How to prepare any Linux distribution for the Japanese language using canna and kinput

(More than) a tutorial on how to make any non-Asian-distribution ready for writing and printing Japanese

Version: 28th December 2005

INDEX

1 Introduction

This is a tutorial both for Linux users and professionals. It demonstrates how to prepare any Linux distribution for typing and printing Japanese using **canna** and **kinput2** in conjunction with a western distribution and keyboard. Furthermore it will be also very useful for Linux distributors in order to make their distributions ready for Japanese.

When taking a look at the multilingualization of Linux, especially concerning Japanese with canna and kinput, it seems quite astonishing that only the big distributions - not taking into consideration the Japanese distributions - offer assistance for Japanese. I could not believe that it would be difficult to prepare any Linux distribution for the Japanese language. But I saw myself that the information on the web pages of the maintainers is still poor and mostly in Japanese. I decided therefore to make the effort to install canna and kinput on a rather unknown little Linux distribution that had no ambitions to support Japanese. A lot of work was entailed to find out which files I really needed and in what order they had to be installed.

Well, this is the result I came up with: It works perfectly and should work on any Linux distribution, not only on Linux but - with some modifications - on any Unix.

I also added a section on how to write any Asian language using scim, such as Chinese, Japanese, Korean, Thai, Punjabi, Hindi, Nepali, Arabic etc. The installation was much easier than the one of canna and kinput. It works fine without any problems. Nevertheless in my opinion the solution of canna and kinput appears much more efficient for Japanese. Scim seems to be more optimized for usage for a variety of languages at the same time. So it is certainly worthwhile to work on the improvement of the packages for canna and kinput.

This tutorial has been tested with Rock Linux and T2 Linux (a derivate of Rock Linux). These two distributions are internationally rather unknown, but they are stable and, as yet, have no ambitions to support any of the Asian languages. Hence they were ideal for testing purposes. The aim of the project is to write and to print the Japanese language with any X-program under Linux, including the usage of Unicode True Type Fonts like in the following example. I have written the hiragana writing of most kanji in brackets.
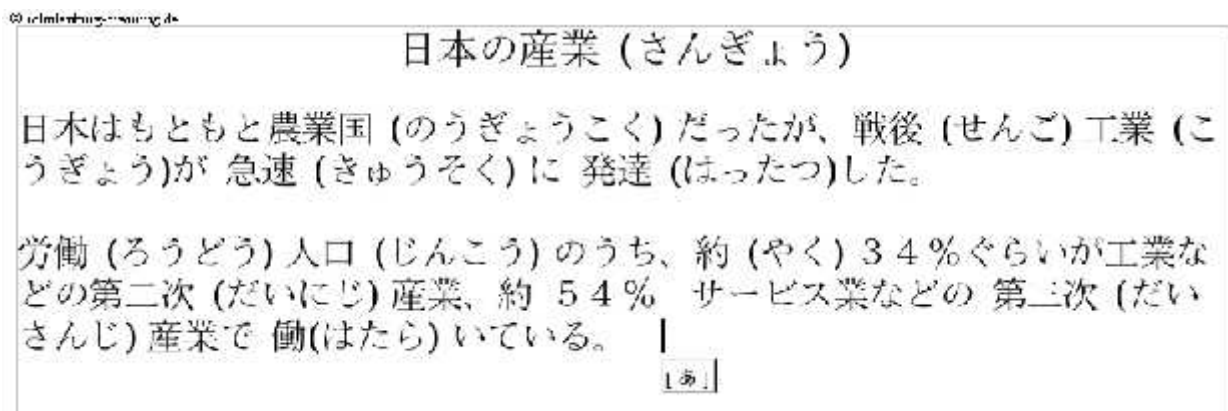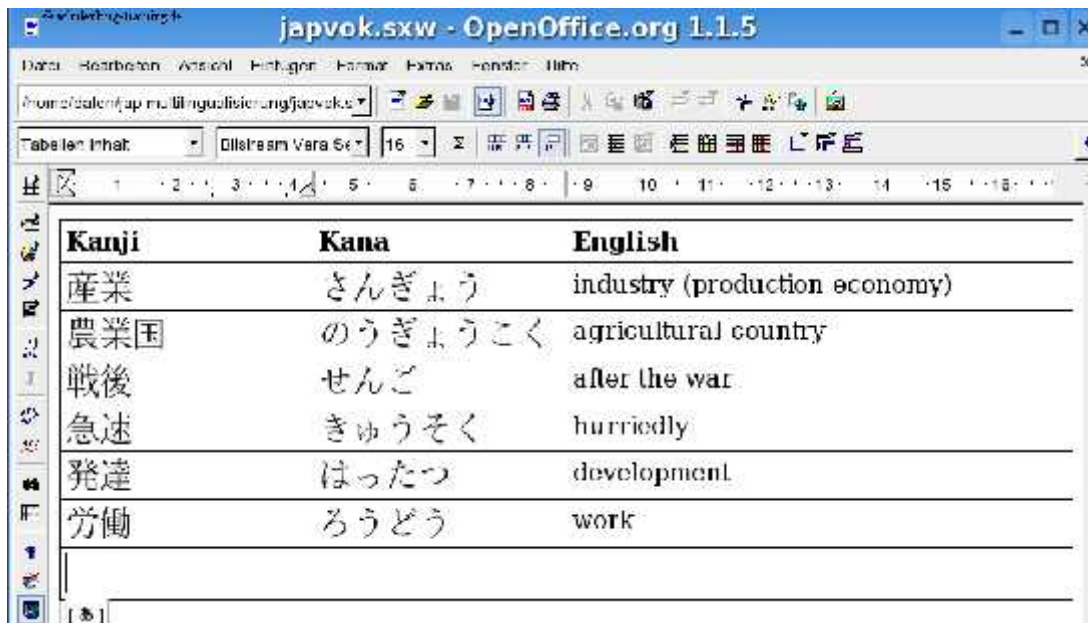
Screen shots of "Open Office" with Japanese language.

2 The Japanese Language

2.1   The sound of the Japanese language

The Japanese language has no relation with the Chinese language. The intonations of the two languages are totally different. While the Chinese language sounds rather odd to Europeans, the Japanese language can be spoken very easily by Europeans. Consonants, apart from the "r", are nearly spoken as in English while the vowels are spoken as in Italian or Spanish. The Japanese in general have no problems to learn to speak Latin languages. Another difference is that while Chinese people normally have difficulties in pronouncing the "r" and replace it by an "l", the Japanese problem is just the other way round. The Japanese "r" is spoken with the tongue as in Spanish or Italian.

2.2   Kanji

Kanji are complex Chinese characters that are used in Japanese. When the ancestors of the present Japanese inhabitants came to the islands that form Japan, they still had no written form of communication. Kanji were introduced into the Japanese language during the Han-Dynasty (between about 206 BC and 9 AC). Kanji have changed quite a bit since the Han era; nevertheless the Japanese and Chinese people these days can generally communicate in written language without any major problem.
Every Kanji has a meaning and is normally spoken as one syllable, at least in composed words. Simple one-Kanji-words normally have two syllables like in the word "hito", which means "man", whereas the Chinese reading of the same Kanji is "jin", like in the word "gaijin", which means "foreigner" and is composed by two Kanji. The word you can read in the Tux above means "Kan-ji" and is composed by Kan (= Han, "Han dynasty") and Ji (= writing).

2.3 Hiragana and Katakana (Kana)

While the Chinese language does not have any alphabet, the Japanese have created one (or better two) by simplification of kanji, which form syllables instead of consonants and vowels like in Latin, Greek or Cyrillic writing.

The only consonant that can stand alone is the "n". Other alone-standing consonants are not necessary in order to speak Japanese. This is the main reason why Japanese have more problems with languages that have relatively many consonants following each other like the German language than with Latin languages like Spanish or Italian. In order to speak my first name "Manfred" for instance Japanese would write (in his syllable-alphabet) "ma-n-fu-re-do", which still sounds acceptable to me, but in the case of my surname he or she would write "shu-re-n-bu-ru-gu", which is nearly unrecognizable.

Hiragana was the first alphabet that was created. It is used to write pre- or suffixes that come together with kanji or in order to write words where the writer does not know the correct Kanji. This is a big advantage for Japanese in comparison with a Chinese. The only way to write something where a Chinese knows the sound but not the correct character would be to use an incorrect character with the same sound, which can lead very easily to confusion.

Katakana is a more sharp-edged syllable-alphabet, which is used for foreign words that are used in Japanese (like "pan" = bread, "arubaito" = manual work or "kurisumasu" = Christmas), foreign names and for publicity.

2.4 Romaji

The word "Roma-ji" is composed by "Roma" (= Rome) and "Ji" (= writing) and means "the writing of the Romans", which is Latin.

The word "Linux" and some few other IT related words are written in Romaji. Imagine the word "ri-nu-ku-su" - almost unrecognizable!

.

Hiragana					Katakana					Romaji				
あ	い	う	え	お	ア	イ	ウ	エ	オ	a	i	u	e	o
か	き	く	け	こ	カ	キ	ク	ケ	コ	ka	ki	ku	ke	ko
さ	し	す	せ	そ	サ	シ	ス	セ	ソ	sa	shi	su	se	so
た	ち	つ	て	と	タ	チ	ツ	テ	ト	ta	chi	tsu	te	to
な	に	ぬ	ね	の	ナ	ニ	ヌ	ネ	ノ	na	ni	nu	ne	no
ま	み	む	め	も	マ	ミ	ム	メ	モ	ya		yu		yo
や		ゆ		よ	ヤ		ユ		ヨ	ra	ri	ru	re	ro
ら	り	る	れ	ろ	ラ	リ	ル	レ	ロ	o				u
お			う		オ				ウ	n				
ん					ン									

2.5 Numbers

There are of course Japanese numbers which are derived from the Chinese. Nevertheless nowadays the Japanese use mostly Arabic numbers in western style.

Arabic	Kanji	Reading
1	一	いち；ひとつ
2	二	に；ふたつ
3	三	さん；みっつ
4	四	し；よっつ
5	五	ご；いつつ
6	六	ろく；むっつ
7	七	しち；ななつ
8	八	はち；やっつ
9	九	きゅう；ここのつ
10	十	じゅう；とう

© schulenburg-trauung.de

The Kanji 死 - read „し" like the number 4 - means „death". This is the reason, why the number 4 in Japanese is a number of misfortune like the number 13 in western countries.

3 Files to be downloaded

3.1 Packages needed

Canna [A Japanese character set input system]:
http://canna.sourceforge.jp/
(http://downloads.sourceforge.jp/canna/9565/Canna37p3.tar.bz2)
canna-libs [The runtime library for Canna]:
http://dc.qut.edu.au/cgi-bin/rpm.cgi
(rpm-package: to be transformed e.g. with 'alien')
Or just download the files I have already converted: Canna-libs-3.7p3.tgz
canna-devel [Header file and library for developing programs which use Canna]:
http://dc.qut.edu.au/cgi-bin/rpm.cgi
(rpm-package: to be transformed e.g. with 'alien')
Or just download the files I have already coverted: Canna-devel-3.7.tgz
cannadic ['cannadic' is a Japanese kana-kanji conversion dictionary for Canna]:
http://cannadic.oucrc.org/cannadic-0.95c.tar.gz
Kinput2 [Kinput2 is an input server for X11 applications in order to enable Japanese text input.]:
ftp://ftp.sra.co.jp/pub/x11/kinput2/
kinput2-canna-wnn6 [kinput2 for both Canna and Wnn6]
http://rpmfind.net/linux/rpm2html/search.php?query=kinput2-canna-wnn6
libspt [With libspt you can drop all privileges of your terminal emulator]:
http://www.j10n.org/libspt
(I downloaded this one: http://www.j10n.org/libspt/libspt-1.1.tar.bz2)
wnn6-SDK [Wnn6 Client Library] => some files are needed
http://rpmfind.net/linux/rpm2html/search.php?query=libwnn6.so.1.0

(rpm-package: to be transformed e.g. with 'alien')
Or just download the files I have already converted: Wnn6-SDK-1.0.tgz

3.2   packages one should know about

The packages listed here are more interesting and beneficial for those who wish to form one's own Japanese distribution.
canna-yubin [postal code extension dictionary for Canna]
http://dennou-k.gaia.h.kyoto-u.ac.jp/library/Linux/debian-jp/pool/main/c/canna-yubin/
Canna-shion
http://packages.debian.org/stable/utils/canna-shion
Shion is a set of dictionaries to assist Canna. It has many more words than the original dictionaries of Canna and renders Canna more intelligible.
cannadict [This package holds the basic Canna conversion dictionaries]:
http://www.openbsd.org/3.5_packages/m68k/cannadict-3.5b2.tgz-long.html
Those dictionaries should installed under /var/dict/canna (or where your dictionaries are). Although it contains the basic dictionaries you do not really need it, because the most important dictionaries are already shipped with canna.

3.3   Scim

See „Scim"

3.4   Fonts

You will find many links to Unicode fonts supporting Japanese here:
http://www.travelphrases.info/gallery/Fonts_Japanese.html
As you are probably in favour of true-type fonts, downloading a Unicode font like kochi-gothic and kochi-mincho is recommended.
kochi-gothic:
ttf-kochi-gothic-naga10_0.2.20020326-1.tar.gz
Source: http://ftp.debian.org/pool/non-free/t/ttf-kochi-gothic-naga10/
kochi-mincho:
8.675.822 kochi-substitute-20030809.tar.bz2
Source: http://sourceforge.jp/projects/efont/files/
Arial Unicode:
23.274.572 ARIALUNI.TTF
Source: Comes with Microsoft's Office 2000, FrontPage 2000, Office XP and Publisher 2002
Cyberbase [It omits the Chinese, Japanese and Korean characters]
309.572 Cyberbas.ttf
Source: ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/
Cyberbit
13.353.848 Cyberbit.ttf
Source: ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/
bdf-unicode:
28.499 CaslonBlack.bdf.tar.gz
152.774 CaslonBold.bdf.tar.gz
326.313 CaslonItalic.bdf.tar.gz
882.539 CaslonRoman.bdf.tar.gz
373.975 CaslonRomanSmallCaps.bdf.tar.gz
Source: http://bibliofile.duhs.duke.edu/gww/fonts/Unicode.html
(ttf also available)

kochi:
6.296.436 kochi-ttf-20020503.tgz
Source: http://www.dan.co.jp/~dankogai/NetBSD/macppc/packages/fonts/
monafont:
808.897 monafont-2.90.tar.bz2
Source: http://sf.gds.tuwien.ac.at/m/mo/monafont/
tuxpaint:
tuxpaint-ttf-japanese-2003.01.24.tar.gz
Source: http://www.newbreedsoftware.com/tuxpaint/download/fonts/

3.5   Other Software

Alien:
Alien is a program that converts between the rpm, dpkg, stampede slp, and slackware tgz file
formats.
If you do not have the RPM Package Manager installed on your favourite distribution, like I
hadn't on Rock Linux, consider to use it via a Live-CD like Knoppix.
http://www.kitenet.net/programs/alien/
Freetype:
The FreeType project develops free, portable and high-quality software solutions for digital
typography.
Source: http://www.freetype.org/
or http://freetype.sourceforge.net
I used the version: freetype-1.3.1.tar.gz
Freetype-devel:
freetype-devel-1.3.1
(rpm-package: to be transformed e.g. with 'alien')
I used: ftp://fr2.rpmfind.net/linux/redhat-archive/6.2/en/os/i386/RedHat/RPMS/freetype-
devel-1.3.1-5.i386.rpm
Or just download the files I have already converted: freetype-devel-1.3.1.tgz
We need the file /usr/lib/libttf.a, which had to be copied - in my case - to /usr/local/lib
Freetype-utils:
freetype-utils-1.3.1
(rpm-package: to be transformed e.g. with 'alien')
You may use: ftp://rpmfind.net/linux/contrib/libc6/i386/freetype-utils-1.3.1-1.i386.rpm
Or just download the files I have already converted: freetype-utils-1.3.1.tgz
ttmkfdir:
ttmkfdir is a tool to create valid and complete fonts.dir files from TrueType fonts.
http://www.joerg-pommnitz.de/TrueType/xfsft.html
(download the file http://www.joerg-pommnitz.de/TrueType/ttmkfdir.tar.gz)
Ghostscript (application, libraries and fonts):
Ghostscript is an interpreter for PostScript and PDF.
http://ghostscript.com/
http://www.ctan.org/tex-archive/support/ghostscript/3rdparty/
http://www.cs.wisc.edu/~ghost/doc/AFPL/get700.htm
I used the following files:
- ghostscript-8.00.tar.gz
- jpegsrc.6vb.tar.gz
- libpng-1.2.5.tar.gz
- zlib-1.1.4.tar.gz
- ghostscript-fonts-std-6.0.tar.gz

GTK+ and glib:
GTK+ is a multi-platform toolkit for creating graphical user interfaces (I used gtk+1.2.9).
GLib is the low-level core library that forms the basis of GTK+ and GNOME. It provides data
structure handling for C, portability wrappers, and interfaces for runtime functionalities such
as an event loop, threads, dynamic loading, and an object system.
ftp://ftp.gtk.org/pub/gtk/v1.2
ftp://ftp.gtk.org/pub/gtk/v1.2/glib-1.2.10.tar.gz
(Omniprint needs glib-config, which is a part of glib).
Omniprint:
The Omni printer driver provides support for over 450 printers using the Ghostscript
framework.
http://omniprint.sourceforge.net/
I used the version Omni-0.9.2.tar.gz.
LPRng:
The LPRng package contains an enhanced, extended and portable implementation of the
Berkeley Line PRinter (LPR) print spooler. This is
useful for queuing print jobs.
http://www.lprng.com/DISTRIB/LPRng/LPRng-3.8.28.tgz
OpenOffice:
OpenOffice is - of course - not necessary in order to show the abilities of canna and kinput2.
Nevertheless, I would consider using it for its great possibilities, as shown in the tutorial. It is
also recommendable to download the java runtime environment before installing OpenOffice.
http://www.openoffice.org/
I used the OpenOffice 1.1.15. OpenOffice 2.x works anologously.
Java Runtime Environment:
As the Sun version could not be installed under Rock Linux I used the Blackdown version,
which can be found here: http://www.blackdown.org/java-linux/java-linux-d2.html
I used the version: j2re-1.4.2-rc1-linux-i586-gcc3.2.bin

4 Installation

4.1 Preliminary Works

4.1.1 paths

Eventually you still have to make sure that programs like „xmkmf" can be found:
```
$ export PATH="$PATH:/usr/X11R6/bin"
```
The command „echo $PATH" shows you which paths are already defined.
If – as happened to me with Rock Linux, obviously an error – after sending this command, all
others paths have been erased, restart.
Manpaths can be added in the same way. The command „echo $MANPATH" shows the
already added manpaths. So with the following command you can add the manpath of canna:
```
$ export MANPATH="$MANPATH:/usr/local/canna/man"
```


4.1.2 Transformation of some rpm-packages

In the case that you use a Linux distribution that does not use rpm you can do the
transformation with the program alien using the command „alien -t file.rpm".
rpm-packages you will need are:
```
Wnn6-SDK-1.0-27.1 (or higher)
canna-libs
```

```
canna-devel
```

4.2   The Installation of the canna and kinput related software

4.2.1   canna-devel

Unpack canna-devel 3.7 for Turbolinux (as a proposal). You will find a folder „usr". Copy it to / (root).
```
cp -R usr /
```

4.2.2   canna-libs

You will find a folder „usr". Copy it to /.
```
cp -R usr /
```

4.2.3   ibspt1.1

Unpack: `$bzip2 -d libspt-1.1.tar.bz2`
```
tar -xvf libspt-1.1.tar
```
Attention: if you install directly, you will receive an error message like „invalid group 'wheel'".
Solution: delete the Makefile and edit Makefile.in. In the file Makefile.in substitute „wheel" by „users", if the user group is called so. Then go on as follows.
```
$./configure
$make
$make install
```

4.2.4   Wnn6-SDK-1.0-27.1

(to be transformed to tar.gz)
You will find a folder „usr". Copy it to /.
```
cp -R usr /
```

4.2.5   Canna27p3

```
xmkmf
make canna
```
`su` (the following commands have to be made by root, the others can be made by a user)
```
make install
make install.man
mkdir /var/lib/canna
mkdir /var/lib/canna/dic
cp /usr/local/canna/share/canna/dic/default.cbp
/var/lib/canna/dic
cp /usr/local/canna/share/canna/default.canna /var/lib/canna
```
as root: `export PATH="$PATH:/usr/local/canna/bin` **(but: see below)**
```
Hint:
```
With T2-Linux and Rock-Linux as well there occurred an error: all other paths were "forgotten":
`echo $PATH` told me, that there was only the new PATH
When I re-logged in, I saw that all the old paths were there again and the new path was lost. The best solution was to add the new path **directly in the file /etc/profile:**
```
PATH=$PATH:/usr/local/canna/bin
```

4.2.6 kinput2-v3.1

Attention: delete the commands concerning sj3-files in Kinput2.conf (important!),
then do:
```
$xmkmf
$make Makefiles
$make
$make install
```

4.2.7 cannadic

1.) add the directory `var/lib/canna/dic/canna`
2.) `cp * /var/lib/canna/dic/canna`
3.) `make maindic`

4.2.8 Copy default.canna

1.) `cp .../Canna37p3/misc/default.canna /var/lib/canna`
2.) `cp /var/lib/canna/default.canna ~/.canna`
3.) Modify the.canna-file in your home directory as you so wish.

4.3 The installation of other - especially typing related – software

4.3.1 Freetype-devel

/usr/lib/libttf.a is needed => to be copied to /usr/local/lib
gunzip freetype-devel1.3.1.tar.gz
tar -xvf freetype-devel1.3.1.tar
cd usr
mkdir local
mv lib local
mv include local
Now copy usr to / (root-directory)
cp -R usr /

4.3.2 Freetype-utils

unpack
```
cp * /usr/bin
```

4.3.3 Freetype

```
gunzip freetype-1.3.1.tar.gz
tar -xvf freetype-1.3.1.tar
cd freetype-1.3.1
cd test
vi ftdump.c
```

Now at "`footprint.##field`" delete the two rhombus in front of the word "field",

```
./configure --enable-static (enable-static is important!)
make
make install
```

4.3.4 ttmkfdir and true type fonts

Unpack ttmkfdir, then change the "Makefile" of ttmkfdir:
(in my case:)
```
FREETYPE_BASE = /usr/local/include
FREETYPE_INCL = /usr/local/include/freetype
FREETYPE_LIB = /usr/local/lib/libttf.a
```
Now install ttmkfdir:
`make` (or: `make FREETYPE_BASE = /usr/local/include/freetype`)
then:
`cp ttmkfdir /usr/X11R6/bin` (where ttmkfdir is the compiled file)
Now copy the ttf-files you need (kochi-mincho, kochi-gothic, Arial unicode) to
/usr/X11R6/lib/X11/fonts/TTF
then: `ttmkfdir > fonts.dir`
restart X
```
ttmkfdir -o fonts.scale
mkfontscale
```
(`mkfontscale -b -s -l` is equivalent to `mkfontdir`)
then as root: `fc-cache --force --verbose`

4.3.5 Ghostscript, glib, Omniprint

4.3.5.1 zlib 1.1 or higher

```
./configure
make
make install
```

4.3.5.2 jpeg

```
mkdir /usr/local/man/man1
```
then:
```
./configure
make
make install
```

4.3.5.3 libpng

```
cp scripts/makefile.std makefile
make
make install
```

4.3.5.4 gtk+1.2.9

```
./configure
make
make install
```

4.3.5.5 glib 1.2.10 or higher (needs GTK)

(if an error occurs: install patch (see typical errors)
If your glib version does not have to be patched go on as follows:
```
./configure
make
make install
```

4.3.5.6 adapting glib-config

Omniprint needs glib-config. I found it in my case under /usr/opt/gnome/bin, but Omni looks for it under /usr/bin.

```
cp /opt/gnome/bin/glib-config /usr/bin
```

(in some distributions it is: `cp /usr/local/bin/glib-config /usr/bin`)

4.3.5.7 Omniprint

```
./setupOmni
```

4.3.5.8 ghostscript 8

```
./configure
make
make install
```

4.4   Installing OpenOffice and Java - not necessary but very useful

4.4.1   Installing Java support

(assuming you use Blackdown Java)
assumption: Installation to /usr/local

1. copy thefile 'j2re-1.4.2-...' to /usr/local (must be directly there, not in a hand-made subfolder)
2. Make it executable, if not done already: `chmod +x j2re-...`
3. extract the contents with the command: `exec ./j2re-1.4.2-...`
4. add j2re1.4.2/bin to the path:

```
export PATH=/usr/local/j2re1.4.2/bin:PATH
```

4.4.2   Installing OpenOffice 1.1.x

`./setup -net` (as root)
If not displayed, select manually the java path (here: /usr/local/j2re1.4.2)
`./setup` (as user in the program directory, in my case /opt)
(Configuration of OpenOffice: see „typing Japanese")

5 Typical Errors

5.1   Errors concerning japanese related software

5.1.1   I get an error message when starting kinput2

```
abc@ajedrea:~$ export XMODIFIERS="@im=kinput2"
abc@ajedrea:~$ LANG=ja_JP LC_ALL=ja_jp /usr/X11/bin/kinput2 -xim -kinput -canna
&
[1] 2020
abc@ajedrea:~$ Warning: gcannaf€ò¥Þ¥Š¥ó¥Ë€Ç€-€Þ€»€ó€Ç€·€¿
Warning: gcanna€ò¥Þ¥Š¥ó¥Ë€Ç€-€Þ€»€ó€Ç€·€¿
abc@ajedrea:~$ █
```

Solution:

gcanna hat to be installed with the command "make maindic".

5.1.2   The command crxdic cannot be found

```
bash-3.00# make maindic
crxdic -m -o gcanna.cbd gcanna.ctd
make: crxdic: Kommando nicht gefunden
make: *** [maindic] Fehler 127
```

Solution:

This error happens only if the path /usr/local/canna/bin is missing.

Hint: With T2 Linux and also Rock Linux when I tried to add the paths with the command export PATH="$PATH:/usr/local/canna/bin" I always lost all other paths. When I logged in again, the old paths were there again, but the new path was missing. So I simply solved this problem putting in the new path in the file /etc/profile.

5.1.3   The cannaserver cannot be started

I get the error message, that the permission is denied.

```
kdecore (KAccel): WARNING: kde_g_bKillAccelOverride set, but received an event
ther than AccelOverride.

abc@ajedrea:~$ //usr/local/canna/sbin/cannaserver
abc@ajedrea:~$ Warning: //usr/local/canna/sbin/cannaserver: /var/log/CANNA0msgs
open faild
Permission denied
```

Solution:

The cannaserver has to be started as root.

5.1.4   I receive the error message "another 'cannaserver' is detected

```
bash-2.05b# /usr/local/canna/sbin/cannaserver

ERROR:
    Another 'cannaserver' is detected.
    If 'cannaserver' is not running,
    "/tmp/.iroha_unix/IROHA" may remain accidentally.
    So, after making sure that 'cannaserver' is not running.
    Please execute following command.

            rm /tmp/.iroha_unix/IROHA

bash-2.05b# ▮
```

Solution:

It just means that the cannaserver has already been started, so that there is no need to start it.

5.1.5   The file default.cbp is missing

I receive an error message, that the file default.cbp cannot be found:

```
bash-3.00$ LANG=ja_JP LC_ALL=ja_JP kinput2 -xim -kinput -canna &
[1] 3542
bash-3.00$ Warning: ¥íiŒ¥Þ»ú€«€ÊÊÑŽ¹¥ÆiŒ¥Ö¥ë(default.cbp)€¬¥ªiŒ¥×¥ó€ÇŒ-€Þ€»€óí£
```

By default, Canna uses this romaji-canna conversion table. If it is not found, you won't be able to create Japanese characters.

Solution:

cp /usr/local/canna/share/canna/dic/default.cbp /var/lib/canna/dic

5.1.6   The file default.canna is missing

```
bash-3.00$ LANG=ja_JP LC_ALL=ja_JP kinput2 -xim -kinput -canna &
[1] 25571
bash-3.00$ Warning: ¥·¥¹¥Œ¥à€Î¥«¥¹¥¿¥Þ¥€¥ª¥Ö¥i¥€¥ë /var/lib/canna/default.canna
€¬Âž²ß€·€Þ€»€óí£
Warning: ¥·¥¹¥Œ¥à€Î¥íiŒ¥Þ»ú€«€ÊÊÑŽ¹¥ÆiŒ¥Ö¥ë€¬¥ª iŒ¥×¥ó€ÇŒ-€Þ€»€óí£
```

Solution:

cp /usr/local/canna/share/canna/default.canna /var/lib/canna
(See also "default.cbp)

5.2   Other Errors not concerning Japanese software

5.2.1   KDE is extremely slow

In some distributions this happens, when you install some network-related software.

Solution:

Check if your /etc/hosts file is correct:
When you have a static IP-address, the domain and hostname must be entered here like:
192.168.1.123 tux.domainname tux

If you have a dynamic IP-address or even if you have no network just enter the hostname after the local network:
127.0.0.1 localhost tux

5.2.2   lpc and lpstat not found

$ LC_CTYPE=ja_JP.UTF-8 /opt/OpenOffice.org1.1.4/program/swriter
... /usr/sbin/lpc: no such file or directory (in german: "lpc: Datei oder Verzeichnis nicht gefunden")
... lpc: command not found
... lpstat: command not found

```
abc@ajedrea:~$ LC_CTYPE=ja_JP.UTF-8 /opt/OpenOffice.org1.1.4/program/swriter
sh: line 1: /usr/sbin/lpc: Datei oder Verzeichnis nicht gefunden
sh: line 1: lpc: command not found
sh: line 1: lpstat: command not found
```

Solution:

Install lprng (see chapters on download and installation)

5.2.3   I receive an error when trying to compile glib-1.2.10.tar.gz

The error message:

```
gstrfuncs.c: In function 'g_printf_string_upper_bound':
gstrfuncs.c:870: error: syntax error before string constant
gstrfuncs.c:1037: error: syntax error before string constant
gstrfuncs.c:1080: error: syntax error before string constant
gstrfuncs.c:1111: error: syntax error before string constant
gstrfuncs.c: In function 'g_strdown':
gstrfuncs.c:1139: warning: pointer targets in assignment
differ in signedness
gstrfuncs.c: In function 'g_strup':
gstrfuncs.c:1155: warning: pointer targets in assignment
differ in signedness
gstrfuncs.c: In function 'g_strchug':
gstrfuncs.c:1314: warning: pointer targets in assignment
differ in signedness
gstrfuncs.c:1317: warning: pointer targets in passing argument
1 of 'strlen' differ in signedness
make[2]: *** [gstrfuncs.lo] Error 1
make[2]: Leaving directory `/home/landuchi/Progs/glib-1.2.10'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory `/home/landuchi/Progs/glib-1.2.10'
make: *** [all-recursive-am] Error 2
```

Solution:

Install the update you can find here:
http://www.linuxfromscratch.org/blfs/view/stable/general/glib.html

- uncompress and unpack the Software to be installed
- enter into the unpacked directory
- enter the following command:
  ```
  patch -Np1 -i ../glib-1.2.10-gcc34-1.patch &&
  ./configure --prefix=/usr &&
  make
  ```
- 4. Test the results: `make check`
- 5. enter as root:
  ```
  make install &&
  chmod -v 755 /usr/lib/libgmodule-1.2.so.0.0.10
  ```

6 Typing Japanese with an occidental keyboard layout

6.1   Preparatory work

Starting the canna server: `/usr/local/canna/sbin/cannaserver` (as root)
then input:
```
export XMODIFIERS="@im=kinput2"
LANG=ja_JP LC_ALL=ja_JP kinput2 -xim -kinput -canna &
```

```
bzw: LANG=ja_JP LC_ALL=ja_JP /usr/X11/bin/kinput2 -xim -kinput
-canna &
```
Leave the command window open.

Try with kwrite:

type: `LANG=ja_JP.UTF-8 kwrite&`

or: `LC_CTYPE=ja_JP.UTF-8 XMODIFIERS="@im=kinput2" kwrite` (or kedit in older distributions)

Try with OpenOffice (configuration of OpenOffice see below):

Assuming that OpenOffice is installed in the directory /opt/OpenOffice.org1.1.5/program/swriter:

`LC_CTYPE=ja_JP.UTF-8 /opt/OpenOffice.org1.1.5/program/swriter`

6.2   Writing with canna and kinput2

With "Shift" and "Space" you can change between Latin (Romaji) and Japanese.
Introduction of Kanji
„word in Romaji" [SPACE] [ENTER]
Introduction of Hiragana
„word in Romaji" [ENTER]
Introduction of Katakana
„word in Romaji" [down-arrow] [ENTER]

6.3   Special hints for OpenOffice Users

use Spadmin:

`/opt/OpenOffice.../spadmin`

Click on Fonts
Make known the fonts that are in the directory /usr/X11R6/lib/X11/fonts/TTF.
Select at least Arial Unicode, Kochi Gothic and Kochi Mincho.

start OpenOffice Writer:

`LC_CTYPE=ja_JP.UTF-8 /opt/OpenOffice.../program/swriter`

Select for instance Kochi Gothic.
Activate Asian language support:
GERMAN: Extras | Optionen | Spracheinstellungen | Sprachen
ENGLISH: Tools | Options | Language | Asian Layout

Make Japanese basic writing:

GERMAN: Extras | Optionen | Textdokument | Grundschriften (asiatisch)
ENGLISH: Tools | Options | text document | basic fonts (asian) [assumed]

and configure substitutional font table:

GERMAN: Extras | Optionen | Schriftarten | Ersetzungstabelle anwenden
ENGLISH: Tools | Options | OpenOffice.org | font replacement
for instance: substitute "Bitstream Vera Sans" through "Kochi Mincho"
(It is not true - as told in some OpenOffice howtos - that the font `Andale Sans UI` has to be substituted.)

6.4   Entering Unicode hex code

Since the dictionary used by kinput2 contains only the around 2000 most important kanji, sometimes it is necessary to use Unicode hex code, because the Japanese fonts offer far more than those basic kanji. You may need it, for instance, for ancient family names. An example:

Mr. Masui is Mr. Schulenburg's Japanese teacher.
桝井さんシュレンブルグさんの日本語の先生です。|

The kanji "masu" was not among the around 2000 most important kanji. So I had to enter the hex code, which in this case was 685D.
How to enter Unicode hex code in X-programs depends on the software.
Input Unicode hex code with **Scribus**: Press **F12**, release and enter the four digit unicode hex code. The symbol appears immediately, without pressing Enter. From there I could copy it to OpenOffice 1.1.x. I did not manage to enter Unicode hex code in OpenOffice 1.1.x.
OpenOffice 2 is unfortunately only distributed as rpm files, so I did not try. There it probably works with **Ctrl**+**Shift**+**Unicode-ID**.

Unicode-Input with **kinput** and **canna**: Press Home button (Pos1). You will get the following window:

Now select No. 2 in order to input Unicode-Codes.

7 Another Method: Scim

In the following I am going to explain how to enter any Asian font with SCIM. Scim is a software that provides an input method for more than 30 languages, including complex Asian fonts. For Japanese I think that canna is better, but when you have to deal with different languages at the same time it provides a good option.
Like in the case of canna my test distribution was T2 Linux, a derivate of Rock Linux. As it has no aim to support Asian languages by default, it should work as described in this manual with any other distribution.
Please consider that before installing SCIM, I installed the Japanese language support with canna and kinput2, so if something is missing, please refer to that installation guide. Please take also into consideration that the paths may differ in your Linux distribution.

7.1 Where to download scim

http://www.scim-im.org/
http://prdownloads.sourceforge.net/scim
http://sourceforge.jp/projects/scim-imengine/files/ (Scim-anthy, for Japanese: connects Scim to Anthy)
http://sourceforge.jp/projects/anthy (Anthy, for Japanese: converts Hiragana text to Kana Kanji mixed text)
http://sourceforge.net/project/showfiles.php?group_id=108454 (Scim-input-pad: add-on to input various symbols)
http://kasumi.sourceforge.jp/ (Kasumi, for Japanese: dictionary maintenance tool for Anthy)

7.2 Installing SCIM

Do not worry about version numbers, just install the latest, typing:
```
./configure --prefix=/usr
make
make install
```

If you just type "`./configure`" it can be, like in the case of T2 Linux, that the program is being copied to /usr/local. In this case "anthy" would not work. This programming style is quit rigid, but please take it into consideration.

scim-1.4.2
anthy-7100bb (for Japanese)
scim-anthy-0.8.0 (for Japanese)
scim-input-pad-0.1.0.tar.gz
kasumi-1.0 (for Japanese)
scim-tables-0.5.5

```
./configure
(cp /usr/local/lib/libscim* /usr/lib)
make
install
```

In T2 Linux I had to copy the libraries from /usr/local/lib to /usr/lib although the LIBPATH was perfectly written in the Makefile. Without this step the libraries could not be found and the `make` command imparted an error message.

scim-Chinese-0.4.2 (for Chinese)
scim-pinyin-0.5.91 (for Chinese)
Install the true type fonts you need

copy ttf-files to /usr/X11R6/lib/X11/fonts/TTF

```
gunzip *
cp *.ttf /usr/X11R6/X11R6/lib/X11/fonts/TTF
ttmkfdir > fonts.dir
ttmkfdir -o fonts.scale
fc-cache --force --verbose
```

If you are in X you have to restart the X font server via:

```
/etc/init.d/xfs restart
```

7.3   Starting the Scim environment

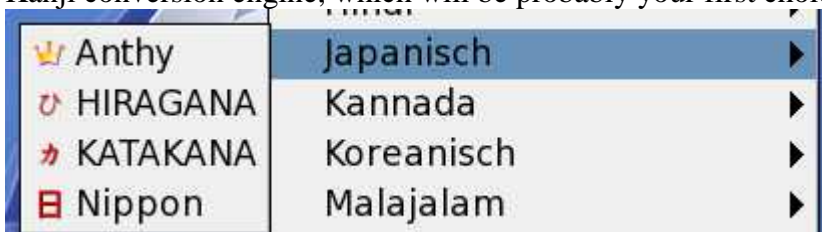1.) `$ /usr/local/bin/scim -d` => starts the SCIM daemon with socket frontend
2.) `$ XMODIFIERS="@im=SCIM"`
[In order to see what is running type "`scim -l`"]

7.4   Writing Japanese

`LC_CTYPE=ja_JP.UTF-8 /opt/OpenOffice.org1.1.5/program/swriter`
You can select between Hiragana, Katakana, Nippon and Anthy. Anthy is a Romaji-Kana-Kanji conversion engine, which will be probably your first choice.

Mr. Masui is Mr. Schulenburg's Japanese teacher.
桝井さんシュレンブルグさんの日本語の先生です。

7.5   Writing Chinese

7.5.1   Starting OpenOffice (or s.th. else) with Chinese Unicode support

```
$ LC_CTYPE = zh_CN.UTF8
/opt/OpenOffice.org.1.1.4/program/swriter
```
[zh_CN means the simplified Chinese, zh_TW means traditional Chinese]
CN = people's republic of China, TW= Republic of China = Taiwan

7.5.2   Writing Chinese

1.) Change between normal and Scim input via "Ctrl + Space"
As soon as you do this a little window appears where you can select language etc.:
2.) Select the Chinese font
3) Try to enter a Chinese word with pinyin

7.6   Writing other languages with Scim:

When you enter "Ctrl+Space" you will get a panel like the one below. It should appear on the bottom on the right side. There you can select the language and make other
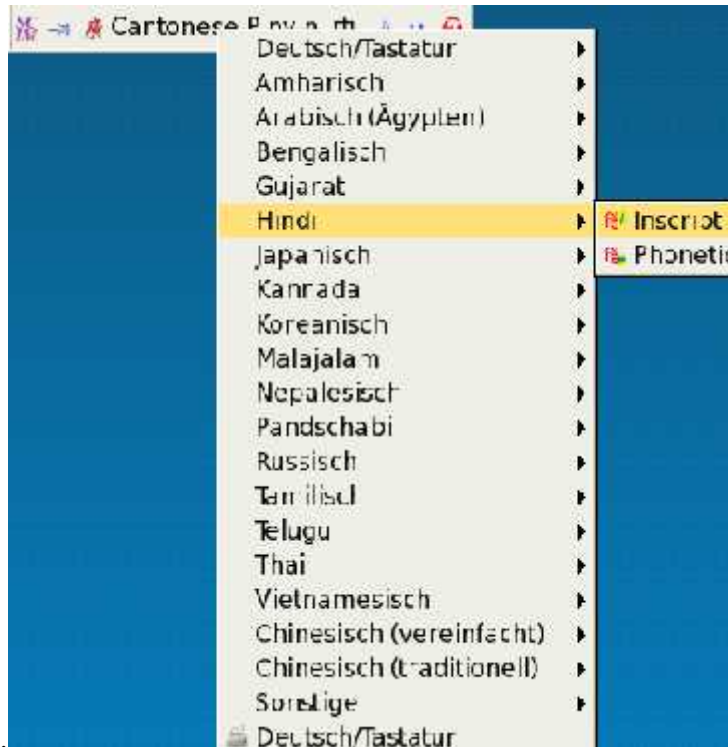
configurations.

7.6.1 Starting OpenOffice with Hangul (Korean):

```
$ LC_CTYPE = ko_KR.UTF8
/opt/OpenOffice.org.1.1.4/program/swriter
```


7.6.2 Starting OpenOffice with Thai:

```
$ LC_CTYPE = th_TH.UTF8
/opt/OpenOffice.org.1.1.4/program/swriter
```


7.6.3 Starting OpenOffice with Saudi or Egytian Arabic:

```
$ LC_CTYPE = ar_SA.UTF8
/opt/OpenOffice.org.1.1.4/program/swriter (Saudi)
$ LC_CTYPE = ar_EG.UTF8
/opt/OpenOffice.org.1.1.4/program/swriter (Egyptian)
```


8 Outlook

In my opinion, canna and kinput are very mature. Unfortunately it is almost only documented in Japanese and even the Japanese documentation seems to be very poor, so that there is still quite a lot to do. Nevertheless I do not doubt that in spite of the efforts of the scim maintainers, canna and kinput will be the most important

software packages for writing Japanese for all Linux distributions. As we could see it is possible to use these software packages with any Linux distribution, so I hope that the maintainers of the relatively small Linux distributions will find the interest in multilingualization.

- some tasks for Linux distributors -
- Forming packages for easier installation: finding out which files are exactly needed and writing installation scripts
- Writing manuals

- some tasks for canna and kinput maintainers -
- maintain the existing pages concerning canna and internationalizing them, i.e. at least in English
- the same for the cannadic pages
- build up a central page for kinput
- writing manuals