The CJK package

Werner Lemberg

29–Dec–2008

Contents

This is the LaTeX2e style package CJK Version 4.8.2 (29–Dec–2008). It is freely distributable under the GNU Public License. **You need LaTeX 2$_\varepsilon$ version 2001/06/01 or newer!**

1 Usage

Use `CJK.sty` as a package, e.g.,

```
\documentclass{article}
\usepackage[<option>]{CJK}
```

See section 'Caveats' below for the available options. Normally, you don't need them.

Two new environments,

```
\begin{CJK}[<fontencoding>]{<encoding>}{<family>}
...
\end{CJK}
```

and

```
\begin{CJK*}[<fontencoding>]{<encoding>}{<family>}
...
\end{CJK*}
```

are defined. The parameters have the following meaning:

1.1 <encoding>

These character sets and encodings are currently implemented in `CJK.enc`:

Name	Description	Character set(s)	Encoding
`Bg5`	For traditional Chinese. Mainly used in Taiwan.	Big 5.	Big 5 without UDA2 and UDA3.
`Bg5+`	For traditional Chinese. Obsolete.	Big 5+.	GBK.
`HK`	For traditional Chinese. Used in Hong Kong.	Big 5 + HKSCS-2004.	Full Big 5.
`GB`	For simplified Chinese. Mainly used in PR China. Also called 'EUC-CN'.	GB 2312-1980.	EUC.
`GBt`	For traditional Chinese. Rarely used in PR China.	GB/T 12345-1990.	EUC.
`GBK`	For Chinese. An extension of GB 2312.	GBK.	GBK.
`JIS`	For Japanese.	JIS X 0208:1997.	EUC.
`JIS2`	Japanese supplementary character set,	JIS X 0212-1990.	EUC.
`SJIS`	For Japanese. Used mainly on PCs. Also known as 'MS Kanji'.	1-byte characters from JIS X 0201-1997 (half-width katakana), 2-byte characters from JIS X 0208:1997.	SJIS.
`KS`	For Korean. Also called 'EUC-KR'.	KS X 1001:1992 = KS C 5601-1992.	EUC.
`UTF8`	Unicode Transformation format 8, also called 'UTF-2' or 'FSS-UTF'.	Unicode.	UTF-8.
`CNS1`	Chinese National Standard Plane 1,	CNS 11643-1992 plane 1.	EUC.
`CNS2...CNS7`		CNS 11643-1992 plane 2–7.	EUC.
`CEFX`	reserved CEF character set for IRIZ.		EUC.
`CEFY`	private CEF character set.		EUC.

Note: The value '`HK`' can be also used for complete Big 5 support which needs user-defined areas 2 and 3 (UDA2 and UDA3), located in the ranges `0x8E40`–`0xA0FE` and `0x8140`–`0x8DFE`, respectively.

For details on HKSCS-2004 see `http://www.info.gov.hk/digital21/eng/hkscs/download/e_sect3_2004.pdf`

These encodings (except Big 5, Big 5+, HK, GBK, SJIS, and UTF-8) are simplified EUC (Extended Unix Code) character sets without single shifts. The used character set slot G1 stands for two-byte encodings with byte

values taken from the GR (Graphic Right) character range `0xA1`–`0xFE` (as defined in ISO 2022).

Note that CNS1 and CNS2 contain almost the same characters in the same order as Big 5 (but in EUC).

For CEF and CNS character sets see `CEF.txt` also.

Big 5+ and GBK have exactly the same encoding layout (but their origins differ).

Additionally, the following encodings *with* single shifts are implemented, using some of the above defined character sets:

Encoding	**Description**	**Character sets**
`EUC-JP`	for Japanese.	Half-width katakana (from JIS X 0201-1997), JIS X 0208:1997, JIS X 0212-1992.
`EUC-TW`	for traditional Chinese.	CNS 11643-1992 planes 1–7.

EUC-JP, EUC-TW, and UTF-8 encodings can't be used in preprocessed mode (see below) because it makes no sense. (To be more precise, UTF-8 sequences with more than two bytes can't be used.)

If you use this parameter it is the same as you would have used `\CJKenc`: Writing e.g.,

```
\begin{CJK}{Bg5}{...}
...
```

is identical to

```
\begin{CJK}{}{...}
\CJKenc{Bg5}
...
```

Note: A 'character set' is an ordered collection of glyphs. The order of the glyphs is just for defining purposes and for reference.

An 'encoding' is an ordering scheme to access a character set. LaTeX 2_ε also uses the term 'input encoding'.

A character set can have many encodings (cf. JIS X 0208 → EUC, SJIS).

An encoding can be used for many character sets (cf. EUC → KS X 1001, GB 2312, etc.).

Sometimes, the character set has the same name as the encoding (Big 5, Big 5+, GBK).

For more details I suggest to read the document `cjk.inf` from Ken Lunde; it is available from `ftp://ftp.ora.com/pub/examples/cjkvinfo/doc/cjk.inf`

A really thorough reference is his latest book 'CJKV Information Processing' (O'Reilly).

Throughout this CJK documentation, 'encoding' refers to the valid encoding/character set combinations defined just above.

1.2 <fontencoding>

These font encodings are currently defined: '' (empty; the default), 'pmC' (available for Bg5, GB, GBt, JIS, and KS), 'dnp' (for JIS and SJIS), 'wn' (for JIS), and 'HL' (for KS).

'Font encoding' means the order of characters in the subfonts itselves. A change of the font encoding neither alters the meaning of a CJK character nor changes the character code in the selected encoding.

The font encoding 'pmC' is defined for compatibility with the pmC package (which is obsolete). It is not encouraged to use this font encoding because of wasting subfonts. If possible, convert your original CJK bitmap fonts with hbf2gf (see hbf2gf.txt) or other tools to CJK encodings.

'dnp' implements the character order of the Dai Nippon Printing fonts and is only available for JIS and SJIS encoding. 'wn' (only available for JIS) is the font encoding for watanabe jfonts. There exists a linking package which maps the watanabe jfonts onto the dnp naming scheme (thus you can use the real dnp fonts for printing and the mapped jfonts for previewing). See the documentation files in the 'japanese' subdirectory for further details.

'HL' allows the use of the new HLaTeX fonts (starting with version 1.0); note that the definition of fonts is rather different compared to HLaTeX. See the section 'Korean input' below for a detailed description.

You can change the font encoding per encoding with the command \CJKfontenc; the first parameter is the encoding, the second the font encoding.

1.3 <family>

It is impossible to know in advance what fonts are available at your site; look at the example FD (font definition) files how to create or modify appropriate FD files suiting your needs. See fonts.txt also for further hints.

If this parameter is empty, the default value given in CJK.enc is selected: 'song' for all encodings except KS (which defaults to 'mj'). If you use this parameter it is the same as you would have used \CJKfamily; all encodings then use this family:

```
\begin{CJK}{...}{song}
...
```

is identical to

```
\begin{CJK}{...}{}
\CJKfamily{song}
...
```

You can change the families per encoding (and font encoding) with the command \CJKencfamily; the first parameter is the encoding, the second

the family, the optional argument is the font encoding. This overrides the default value.

Note that `\CJKfamily` or a non-empty '`family`' parameter of the `CJK` environment overrides any `\CJKencfamily` commands. Say '`\CJKfamily{}`' to enable `\CJKencfamily` again.

The `CJK*` environment swallows unprotected spaces and newlines after a CJK character (the usual habit for Chinese and Japanese text), whereas CJK does not (for European and Korean text). You can switch between these two 'modes' with `\CJKspace` (`CJK*` → `CJK`) and `\CJKnospace` (`CJK` → `CJK*`).

If you use `cjk-enc.el`, you don't need to specify a `CJK` environment. This is done automatically. See `cjk-enc.txt` for details.

This is a typical example:

```
\begin{CJK*}{GB}{kai}
  ...
  Chinese simplified text in GB encoding
  ...
\end{CJK*}
```

2 How it works

Asian logographs can't be represented completely with one byte per character. (At least) two bytes are needed, and the most common encoding schemes (UTF-8, GB, Big 5, JIS, KS, etc.) have a certain range for the first byte (usually `0xA1`–`0xFE` or a part of it) which signals that this and the next byte represent an Asian logograph. This means almost all plain ASCII characters (characters between `0x00` and `0x7E`) are left undisturbed, and the remaining character codes (`0x80`–`0xFF`) are assigned to a CJK encoding, creating a multiple-byte encoding with 1-byte and 2-byte characters (and even 3-byte and 4-byte characters for UTF-8).

The character `0x7F` is reserved also for the CJK package. See the section 'Preprocessors' below.

Encodings like EUC-TW access additional character sets using escape characters (`0x8E` and `0x8F`) which signals that the next character comes from another character set (which is 'shifted' to the GR range); up to four bytes are needed for a single character. Example:

> `0x8E 0xA3 0xB7 0xCE`
> `0x8E` is a single shift escape character; `0xA3` selects CNS plane 3, and `0xB7CE` is the character code (in GR representation) in this plane.

`CJK.sty` makes the character codes `0x7F` and `0x81`–`0xFE` active inside of the `CJK` environment and assigns macros to the active characters which

then select the proper font and character. The real mechanism is a bit more complex to assure robustness (it was borrowed and modified from LaTeX 2_ε's `inputenc.sty`) and correct handling of punctuation characters.

emTeX users: you must activate 8bit input and output while creating the LaTeX 2_ε format file! Do this by using the switches `-o` and `-8` (additional to the iniTeX switch `-i`). Example: `tex386 -i -o -8 latex.ltx`.

3 Some internals

Internally three levels (bindings, encodings, character macro sets) are defined:

```
active characters
    |
    +----------> bindings (standard, SJIS, UTF8)
    |
active character macros
    |
    +----------> encodings (GB, Big 5, ...) +
    |                font encodings (none, dnp, wn, pmC, HL)
    |
subfont selecting macros
    |
    +----------> character macro sets (standard, Big 5, ...)
    |
character selecting macros
```

User-selectable are only the encoding and the font encoding (as explained above); the other levels are selected by the CJK package.

These levels correspond to the following internal macros:

`\CJK@xxxxBinding` (‘`xxxx.bdg`’ files): Possible values for ‘*xxxx*’ are: `standard`, `SJIS`, `UTF8`, `EUC-JP`, and `EUC-TW`.

`\CJK@xxxxEncoding` (‘`xxxx.enc`’ files): Possible values for ‘*xxxx*’ are: `standard`, `extended`, `Bg5`, `SJIS`, `KS`, `UTF8`, `pmCsmall`, `pmCbig`, `JISdnp`, `SJISdnp`, `KSHL`, `EUC-JP`, and `EUC-TW`.

`\CJK@xxxxChr` (‘`xxxx.chr`’ files): Possible values for ‘*xxxx*’ are: `standard`, `Bg5`, `KS`, `SJIS`, `UTF8`, `pmC`, `HLaTeX`, `EUC-JP`, and `EUC-TW`.

In preprocessed mode (see below), no bindings are used.

And now a more detailed description of the various encodings. Please note that you should never access these macros directly.

- `\CJK@standardEncoding` is used for EUC encodings with the first and second byte in the range `0xA1`–`0xFE` (`GB`, `GBt`, `JIS`, `JIS2`, `CNS`, `CEF`).

- `\CJK@extendedEncoding` is used for Big 5+ and GBK encodings. The first byte is in the range `0x81`–`0xFE`, the second byte in the range `0x40`–`0xFE` (with a gap at 0x7F).

- `\CJK@Bg5Encoding` is used for Big 5 encoding with the first byte in the range `0xA1`–`0xFE` and the second byte in the range `0x40`–`0xFE` (with a gap from `0x7F`–`0xA0`).

- `\CJK@SJISEncoding` is used for SJIS encoding; one-byte characters are in the range `0xA1`–`0xDF`, two-byte characters have the first byte in the ranges `0x81`–`0x9F` and `0xE0`–`0xEF`, the second byte runs from `0x40` to `0xFC` except `0x7F`. Since SJIS only squeezes the JIS X 0208 character set into a new scheme without changing the ordering, fonts produced by `hbf2gf` or `ttf2pk` look the same for EUC and SJIS encoding except one-byte SJIS characters. For more details see below the section 'SJIS encoding'.

- `\CJK@KSEncoding` is used for the KS X 1001 character set in EUC encoding. Two sets of subfonts are defined, one for Hangul syllables and elements, and a second for Hanja. For more details see below the section 'Korean input'.

- `\CJK@UTF8Encoding` is used for Unicode in UTF-8 encoding. The first byte is in the range `0xC0`–`0xDF` for two-byte values, `0xE0`–`0xEF` for three-byte values, and `0xF0`–`0xF4` for four-byte values. The other byte(s) are in the range `0x80`–`0xBF`. Note that CJK expects two hexadecimal digits as a running number in the font name (as defined in `UTF8.enc`) instead of two decimal digits for subfonts covering characters up to `U+FFFF`. Subfonts for Unicode values greater than `0xFFFF` use four hexadecimal digits in the font name. Select the option '`unicode yes`' in the `hbf2gf` config file if you use `hbf2gf` to transform bitmap fonts in HBF format to PK fonts as used by `CJK.sty`. Three commands (`\CJKCJKchar`, `\CJKhangulchar`, and `\CJKlatinchar`) control the handling of intercharacter glue: `\CJKCJKchar` (the default) selects CJK style (using `\CJKglue`), `\CJKhangulchar` selects hangul style (using `\CJKtolerance`), and `\CJKlatinchar` selects none of them. This encoding does not work in preprocessed mode.

- `\CJK@pmCsmallEncoding` and `\CJK@pmCbigEncoding` can be activated with `\pmCsmall` (this is the default) and `\pmCbig` inside the CJK environment. Note that the original pmC fonts have two character sizes per font (the bigger ones with an offset of −128); `Bg5pmC` encoded fonts cannot contain big characters. The names of the fonts in the FD files reflect the modifications added by Marc Leisher `<mleisher@nmsu.edu>` to the original poor man's Chinese (pmC) package written by Thomas Ridgeway `<ridgeway@blackbox.hacc.washington.edu>`.

- `\CJK@JISdnpEncoding` is the JIS X 0208 character set in EUC encoding with dnp fonts. The main difference (besides the offsets) is the composition of real font names; a dnp font name consists of name stem + subfont name + designsize: an example is `dmjkata10`. Note that the wadalab PS fonts omit the designsize part in the font names, thus it is sufficient (and even better) to use the 'CJK' size functions in FD files instead of the 'DNP' ones. `\CJK@JISwnEncoding` is similar to `JISdnp` encoding but uses Watanabe jfonts; `\CJK@SJISdnpEncoding` maps SJIS onto `dnp` encoded fonts.

- `\CJK@KSHLEncoding` finally uses the new fonts of the HLaTeX package for Korean; three internal encodings are necessary to represent it. See the next section for details.

- `\CJK@EUC-TWEncoding` and `\CJK@EUC-JPEncoding` are quite similar to `\CJK@standardEncoding` but implement single shift access additionally. They can't be used in preprocessed mode.