

# Flib, the Fontlibrary program for the AMIGA

**Version 1.03**

**vom  
27. July 1995**

by  
Georg Heßmann  
translation by Thomas Tavoly

## **Abstract**

Flib is a program for making and administration of fontlibraries. These are libraries, in which many PK-Fonts are bundled together. The advantage of libraries in respect to single files is the space savings on disk or harddisk, the better overview and the faster access to the particular elements. These libraries are used by the  $\text{\TeX}$ -drivers ShowDVI and DVIprint.

## Table of contents:

1. Copyright and similar .....	2
2. Flib - generals .....	2
3. Calling Flib .....	3
4. The structure of the libraries .....	3
5. Portability .....	4

## 1. Copyright and similar

The program Flib is “copyrighted Public Domain”. This means anyone can distribute the program and change it considering some limitations. Firstly, the program retains my Copyright. The Copyright must not be deleted from the program text. and the Copyright mention must be further at least per option be displayable. All alterations, which have been made to the program, must be conferred to me, as for me to be able to adjust the versions as possible. These Copyright stipulations are also present on the beginning of the file “flib.c” and must not be deleted from there either. Also, the files flib.c, flib.tex as well as the corresponding README file must always be distributed together. In return anyone can use the program as he/she likes and distribute it commercially or non-commercially<sup>1</sup>.

For further explanation see the README file.

## 2. Flib - generals

Flib is a utility program for the T<sub>E</sub>X-drivers ShowDVI and DVIPrint. It is there for bundling the PK-Fonts of a particular resolution in one file, since this has some convincing advantages.

Firstly, you could name the space savings. As every single file always consists of a multiple of the block size of the current File System, on average half a block is thrown away at each file too. On certain operating systems, an entire block is needed for every directory entry also. That means, that per file a waste of one and a half blocks is to be expected. In case of about 100 modules per library, approximately 150 blocks are saved in this manner.

But there are other advantages. There are the better overview and also the faster access just to name a few. As you mostly need several fonts of one resolution, you only need to open one file, and read the directory there. After that every access to a font is only an “fseek” command. As this way for each resolution only one file has to be opened, you can keep this open all the time the program is running. Thus you avoid a constant opening and closing of files.

The Flib program is however not only specialized in PK-Fonts. In theory you can archive every type of file in a library. As the program source is available for the Flibprogram, other programs could be adjusted to it, who load a lot of help files.

---

<sup>1</sup> This goes only for the Flib program and *not* for the other programs

### 3. Calling Flib

The syntax for calling is as follows:

```
usage:  flib c|a|x|d|z|p|l|t|v] library [-|[modules]*]
```

- c,a** appends a module to a library. Does the library not yet exist, then it is created. If the module is already in the library, it is replaced.
- x** copies the module from the library to an own file with the name of the module. If no module is supplied, then all modules are copied from the library.
- d** deletes a module from a library. If the module was the last one in the library, then the library is deleted.
- t** outputs the list of all modules in the library.
- r** renames all indicated modules. To this end the new name is expected from the console (stdin).
- z** transforms fontlibraries of the old type to the current version. All newly created libraries are automatically of the new type.
- p** tests all modules, whether they are still correct. To this end a checksum is created at the creation of the library and stored in the directory of the library.
- l** creates a library-link. The call looks in this case like this:

```
flib l link-name dest-name
```

At this, a library *link-name* is created, which is actually a pointer to a library with the name *dest-name*. This is used for example in case you need fontlibraries of the resolutions 109 dpi and 110 dpi, but due to memory limitations you only want one of them to be held on the harddisk. As the difference between the both fontlibraries is not so big, you could decide to hold only one of them on harddisk, and create the other only as a link.

- v** switches verbose mode on. This flag can be switched with every one of the above options. Normally only error messages are displayed. With the **v** option all actions are protocolled. This is especially useful in case of more complicated operations.

In case of all commands you can indicate an arbitrary number of modules behind the library name. The possibility exists however, to supply on a minus (-). Then all module names are read from the console (stdin) (one module per line), until an end of file character comes.

## 4. The structure of the libraries

In case of the libraries, two different types must be recognized: Firstly the library, in which the modules are actually stored, and secondly all the other libraries, which consist of only a link to another library.

### 4.1. Real libraries

The fontlibraries consist of three parts. At the beginning of the library a library header is present, then comes the library-directory and then the modules themselves.

The library header exists from two 32-bit words (longwords). The first one is the “magic-number”, which consists of the four characters “FLIB”. After that the directory size. It

has not have to be exactly equal to the number of modules in the library. Directory entries can be shown as unused.

The directory of the library consists of the directory entries. The number is stored in two longwords. A directory entry looks like this:

```
struct new_flib_dirent { char  mname[22];          /* module name  */
                        unsigned short checksum;    /* checksum      */
                        long  size;                 /* size in bytes */
                        long  where;                /* position      */
};
```

At which **mname** is the modulename, **checksum** the checksum, **size** the size of the module and **where** the position of the module inside the library. When **where** is equal zero, then the directory entry is unused.

In case of all these number indications, the byte order is to be paid attention to. The 680x0 processor format is used. Thus first always the “least significant byte” and then the “most significant byte”. The output routines of the Flib program are written in a way that they create and read the wanted format on other processors, too.

The third and last (largest) part of the library are the particular modules, which are simply put behind one another.

## 4.2. Library Links

Library links have a very simple structure, which can in theory also be created with a text editor. For example:

“FLNK<PK0110>”

This is a library link to a library with the name *PK0110*.

So, the magic word is here **FLNK**. After that the actual name of the directory is coming, to which the link should be pointing, enclosed between smaller than and greater than signs.

## 5. Portability

The program is written in ANSI-C, but can be compiled with a normal C-Compiler by altering a define. The program has been compiled and tested on the Amiga, Atari, Sun 3, Sun SPARC and a DECstation under AMIGA DOS, ATARI TOS and UNIX (SunOS/ULTRIX).

Translation from German to English by Thomas Tavoly -Sep.'90.