MOKE - Mark's Own Kanji Editor

User Reference Guide - 1.1

Mark Edwards

KiCompWare

1.0 Overview

MOKE is Japanese language editor for the PC with a command set similar to the UNIX editor 'vi'. The name MOKE is partially derived from JOVE (Jonathan's Own Version of Emacs) and shows some lack of creativity. However the only reward I may get in writing this editor may be that it bares my first name. This User's Reference Guide was entered using MOKE and MicroEmacs 3.10.

This program is the result of many long nights in front of my PC and is based on several other programs (view, rtk, fk). This could not really be a Japanese editor without the Simple Kana to Kanji conversion program by Masahiko Sato (masahiko@sato.riec.tohoku.junet) and the Wnn dictionaries.

MOKE allows input of english, hiragana, katakana, and most importantly kanji. As far as I know there is "no good" way of entering kanji on a computer. All methods require a number of key strokes first to enter kana, then more strokes to select a kanji character. In that respect this editor is no different. It should be noted that this is not a word processor, but a text editor.

MOKE can also aid in the entry of Kanji. When writing japanese I frequently must consult a dictionary (or my wife) to find out where the kanji stops and the kana starts. Many times I know the onyomi of the kanji. If the kanji is displayed in on the screen and the cursor is below it by using the F7 key all the on/kun yomi's of the kanji will be displayed in the ENTERLINE line. By using the ALT-m combination to mark a place and the F7 key the japanese to english translation of the word (if found in the dictionary) will be displayed as well as the yomikata of a kanji word.

As the UNIX "vi" editor is a mode editor so is MOKE. Perhaps the major reason for this is the need to enter Japanese kana and kanji. Though it doesn't necessitate it. MOKE has two modes: CURSOR mode, and INSERT mode. In CURSOR mode all movement in the file is done, as well as editing of existing text. In INSERT mode all entering or inputing of new text is done.

MOKE may be freely distributed. The source isn't copyrighted or restricted in any way. If you pass the program along, please include all the documentation and source. The only connection that this program has with the University of Wisconsin is that I am a full time employee there.

Section 2.0 states the System Requirements and Section 3. details the installation of MOKE. While the remainder of this document describes the operation of the editor. If you have any questions, problems, bug reports, or other please report them to me.

mark edwards [マーク。エドワズ]

Systems Programmer
University of Wisconsin - Madison (MACC)
1210 W. Dayton St.
Madison, Wi. 53706 (USA)
Phone: (608) 262-7585

Email:
Internet: edwards@macc.wisc.edu
Bitnet: edwards@wiscmacc.bitnet

Note: MOKE and its utility programs have more than 5000 lines of code. The code is still undergoing development. MOKE the editor was started around the middle of January 1990. Being that this code is only around 3 months old it is not surprising that MOKE still has some bugs. Most of the bugs while being bothersome are harmless. Some bugs, however, may lock up your PC causing loss of text. So please insure that you have adequate backups and that you save the current buffer every once and a while. (If you keep me informed of bugs you find I'll keep everyone else informed, if not fix them as soon as the bugs surface.)

2.0 System Requirements

- IBM PC (XT, AT, 386, PS/2, or clone)
- MSDOS 3.1 or higher
- 640k Memory (Probably works with less)
- Graphics Adapter (Hercules, CGA, EGA, VGA)
- Harddrive or one high density floppy disk drive (depending on which Kana to Kanji dictionary used)

Note: Because MOKE uses bitmapped fonts the faster your PC the better the response time will be.

3.0 Installing MOKE

MOKE is fairly easy to install, but since it uses many files the installation is not trivial. There are several data files including the Kana/Kanji bitmaps, the SKK or Wnn jishou, the indexes for the SKK and Wnn jishou, the reverse lookup file for kanji on/kun yomi, and the japanese to english dictionary, and also has kana yomikata for some japanese words. For ease put all the data files in a directory called "\kanji". See section 3.1 for a list of data files for more information.

3.1 Data Files

The following is the list of data file for MOKE, the distribution you receive may not have all these files. The files that are marked with a plus are necessary for MOKE to work as a Japanese Editor. The ".ind" files can be built from the Utility programs.

	FILE	Description
+	\kanji\k16jis1.fnt	- 16x16 JIS Level 1 bitmap fonts (from KD dist)
+	\kanji\k16jis2.fnt	- 16x16 JIS Level 2 bitmap fonts (from KD dist)
	\kanji\k14bdf	- 14x14 JIS Level 1 & 2 bitmap fonts (from X-window)
	\kanji\k24bdf	- 24x24 JIS Level 1 & 2 bitmap fonts (from X-window)

(k14, k24 are still under development and are not being

release yet)

```
+  \kanji\skkjis        - Simple Kana to Kanji dictionary
+  \kanji\skkjis.ind    - index into skkjis
   \kanji\wnndict       - Wnn and SKK combined dictionaries
   \kanji\wnndict.ind   - Index into wnndict dictionary

   \kanji\mydict        - Alternate Dictionary to search for Kana to kanji
                          lookup

   \kanji\revtab        - reverse lookup index to skkjis
                          (to display on/kun)
   \kanji\wsktok        - combined Wnn & SKK Kanji to kana dictionary
   \kanji\wsktok.ind    - index into above file
   \kanji\skkktok.dat   - SKK Kanji to kana dictionary
   \kanji\skkktok.ind   - index to above file

   \kanji\edict         - japanese to english dictionary
```

3.1.1 Data Files and Configuration

A. When disk space is at a premium (ex. floppy based systems)

skkjis, skkjis.ind

On the command line specify the -s option or use the "dictionary" parameter in "moke.rc".

B. When you have plenty of disk space

wnndict, wnndict.ind, wsktok, wsktok.ind, revtab, edict

This is the default and my normal configuration.

3.1.2 Making the index files

The most important data file for MOKE is the kana to Kanji dictionaries, "skkjis", or "Wnndict" . I would like to distribute the Wnn dictionaries and skkjis dictionaries as is but I haven't found a suitable sorting program. I may have to write my own. In the mean time I am distributing a sorted version of each dictionary. If the index file to these files is not included with the distribution you receive you can make it by running the "wnn_ind" program.

```
wnn_ind \kanji\skkjis  (or \kanji\wnndict)
```

yields the file "wnn_ind" then copy and rename the file appropriately.

```
copy wnn_ind \kanji\skkjis.ind (or \kanji\wnndict.ind)
```

Making "revtab" for skkjis

```
revtab  \kanji\skkjis
copy revtab \kanji
```

for wnndict

```
revtab \kanji\wnndict
copy revtab \kanji
```

3.2 Graphics Driver

The Graphics Driver file (BGI) is located by default in current the directory or by the environment variable "bgi". For example if the bgi graphics interface files are located in the \kanji directory

3.3 Moke.rc

Like the Unix VI MOKE also has a startup file where additional options can be specified. Some of the options are similar to the switches on the command line, but provide an easier way to start MOKE up rather than specifying each on the command line. Each of the option should be on a single line.

Option	Possible values (The default value are listed first)
filetype	Unix/MSDOS
codetype	EUC/JIS
graphicstype	HGC/CGA/EGA/VGA
drivepath	(not implemented yet)
dictionary	SKKJIS/WNNDICT (wnndict is combination of skkjis & wnn)
edict	Yes/no (If using the Jdict functions and have a edict)
mydict	Yes/no (default is no)
fonthalfsize	9 for readibility (default is 8) The font size is 16x16 for kanji, 8x8 for ascii

4.0 Using MOKE

See Section 3.0 for information on how to install moke before trying to start MOKE. MOKE can be started up at the MSDOS prompt by simply typing "moke". You can also ask it to load a file into the buffer upon startup as shown below.

```
moke myfile
```

MOKE can be given some arguments (or parameters) on the command line. The arguments include "-j" to load in a JIS compatible file or "-e" for a Extended Unix Code (EUC) compatible file.

4.1 MOKE’s Screen

MOKE’s screen is currently divided into three parts, the editing window, the STATUS line, and the ENTERLINE. The editing window or screen is where all editing, and displaying of the buffer takes place. The STATUS line displays information useful in using the program. It tells you the current mode (INSERT or CURSOR), the current INSERT mode state (hiragana, katakana, or ascii), and a one or two word help for some of the command keys or function keys.

The ENTERLINE is a general purpose line where the entering of romaji or ascii takes place, displaying and selecting of the Kanji Choices for a particular romaji string, the displaying of on/kun yomi of a single Kanji, and the display of the english translation of words found in the japanese to english dictionary as well as the yomikata of that word.

4.2 Getting Help in MOKE

Help is available in both CURSOR and INSERT mode. To get a brief display of the available commands for each mode press F1. For more detailed help press "h". To continue on in the current mode press any other key.

The following is a list of the current help topics.

1. Entering Hiragana
2. Entering Katakana
3. Entering ascii
4. Entering JIS ascii
5. Entering Kanji
6. Entering Punctuation etc.
7. Kanji Guess Lookup
8. English to katakana lookup
9. Entering Tutorial #1
10. Verb Conjugation

You may enter additional help topics up to Twenty. Twenty is the number of topics that will fit in a single column. This interface is still primitive. There will be enhancements to this in the future.

4.2.1 The help.hlp file

A help file is potentially any EUC file. Most of the current help files were derived from this document. The format of "help.hlp" is as follows.

<help file name>.hlp <40 character or less description of the help>

Example:

```
        tut.hlp         Entering Tutorial #1
```

5.0 Quick Tutorial

To run program type:

```
  moke <filename>
```

where <filename> is an optional argument. If <filename> is supplied it will read the file into the buffer.

It will open various files and load the kanji fonts. If everything goes right the text in the file will be displayed on the screen with the the cursor ('^' for ascii characters, '^^' for japanese characters). Initially you are in the CURSOR mode. In the CURSOR mode you can move about in the file with the arrow keys. To enter something type i (for INSERT MODE). In order to exit moke you must be in CURSOR mode. To change INSERT mode to CURSOR mode press the ESCAPE key. To exit press F9. If the buffer has been modified it will ask you if you really want to leave. If you want to save the buffer type 'n' and use the F10 key to save the file (it prompts for a file name). Then press F9 and MOKE will exit.

The rest of this section will take you through entering two lines of japanese text.

5.1 Entering a simple Japanese phrase

Let's see how to enter the following japanese sentence using moke

あした、わたしは東京に行きます。

(Ashita, watashi wa toukyou ni ikimasu)

Start moke with no filename and we will then have a fresh buffer to play around with. The first word to enter is あした. To enter we must first go to the INSERT mode. Typing an 'i' changes the mode to INSERT mode. When you pressed the 'i' key the "Enter:" prompt was displayed on the ENTERLINE or the last line of your screen. Now to enter あした simply type "ashita" using romaji as shown below.

Enter: ashita

Press return and the hiragana for ashita appears on the first line of the screen. The comma can be entered as is. Enter it and press return. Next enter "watashi", press return. The "wa" must be entered like it appears in hiragana that is は (ha), press return.
Now we get to the first kanji word in the sentence. Enter toukyou just as you would any other word but instead of using the return key to send the line to the screen use the F2 function key. The romaji string toukyou has only one mapping in the SKK jishou so it is displayed immediatedly. Had there been more than one mapping the choices would have been displayed in the ENTERLINE and selection is done with the arrow keys and the return key.

Now let's finish the sentence. Next is "ni", enter it like "ashita" and press the return key. Now we have the last word "ikimasu". Since this word uses both kanji and hiragana to represent the word entering it is slight different from kanji only words. To lookup the kanji あ in the dictionary enter the romaji for the kanji part of the word then enter the first romaji character of the hiragana portion of the word which is a "k". But capitalize it. Capitalizing it tells the program that this character should be treated differently and should not be translated into hiragana. Press the F2 key. Two choice will appear on the ENTERLINE. The first choice is the question mark (?) if you select this by pressing the SPACE BAR or the return key nothing will be selected. So use the right arrow and move the cursor under あ. Press the SPACE BAR or the return key. Lastly enter the hiragana porition of "ikimasu" with a period and press the return key.

If you think you've got it skip the next section otherwise for another look at entering continue on.

5.2 A second example

今、私はカルフォルニアに住んでいます。

(Ima, watashi ha California ni sunde imasu.)

If you have just finished the first sentence press F5 for a new line. Having to press a key other than the enter key for a newline is rather kludgy, but that is how it currently works. The first word to enter is

今. On the enterline type "ima" and then press the F2 key. The kanji for "ima" should appear. Enter "watashi" the same as "ima" and press the F2 key. We want the katakana version of California so we do this by surrounding "California" with tildes.

Enter: ~california~

The katakana version of California is in the SKKjis, but if we didn't know we could check anyways. Press the F2 key. The katakana for California should have appeared on the upper screen. Next is "ni". Enter it and press return. Now we get to "sunde" which is a verb. The dictionary form of the verb is "sumu", so we type in "suM" and press the F2 key. This time there are three possible choices. The correct choice is the 2nd one. Position the cursor underneath it and press return. You can finish the rest of the sentence by typing in "ndeimasu.".

6.0 INSERT mode in detail

This section describes the input method of MOKE. Entering text in MOKE is done in the INSERT mode. The normal mode of MOKE is CURSOR mode or command mode. Several command keys invoke the INSERT mode including "i" for insert, "a" for append, and "o" for open. To quit INSERT and return to CURSOR mode press the ESCAPE key.

Note: This section primarily is valid only for the SKKJIS dictionary. It is also valid for the WNNDICT, however the kanji in WNNDICT are listed in other ways also. I haven't done extensive experimentation with it though.

6.1 Hiragana

There are three states for entering text. The normal state is hiragana. This state can be changed via the F3 key or via special non alpha flags which will be described below. Hiragana is enter via romaji. That is you use a ordinary ascii keyboard and enter romaji then you press the return key to change the romaji into hiragana. The hiragana state accepts only romaji and a few punctuation characters. You can enter as few as one romajinized hiragana or a whole line of romajinized hiragana. DO NOT USE SPACES to separate words.

Example:

Enter: gohan ごはん

For words like しっぱい (shippai) or しったら (ittara) use "t-" (っ) to denote that the next consonant is long or that it is doubled.

Example:

Enter: shit-pai しっぱい

Enter: it-tara しったら

6.2 Katakana

Katakana is the second state for entering text. Katakana words are entered similar to hiragana words. That is with romajinized katakana. To enter katakana either press the F3 key until katakana is listed under the INSERT mode on the status line or toggle the KATAKANA flag by first typing the "|". To end the katakana state type another "|".

Example:

Enter: |programming| プログラミング

The long vowel sound is denoted by the "-" character.

Example:

Enter: |konpyu-ta| コンピュータ

6.3 Ascii

Ascii is the last state for entering text. In the Ascii state any printable Ascii character can be entered. The Ascii state, like the katakan state, is invoked either with the F3 key or with a Ascii flag, the Ascii flag. The Ascii flag is the tilde character ("~") and is used in the same manner as the katakana flag.

Example:

Enter: ~Example~ Example

Since the "~" and "|" characters are special control characters they cannot be entered normally. To enter these two characters precede them with the backslash ("\") character. The backslash will escape the normal meaning of these characters. Using the backslash with any other character will enter the backslash and that character.

6.3.1 JIS Ascii

While in INSERT mode press the F3 key (the k/h/a key) until "jascii" is displayed under INSERT mode. The English Alphabet in lower and upper case can be entered. The characters will be the same size as hiragana, katakana, and kanji.

Example:

ＡＢＣＤＥＦＧＨＩＪＫＬＭＮＯＰＱＲＳＴＵＶＷＸＹＺ
ａｂｃｄｅｆｇｈｉｊｋｌｍｎｏｐｑｒｓｔｕｖｗｘｙｚ

6.4 Entering Kanji

To enter Kanji start by entering the yomikata of the kanji in romaji. Enter the yomikata for a single kanji or the romaji string that represents a combination of two or more kanji. This time instead of pressing the return key in order to change the romaji into kana,

press the F2 key. If the SKK Jishou only maps to one possible Kanji(s) then it will be displayed directly in the window. If the string maps into more than one possible Kanji then a list of possible kanji will be displayed in the ENTERLINE.

Use the left and right arrow keys to move the cursor to the correct Kanji. If the Kanji you desire is not listed in the ENTERLINE use the up and down arrows to see if any more possible kanji are available. When you have moved the cursor underneath the correct kanji press the SPACE BAR or the return key to select that KANJI. The selected kanji will then appear at the end of the current line in the window. If the Kanji you want is not listed in any of the lines move the cursor underneath the question mark ("?") and press the SPACE BAR or the return key.

6.4.1 Entering Kanji only words

In this section I will detail how to enter the yomikata of a kanji and then select the desired kanji from the possibilities list. The kanji I wish to look up is 報 . The yomikata for this kanji is ho- (ほう). So I type:

Enter: ho-

and then press the F2 function key. The ENTERLINE is erased and the following list of kanji is displayed.

ほう ? 方 法 報 放 邦 泡 縫 砲 包 抱 豊 倣 俸 呆 朋 峯

The kanji I wish to enter is the third kanji from the left. I use the right arrow key and move the cursor underneath it. Then I press the return key. If I had pressed the down arrow the following list would be displayed.

ほう ? 訪 崩 鵬 宝 胞 蓬 芳 捧 庖 峰 奉 烹 萌 蜂 褒 鋒

If I had pressed the down arrow key again the remainder would have been displayed.

ほう ? 飽 鳳

6.4.2 Entering Verbs and Adjectives with a Kanji part

Entering verbs and adjectives is some what different from entering words that are all kanji. How verbs are entered in MOKE relys much on how the the entry appears in the skk dictionary. Verbs that have a kanji part and a hiragana part are looked up in the skkjis with hiragana for the kanji and an ascii character representing the start of the hiragana part of the verb. For the verb 見る (miru) it is listed as みr, 書く is listed as かk, and 思う is listed as おもu. For MOKE just capitalize the character representing the start of the hiragana part of the verb or the 'r', 'k', and 'u' of the above verbs. Below is a couple of examples and how they are to be entered.

kanji	romaji	enter	displayed

書く	kaku	kaK	書
見る	miru	miR	見
思う	omou	omoU	思
食べる	taberu	taB	食
待つ	matsu	maT	待
出す	dasu	daS	出
新しい	atarashii	ataraS	新
美しい	utsukushi	utsukuS	美
大きい	ookii	ooK	大
高い	takai	takaI	高
低い	hikui	hikuI	低

6.4.3 Entering Katakana versions of English words.

The skkjis dictionary also has some english words. Try computer, path, program, programming, etc (computer related words). For more words do a strings on skkjis. Enter english words as in rtk.

Example:

Enter: ~computer~

6.4.4 Guess Kanji command

Many times you will know only one kanji of the word. In this case MOKE can assist you in finding the right word. To use this function type in the known kanji and use ALT-g (for guess). It will act similar to the F2 command and ask you to choose a kanji if the kana maps to more than one kanji. Select the desired kanji as you normally do. Next you'll get the following message.

Which kanji in the word (1)st, 2)nd, ...):

Enter the position number of the known kanji (Example: 1). The disk will spin and if the kanji is found (in the wsktok.dat dictionary) it will display one entry. To go to the next entry hit any key except ESCAPE or RETURN. To select the entry hit RETURN. To Exit without selecting anything press ESCAPE.

Of course this function is only as good as the dictionary itself. If the word is not in the dictionary you won't find it. You should note that searches on the first kanji of the word is much faster then searching on subsequent kanji. This is because the dictionary is sorted on the first kanji and therefore the program can jump to the general area of that kanji. An other kanji in the word must necessarily search the entire dictionary and the dictionary is over 400k.

6.4.5 English to kanji

Kanji characters and compounds can be entered via english also. This presupposes that the english already exists in the edict dictionary however. The english word(s) is entered either in the ascii state of the INSERT mode or by surrounding the word with "~" (tildes) and then pressing ALT-e (for english lookup) to lookup the word.

Example:

~idea~

To go to the next entry hit any key except ESCAPE or RETURN. To select the entry hit RETURN. To Exit without selecting anything press ESCAPE.

To find any match the entire dictionary is searched.

Note: The pattern matching is currently for any string in a particular english entry in edict. This means that "plan" will also return explanation. (This will be fixed in upcoming releases).

6.5 Punctuation

Some japanese punctuation character are also available. Many of the punctuations are available by name. (In latter versions of documentation I'll try to include a list)

Punctuation	Enter	Description
「	‘	kagikakko
」	’	kagikakko
＃	#	
？	?	
。	.	period
、	,	comma

6.6 Other

Greek and Russian and other misc characters are available, and various other things also by japanese name. Try "shikaku", "maru".....

You can use moke with the "fonttest.jis" file to find out how to enter special characters also. The fonttest.jis file is part of the kd1.0 distribution.

7.0 Kanji to kana/english lookup

The readings (yomikata) of Kanji characters as well as Kanji words can be looked up by MOKE. The possible readings of a single Kanji character may be looked up via the F7 key in Cursor Mode. The readings if found will be displayed on the ENTERLINE. To lookup a Kanji word the start of the word must be marked with ALT-m. Then move the cursor to the end of the word. The word will become underlined as you move the cursor. Use the F7 key here to display the reading (if any).

If edict is "yes" in moke.rc first the edict file will be searched and the reading along with any english translations for the word will be displayed. Press ESCAPE or RETURN to resume CURSOR MODE or any other key to continue searching the file for additional readings.

Note: To get the english translation of a single kanji character you must always mark it with ALT-m. Using F7 without ALT-m will always search only the "revtab" dictionary.

7.1 Entering English translations while in the editor

Brute Force

English can be entered via the ALT-d (for dictionary entry) key. Entering is done as usual but to lookup a word use ALT-d instead of F2. Select the kanji word if there are multiple mappings and if not it will be displayed in the following format

<kanji word> [<hiragana yomikata>] /

The ENTERLINE is now ready to accept the english (ascii mode). Enter individual translation separated by "/". To end close the last translation with "/" and "~". This changes the mode back to hiragana. Press F5 to go to the next line.

While Editing a Text

The english translation for a kanji word can also be entered while editing a text. To do this lookup a kanji word marking it with ALT-m and instead of using the F7 key use F8. If there is no english it will ask you for the english. Even if there is english for this word it is possbile that there are multiple readings. Hit any character (except for ESCAPE or RETURN which will bring you back to CURSOR MODE) to get the next reading (I usually press the Space Bar).

When entering english you'll get the prompt

English (1):

then

English (2):

Enter however many english translations there are pressing return to go to the next one. When no more press ESCAPE on right after presing RETURN.

8.0 MOKE Command Keys

This is a visual line editor. I looked at patterning commands after ed, ex or vi, but choose it to be vi like. If you don't like the letters I choose you can change them in the moke.h file and recompile the program.

Like vi when a line is deleted it is copied to a buffer. This buffer

can be put (p) anywhere you like. Another delete line will destroy
the current buffer and again make a copy of the line just deleted.

Copying a line, like vi, can be done with the yank line command (y).

8.1 CURSOR MODE keys

a - append (insert to the left of cursor)

d - delete current line

i - insert (insert to the right of cursor)

o - open line for editing (insert line after currentline)

p - put last line deleted or yanked below current line

x - delete cursor character

y - yank line (copies line into buffer)

j - join lines (currentline with next line)

D - clear from cursor position to the end of line

b - goto beginning of line

e - goto end of line

g - goto line number (prompts for the line number)

/ - search forward from currentline

? - search reverse from currentline

n - search forward next

N - search forward reverse

ALT-g report current line of how many (not implemented yet)

ALT-r insert file after current line

ALT-w write current file (does not ask)

ALT-z write current file and quit.

F1 - Help for CURSOR mode

F6 - toggle between EUC and JIS output

F7 - on/kun yomi lookup of single/(string of) kanji

F8 - on/kun yomi lookup of single/(string of) kanji and asks you if you'd
like to enter the english of the string if it is not found in the
edict dictionary.

F9 - Quit (If buffer has been modified it asks if you wish to save file)

F10 - Save file (always asks you the file name to save to)

ALT-M - Used with the F7 key to look up a japanese string.

8.1.1 Searching

MOKE can primatively search forward and backwards. To search forward use the "/" key and to search backwards use the "?" key. When you press the forward search in the ENTERLINE "Fsearch: " will appear type in the Japanese (or english) string to search for. You can use the F2 key to get kanji. The search string as you build it will appear on the enterline also. You can combined kanji, kana, english.

To find the next occurrence of the string press "n", to search reverse use "N".

8.2 INSERT mode keys

F1 - Help for INSERT mode

F2 - lookup word on enter line

F3 - toggle between hiragana/katakana/ascii input modes
(default is hiragana)

F4 - copy last string entered in screen window to enterline
(erases anything currently displayed in the enterline)

F5 - Start Newline (very kludgy)

bs - (back space) erase one character on enter line

q - quit and save

e - quit (doesn’t ask to save)

ALT-d - like F2. For input of dictionary words for the Japanese->english dictionary project.

ALT-e Looks kanji by way of english using the EDICT dictionary

ALT-g - guess kanji (used like F2 or ALT-o, if you know one of the kanji in a word use this and MOKE will display possible kanji using the kanji you give it.)

ALT-o - search other dictionary (\kanj\mydict) for word

9. Future Enhancements

Some future enhancements may include getting rid of the entering only on the enterline and do the kana/kanji editing inline, and possibly multi-buffer capability.

10. Utility Programs

jc - Converts Japanese files between the various methods of coding
EUC, JIS, SJIS

Source of the code is Japan but I don't have anything more then that.
The program guesses what form the input file is and changes it to
the format that you request.

Syntax: jc <-e, -j, -s> <input file> > <output file>

-e some format to EUC
-j some format to JIS
-s some format to SJIS

Note: Use the output redirection switch (>) to direct output to the
output file name.

makelist Program to create a file suitable for use with the japanese
to english portion of MOKE. It doesn't insert english or
anything AIish like that all it does is create a file in
the format for jdict which at least give the readings of
kanji words.

Syntax: makelist skkjis

revtab Creates the Reverse lookup table. This is for the lookup of
of the on/kun yomi of an individual file. There are 7305
kanji entries. (Creates a file named "revtab" size 224000
bytes)

Syntax: revtab skkjis

wnn_ind

ktok_ind

wnn2skk

makeskk

11. Special thanks and Acknowledgements

Special thanks to my wife Ritsuko [律子] who learned
well a new english phrase and the meaning, computer widow.
You see since receiving the Junet Reader program middle December
I've bought a faster PC for home, bought Minix 1.3 and am trying
to update it to 1.5.5 with cdifs posted to the net, play with our
first baby (almost seven months old now, Akane [茜]), and
write all these programs.

11.1 Acknowledgements

This program would not be useful without the SKK dictionary provided in
the SKK interface to Nemacs by masahiko sato or the Wnn dictionaries
provided with from Kyoto University.

11.1.1 SKK

Below is the first few lines from the SKK manual.

かな漢字変換システム SKK (Simple Kana to Kanji conversion system)

東北大学電気通信研究所 佐藤 雅彦
(masahiko@sato.riec.tohoku.junet)

SKK は NEmacs 上で動く簡単なかな漢字変換システムである。以下では SKK の特徴および使用法について述べる。

11.1.2 Wnn

Below is the Copyright notice with Wnn.

Wnn Version 4.0

This distribution contains Version 4.0 of Wnn Japanese Input System. Wnn is a network-extensible Kana-to-Kanji conversion system and was jointly developed and released by the Software Research Group of Kyoto University Research Institute for Mathematical Science, OMRON TATEISI ELECTRONICS CO., and Astec, Inc. If you have Wnn up to Version 3.3, refer to "manual/intro" for revision information.

This distribution has been tested on the following systems.
Sun-3 Sun-4, SunOS3.4, SunOS4.0
OMRON SX9100, UniOS-B (4.3BSD) UniOS-U (SystemV R2.1) (DT DS M90)

/*
Copyright
Kyoto University Research Institute for Mathematical Sciences
1987, 1988, 1989
Copyright OMRON TATEISI Electronics, CO. 1987, 1988, 1989
Copyright ASTEC,Inc. 1987, 1988, 1989

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without any fee is hereby granted, subject to the following restrictions:

The above copyright notice and this premisson notice must appear in all version of this software;

The name of "Wnn" may not be changed;

All documentation of software based on "Wnn" must contain the wording "This software is based on the original version of Wnn developed by Kyoto University Research Institute for Mathematical Sciences (KURIMS), OMRON TATEISI Electronics, CO. and ASTEC,Inc.", followed by the above copyright notice;

The name of KURIMS, OMRON and ASTEC may not be used for any purposes related to the marketing or advertising of any product based on this software.

KURIMS, OMRON and ASTEC make no representations about the suitability of this software for any purpose.

It is provided "as is" without express or implied warranty.

KURIMS, OMRON and ASTEC DISCLAIM ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS,
IN NO EVENT SHALL OMRON, ASTEC and K.U.R.I.M.S BE LIABLE FOR ANY SPECIAL,
INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

Author: Hideki TsuikiKyoto University
tsuiki%kaba.or.jp@uunet.uu.net

Hiroshi Kuribayashi Omron Tateisi Electronics, Co.
kuri@frf.omron.co.jp
uunet!nff.ncl.omron.co.jp!kuri

Naouki Nide Kyoto University
nide%kaba.or.jp@uunet.uu.net

Shozo Takeoka ASTEC, Inc
take%astec.co.jp@uunet.uu.net

Takasi Suzuki Advanced Software Technology & Mechatronics
Research Institute of KYOTO
suzuki%astem.or.jp@uunet.uu.net

*/

Appendix A Alternate Dictionary (mydict)

MOKE will search an alternate dictionary if the mydict variable is set in moke.rc. The dictionary form is similar to the SKKJIS or WNNDICT format.

FORMAT

<hiragan/english> /<kanji/katakana>/<repeat as many times as needed>/

Examples

mark /マーク/
edwards /エドワース/
りつこ /律子/

The dictionary should be \kanji and should be named "mydict".
Be sure to turn the Alternate dictionary on in the startup file "moke.rc". The following line is all that is needed.

```
mydict  yes
```

Appendix B Dictionary of Technical Words

Term	Meaning
baseline	The horizontal line where characters without descenders

	rest. An example of a character with a descender is p.
bitmap	a bitmap in its simplistic form is essentially a two dimensional array of pixels or bits where an image is formed depending on the binary on or off state of each pixel in the array.
bitmap font	uses pattern of pixels for each character. With bitmap fonts there exists a different set of pixels for each character in every fonts size you want. Sometimes called raster
buffer	Area in the computers memory used to store what the computer is currently working on.
EUC	Extended UNIX CODE
font metric	The physical characteristics of each character in a font including height, width, .
glyph	a symbol that conveys information nonverbally. (Websters)
graphic font	See vector font.
hiragana	One the two kanas. Hiragana is used to write japanese words and to decline japanese verbs, put endings on adjectives, etc. Characterized by more rounded and artistic strokes.
JIS	Japanese Industrial Standard
kana	The alphabet like symbols used with Kanji to represent the Japanese langugage. There are two parallel systems hiragana and katakana.
katakana	One of the two kanas. katakana is usually used to represent foreign words, but not always. Characterized by straighter strokes.
kanji	The Chinese characters used with the kanas to represent the Japanese characters. There are around 2000 basic kanji with an additional 5000+ other kanji. Some of the kanji were greatly simplified after WWII.
kerning	adjusting the amount of space between letter.
pixel	the individual elements or dots of the digitized image. Derived from "picture elements". Some times called dots as in dots per inch.
raster	The area where a picture is displayed on a crt.
raster font	See bitmap font.
raster imaging	The creation of a complete dot-for-dot image of a page or graphic. In most cases one dot or pixel represents one unit in memory.
resolution	the process or capability of making distinguishable the individual parts of an object or closely adjacent optical images. (Websters)

Resolution is usually measured in dots per inch or pixels per inch.

computer displays are typically 50 to 100 dpi
dot matrix printers are typically 100 to 200 dpi
laser printers are typically 300 to 600 dpi
Photographic technology, above 1000 dpi

romaji — Method of writting Japanese with roman characters. It is highly ambiguous due to the number of homonyms in the japanese language.

SJIS — Shift JIS.

stroke — a mark or dash made by a single movement. One of the lines of a letter of the alphabet. (websters)

vector font — define lines that make up characters. Only need one set of characters for each font. They can be drawn at other sizes by varying the length of the lines. Vector fonts can be easily rotated and scaled.

vi — The UNIX visual editor developed by UC Berkeley. Is now widely available on other Operating Systems. The version of vi called Stevie also has also been patched for the Japanese language.

UNIX — Operating System originally developed by AT&T. There are now several different flavors (VERSIONS) of UNIX the two most popular are V R3 by AT&T and the Berkeley Software Development (BSD) now at level 4.4. UNIX is responsible for many of the current programming tools like AWK, LEX, YACC. But is probably most noted for the C Programming Language.

Appendix B. Related Programs

B.1 Programs to display Japanese on a PC

C.1.1 KD 1.0 - Kanji Driver by Izumi Ohzawa

KD is probably the best and most powerful program for displaying Japanese on a PC. It really is different from the rest of the display programs here in that it allows one to display japanese from program on a remote computer without having to download the file to the PC. In such it is kind of a terminal emulator.

Izumi Ohzawa [大沢五住]
University of California, Berkeley
izumi@violet.berkeley.edu or ...!ucbvax!violet!izumi

C.1.2. Reader - Junet Reader Program by Seiichi Nomura and Seke Wei

This is the program where Izumi Ohzawa and I got our inspiration. In that respect this is probably the most important program of the group. It is mainly aimed at reading News from Junet. It does have some capabilities of printing Japanese on a Dot Matrix printer which of have not explored.

Seke Wei
Institute of Computer Science
TsingHua University
HsinChu, Taiwan 30043
TH774339@TWNCTU01.BITNET

Seiichi Nomura
Associate Professor of
Mechanical Engineering
University of Texas at Arlington
B470SSN@UTARLVM1.BITNET
B460NOM@UTARLG.ARL.UTEXAS.EDU

C.1.3. View - JIS viewer by me [Mark Edwards]

I first translated the Junet Reader program into TC 2.0 and then further enhanced it to my liking. I improved the speed at which the bitmapped fonts were displayed probably over 50% by converting the kjfont16.ver file into the form that it is displayed in. This program serves as the basis to all my other programs. I also have added support for the k14.bdf file that is distributed by X11R4.

C.2. Programs to print to a printer

As stated in C.1.2. the Junet Reader program has some facilities for this. The best program is Kanjips - JIS to Postscript translater by Toshiyuki Masui. His development platform isa Unix machine capable of running PERL. PERL isn't available on the PC so I did a quick hack to make it work on the PC. It is still buggy, and I plan to work on it in the future to make it more usable.

Toshiyuki Masui
toshiyuki.masui@a.nl.cs.cmu.edu.

C.3 Miscellaneous Programs and Guides

C.3.1 jis.pas by Ken Lunde

Ken has written a pascal program for the VAX running VMS to convert between the various JIS coding techniques.

klunde@vms.macc.wisc.edu

C.3.1 Electronic Transfer of Japanese by Ken Lunde

This is a good introduction for sending and receiving mail from Japan and how to do it in Japanese. (See above for mail address)

D. Japanese to English Dictionary Project

The on/kun yomi enhancement to view (and moke) led me to think of another enhancement, namely a lookup based on a japanese word that would yield english. What would be needed here is a kanji/kana

to english dictionary much like the kana to kanji dictionary that the SKK dictionary provides.

I am curious to what kind of interest the readers of this news group might have for this facility. A workable online dictionary would require much effort and is not a one-man job. If there is any interest at all I would be glad to collect contributions from anyone and then build the interface and distribute the dictionary to all.

Tentatively I think a good submital format would be best patterned after the SKK dictionary format.

Example:

<kanji> [on/kun yomi] /english one/ english two/ english n/

miru in kanji [miru in hiragana] /to look/to see/

What do you think?

moke or nemacs or whatever program for editing japanese could be used to construct the entries.

E. Using Japanese on other Platforms

E.1 Unix Machines

Nemacs
kterm - uses kinput and wnn server
SKK
Wnn

E.2 Macintosh

Japanese System Software available from ADPA (Apple Computer)

E.3 Other

F. Problems

I have tested MOKE on Hercules and VGA graphics cards. The program will not work on a monochrome only machine, and it doesn't work on at least some Toshiba portables.

Bug reports, comments, or to get on the mailing list to receive bug reports and bug fixes.

Send to:

mark edwards [マーク。エドワズ]
Systems Programmer
University of Wisconsin - Madison (MACC)
1210 W. Dayton St.
Madison, Wi. 53706 (USA)
Phone: (608) 262-7585

Email:
edwards@macc.wisc.edu
edwards@wiscmacc.bitnet