

ShowDVI, the T_EX-Previewer for the AMIGA

Version 1.06

**of
27. July 1995**

by
Georg Heßmann
translation by Thomas Tavoly

Abstract

ShowDVI is a T_EX-Previewer with loads of special features: it works fast, is convenient to use and flexible to configure. It now also uses some of the features of the new Operating System 2.0. Furthermore there is a Library-administration integrated for the fonts, to ease the administration of fonts, to visibly decrease usage of space on the harddisk and to increase the access speed. Plus, the driver now also supports the new Font standard and can work with fonts that have more than 128 characters.

Table of contents:

1. Copyright and similar	3
2. About ShowDVI	3
3. Executing from the CLI	3
4. The usage of ShowDVI	4
4.1. Popup-menu	4
4.2. The Main menu	5
4.3. Gadgets, Scrollbars and Requesters	6
4.4. Operation with the use of the keyboard	8
5. ARexx Port	10
5.1. The commands	11
5.2. Map of funktion keys	11
5.3. Generation of not found fonts with Metafont	12
6. Configuration files	12
6.1. The Startup File	12
6.2. The Fontdefinition File	14
6.3. The AmigaTeX-Configuration-File	17
7. “2.0” Features	17
7.1. Screen modes	17
7.2. Application Window	17
8. Internals	18
8.1. Environment Variables	18
8.2. Font Search Paths	18
8.3. Font Caching	20
8.4. Font Memory	20
8.5. Pagebitmap - Resolutions	20
8.6. Printing a page	21
8.7. Taskname and Signals	21
8.8. Required Stack	22
8.9. Starting in the background	22
8.10. ARP Interface	22
9. The Logfile	22
10. Required Soft- Hardware	23
11. Known bugs	23

1. Copyright and similar

The program ShowDVI is “freely distributable copyrighted software”. Which means, that you may copy and use it, but the Copyright remains entirely with me. The program can also be redistributed unaltered only.

The program can not be sold commercially without my explicit and written prior consent. If distributed as PD-Software, the price must not be higher than 5.- DM (=3 US\$) per disk.

It also can only be redistributed as a complete and unaltered package.

For further explanation see the README file.

2. About ShowDVI

ShowDVI, What do you need it for?

ShowDVI is a program, with which you can view the output of the \TeX -program on the screen. Usually such a program is called \TeX -Previewer

What do you need this program for? Though \TeX is an excellent wordprocessor, but not a WYSIWYG-program¹. You have to enter the \TeX input in a normal ASCII editor and then process it with `virtex` or `tex`. The program \TeX produces from this – provided the input is error free – a DVI file². This can be printed with a printerdriver. But as you wouldn't want to print the entire document again after correcting an error, there has been made a beginning with developing previewers, with which you can display the document on screen before printing it.

ShowDVI is my contribution to the already existing previewers, and as you shall see in this manual not a bad one.

A few particularities upfront:

- Support of fontlibraries (saves memory and speeds up access).
- Elaborate search-algorithm with the help of environment-variables and definition files, to locate fonts.
- Can work with PK-files also.
- Fast on screen display.
- ARexx port.
- Highly configurable. Display of current status in a configuration file.

3. Executing from the CLI

ShowDVI should be always called from the CLI. Though start from the Workbench is possible, it is not further supported. But this is not tragical though, as \TeX itself can only be started from the CLI too, which makes the CLI indispensable anyway.

For that there is a multitude of command line options. The call format looks like this:

¹ what you see is what you get

² device independent

- ```
showdvi [-?] [-a fontarea] [-b byte] [-f page] [-z dpi] [-l] [-s]
 [-h offset] [-v offset] [-p] dvifile
 and offset = real number{pt|pc|in|bp|cm|mm|dd|cc}
```
- ? Displays a short explanation of the options, without starting the program.
  - a Inserts an additional directory in the search path of the fontlibraries and the PK-directories. For more see chapter 7.2.
  - b Size of the memory range, in which loaded fonts are kept. The display is in bytes. The default value can be displayed with the -? option.
  - f *n* Starts the display at page *n*.
  - z *n* Overrides the default resolution and sets the value to *n* dpi instead. The default resolution is 100 dpi.
  - h *xxx* Sets the horizontal offset to *r*. *xx* is the unit. These are the units you can choose from:
    - pt point (is used by T<sub>E</sub>X)
    - pc pica 1pc = 12pt
    - in inch 1in = 72.27pt
    - bp big point 72bp = 1in
    - cm centimeter
    - mm millimeter
    - dd didot point 1157dd = 1238pt
    - cc cicero 1cc = 12dd
  - v *xxx* Sets the vertical offset to *r*. *xx* is units, like at the -h option.
  - p Loads all fonts, which are used in the document, at the start of the program in the internal font memory.
  - s Outputs some additional information to the logfile. See also chapter 8.
  - l Prohibits the use of a logfile.
  - dvifile* The DVI file to be printed or a directory, that becomes home directory of ShowDVI. May be omitted.

## 4. The usage of ShowDVI

On the contrary to its calling, operation of ShowDVI is easy and very versatile. At that there is one particularity: No pulldown-menus have been used – as usual on the Amiga – but popup-menus. In my opinion, popup-menus are significantly easier and faster to operate as the normal ones, and the distances, that the mouse has to travel are shorter.

### 4.1. Popup-menu

The popup-menu appears after pushing the right mouse button, directly at the mouse pointer. The separate entries of the menu are placed in rows beneath each other. The rows, which end in an arrow, have another submenu. If you bring the pointer in the vicinity of one of these arrows the submenu will unfold. Menu entries are selected by releasing the right mouse button on top of them. Should you want to chose the first entry in a submenu, it suffices to select the submenu entry in the popup menu.

Two other, slightly different manners of operation of the menu can be set through the configuration file (see chapter 6). The menu either always appears slightly to the lower

right of the mouse pointer, or the program remembers the last menu entry which was selected. After repeated activation of the menu this entry is then selected immediately. This mode is also the default mode.

## 4.2. The Main menu

The main menu contains the following entries:

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>about</b>            | Shows a requester with a highly interesting piece of text. With this requester you can bring up a short help screen on keyboard sequences.                                                                                                                                                                                                                                                                                                      |
| <b>+ ++</b>             | This is a two part menu entry. With <b>+</b> you get to the next page, and with <b>++</b> to the last page.                                                                                                                                                                                                                                                                                                                                     |
| <b>- --</b>             | Equals the above menu entry, except that it does go to either the previous or the first page.                                                                                                                                                                                                                                                                                                                                                   |
| <b>show full page</b>   | Reduces the current page, in a manner that it fits the screen entirely.                                                                                                                                                                                                                                                                                                                                                                         |
| <b>outfit =&gt;</b>     | This is a submenu, in which the commands to change the looks of the ShowDVI screen are located.                                                                                                                                                                                                                                                                                                                                                 |
| <b>load DVI =&gt;</b>   | A submenu to load a new DVI file.                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>resolution =&gt;</b> | With this submenu the resolution of the screen display can be altered.                                                                                                                                                                                                                                                                                                                                                                          |
| <b>print page</b>       | Command to print the current page. This happens through a system call. Thus you can select the printer type and resolution required through the Preferences program. Is there already a printing process running then the menu will look like this: <b>abort print</b> . With this option you can abort the currently running printing process. (Though your printer may go on printing for a while until the printer's buffers are empty. -TT) |
| <b>hide quit</b>        | Another two part menu entry. With <b>hide</b> the ShowDVI screen is pushed to the background and <b>quit</b> ends the program after checking with the user if this should really happen. You can by the way end the program with Control-C without the safety check.                                                                                                                                                                            |

Now an overview of the submenus:

### 4.2.1. The submenu “outfit”

This submenu contains the next entries:

|                       |                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------|
| <b>unset lace</b>     | Switches from interlaced mode to the noninterlaced mode. After this the entry is changed to <b>set lace</b> . |
| <b>hide scrollbar</b> | Removes the scrollbars. Should the scrollbars not be active, the entry is called <b>show scrollbar</b> .      |
| <b>change color</b>   | calls a color requester, with which the colors of the ShowDVI-screen can be altered.                          |
| <b>save config</b>    | Saves the current configuration in a file. The file format will be explained in chapter 6.                    |

As said before, to call the entry **(un)set lace**, it is not necessary to explicitly select it, it suffices to select the main menu entry **outfit**. Because **(un)set lace** is the first entry in this submenu, it is selected. Of course, exactly this way do the other submenu entries that are first function.

### 4.2.2. The submenu “load DVI”

This submenu is comprised of two entries. Firstly the point `load file again`, Which loads the current file again and jumps to the exact same position as you were on again. Secondly the point `load new file`. This one calls the ARP file-requester, with which you can choose a new DVI file. Of course this only functions if you are in the possession of the “arp.library”.

### 4.2.3. The submenu “resolution”

This is a special menu, as it can be defined by the user. It consists of up to 10 DPI-entries, with which you can alter the resolution, in which the DVI file will be displayed. Besides this though the fonts of the particular resolutions must be available to you. Without alterations of the user the menu has three entries: `100 dpi`, `83 dpi` and `120 dpi`. You can read about generating your own entries in chapter 6.

## 4.3. Gadgets, Scrollbars and Requesters

The program has about 18 different gadgets, scrollbars and requesters. Though most of them are self-explanatory, some also have a hidden function. Thus every single element is going to be explained now.

### 4.3.1. The Gadgets

The biggest part of the gadgets is within the menu bar. From left to right:

- The “Window-Close” gadget. With this one you can end the program after a safety check by the program.
- The “--” gadget. By clicking on this gadget you will get to the first page of the document.
- The “-” gadget. This gadget folds to the next page.
- The “+” gadget. This corresponds to the “-” gadget. With it you get to the next page.
- The “++” gadget. With it, you can get to the last page of the DVI-file.
- The “D” gadget. This is a string gadget. Normally it displays the current directory, e.g. the directory in which the current DVI file is to be found. Of course you can also alter it. This directory is also the current directory for the file requester.
- The “F” gadget. In this string gadget the current DVI file is displayed. You can also enter a new file here. If you only click on it and `RETURN` is pressed the current file will be reloaded. This is only paractical if you have edited and compiled the `TEX`-File in the background in the mean time.
- The “#” gadget. Here the current page number is displayed. Of course you can select a new page number with this. In case a logical page number occurs more than once in the DVI file, then it is displayed to the right of the gadget, which consecutive page with this number is the current page.

On the rightmost border, under circumstances, a button called “OK” can appear. More about this at the review of the scrollbars The other gadgets will be explained together with the scrollbars too.

### 4.3.2. The Scrollbars

On the lower and right border the scrollbars are located. Normally they behave no differently than usual. If, however, the lower-rightmost gadget is selected, the vertical scrollbar will be switched to the “Page Scroll Mode”. At that the appearance will change as follows: Above the scrollbar an OK-button will appear, the previously selected button will change in a NO-switch and the scrollbar itself will display the current page. Now you can choose a page with the mouse. And lastly, with the OK gadget you jump to the marked page or you quit the Page Scroll Mode with the NO button.

This can be fully done with the keyboard as well. The “s” key switches on the Page Scroll Mode. An arbitrary page can then be selected with the cursor keys. It is actuated with all the keys that otherwise result in a change of the page (e.g. RETURN/ENTER/BACKSPACE). Repeated pressing of the “s” key equals the NO gadget, the “Page Scroll Mode” is left without change in the page number.

Another particularity keep the up- and down- arrows of the vertical scrollbar in store for you. If located on the end of a page, you can jump to the beginning of the next page by double-clicking on the arrow pointing down. Are you on the other hand on the beginning of a page, you can get to the end of the previous page with a double-click on the arrow pointing upward. This way you can, by only clicking the both arrows, scroll through the entire DVI file.

Of course, you can also switch off the scrollbars. Either with the key “o” or per menu entry “hide scrollbar” in the outfit-Menü. The state of the scrollbars, visible or not, can be saved in the configuration file, naturally.

### 4.3.3. The Requesters

There are four requesters in the program:

- The color requester. With it both the colors of the ShowDVI screen can be altered.
- The “about” requester. With it you can get to the copyright notice on one hand, and to a short help on the keyboard commands on the other hand.
- The file requester and
- the program-end requester.

The color requester consists of seven gadgets. The upper two are for the selection of the color, which should be changed. Beneath these two, there are three sliding-regulators, with which the individual color components (red, green and blue) can be adjusted. With the OK-gadget, below on the left, a new color set can be accepted then, or with the CANCEL-gadget the requester can be left without changes in the colors. To call the color requester press either AMIGA-c or select the menu entry **change color** in the submenu **outfit**.

The “about”-requester should be read at least once, and a conclusion should be drawn. Should you want to read a small help set on the keyboard commands, then you should select the **HELP** button below on the right. You can leave the requester in any case when selecting the double lined OK button. You can call the “about”-requester either with the **HELP** key or with the upper menu entry **anout**.

To be able to work with the file requester, you need the “arp.library”. The use of the requester is quite easy: To choose a file, either click on it twice, or just once and then

select the OK button. To change the directory, either click on it, select the button **DRIVES** or the button **PARENT**. Only files with the ending “.dvi” and directories are displayed by the way. The file requester is called either with **AMIGA-o** or with the menu entry **load new file** in the submenu **load DVI**.

Lastly, the program-end requester comes always <sup>3</sup> then, when you want to leave the program. With **YES** you end the program definitely and with **Oh, No!** you can resume your work.

## 4.4. Operation with the use of the keyboard

Everything that you can execute with use of the mouse, is also available through keyboard commands.

### 4.4.1. Scrolling within a page

Inside a page, you either scroll, if scrollbars are active, with the mouse or with the cursor keys. With the cursor up key you jump to the beginning of the page, and with the cursor down key you jump to the end of the page. Similarly for left-right. If the shift key is pressed the scrolling speed is smaller. With Alt and a cursor key you get to the appropriate border of the page.

### 4.4.2. Simple turning of a page

You can leave the current page and jump to a new one with 7 different keys. The simplest of these are:

- **RETURN**; With it you get to the next page, but remain on the same height in the page. Together with a key like Shift or Alt you jump to the last page.
- **BACKSPACE** or **-** on the normal keyboard (i.e. not on the number keypad. **-TT**). With these two keys you get to the previous page and stay on the same height in the page again. Also with the keys Shift or Alt you jump to the first page.
- **ENTER** has the same function as **RETURN** except for the fact that it gets you to the beginning of the page. This is handy for scrolling through the file.
- **-** on the number keypad has the same function as **BACKSPACE**, except that you always end up at the bottom of a page.

If you keep the **Ctrl** key pressed while you press any of the above mentioned keys, you get temporarily in the “physical mode”. An explanation of this is following in the next few paragraphs. For now this much: you then do not get to the page with the next highest page number when pressing **RETURN**, but to that one which is next in line after the current page, in the DVI file.

The rest of the possibilities need a further explanation.

### 4.4.3. Page-turning with the “PJC”

“PJC” or “Page Scroll Counter” is an at most four digits long numerical variable, which can be set by the keys 0 through 9. Its current value, if set, is displayed at the top on the right of the window-titlebar next to the **#** gadget. With the key **c**, its contents can be erased.

---

<sup>3</sup> Ctrl-C leaves the program immediately



What do you need this counter for?

With **g** or the ‘.’ on the number keypad you jump to the page that the PJC indicates. Thus, by entering **c 1 5 g** you jump to page 15. After that, the counter is set back to zero automatically.

On top of this, it is also used with the other keys for changing a page. With **3 RETURN** you jump forward three pages with **7** – seven pages back.

With the help of this counter, the number keypad and cursor keys are completely sufficient, to move swiftly in a DVI file.

The keys **g** and **.** too can be switched temporarily to the use of the physical page numbers by pressing (and holding) the **Ctrl** key. Thus, with **1 5 Ctrl-g** you get to the 15th page within the DVI file. This can have without problem one of fifteen different (logical) page numbers.

#### 4.4.4. Miscellaneous

There is a list of other keys, which are inhibited with functions:

- f** Displays the current page in the “whole-page-mode”. To this end the page is reduced in a manner that the current screen resolution can be fully displayed. Besides the page itself, its dimensions are displayed. The unit which is used at this, can be saved in the ShowDVI-configuration file. You have the choice between: inch (in), centimeter (cm) and point (pt). If you change to another page in the “full-page-mode”, this will be also displayed in this mode. You can get back to the normal display mode by pressing the key **f**. For confirmation that you really are in the “Full Page Mode”, an ‘F’ is displayed in the status field, to the right, next to the **#** gadget.
- i** Switches between interlaced and noninterlaced modes.
- o** Switches the scrollbars on and off.
- c** Erases the Page Jump Counter, i.e. sets it to zero.
- a** Reloads the current DVI file, but remains on the same page.
- s** Switches the Page Scroll Mode on. With the cursor up and down keys you can then select the wanted page and with one of the page switching keys (**RETURN ENTER BACKSPACE - g .**) jump to it. Repeated pressing of the key **s** works as the **NO** gadget.
- y** With this key you switch between the two different possible page number interpretations. For one, there is the possibility of interpreting each page number as that one which it has in the **T<sub>E</sub>X**text (logical number). If, for example, you fix the page number on seven with **\pageno=7** at the beginning of the **T<sub>E</sub>X**text, then, normally, the previewer interprets the physical first page of the DVI file as the page with the number seven. However, if you switch to the physical page number mode with the ‘y’ key, then the first page of the DVI file will be labeled with the page number one. This mode is shown with a ‘Y’ on the right side of the window titlebar.
- , ,** (SPACE) Similar to the **more** utility. Scrolls down about one screen. If you were on the lower border of the page, the next page is shown.
- p** Prints out the current page. As long as the printing process has not been finished, a “P” is displayed next to the **#** gadget to the top right of the screen

- (As mentioned above, your printer might print on after the computer thinks the printing is completed, because of an internal printer buffer. -TT). By repeated pressing of the **p** key the printing process can be aborted.
- d** Brings the workbench screen to the front and activates the window, that was active before ShowDVI was activated.
  - !** Opens a shell window on the ShowDVI screen. ShowDVI waits, until the shell has been closed again with `endcli`. Warning: This is the only function in the ShowDVI program which does not adhere to common programmer guidelines. Thus it is very well possible, that this is not going to function in a new operating system version. (Or leads to a crash. -TT)
  - AMIGA-c** Calls the color requester.
  - AMIGA-o** Opens the file requester.
  - AMIGA-q** Ends the program.
  - ESC** This key has two functions depending on its setting in the configuration file. Either the key ends the program, or (normally) pushes the ShowDVI screen to the background and activates the window, which was active before ShowDVI was activated.

Besides this you can overlay the function keys with ARexx scripts More about this in the next chapter.

#### 4.4.5. About the physical page numbering...

As mentioned above, you can switch between a logical page numbering (default) and a physical one with the 'y' key. But what are the major differences between these two?

In a normal plain- $\text{\TeX}$  text they do not show at all. Only when you alter the page numbering inside a  $\text{\TeX}$  file do differences occur.

For one, at turning pages. The same way as, in case of the logical numbering, you always get to the page with the next biggest number with the **RETURN** key, you always get, in case of the physical numbering, to the page which is coming next after the current page in the DVI file.

But also at jumping to the end of the text differences occur. In the logical mode, when pressing **SHIFT RETURN**, you jump to the page with the biggest logical page number. In the physical mode on the other hand, to the last page within the DVI file.

As mentioned, you switch between the two kinds of page numbering methods with the 'y' key. But as also already mentioned, you can only switch from the logical mode to the physical mode temporarily (only for the command to be executed). This happens through the Control-key. Pressed together with one of the keys **RETURN ENTER - BACKSPACE . g** the next change of page is executed in the physical mode. After that you are back in the logical mode immediately.

## 5. ARexx Port

As every program that regards itself as respectable, the ShowDVI program has an ARexx port too. The name of the port is "showdvi". The file extension for scripts is ".sd".

## 5.1. The commands

At this moment, the program has next commands to its disposal:

|                                |                                                                                                    |
|--------------------------------|----------------------------------------------------------------------------------------------------|
| <code>first</code>             | go to the fist page                                                                                |
| <code>last</code>              | go to the last page                                                                                |
| <code>prev [nr]</code>         | go back <i>nr</i> pages, or if <i>nr</i> is missing, to the previous page                          |
| <code>next [nr]</code>         | go forward <i>nr</i> pages, or if <i>nr</i> is missing, to the next page                           |
| <code>goto nr</code>           | go to page number <i>nr</i>                                                                        |
| <code>tofront</code>           | screen to the front                                                                                |
| <code>toback</code>            | screen to the background                                                                           |
| <code>wbtofront</code>         | workbench screen to the front                                                                      |
| <code>screen</code>            | display the hexadecimal address of the sreen                                                       |
| <code>fullpage</code>          | change to display in the full page mode                                                            |
| <code>printpage</code>         | print the current page, or if a printing process is running already, abort it.                     |
| <code>setdir dir</code>        | change current directory to <i>dir</i>                                                             |
| <code>loadnew file</code>      | load a new DVI file                                                                                |
| <code>loadagain</code>         | reload the current DVI file                                                                        |
| <code>getdir</code>            | get the current directory name                                                                     |
| <code>getfile</code>           | get the filename of the current DVI file                                                           |
| <code>getpage</code>           | get the current page number                                                                        |
| <code>togglelace</code>        | change the display mode (lace ↔ hires)                                                             |
| <code>togglescrollbar</code>   | scrollbar on/off                                                                                   |
| <code>setcolor nr r g b</code> | set the screen color <i>nr</i> to <i>r g b</i>                                                     |
| <code>setresolution dpi</code> | set the current resolution to <i>dpi</i>                                                           |
| <code>beep</code>              | abort a DisplayBeep                                                                                |
| <code>message string</code>    | display a message. This will be written on the screen menubar and overwritten after seven seconds. |
| <code>refresh</code>           | screen refresh (switches off the fullpage mode)                                                    |
| <code>spawn procedure</code>   | interpret <i>procedure.sd</i>                                                                      |
| <code>version</code>           | returns the version string                                                                         |
| <code>saveconfig</code>        | save the current configuration                                                                     |
| <code>exit</code>              | end ShowDVI                                                                                        |

The output of commands like `getfile` or `getpage` are at `options results` set, transferred to the `result` variable

Should important commands in the present implementation of the ARexx port not be present, you should contact me. In case they seem meaningful, I will gladly implement them.

If you possess the commercial AmigaTeXversion: The few ARexx commands present in its previewer are implemented in mine too. These are: `getcwd` (equals `getdir`), `filename` (= `getfile`) and `pagenum` (= `getpage`). `screen` and `exit` are present anyway.

## 5.2. Map of funktion keys

In the configuration file an ARexx script can be given for every function key with and without Shift, which will be executed at the pressing of the appropriate function key.

More on this in the chapter about the configuration file.

### 5.3. Generation of not found fonts with Metafont

If the previewer cannot find a font needed for displaying, and the environment variable “CALLMF” is set, the contents of this variable is used as the name of an ARexx script to execute.

This is called for the generation of not found fonts with the three following parameters:

- the fontname of the font to be generated,
- the resolution of the font in DPI (Dots Per Inch -TT)
- the basic resolution, which is presently set in the previewer.

This means that after

```
setenv CALLMF callmf
```

the previewer would call the script with

```
callmf cmr10 120 100
```

in case the font “cmr10” with the resolution “magstep1” has not been found and the previewer would presently work with a ground resolution of 100dpi.

## 6. Configuration files

ShowDVI works together with three configuration files. Firstly this is the “Startup-File”. There, amongst others, the looks of the ShowDVI screen is recorded.

In the other two it is defined where fonts and font libraries are to be found. One of them is in an own ShowDVI/DVIprint format.

The other is in AmigaTeX<sup>4</sup>-format. This is being looked for in the directory `TeX:pk`.

Both the private ShowDVI/DVIprint files are searched for in the directory `TeX:config`. They do not have to be explicitly there though, this can be set by the environment variable “DVICONFIG”. After `setenv DVICONFIG ram:` these configuration files are searched for only in the directory `RAM:.`

Both definition files abide the same rule: everything after a semicolon is ignored together with the semicolon itself. Besides this, spaces are not allowed in file- or directory names. This goes not only for the definition files, but also for the environment variables.

The AmigaTeXconfiguration file `fontvols` is only read then, when a font is not found. The other two are only read once at the start of the program.

### 6.1. The Startup File

The name of the ShowDVI configuration file is `ShowDVI.config`.

---

<sup>4</sup> AmigaTeX is a commercial product by Radical Eye Software

The file is interpreted by the ShowDVI program line by line. Lines beginning with a semicolon, or have tabs or spaces only in front of a semicolon, are ignored. All other lines consist of a keyword followed by the appropriate values. After this a remark could follow.

Warning: If you save the configuration file from within the ShowDVIprogram, the old one is simply overwritten. That means, Comments inserted per hand are lost.

An example for such a startup file is:

|                    |                           |                                       |
|--------------------|---------------------------|---------------------------------------|
| color_0            | 3,2,2                     | ; values of background color          |
| color_1            | 6,7,8                     | ; values of foreground color          |
| scrollbar          | on                        | ; state of scrollbars                 |
| interlace          | on                        | ; interlace                           |
| beep               | on                        | ; beep on warnings                    |
| esc-exit           | off                       | ; exit prog on ESC                    |
| int-menu           | on                        | ; is the menu intelligent             |
| big-menu           | on                        | ; menu with Topaz 11                  |
| physical-numbers   | off                       | ; on == use physical numbering        |
| unit               | in                        | ; used unit                           |
| application-window | on                        | ; application window on/off           |
| app-icon-name      | TeX:config/AppWin-ShowDVI | ; icon file for app-win               |
| screen-mode        | pal                       | ; screen mode                         |
|                    |                           | ; (pal   ntsc   productivity   a2024) |
| default_resolution | 91                        | ; default start resolution            |
| resolution_menu    | 100,91,83,120             | ; entries for the resolution menu     |
| f1                 | testf1                    |                                       |
| f2                 | testf2                    |                                       |
| F1                 | testF1                    |                                       |
| F10                | testF10                   |                                       |

Here the screen colors (red, green, blue) are set with `color_?`.

With `scrollbar` (on/off) is set, whether ShowDVI directly after the start of the program scrollbars should appear or not.

`interlace` can be on or off. Resp. if either you want to work in the interlace mode or not.

With `beep` (on/off) is shown, if the user should be alerted to certain errors especially or not.

The on/off switch `esc-exit` sets, whether the program should end if the ESC key is pressed, or only the screen should be pushed to the background.

The next line with the somewhat exotic keyword `int-menu` determines the mode, with which the menu should work. With `off` the menu is fixed in a way that it always appears on the same spot, just below the pointer. With `on` the menu becomes “pseude-intelligent”, it then remembers the last selected entry and selects this one immediately after a repeated activation of the menu.

With `big-menu` you can determine, whether the popup menu should be in Topaz 11 (on)

or Topaz 8 (off) fonts.

The flag **physical-numbers** fixes whether the program should be in the “physical” page number mode directly after it is called.<sup>5</sup>

**unit** determines, in which unit dimensions within the ShowDVI program are to be measured. There is **in** for inch, **cm** for centimeter and **pt** for point<sup>6</sup>.

The next two lines **application-window** and **app-icon-name** belong together. With “**application-window on**” you tell ShowDVI to open an application-window on the Workbench screen. This has two functions. Firstly, you can bring the ShowDVIscreen to the front by double clicking in this window. Secondly you can make ShowDVI to load a DVI file by dragging the icon of the DVI file onto the Application-Window. This function however only works under an operating system version  $\geq 2.0$ . The size, the looks and the position of the Application-Window can be fixed by an “.info” file.<sup>7</sup> Normally, the file `TeX:config/AppWin-ShowDVI.info` is taken up for this. Through **app-icon-name** an arbitrary, different icon file can be used. The keyword **screen-mode** has any meaning only under the new operating system 2.0! With it, it can be indicated, in which mode the ShowDVIscreen should be opened. The choice consists of **pal** for PAL resolution, **ntsc** for the NTSC resolution, **productivity** for the new productivity mode and **a2024** (monochrome Hedley monitor -TT) for the new full page screen<sup>8</sup> The size of the screen is still influenced by the overscan preferences, which can be set in Preferences. In case ShowDVI finds an operating system version lower than 2.0, this line of the configuration file is ignored. In case the line is missing and Operating System 2.0 or higher is employed, then the “default Monitor” is used. This is at the time either **pal** or **ntsc**, depending on your location.

**default\_resolution** indicates, in what resolution the program should be started in.

After the keyword **resolution\_menu** up to 10 numbers follow. These determine the looks of the **resolution** submenu.

The rest of the lines, finally, define the mapping of four function keys. The procedures are searched for with the usual ARexx search strategy. The postfix used is “.sd”.

With the menu entry **save config** or per keyboard sequence **AMIGA-w** the present state is saved. That is, the at present resolution in use will become **default\_resolution**. Also the current mode of the screen resolution (interlace), the colors selected, as well as the state (on/off) of the scrollbars is saved.

## 6.2. The Fontdefinition File

In this file additional data can be provided, where single fonts or certain font libraries are located. The full search path will be explained thoroughly in the next chapter “Internas”.

---

<sup>5</sup> See the description of the ‘Y’ key and chapter 4.4.5

<sup>6</sup> pt is a unit, which stems from the printing business. A normal font is 10pt large. The line distance is 12pt then.

<sup>7</sup> More about this in chapter 7.2

<sup>8</sup> This mode has not been tested yet, as I do not have such a monitor.

The fontdefinition file is normally searched for in `TeX:config`. You can overwrite this directory though with the variable `DVICONFIG` too. But there is searched in one directory only though. So it is not possible to enter a search path. The name of the configuration file is `showdvi.fnt`.

Now to the structure of this file. Here an example:

```

flibenv dir1 dir2 ; directories with flib's
pkenv dir1 dir2 ; directories with pk-files

flib_str PK%04d ; formatstring for flib's
pk_str %d/%s.%d ; formatstring for pk-files
pkdir_str %s.pk ; fnt for pk-files defined in pkdir

flib dpi directory ; two args
font font dpi directory ; three args
pkdir dpi directory ; two args

flib 100 ram:fastflib ; example
font cmr10 100 ram:fastpk ; example
pkdir 109 TeX:pk/110 ; example
pkdir 110 TeX:pk/110 ; example

```

With the first two keywords you can define additional directories, in which should be searched either for fontlibraries, or for PK-Fonts. According to the function these two lines correspond to the environment variables `FLIBDIR` and `PKDIR`. More on this in chapter 7.1.

The next three lines define the filename of the fontlibraries and PK-Fonts. As the second argument a C like Format String is given. At this, the following modifiers can be applied.

- “%s” steht für den Font Namen (z.B. `cmr10`),
- “%d” steht für die aktuelle dpi-Zahl,
- “%h” steht für die dpi-Auflösung in horizontaler Richtung (entspricht “%d”),
- “%v” steht für die dpi-Auflösung in vertikaler Richtung,
- “%x” ist eine ältere Einheit, entspricht der horizontalen dpi mal fünf,
- “%y” wie “%x” nur für die vertikale Auflösung,
- “%%” steht schließlich für ein “%” Zeichen.

The “%h” “%v” “%x” “%y” modifiers are of no great importance to the previewer. They are used mostly for non quadratic fonts (e.g. the Epson FX-80 fonts).

The three format strings of the above table are at the same time the default values of the program.

Overall, the formatstring is completely C compatible. Examples:

| Flib-fmt | PK-fmt   | dpi | Name  | Flib     | Font            |
|----------|----------|-----|-------|----------|-----------------|
| PK%04d   | %d/%s.%d | 100 | cmr10 | PK0100   | 100/cmr10.100pk |
| PK%d     | %s.%d    | 83  | cmbx8 | PK83     | cmbx8.100       |
| %x.lib   | %s/%d    | 360 | lasy8 | 1800.lib | lasy/360        |

The lines with the keyword “flib” define the directory of the fontlibraries. After `flib` two parameters follow: first a numerical entry for the DPI of the fontlibrary, then the directory, in which the fontlibrary is to be found.

The lines with the keyword “font” are constructed similarly. This time there are three parameters. The first parameter indicates the font name, as it is defined in a `TEX`file. After that the DPI and the directory, in which the font is to be found.

The lines, beginning with “pkdir” are there to be able to use the very same fonts for different resolutions. If a font is looked for in the directory defined there for this purpose, then the formatstring “pkdir\_str” is used. In the above example the font `cmr10` would be searched in the resolution 109 dpi as well as in 110 dpi in the directory `TeX:pk/110` under the name `cmr10.pk` (if it was not found before in a fontlibrary or as “normal” PK-Font).

Note two particularities. Firstly, the line of order in which the entries are written in the file *within a* group of entries (e.g. all `flib` entries) is taken in account when searching for fonts. Thus, you can define several fontlibraries of the same resolution and fix the order in which they should be searched for.

A further particularity is, that you can define the format strings more than once. The last defined string is then used. An example:

|                       |                          |                          |                                  |
|-----------------------|--------------------------|--------------------------|----------------------------------|
| <code>flib_str</code> | <code>PK%04hx%04v</code> |                          | ; due to non-symmetric fonts     |
| <code>flib</code>     | <code>120</code>         | <code>TeX:fontlib</code> | ; use format string from above   |
| <code>flib</code>     | <code>131</code>         | <code>TeX:fontlib</code> | ; same                           |
| <code>flib_str</code> | <code>PK%04d</code>      |                          | ; resets format string           |
| <code>flib</code>     | <code>100</code>         | <code>TeX:fontlib</code> | ; now uses the new format string |

The first two fontlibraries defined now use the format string with the horizontal and vertical resolution. The library defined third, and *all*, which were defined while searching for the fonts during program execution, use the format string defined second.

The greatest use of this definition file will get those users with limited disk space. You can first determine, which fonts are definitely necessary. A great help at that is the logfile, which lists each font needed. Then you create for each resolution needed a fontlibrary, in which the previously determined fonts are then embedded. These are then copied to the hard-disk. The complete fontlibraries can be kept on disk or, in case they get to large, spread on several disks. If you define all libraries explicitly, then the fonts are searched in the file, in the order, in which they have been mentioned. Thus, first on hard-disk, and then - if not found - on the disk(s) which you have specified.

For users of METAFONT it can be of great help to define single fonts. Should you for example want to reproduce the font `cmr10`, then you don't have to erase it from all fontlibraries during the testphase already and reenter it again. Instead the font is defined in the definition file. Because it is only found then, ShowDVI does not regard the font with the same name in the fontlibraries.

Should you want to test the fontdefinition file, you should start ShowDVI with the option `-s`. Before the end of the program the definition file is written in the logfile together with comments (used/found/not found).



### 6.3. The AmigaTeX-Configuration-File

If a font has not been found after all the tries till presently, then the AmigaTeXconfiguration file `fontvols` is called in for advice. This consists of several lines in the following format:

*Volume-Name=unusedd.dpi<sub>1</sub>.dpi<sub>2</sub> . . . . dpi<sub>n</sub>*

“Volume-Name” is the name of a disk. “unused” is a random number. It is not used in the current version. The particular “dpi” values indicate, in what resolutions fonts are to be found on the disk.

A short example:

With “FontDisk=222d.131.207” it is defined, that a disk with name “FontDisk:” exists, which contains two directories. Firstly the directory “FontDisk:131” with the fonts of the resolution 131dpi, and secondly the directory “FontDisk:207”, in which the stored fonts of resolution 207dpi are to be found.

One resolution can exist more than one time at that. The search path is from up to down and from left to right.

This configuration file has to be in the directory `TeX:pk` with the name `fontvols`.

## 7. “2.0” Features

A few particularities of the new AMIGA Operating System could already be considered in this version. They have been mentioned already at the appropriate place in this manual, but should be summarized again here shortly.

### 7.1. Screen modes

All the AMIGA screen modes known up to now under 2.0<sup>9</sup> are supported. These are: *PAL*, *NTSC*, *Productivity (VGA)* and *A2024*<sup>10</sup>.

These modes can be used in either *interlaced* or *non-interlaced* mode. In case of the A2024 selection this means the difference between the 10 Hz and the 15 Hz resolution.

The required monitor and resolution can be selected through the configuration file. For that, see chapter 6.1.

Under 2.0, the eventual resolution is still deduced over the `Text-Overscan`. This can be set through Preferences.

### 7.2. Application Window

The Application Window is a very practical institution for people, who enjoy working with the Workbench. Now you can pull ShowDVI to the front by mouse from the Workbench or by dragging an icon you can load a new DVI file.

---

<sup>9</sup> multicolored ones like HAM etc.

<sup>10</sup> The A2024 Mode has not been tested, as I do not have such a monitor.

The looks, size and position of the application window can be determined entirely from the Workbench. And for this purpose the icon file exists which you can also indicate in the configuration file. The file `TeX:config/AppWin-ShowDVI.info` is used per default. You can edit this icon with `IconEdit` and adapt it to your own taste. To determine the position of the Apps-Window, you simply place it there where you want it and choose the menu entry `snapshot`.

Take note: These operations must be executed on the “.info” file and *not* on the Applications Window of ShowDVI.

## 8. Internals

This chapter is meant to convey some of the background information on the functioning of ShowDVI. Though this information is not necessarily needed for the simple use of the program, but can be quite useful.

### 8.1. Environment Variables

Only the ENV: Environment Variables supported Commodore are used.

Following variables are used:

DVICONFIG indicates the directory of the definition file.

FLIBDIR List of paths for finding the fontlibraries.

PKDIR List of paths for finding the PK-Fonts.

CALLMF when this variable is set, an ARexx script is called in case a font is not found, whose name is contained in the environment variable.

A list of paths exists from several directories separated by “!” or by “,”. For example:

```
ram:dir1!df1:dir2,fonts:
```

Spaces inside a directory are not allowed.

### 8.2. Font Search Paths

Beforehand: the program always favours the fontlibraries if possible, which has many advantages in my view. These are (summarized):

- some memory space is saved
- fast access is possible
- the directory structure is conveniently arranged

Where are the fonts searched for?

If you start ShowDVI without further preparations, then fonts are looked for first in the fontlibraries in `TeX:FontLib` and then as a independant PK file in `TeX:pk`. The particular filename is determined with the next format strings

`PK%04d` for the fontlibraries, and

`%d/%s.%dpk` for the particular PK-Files.

Of course, as already mentioned in the chapter about the fontdefinition file, these format strings can be redefined.

The first possibility to redefine are two environment variables. With `FLIBDIR` you can define further directories, in which the fontlibraries are to be found and with `PKDIR` directories in which PK-Files can be found. After

```
setenv FLIBDIR DH0:flib1!DH1:flib2!FlibDisk:
setenv PKDIR DH0:pk1!PKDisk:
```

a font is searched on the following directories:

- `DH0:flib1`, within a fontlibrary.
- `DH1:flib2`, within a fontlibrary.
- `FlibDisk:`, within a fontlibrary.
- `TeX:FontLib`, within a fontlibrary.
- `DH0:pk1`, as independant PK file.
- `PKDisk:`, as independant PK file.
- `TeX:pk`, as independant PK file.
- Should the font still not have been found, then the file `TeX:pk/fontvols` is searched for a disk, on which the font could be found.

If another directory was selected on the command line with the `-a` option, then this one will be searched before the current directory. Thus a fitting library is searched in the directory, before the `TeX:FontLib`-directory, and if till the end of the search-list the font has still not been found, then before the `TeX:pk`-directory, in the directory that comes after the `-a` option is searched.

Are these possibilities still not sufficient, then fonts and fontlibraries can be defined one by one in the fontdefinition file.

In that case the following search path is used:

- Firstly there is searched in the list of fonts defined one by one.
- After that, in the list of predefined fontlibraries is searched for libraries with the correct DPI-value and within these for the font.
- Should the font not be found, the Flib directories are searched now. These are defined with the environment variable `FLIBDIR` and the entry `flibenv` of the font-definition file.
- Now in the list of the PK-directories is searched for the fitting PK-file. This list can be changed through the environment variable `PKDIR` and the entry `pkenv` of the font-definition file.
- As the last but one try to find the font there is searched in the PK-directories as defined through the `pkdir` entries.
- Last but not least, the file `TeX:pk/fontvols` consulted. In case the font has been found at this, it is copied to `TeX:pk`. This is also known as *Font Caching*, as you can simply keep your fonts on disk and only those, that you really need are copied to the `TeX:pk` field. See also chapter 7.3 for this.

Should all of these attempts, to find a font, fail, and the environment variable “CALLMF” is containing an ARExx scriptname, then this script is executed with the three parameters “Font Name”, “Font Resolution” and “Basic Resolution”. As soon as this script has returned with the return-code zero, the same search algorithm is executed again. If thus

the script has produced the font with Metafont, it only needs to copy it to an arbitrary place within the search path, and ShowDVI uses it immediately.

### 8.3. Font Caching

Also taken from Amiga $\text{\TeX}$  is the idea of “Font Caching”. If a font was only found through the `fontvols`-file on a disk, then it is copied to the directory `TeX:pk/ldots`. This eases the selection for users who must get around without or with little harddisk space, of fonts which they have to be able to access fast.

A  $\text{\TeX}$ user, who only has two diskdrives and no harddisk, might get his act together in the following manner:

On the first disk the Workbench resides with the programs necessary, like `virtex` and `showdvi`. On the second disk the tfm files and the format files needed by `virtex` should be copied. Plus a directory with fontlibraries, which only contains the fonts which are rally needed all the time. Apart from these a directory `TeX:pk` has to exist on this disk, in which the `fontvols` file resides. In this file all disks are listed, which contain PK files.

If a font is not found on the working disk now, then the additional disk needed is requested and the font copies to the directory `TeX:pk`. Thus, next time this font will be available immediately. If, after a period of time, several fonts have been collected into `TeX:pk`, then you can enter them into the fontlibraries with the help of the fontlibrary program `flib`. This has the advantage, that you need somewhat less space on the disk<sup>11</sup> and the access will be somewhat faster.

### 8.4. Font Memory

To make a change of pages as fast as possible, the already loaded fonts are kept in RAM. This is practiced as long as the memory space reserved for this purpose is full. After that the LRU<sup>12</sup> method is used, and the font, which has not been needed for the longest time will be removed from the memory

Of course, it depends on your particular setup and configuration, to determine how much memory you can or want to delegate to the fonts. With the option `-?` the default value can be requested, and with `bbytes` an arbitrary value can be set at execution.

To find the optimal value for your setup, the logfile is the greatest help in this case too. There after every single entry of a loaded font the percentage of contention of the font memory is displayed.

### 8.5. Pagebitmap - Resolutions

For the scrolling on one page to be as fluent as possible, the entire bitmap of the current page is held in CHIP-RAM. As the CHIP-RAM is unfortunately limited, a limitation of usable resolutions occurs. Normal dimensioned pages can thus be displayed at 180 DPI, if not much more CHIP-RAM is needed for other applications. In my case then about 5 KB CHIP-RAM is left (from the 512 KB).

---

<sup>11</sup> on average about 768 Bytes per PK file

<sup>12</sup> last recently used

The size of the ShowDVI screen depends on the size of the Workbench screen. If the Workbench has been started in the overscan mode, then ShowDVI uses that mode too. Is the Workbench screen in noninterlaced-mode and should the ShowDVI screen be opened in the interlace-mode, then the y-resolution of the Workbench screen is doubled.

## 8.6. Printing a page

As internally the page is stored as a RastPort, it can be printed very easily with the io\_command PRD\_DUMPSPORT. As this functions over the printerdriver as selected in the preferences, this function can be used on every printer capable of printing graphics. The resolution is not as good though, since ShowDVI works with a low resolution. With 1 MB CHIP-RAM you should be able to print with up to 300 DPI.

A print that is not ready yet can be recognized on the “P” in the right-upper corner on the menu bar. During this you can neither leave the current page, nor load a new DVI file. A running printing process can be aborted with the menu entry or key with which it has been started.<sup>13</sup>

## 8.7. Taskname and Signals

The taskname of the program is set to “ShowDVI-Task” immediately after the program has started. Before this however, a test is performed, whether a task is already running under this name. If so, then this task is sent a CTRL-E signal. This persuades the other ShowDVI program, to bring its screen to the front and activate its window. The ShowDVI program that has just been started ends its activities right after it has sent the signal<sup>14</sup>.

If, when calling ShowDVI for the second time a filename has been given (Important: Filename *only* parameter), then, after the Ctrl-E signal the ARexx command

```
address "showdvi" "loadnew %s"
```

is executed, at which event the filename is substituted for %s, which has been passed on to the program as parameter. However, it is important that the file is searched for, relative to the directory, which has been set at the first started ShowDVI.

ShowDVI uses another signal – the Ctrl-F signal. If ShowDVI receives this signal, then the ShowDVI screen is pulled to the front and the current file is loaded again.

Thus you can communicate with the ShowDVI program in a limited fashion, even if you are not the owner of ARexx. You can at least bring it to the front, activate it and reload the current DVI file with a signal. You can send the signal from your own program, terminate with the CLI command **break**, or simply transmit by calling ShowDVI again.

Another word to the automatic activation of ShowDVI. If you activate ShowDVI with Ctrl-E or Ctrl-F, then it will remember the window active before it. Now, if you push the ShowDVI screen to the back again with the ESC key, then the window active previously is activated automatically.

---

<sup>13</sup> Warning: Due to a very well hidden bug (unintentionally! :) sometimes a system crash is caused

<sup>14</sup> This means that only one ShowDVI program can be running at the same time

ShowDVI takes another signal in consideration. Being the Ctrl-C. With this key combination you can end ShowDVI anytime in practically every situation, without a question.

## 8.8. Required Stack

Contrary to earlier versions of ShowDVI, this one is satisfied with only 4000 bytes of stack. (Your CLI/Shell starts with that amount anyway, so nothing special has to be done. -TT)

## 8.9. Starting in the background

If ShowDVI has been started as a background task, no output is sent to “stdout”. This means, that in a case of error, no error messages appear visibly (in case the ShowDVI screen closes so fast, that you could not read the message there). Thus it is recommended not to switch off the logfile and to examine it then. The exact cause of error is listed there.

If the program crashes uncontrolled, the logfile is unfortunately not readable anymore. Then you should start it once again in the foreground to see if at least in the window an error message appears. In any case, an uncontrollable crash is something so odd, that you should contact me at once. ( :-) -TT) But please with detailed information, so that I can reproduce the error.

## 8.10. ARP Interface

The program works in theory also without the ARP library. Only two functions are not available then anymore. Firstly the file requester and secondly the environment variables can't be read.

Even if you are not working with ARP completely, the arp.library is recommended for a well cared for LIBS: directory. <sup>15</sup>

# 9. The Logfile

This file has been appraised some times already in this manual. Should you work with a harddisk (which is recommended anyway), you should abandon the option -l, which switches off the logfile. The speed loss of setting the logfile up is so minimal, that the potential use of it surmounts at all times.

The logfile is opened in the directory, in which ShowDVI was started, it's name is “showdvi.log”.

What information does this logfile contain?

On one hand warnings like:

- configuration file not found,
- predefined font/fontlibraries not found
- and erroneous DVI files.

Information like:

- what fonts were loaded,

---

<sup>15</sup> ARP : AmigaDOS Replacement Project ( -TT)

- how much the font memory was stressed,
- which DVI file was loaded
- and - with the `-s` option set - general statistics about the DVI file and especially about the font-definition file. There, then for example it is indicated, where the particular fonts can be found.

And lastly error messages, which caused the ShowDVI program to terminate its activities.

## 10. Required Soft- Hardware

As software is concerned, first you need a program which produces the DVI file. In general this program is called `virtex` or simply `tex`. Of great use is also the “arp.library”, as you would otherwise have to do without the environment variables and the file requester. A great help in easing your work is ARexx by William S. Hawes.

As hardware goes, above all you need an Amiga. The most important expansion is then a memory expansion. 1 MB main memory is then the absolute minimum. A hard disk is, if you want to work sensibly a must. For the ShowDVI program you don't need more than about 3 MByte on your harddisk, But if you would want to install the character sets for your printerdriver completely on the harddisk, then you need definitely more.

## 11. Known bugs

The abortion of a printing process can lead to a crash.

This is the only real bug of the program (that is known to me). However, there are a few minor flaws. For example, that error messages on the screen can be overwritten by something else very quickly, without you having the chance to read them. As a remedy at the moment you can use only the logfile, in which all important messages are recorded.

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

*(The rest of the page remains free for own bugs, notes)*