NAME

hbf2gf – convert a CJK bitmap font into subfonts usable by TeX and Omega.

SYNOPSIS

hbf2gf [**-q**] *configuration-file*[*.cfg*]
hbf2gf [**-q**] [**-p**] [**-g**] [**-n**] *subfont-name x-resolution* [*y-scale* | *y-resolution*]
hbf2gf -t [**-q**] *subfont-name*
hbf2gf --version | --help

DESCRIPTION

CJK bitmap fonts can't be directly used with TEX because the number of characters in such fonts exceeds 256, the limit of a TEX font. Thus it is necessary to split these fonts into subfonts, and this is exactly what **hbf2gf** does.

As the name says, **hbf2gf** uses CJK fonts in a certain format which is called **Hanzi Bitmap Font** (**HBF**) format. It simply consists of the CJK bitmap file(s) and a text file in a format very similar to the BDF format of the X Window System which describes the bitmap font files: the encoding, the size, etc. The produced `GF` files can then be converted with **gftopk** into standard `PK` files.

hbf2gf can be called in three modes:

hbf2gf [**-q**] *configuration-file*[*.cfg*]

This call normally creates a set of `GF` files, one `PL` file, and a batch file which must be executed after **hbf2gf** has finished. This script will then call **gftopk** to convert all `GF` files into `PK` files, and it will call **pltotf** to convert the `PL` file into a `TFM` file. Finally it will copy the `TFM` file so that each `PK` file has its `TFM` file (which are all identical).

If **ofm_file** is set to 'yes' in the configuration file, `OFM` and `OVF` files will be created too.

-q makes **hbf2gf** quiet.

hbf2gf [**-q**] [**-p**] [**-g**] [**-n**] *subfont-name x-resolution* [*y-scale* | *y-resolution*]

This mode is intended for use with **mktexpk** and its derivates. Only one `GF` file together with a `PL` file for the given subfont will be computed, taking the horizontal resolution and a vertical scalingfactor (if the value is smaller than 10) resp. the vertical resolution (otherwise) from the command line, ignoring the **nmb_fonts** parameter of the configuration file. The last two characters (which are interpreted as the subfont number) are stripped to get the name for the configuration file (which must end with '`.cfg`'). No job file will be created. If option **-p** is set, no `PL` file is created. If **-g** is set, no `GF` file is created. The extension can be controlled with **-n**; if set, the extension is '`.gf`', otherwise '`.`<*resolution*>`gf`'. **-q** makes **hbf2gf** quiet.

hbf2gf -t [**-q**] *subfont-name*

This mode is intended for use with scripts like **mktexpk**; it tests whether the specified subfont name leads to an **hbf2gf** configuration file. It returns 0 on success and prints out the name of that configuration file (provided the **-q** switch isn't set). This test isn't a thorough one; it only removes the last two characters and checks whether a configuration file with that name exists.

See the next section for more details about configuration files.

Specifying the option **--version** returns the current version of **hbf2gf** and the used file search library (e.g. **kpathsea**). Usage information is shown with the **--help** parameter.

CONFIGURATION FILE

Here a sample configuration file (`gsfs14.cfg`) for a 56x56 Chinese font in GB encoding; note that all information about the font is in the `jfs56.hbf` file. See the **FILE SEARCHING** section how HBF fonts and **hbf2gf** configuration files are found. See the **AVAILABILITY** section where to get CJK fonts together with its `HBF` files:

```
hbf_header      jfs56.hbf
mag_x           1
```

```
threshold      128
comment        jianti fansongti 56x56 pixel font

design_size    14.4

y_offset       -13

nmb_files      -1

output_name    gsfs14

checksum       123456789

dpi_x          300

pk_files       no
tfm_files      yes

coding         codingscheme GuoBiao encoded TeX text

pk_directory   $HBF_TARGET/pk/modeless/gb2312/gsfs14/
tfm_directory  $HBF_TARGET/tfm/gb2312/gsfs14/
```

A configuration file is a plain text file consisting of keywords and its arguments. A keyword must start a line, otherwise the whole line will be ignored. If the word starting a line is not a keyword, the line will be ignored too. Empty lines will also be skipped. The search for keywords is case insensitive; in contrast, the arguments will be taken exactly as given (except 'yes' and 'no' which can be written with uppercase or lowercase letters). Each keyword has one argument which must be separated by whitespace (blanks or tabs) from the keyword and must be on the same line. Each line must not be longer than 256 characters.

You can use environment variables in the configuration file. The escape character starting an environment variable in the configuration file is always '`$`', even for operating systems like DOS which has other conventions. **hbf2gf** recognizes only environment variable names which start with a letter or an underscore, followed by alphanumeric characters or underscores. You can surround the variable with braces to indicate where the variable name ends, for example `${FOO}`. To get a dollar sign you must write '`$$`'. The expansion of environment variables in hbf2gf itself (without the help of either kpathsea, emtexdir, or MiKTeX searching routines) is very limited; this feature has been carried over from previous versions. It can't expand variables set in texmf.cnf; it also can't handle more than one directory as the variable's value. **Don't use it except for the 'pk_directory' and 'tfm_directory' parameters!**

This is the list of all necessary keywords:

hbf_header
: the HBF header file name of the input font(s). **hbf2gf** uses the given searching mechanism (kpathsea, emtexdir, or MiKTeX) to locate this file.

output_name
: the name stem of the output files. A running two digit decimal number starting with '`01`' will be appended. For Unicode fonts see the keyword **unicode** below. This value is in almost all cases identical to the name of the configuration file.

And now all optional keywords:

x_offset
: increases the character width. Will be applied on both sides; default for non-rotated glyphs is the value given in the HBF header (**HBF_BITMAP_BOUNDING_BOX**) scaled to **design_size** (in pixels).

y_offset
shifts all characters up or down; default for non-rotated glyphs is the value given in the HBF header (**HBF_BITMAP_BOUNDING_BOX**) scaled to **design_size** (in pixels).

design_size
the design size (in points) of the font. **x_offset** and **y_offset** refer to this size. Default is 10.0

slant the slant of the font (given as Δx / Δy). Only values in the range $0 \leq$ **slant** ≤ 1 are allowed. Default is 0.0

rotation
if set to ‘yes’, all glyphs will be rotated 90 degrees counter-clockwise. The default offsets as given in the HBF header will be ignored (and set to 0). Default is ‘no’.

mag_x
mag_y scaling values of the characters to reach design size. If only one magnification is given, x and y values are assumed to be equal. Default is **mag_x** = **mag_y** = 1.0

threshold
A value between 1 and 254 defining a threshold for converting the internal graymap into the output bitmap; lower values cut more pixels. Default value is 128.

comment
a comment describing the font; default is none.

nmb_fonts
the number of subfonts to create. Default value is -1 for creating all fonts.

unicode
if ‘yes’, a two digit hexadecimal number will be used as a running number, starting with the value of the first byte of the first code range. Default is ‘no’.

min_char
the minimum value of the encoding. You should set this value to get correct subfile offsets if it is not identical to the lowest character code in the HBF file.

dpi_x
dpi_y the horizontal and vertical resolution (in dpi) of the printer. If only one resolution is given, x and y values are assumed to be equal. Default is 300.

checksum
a checksum to identify the `GF` files with the appropriate `TFM` files. The default value of this unsigned 32bit integer is 0.

coding a comment describing the coding scheme; default is none.

pk_directory
the destination directory of the `PK` files; default: none. Attention! The batch file will not check whether this directory exists.

tfm_directory
the destination directory of the `TFM` files; default: none. Attention! The batch file will not check whether this directory exists.

pk_files
whether to create `PK` files or not; default is ‘yes’.

tfm_files
whether to create `TFM` files or not; default is ‘yes’.

ofm_file
whether to create an `OPL` file or not; default is 'no'. The batch file will then use **ovp2ovf** of the Omega distribution to convert it into an `OFM` and an `OVF` file. The `OPL` file simply maps all subfonts back to a single Omega font.

long_extension
if 'yes', `PK` files will include the resolution in the extension (e.g. `gsso1201.300pk`). This affects the batch file only (default is 'yes').

rm_command
the shell command to remove files; default: 'rm'.

cp_command
the shell command to copy files; default: 'cp'.

job_extension
the extension of the batch file which calls **gftopk** and **pltotf** to convert the `GF` and the `PL` files into `PK` and `TFM` files respectively; default is none.

FILE SEARCHING

hbf2gf uses either the **kpathsea**, **emtexdir**, or **MiKTeX** library for searching files (**emtexdir** will work only on operating systems which have an MS-DOSish background, i.e., MS-DOS, OS/2, Windows; **MiKTeX** is for Win32 systems).

KPATHSEA

Please note that older versions of **kpathsea** (<3.2) have no special means to seach for program related files. Additionally, versions older than 3.3 have no default path for miscellaneous fonts, thus we use the paths for PostScript related stuff if necessary for fonts resp. configuration files. The actual version of kpathsea is displayed on screen if you call **hbf2gf --version**.

Here is a table of the file type and the corresponding **kpathsea** variables.

Version 3.3 and newer (this won't change again in the future!):

```
.hbf        MISCFONTS
.cfg        HBF2GFINPUTS
```

Version 3.2:

```
.hbf        T1FONTS
.cfg        HBF2GFINPUTS
```

And here the same for pre-3.2-versions of **kpathsea:**

```
.hbf        T1FONTS
.cfg        TEXCONFIG
```

Finally, the same for versions ≤2.6:

```
.hbf        DVIPSHEADERS
.cfg        TEXCONFIG
```

Please consult the info files of **kpathsea** for details on these variables. The decision which naming scheme to use for variables will be done during compilation.

You should set the `TEXMFCNF` variable to the directory where your `texmf.cnf` configuration file resides.

Here is the proper command to find out to which value a **kpathsea** variable is set (we use `MISCFONTS` as an example). This is especially useful if a variable isn't set in `texmf.cnf` or in the environment, thus pointing to the default value which is hard-coded into the **kpathsea** library.

```
kpsewhich -progname=hbf2gf -expand-var='$MISCFONTS'
```

We select the program name also since it is possible to specify variables which are searched only for a certain program -- in our example it would be `MISCFONTS.hbf2gf`.

A similar but not identical method is to say

```
kpsewhich -progname=hbf2gf -show-path='misc fonts'
```

[A full list of format types can be obtained by saying '`kpsewhich --help`' on the command line prompt.] This is exactly the how **hbf2gf** searches for files; the disadvantage is that all variables are expanded which can cause very long strings.

EMTEXDIR

Here the list of suffixes and its related environment variables to be set in `autoexec.bat` (resp. in `config.sys` for OS/2):

`.hbf`	`HBFONTS`
`.cfg`	`HBFCFG`

If one of the variables isn't set, a warning message is emitted. The current directory will always be searched. As usual, one exclamation mark appended to a directory path causes subdirectories one level deep to be searched, two exclamation marks causes all subdirectories to be searched. Example:

```
HBFONTS=c:\fonts\hbf!!;d:\myfonts\hbf!
```

Constructions like '`c:\fonts!!\hbf`' aren't possible.

MIKTEX

Please consult the documentation files of **MiKTeX** for more details.

LIMITATIONS

The x and y output size must not exceed **MAX_CHAR_SIZE**, which is defined at compile time; its default value is 1023 (pixel).

SEE ALSO

ttf2pk(1)

`hbf2gf.w`: this is the source code written in **CWEB** which can be converted into a pretty-printed TEX document using **cweave**. The CJK package also contains a preformatted `hbf2gf.dvi` file.

the **CJK** documentation files (`hbf2gf.doc`).

the **Hanzi Bitmap File** (**HBF**) standard version 1.3; available at `ftp.ifcss.org`

the Omega documentation available at `ftp.ens.fr` and the CTAN hosts and mirrors.

FILES

`*.cfg` The **hbf2gf** configuration scripts

`*.hbf` HBF header files which describe fixed-width bitmap fonts. Note that the bitmap font name(s) themselves as specified in the header files are irrelevant for **hbf2gf**.

AVAILABILITY

hbf2gf is part of the CJK macro package for LATEX 2_ε available at the CTAN hosts and its mirrors.

CJK fonts together with HBF header files can be found at `ftp.ifcss.org` and its mirrors.

AUTHORS

Werner Lemberg <wl@gnu.org>
Ross Paterson (the HBF API) <ross@soi.city.ac.uk>