work as expected in XFree86 v3.2 and higher if you choose Danish keyboard support.

2.2.1 Making {, [,] and } work under Metro-X

You can't input the characters "{" (<AltGr><7>), "[" (<AltGr><8>), "]" (<AltGr><9>) and "}" (<AltGr><0>) under the Metro-X server. This bug has been observed under versions 3.1.5 and 3.1.8 of the server.

To correct this bug you have to edit the file `/usr/X11R6/lib/X11/xkb/symbols/dk` and change the lines

```
key <AE07> {    [              7,          slash     ]       };
key <AE08> {    [              8,       parenleft    ]       };
key <AE09> {    [              9,      parenright    ]       };
key <AE10> {    [              0,          equal     ]       };
```

to

```
key <AE07> {    [              7,          slash     ],
                [      braceleft,       NoSymbol     ]       };
key <AE08> {    [              8,       parenleft    ],
                [    bracketleft,       NoSymbol     ]       };
key <AE09> {    [              9,      parenright    ],
                [   bracketright,       NoSymbol     ]       };
key <AE10> {    [              0,          equal     ],
                [     braceright,       NoSymbol     ]       };
```

2.3 Dead keys and accented characters

Dead keys are those that does not type anything until you hit another key. Tildes and umlauts are like this by default under plain Linux if you use the `dk-latin1.map` keymap. This is the default behaviour for these keys under Microsoft Windows as well.

2.3.1 Removing dead key functionality

- Removing dead key functionality under plain Linux and XFree86 v3.1.2

  Under plain Linux type

  ```
  loadkeys dk.map
  ```

- Removing dead key functionality under XFree86 v3.2 and higher

  Put the following line in the `Keyboard` section of your `/etc/XF86Config` file:

  ```
  XkbVariant      "nodeadkeys"
  ```

2.3.2 Invoking dead key functionality

- Invoking dead key functionality under plain Linux

  Under plain Linux type

```
loadkeys dk-latin1.map
```

- Invoking dead key functionality under X11R6 sessions

  First you must make sure you are running XFree86 v3.1.2 or higher. Download and install everything related to the newest release if you have a lower version number. Neither compose nor dead keys will work in X11R6 applications unless these are compiled with support for accented (8-bit) character input. A useful example of such an application is `GNU emacs` version 19.30 (or higher.)

  Some X11 applications still does not support this input method. Eventually this situation might improve, but until that happens you can either hack your applications or submit polite bug reports to the program authors. The latter approach is often the most efficient. See section 6 for some advice on what needs to be done.

  Next you will have to map a key to `Multi_key` (`Compose`.) The `Scroll Lock` key is most likely already mapped as such if you use XFree86 v3.1.2 (you can verify this with the program `xev`,) and it is easy to map the right `Control` key by uncommenting the appropriate line in the `Keyboard` section of the XFree86 configuration file (often `/etc/XF86Config`.) If you wish to use some other key, or if you are using XFree86 v3.2 or higher and want to change the default, you should put something like

  ```
  keycode 78 = Multi_key
  ```

  in your `~/.Xmodmap` file. The statement in the example defines `Scroll Lock` as the `Compose` key. The default `Compose` key in XFree86 v3.2 and higher is <`Shift`><`AltGr`>.

  XFree86 v3.2 and higher comes without support for the dead keys on the standard Danish keyboard. To get this support you have to change a few lines in the `xkb_symbols "basic"` section of the file `/usr/X11R6/lib/X11/xkb/symbols/dk`. The lines

  ```
  key <AE12> {    [          acute,          grave     ],
                  [            bar,     dead_ogonek     ]       };
  key <AD12> {    [      diaeresis,     asciicircum     ],
                  [     asciitilde,     dead_macron     ]       };
  ```

  should be changed to

  ```
  key <AE12> {    [     dead_acute,      dead_grave     ],
                  [            bar,     dead_ogonek     ]       };
  key <AD12> {    [ dead_diaeresis, dead_circumflex     ],
                  [     dead_tilde,     dead_macron     ]       };
  ```

  After these changes you can get support for dead keys by removing the line

  ```
  XkbVariant      "nodeadkeys"
  ```

  from the `Keyboard` section of your `/etc/XF86Config` file.

  (Note for non-Danish readers: There are files for many local keyboards in `/usr/X11R6/lib/X11/xkb/symbols`.)

  The available keystroke combinations are compiled in `/usr/X11R6/lib/X11/locale/iso8859-1/Compose`. There are some bugs in that file you will want to fix:

1. The line reading

   ```
   <dead_tilde> <space>                    : "~"   tilde
   ```

   should be changed to

   ```
   <dead_tilde> <space>                    : "~"   asciitilde
   ```

2. In several places `asciicircum` is misspelled as `asciicirum`

Finally make sure your shells and/or applications is set up for ISO-8859-1 compatibility as described in section 3.3 and you should be all set.

2.4 Making $ (the dollar sign), ø (oslash) and Ø (Oslash) work

2.4.1 $ (the dollar sign)

There is a bug in the Danish keymaps causing the dollar sign to be accessed with <Shift><4> instead of <AltGr><4> by default. If this is a problem for you, determine what keymap you load at boot-time. You can find it by looking around in the directory `/etc/rc.d/` or simply by paying attention to what happens at boot-time. On my computer the relevant keymap is called `/usr/lib/kbd/keytables/dk-latin1.map`. You can fix the problem by changing the line

```
keycode   5 = four            dollar          dollar
```

in the keymap file to

```
keycode   5 = four            currency        dollar
```

and then loading the keytable as described in section 2.1. Currency (dansk: "soltegn") is the default <Shift><4> character on a Danish keyboard.

This should fix the problem for both X11 and plain Linux.

2.4.2 ø (oslash) and Ø (Oslash)

In some older distributions "ø" and "Ø" appear as cent and yen. Find the line for keycode 40 in the keymap file and change it from

```
keycode  40 = cent            yen
```

to

```
keycode  40 = +oslash         +Ooblique
```

Note: This bug appears to have been fixed in `kbd-0.88.tar.gz` and newer versions.

The plus signs are necessary to get `Caps Lock` working properly. "Oslash" can be used as an alias for "Ooblique" in `kbd-0.90.tar.gz` and newer versions.

3 Display and application setup

Most applications need to be compiled as "8-bit-clean" to work well with European characters. Some need a few extra hints to get it right.

3.1 Loading the Latin-1 character set on the console

Execute the following commands under the `bash` shell:

```
setfont /usr/lib/kbd/consolefonts/lat1-16.psf
mapscrn /usr/lib/kbd/consoletrans/trivial
echo -ne '\033(K'
```

You could also choose to load a unicode font, which (in my opinion) is much nicer than `lat1-16.psf` and seems to be more stable. Execute the following commands to do that:

```
setfont /usr/lib/kbd/consolefonts/iso01.f16
mapscrn /usr/lib/kbd/consoletrans/iso01.uni
echo -ne '\033(K'
```

If you use Linux kernels v1.3.1 or higher, you do not need the `mapscrn` and `echo` commands when you use unicode fonts.

Note: This only has effect under plain Linux.

3.2 Characters you can display under Linux

Type `dumpkeys -l | less` at the prompt to find out which characters that are readily available. You can map them to your keyboard via the keymap files mentioned in section 2.1.

3.3 International character sets in specific applications

A number of applications demand special attention. This section describes how to set up configuration files for them.

`bash`:

Put the following in your `~/.inputrc` file:

```
set meta-flag on
set convert-meta off
set output-meta on
```

`elm`:

Put the following definitions in your `~/.elm/elmrc` file:

```
charset = iso-8859-1
displaycharset = iso-8859-1
textencoding = 8bit
```

This may not work on some versions of `elm`.

`emacs`:

Put the following in your `~/.emacs` or the the system-wide initialization file (probably `/usr/lib/emacs/site-lisp/default.el` or `/usr/share/emacs/site-lisp/default.el`):

```
(standard-display-european t)
(require 'iso-syntax)
(set-input-mode (car (current-input-mode))
        (nth 1 (current-input-mode))
        0)
```

You can leave out the first two of the lines above if you have installed locale support, and your `LC_CTYPE` environment variable includes one of the strings `8859-1` or `88591`. See section 5 for some information on locales.

Dead keys should work under GNU emacs provided you use GNU emacs v19.30 or higher and XFree86 v3.1.2 or higher (it works for me anyway,) so do not start researching available elisp packages implementing "electric keys" or anything like that. If you want to implement European keyboard conventions in emacs without upgrading, the best choice is probably the `remap` package available from the SunSite DK server (see section 8.2.) There are also two packages called `iso-acc.elc` and `iso-trans.elc` included with emacs that has similar functionality, but they are not nearly as powerful.

`groff`:

Issue the command as

```
groff -Tlatin1 <your_groff_input_file>
```

Remember to change this in `/etc/man.config` to get latin1 characters working in `man` (don't remove the `-mandoc` switch.)

`ispell` — **Spell checking in Danish:**

First make sure that you install version 3.1 instead of version 4.0 of `ispell`. The latter is obsolete and multiple brain-damaged. You can download the sources for `ispell` at the GNU archive at `prep.ai.mit.edu`, and you can get a Danish dictionary via FTP from *Aalborg University Center* `<ftp://ftp.iesd.auc.dk/pub/packages/dkispell/>`. Follow the compilation instructions and you should have no trouble (One caveat: When defining the variables necessary for compilation you must tell ispell that Linux is a SysV type OS by defining the variable `USG`.)

When you have installed the Danish dictionary for `ispell` you can check the spelling of a Danish language file by executing the command:

```
ispell -d danish -T latin1 -w "{\ae}{\o}{\aa}{\AE}{\O}{\AA}" <your_danish_text_file>
```

(Note for non-Danish readers: You can find dictionaries for most Western languages by reading the file `Where` included with the sources for `ispell`.)

`joe`:

Issue the command as

```
joe -asis
```

or put the following in your `~/.joerc` file:

```
-asis
```

The hyphen character *must* be in the first column.

`kermit`:

This is as close as I can get, but not completely satisfying yet. Put the following in your `~/.kermrc` file:

```
set terminal bytesize 8
set command bytesize 8
set file bytesize 8
set language danish
set file character-set latin1-iso
set transfer character-set latin1-iso
set terminal character-set latin1-iso
```

I think there are more variables to set, but they are hiding. You would have to modify these settings if the remote system is DOS or OS/2 based.

`less`:

Set the following environment variable:

```
LESSCHARSET=latin1
```

`ls`:

Issue the command as

```
ls -N
```

or possibly

```
ls --8bit
```

`lynx`:

Put the following definition in your `~/.lynxrc` file:

```
character_set=ISO Latin 1
```

This can also be set via the `Options` menu in `lynx`. Type 'o' and set the relevant option.

`man`:

See entry for `groff` in this section.

`metamail`:

Set the following environment variable:

```
MM_CHARSET=ISO-8859-1
```

`nn`:

Put the following in your `~/.nn/init` file:

```
set data-bits 8
```

`pine`:

Put the following definition in your `~/.pinerc` file:

```
character-set=ISO-8859-1
```

This can also be set via the `Setup`, `Config` menu option in `pine`.

`rlogin`:

Issue the command as

```
rlogin -8 foo.bar.dk
```

`tcsh`:

Put the following in your `/etc/csh.login` or `~/.tcshrc` file:

```
setenv LANG C
```

Actually you just have to define one of the environment variables `LANG` or `LC_CTYPE`. The value does not matter. Read the `tcsh` man page for more information.

`telnet`:

Put one line of the following type in your `~/.telnetrc` file for **each** host you want to log on to using `telnet`:

```
<hostname> set outbinary true
```

Example:

```
localhost set outbinary true
foo.bar.dk set outbinary true
```

TeX/LaTeX:

There are several problems with TeX/LaTeX: You want LaTeX to understand the special characters and you do not want LaTeX to put in English words like "Chapter" at the beginning of every chapter or use English typesetting conventions.

Under LaTeX2e the header of your input file should look something like this:

```
\documentclass[a4paper]{article}

\usepackage[latin1]{inputenc}
\usepackage{t1enc}
\usepackage[danish]{babel}
```

The first `usepackage` statement ensures that LaTeX will interpret European characters correctly, so you do not have to use escape codes for European characters. The second is not strictly necessary; but it is recommended to include it to use the DC fonts (which of course must be installed.) The DC fonts should soon be replaced by the newer EC fonts. These two packages are most likely included in your LaTeX distribution. The last `usepackage` statement defines a range of standards for typesetting Danish texts.

If you use the Debian distribution (or older Slackware) you will have to install Danish hyphenation tables yourself (dansk: "hyphenation"="orddeling".) These are available from *Aalborg University Center* <`ftp://ftp.iesd.auc.dk/pub/packages/`>. The files you need are `dkhyphen.tex`, `dkcommon.tex` and `dkspecial.tex`. If you use the teTeX distribution (distributed with e.g. Red Hat and S.u.S.E.) you already have the relevant files. Essentially you need to put these into the directory containing international hyphenation tables, edit the appropriate language dependency file (usually called `language.dat`) and finally rebuild LaTeX with `initex`. Before you do anything, please make sure you know what files you are changing and back them up in advance.

If you use NTeX (distributed with the Slackware distribution) you will have a configuration script called `ntm-ltx.cfg` located in `/usr/lib/texmf/tools/`. In that case put the hyphenation tables in the the relevant directory (most likely called `/usr/lib/texmf/tex/hyphenation/`) and run the script. It will guide you through the various steps described below. If you use teTeX there is a somewhat more advanced program called `texconfig` to help you.

Below is a description for enabling Danish hyphenation by hand. If it looks vague it is because TeX/LaTeX installations differ very much in their choice of path-names.

1. Find out where you have the hyphenation tables. Under NTeX they are in `/usr/lib/texmf/tex/hyphenation/`, under teTeX in `/usr/lib/texmf/texmf/tex/generic/hyphen/`. Try issuing the command `find /usr/lib/ -iname '*hyph*' -print` if you cannot find the directory.
2. Check if the hyphenation tables are already there. If not put the hyphenation tables mentioned above in this directory.
3. Edit the file (probably) called `language.dat`. In the teTeX distribution you can just uncomment the appropriate line. Otherwise insert a line reading

   ```
   danish dkhyphen.tex
   ```

   If you have difficulty finding `language.dat` try issuing the command `find /usr/lib/ -name language.dat -print`
4. Find and back up the file `latex.fmt`. It could be in a variety of places. Use `find /usr/lib/ -name latex.fmt -print` to find it.
5. Go to the directory where you found `latex.fmt`. Issue the command `initex latex.ltx`. Pray. If everything went well you now have a new version of `latex.fmt`.

6. You can now use

   ```
   \usepackage[danish]{babel}
   ```

   in your LaTeX headers. Hyphenation should be reasonably correct, quotation marks follow Danish conventions, chapters are now called "Kapitel" instead of "Chapter" etc.

All new Linux distributions now includes LaTeX2e.

(Note for non-Danish readers: The process is similar for other Western European languages, and the necessary files are normally included in the Linux distributions.)

In LaTeX 2.09, use

```
\documentstyle[a4,isolatin]{article}
```

to include support for ISO-8859-1 characters and European paper sizes.

`isolatin.sty` is available from all DANTE servers (see section 8.2) and from *Michael Gschwind's FTP site* `<ftp://ftp.vlsivie.tuwien.ac.at/pub/8bit>`. It should also be included in the standard Linux distributions.

Note: Some people prefer using emacs in a special mode which translates "special" letters into TeX escape codes, but this method should be obsolete by now.

`tin`:

Put the following definitions in your `~/.tin/headers` file:

```
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
```

Now you can post messages with the proper Danish characters in the message body.

4 Miscellaneous problems

4.1 Time zone

Denmark is placed in the Central European Time zone (CET or MET,) which (in the winter) is equivalent to Greenwich Mean Time plus 1 (GMT+1.) You set the time zone on a Linux system by making a symbolic link between `/usr/lib/zoneinfo/localtime` and the file in `/usr/lib/zoneinfo/` with a name corresponding to your zone or country. Danes will want to execute one of the commands

```
ln -sf /usr/lib/zoneinfo/MET /etc/localtime
```

or

```
ln -sf /usr/lib/zoneinfo/Europe/Copenhagen /etc/localtime
```

This automatically sets Daylight Saving Time (GMT+2) in the summer.

You synchronize the system time with the CMOS clock by issuing the command `clock` as root. If your CMOS clock is set to GMT (a.k.a. UTC — the standard on proper Unix systems) use

```
clock -u -s
```

or if your CMOS clock is set to local time use

```
clock -s
```

4.2 A4 papersize

- `dvips`: Edit the file `/usr/lib/texmf/dvips/config.ps` or `~/.dvips`.
- `ghostscript`: Add the command line option `-sPAPERSIZE=a4`.
- `ghostview`: Define the following Xresource:

  ```
  Ghostview.pageMedia:  A4
  ```

- `TeX/LaTeX`: See the entry for TeX/LaTeX in section 3.3.
- `xdvi`: Define the Xresource

  ```
  XDvi.paper:     a4
  ```

4.3 Text file formats for other platforms

You can translate files between an ISO-8859-1 formatted text file and e.g. a DOS text file using codepage 850 with the `recode` package. A DOS file called `foo.txt` would be translated into a proper Unix file with the command

```
recode cp850:latin1 foo.txt
```

`Recode` is available as `recode-3.4.tar.gz` from all mirrors of `prep.ai.mit.edu`.

5 Locale support in libc 5.4.x

The locale support has been updated in `libc 5.4.x`. You can avoid many of the individual programs setups described in section 3.3 if the programs on your system is prepared for locale support. The Debian distribution comes with this support if you install the `wg15-locale` package. Read the Locales mini-HOWTO if you want to set up locale support on a non-Debian system.

To enable support for the Danish locale you just have to set one of the following environment variables:

```
LANG=da_DK.ISO_8859-1
```

or

```
LC_ALL=da_DK.ISO_8859-1
```

The difference between `LANG` and `LC_ALL` is, that while `LANG` can be overridden for the individual locale categories, `LC_ALL` can't. Both environment variables set all the individual locale catgories. See the Locales mini-HOWTO for information about locale categories.

A few programs such as `bash` and `GNU emacs` still needs specific setup, but most should work without further attention. Programs such as `nvi` which didn't work with 8 bit characters before should work now.

Locale support should be more common as distributions based on the new `GNU libc 2` (`libc 6.x`) becomes available.

6 Programming tips for X11

Displaying 8-bit charaters is easy. You can use them just as you would use 7-bit ASCII. Getting applications to accept input of special characters is an entirely different matter.

If you are using e.g. the Xt toolkit and a widget set like Motif you need only add one line to your program. As your **first** call to Xt use `XtSetLanguageProc`. Like this:

```
int main (int argc, char** argv)
{
    ...
    XtSetLanguageProc (NULL, NULL, NULL);
    top = XtAppInitialize ( ... );
    ...
}
```

Now your program will automagically look up the `LC_CTYPE` variable and interpret dead keys etc. according to the `Compose` tables in `/usr/lib/X11/locale/`. This should work for all Western European keyboard layouts and is entirely portable. As XFree86 multilanguage support gets better your program will also be useful in Eastern Europe and the Middle East.

This method of input is supported by Xt, Xlib and Motif v1.2 (and higher.) According to the information I have available it is only partially supported by Xaw. If you have further information on this subject I would like to hear from you.

This section was adapted from a more extensive discussion in Michael Gschwind's Programming for Internationalization. See section 8.1 for a pointer to that document.

7 Getting X11 applications to speak Danish

To get Danish texts on menus, buttons, etc. in a well behaved X11 application, you just have to translate the resource strings defining the texts. Jacob Nordfalk, `nordfalk@cats.nbi.dk` has done such translations for a lot of applications including `Netscape` and `Ghostview`. The translations and a description of how to install them can be found *on his homepage* `<http://alf.nbi.dk/~nordfalk/ovs/>`.

8 References and FTP sites

8.1 Other documents of relevance

The HOWTOs ought to be available from all mirrors of `sunsite.unc.edu` and `tsx-11.mit.edu`. A Danish mirror at the Web is *SunSite DK* `<http://sunsite.auc.dk/ldp/HOWTO/>`.

The German HOWTO (in German) by Winfried Trümper. A lot of other national HOWTOs such as Finnish, Spanish and Polish are also available in the native languages.

The Linux Keyboard and Console HOWTO by Andries Brouwer.

The Locales mini-HOWTO by Peeter Joot.

The ISO 8859-1 National Character Set FAQ and Programming for Internationalization (plus much more) by Michael Gschwind is available from *his homepage* `<http://www.vlsivie.tuwien.ac.at/mike/i18n.html>`.

8.2 FTP and Web sites

The Linux Danish/International HOWTO has its own *homepage* `<http://www.image.dk/~nkbj/>`, which always has the latest version on-line. It also has other informations for Danish users of Linux.

This FTP site at Aalborg University Center (AUC) `<ftp://ftp.iesd.auc.dk/pub/packages/>` has Danish hyphenation tables, dictionary for ispell etc. AUC is also the home of *SunSite DK* `<ftp://sunsite.auc.dk/pub/os/linux/>` which has the Debian and Red Hat distributions, the latest kernels, the *Linux Documentation Project* `<http://sunsite.auc.dk/ldp/>`, mirrors of *sunsite.unc.edu* `<ftp://sunsite.auc.dk/pub/os/linux/sunsite/>` and *the GNU archives* `<ftp://sunsite.auc.dk/pub/gnu/>`, and *the remap package for emacs* `<ftp://sunsite.auc.dk/packages/auctex/>`.

SunSite `<ftp://sunsite.unc.edu/pub/Linux/>` and mirrors. `doc/howto` has the above mentioned HOWTOs. `utils/nls` and subdirectories contain files related to National Language Support. Developers should take a look at `locale-tutorial-0.8.txt.gz`, `locale-pack-0.8.tar.gz` and `cat-pack.tar.gz`.

The GNU archives `<ftp://prep.ai.mit.edu/pub/gnu/>` has the `recode` package for character table conversion, the `ABOUT-NLS` file and the `gettext` package for message translation of some GNU applications and (of course) the latest versions of GNU emacs.

The DANTE FTP site `<ftp://ftp.dante.de/>` has everything needed for TeX and LaTeX support.

9 Post-amble: Acknowledgments and Copyright

Thanks to Peter Dalgaard, Anders Majland, Jon Haugsand, Jacob Nordfalk, the authors of the German HOWTO, Michael Gschwind and numerous others for suggestions and help with several questions. And a big thanks to the people at Aalborg University Center for writing and making available several of the packages described in this document.

This HOWTO is copyrighted by Thomas Petersen, petersen@post1.tele.dk (the original author) and Niels Kristian Bech Jensen, nkbj@image.dk. It is distributed as part of *The Linux Documentation Project* <http://sunsite.unc.edu/LDP/> under the terms described below.

Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution and/or printing is allowed and encouraged; however, the authors would like to be notified of any such activities.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

If you have questions, please contact Greg Hankins, the Linux HOWTO coordinator, via email at gregh@sunsite.unc.edu.

9.1 Disclaimer

Although the information given in this document is believed to be correct, the authors will accept no liability for the content of this document. Use the tips and examples given herein at your own risk.