The Linux Danish/International HOWTO
# Table of Contents

The Linux Danish/International HOWTO ........................................................................................................... 1

Niels Kristian Bech Jensen nkjb@sslug.dk ........................................................................................................... 1

1. Introduction ........................................................................................................................................... 1

2. Keyboard setup ......................................................................................................................................... 1

3. Display and application setup .................................................................................................................. 1

4. Miscellaneous problems .......................................................................................................................... 1

5. Locale support in libc 5.4.x and higher ..................................................................................................... 2

6. Programming tips for X11 ....................................................................................................................... 2

7. Getting X11 applications to speak Danish ............................................................................................... 2

8. Information resources .............................................................................................................................. 2

9. Credits and legal stuff .................................................................................................................................. 2

1. Introduction ........................................................................................................................................... 2

2. Keyboard setup ......................................................................................................................................... 3

2.1 Loading a keytable .................................................................................................................................. 3

2.2 Getting the AltGr key to work under X11 ............................................................................................. 4

2.3 Dead keys and accented characters ....................................................................................................... 4

3. Display and application setup .................................................................................................................. 7

3.1 Loading the ISO−8859−1 font on the console ....................................................................................... 7

3.2 The Euro symbol ...................................................................................................................................... 7

3.3 Characters you can display under Linux ................................................................................................. 7

3.4 International character sets in specific applications ............................................................................... 7

4. Miscellaneous problems .......................................................................................................................... 11

4.1 Time zone ............................................................................................................................................... 11

4.2 A4 papercsize ........................................................................................................................................ 11

4.3 Text file formats for other platforms ..................................................................................................... 11

5. Locale support in libc 5.4.x and higher ..................................................................................................... 12

6. Programming tips for X11 ....................................................................................................................... 13

7. Getting X11 applications to speak Danish ............................................................................................... 13

8. Information resources .............................................................................................................................. 14

8.1 Other documents of relevance ............................................................................................................... 14

8.2 FTP and Web sites .................................................................................................................................. 14

9. Credits and legal stuff .................................................................................................................................. 14

9.1 Legal stuff ................................................................................................................................................ 14
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.

1. Introduction

2. Keyboard setup
   - 2.1 Loading a keytable
   - 2.2 Getting the AltGr key to work under X11
   - 2.3 Dead keys and accented characters
   - 2.4 Making $ (the dollar sign), ø (oslash) and Ø (Oslash) work

3. Display and application setup
   - 3.1 Loading the ISO−8859−1 font on the console
   - 3.2 The Euro symbol
   - 3.3 Characters you can display under Linux
   - 3.4 International character sets in specific applications

4. Miscellaneous problems
   - 4.1 Time zone
   - 4.2 A4 papersize
   - 4.3 Text file formats for other platforms
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 Other documents of relevance 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 the GNU archive. This project needs volunteers for the translations. Send a mail to
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 Dead keys and accented characters. 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 (or /etc/X11/XF86Config) file (it should be 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 Dead keys and accented characters 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/XF86Config (or /etc/X11/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.

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

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

To

```plaintext
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 do 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.

Removing dead key functionality

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

  ```bash
  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 (or /etc/X11/XF86Config) file:

```
XkbVariant "nodeadkeys"
```

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. An example of such an application is GNU emacs version 19.30 (or higher.) Some X11 applications still do 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 Programming tips for X11 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 or /etc/X11/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 (or /etc/X11/XF86Config) file. (Note for non–Danish readers: There are files for many local keyboard maps in /usr/X11R6/lib/X11/xkb/symbols.) The available keystroke combinations are listed in /usr/X11R6/lib/X11/locale/iso8859-1/Compose. There are some bugs in that file you will want to fix:
2.4 Making $ (the dollar sign), ø (oslash) and Ø (Oslash) work

$ (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 (re-)loading the keytable as described in section Loading a keytable. Currency (dansk: ``soltegn'') is the default <Shift><4> character on a Danish keyboard.

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

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
```

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.

You can read more about keyboard configuration at this site.
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 ISO–8859–1 font on the console

Execute the following commands from your shell prompt:

```
setfont lat1u-16.psf
```

In Red Hat Linux 5.2 and higher you can do this by adding these lines to `/etc/sysconfig/i18n`:

```
SYSFONT=lat1u-16.psf
SYSTERM=linux
```

Due to a bug in the `ncurses` package on Red Hat Linux 5.2, you also have to change "linux-lat" to "linux" in `/etc/profile.d/lang.sh`. This is not necessary in Red Hat Linux 6.0.

### 3.2 The Euro symbol

A new symbol has been added to the Danish character set: The symbol for the Euro (the new currency of the European Monetary Union.) A new character set called ISO–8859–15 a.k.a. latin0 (or latin9) has been created to replace ISO–8859–1 (latin1.) You must use the `EURO` package to get support for latin0. The package includes both fonts and keymaps.

### 3.3 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 [Loading a keytable](#).

### 3.4 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
```

3. Display and application setup
elm:

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

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

This may not work on some versions of elm. You can get partial MIME support in elm if you use metamail.

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):

```lisp
(standard-display-european t)
(set-input-mode (car (current-input-mode))
 (nth 1 (current-input-mode)))
```

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 SunSite DK. There are also two packages called iso-acc.elc and iso-trans.elc included with emacs that have similar functionality, but they are not nearly as powerful.

groff:

Issue the command as

groff −Tlatin1 <your_groff_input_file> if you want ISO–8859–1 text output.
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 3.1.20 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 and you can get a Danish dictionary from SSLUG. 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:

```plaintext
ispell -d danish -T latin1 -w "æåø" <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

```plaintext
joe −asis or put the following in your ~/.joerc file:
−asis
```

The hyphen character must be in the first column.

kermit:

The Linux Danish/International HOWTO
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 This is not necessary if your system support locales. Then you should just set LANG, LC_CTYPE or LC_ALL (see section Locale support in libc 5.4.x and higher.)

**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. It won't hurt to enable enable-8bit-esmtp-negotiation and enable-8bit-nntp-posting (for news) in that menu too.

**rlogin:**

Issue the command as
rlogin -8 foo.bar.dk

**sendmail:**
Put (or uncomment) the following in your `/etc/sendmail.cf` file:

```bash
O SevenBitInput=False
O EightBitMode=pass80
O DefaultCharSet=iso-8859-1
```

**tcsh:**

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

```bash
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`:

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

Example:

```bash
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:

```latex
\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 one is not strictly necessary, but it is recommended including it to use the new EC fonts (previously called DC fonts.) The third `usepackage` statement defines a range of standards for typesetting texts in Danish.

All the major Linux distributions now includes the `texTeX` package. To set up `texTeX` you must run the script `texconfig`. Here you can choose Danish hyphenation (dansk: `"orddeling"), A4 papersize for `dvips` and `xdvi` etc.

All new Linux distributions include LaTeX2e, but on older systems you might come across LaTeX 2.09. If that happens you can use

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

to include support for ISO–8859–1 characters and European paper sizes. A better thing to do would be to ask your system administrator to upgrade to LaTeX2e.

`isolatin.sty` is available from all CTAN servers.

Some people prefer to use `emacs` in a special mode which translates `"special"` letters into `TeX` escape codes, but this method is obsolete.

**tin:**

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

```bash
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**

- **ghostscript**: Add the command line option `-sPAPERSIZE=a4`.
- **ghostview**: Define the following Xresource:

  ```
  Ghostview.pageMedia: A4
  ```

- **TeX/LaTeX, dvips, xdvi**: See the entry for TeX/LaTeX in section *International character sets in specific applications*.

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 the [GNU archive](http://www.gnu.org).
5. **Locale support in libc 5.4.x and higher**

The locale support has been updated in libc 5.4.x. You can avoid many of the individual program setups described in section *International character sets in specific applications* 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. Systems with GNU libc 2 (libc 6.x) support locales by default (see remarks about Red Hat Linux release 5.0 later in this section.)

If you use a system without locale support, you can add such support using the following method:

1. Make sure you have the latest libc 5.4.x library. You can get this from Yggdrasil Computing.
2. Make sure you have the `localedef` program installed. It should come with the library.
3. Get the locale sources. You can get them from DKUUG. You need to get both `locale` and `charmap` sources.
4. Put the locale sources in `/usr/share/i18n/locales/` and the charmap sources in `/usr/share/i18n/charmaps/`
5. Execute the `localedef` program to build the locale data files:

   ```bash
   ```

   (Note for non–Danish readers: You can build locale data files for other locales in the same way. All locale and charmap sources are at the DKUUG site.)

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

- `LANG=da_DK`
- `LC_ALL=da_DK`

Try `da_DK.ISO_8859-1` if `da_DK` does not work.

Both environment variables set all the individual locale categories. You can also set a single locale category by using the name of the category as an environment variable. The locale categories are:

<table>
<thead>
<tr>
<th>Locale category</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LC_COLLATE</code></td>
<td>Collation of strings (sort order.)</td>
</tr>
<tr>
<td><code>LC_CTYPE</code></td>
<td>Classification and conversion of characters.</td>
</tr>
<tr>
<td><code>LC_MESSAGES</code></td>
<td>Translations of yes and no.</td>
</tr>
<tr>
<td><code>LC_MONETARY</code></td>
<td>Format of monetary values.</td>
</tr>
<tr>
<td><code>LC_NUMERIC</code></td>
<td>Format of non-monetary numeric values.</td>
</tr>
<tr>
<td><code>LC_TIME</code></td>
<td>Date and time formats.</td>
</tr>
<tr>
<td><code>LC_ALL</code></td>
<td>Sets all of the above (overrides all of them.)</td>
</tr>
<tr>
<td><code>LANG</code></td>
<td>Sets all the categories, but can be overridden</td>
</tr>
<tr>
<td></td>
<td>by the individual locale categories.</td>
</tr>
</tbody>
</table>

In Red Hat Linux 5.2 you can set the environment variables `LANG` and/or `LC_ALL` in the file `/etc/sysconfig/i18n` by adding lines such as this:

```
LC_ALL=da_DK
```

A few programs such as `bash` and GNU `emacs` still need specific setup as described in section *International character sets in specific applications*, but most should work without further attention. Programs such as
nvi which did not work with 8 bit characters before should work now.

Locale support should be more common as distributions based on the new GNU libc 2 become available. Beware that although Red Hat Linux release 5.0 comes with GNU libc 2, the locale support is not working. You have to build the locale data files by executing localedef yourself. You can build the Danish locale data files with the following command:

```bash
localedef -c -i da_DK -f ISO-8859-1 da_DK
```

As of glibc-2.0.7-4.i386.rpm the locale data files are included with the libraries and this is no longer necessary.

---

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:

```c
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/X11R6/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 Other documents of relevance 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 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 at this site.
8. **Information resources**

8.1 **Other documents of relevance**

The HOWTOs are available from all mirrors of www.linuxdoc.org. There is a Danish mirror at SunSite DK.

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 ISO 8859–1 National Character Set FAQ and Programming for Internationalization (plus much more) by Michael Gschwind is available from this site.

8.2 **FTP and Web sites**

SSLUG (Skåne Sjælland Linux User Group) is a Swedish/Danish Linux user group. Their mailing list is a good place to get help with Linux in Danish (or Swedish.) They are also hosts for this document.

AUC in Ålborg is the home of SunSite DK which has the Debian and Red Hat distributions, the latest kernels, a mirror of the Linux Documentation Project and mirrors of metalab.unc.edu and the GNU archive. There is also a mirror of the CTAN archive with everything you need to get TeX and LaTeX running.

9. **Credits and legal stuff**

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. A special Thank You to Thomas Petersen; the original author of this document.

9.1 **Legal stuff**

Trademarks are owned by their owners.

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