

Usage notes for DVIWIN 2.5

Hippocrates Sendoukas
March 15, 1993

General Topics

[Introduction](#)
[Moving around](#)
[TeX Specials](#)
[Printer Alignment](#)
[Interaction with TeX](#)
[Caveats](#)
[Packing List](#)
[Acknowledgements](#)
[Licensing Agreement](#)

Commands

[The File submenu](#)
[The Page submenu](#)
[The Size submenu](#)
[The Options submenu](#)
[Log](#)

Introduction

This document summarizes the use of DVIWIN which is a driver for DVI files under MS-Windows. Some portions of the code have been adapted from the DVI driver family by Nelson Beebe at science.utah.edu. This version is more optimized for smaller processors and exploits some unique characteristics of the operating environment. Most program functions can be accessed through standard menus or accelerator keys that are shown after each menu entry. The general idea is to build a bitmap with the current page contents and display it on the standard window. You can move around the document by using the scrollbars, menu commands or shortcut keys. You can also print the file on any printer supported by Windows, provided that the printer driver supports bitmaps (and the printer has memory for a full page of graphics), and your machine has enough memory to construct the bitmap. Finally, it supports TeX `\special{}` commands for inserting arbitrary graphics in a document. The program requires Windows 3.1 for proper operation; it will also work under Windows 3.0, but it will be unable to display or print any included graphics files (unless they are very small) because of a bug in Windows 3.0. The program opens as many font files as possible: please make sure that you set the maximum number of files in your system to at least 50 through the "FILES=" statement in your config.sys file.

Moving around

There are many ways to position the current window using either the mouse or the keyboard. When moving within a page, you can move by two different factors or a single pixel. The two factors are adjustable from the "Options" submenu; therefore, it is better to talk about a "primary" and a "secondary" factor (instead of a "large" and a "small" factor), because you can assign any values to them. The default settings associate the primary factor with a large movement and the secondary factor with a small movement. You can also move directly to each margin or to each corner. The following table lists the available commands and the mouse and keyboard bindings.

Operation	Mouse	Keyboard
Move by the primary scroll factor	Left button on scrollbar	Arrow keys
Move by the secondary scroll factor	Left button on small arrow at end of scrollbar	Shift-Arrow keys
Move by a single pixel	Right button on small arrow at end of scrollbar	Control-Arrow keys
Move to margin	Middle button on scrollbar; if you click on the upper part of the scrollbar, it goes up; otherwise, it goes down	ifAlt-Arrow keys
Move to corner	None	Home/End, Control-Home/End
Next/Previous page	Right button on vert. scrollbar; if you click on the upper part of the scrollbar, it goes to the previous page; otherwise, it goes to the next page	PgUp/PgDn
First/Last page	None	Control-PgUp/PgDn

If you double-click the left button at any location, the viewer will scroll the window down and to the right, so that the selected location is on the top left corner. If you double-click the right button at any location, the viewer will scroll the window up and to the left, so that the selected location is on the bottom right corner. If you double-click on a ruler, you will move only on the direction of that ruler. There is no keyboard equivalent for this method.

Finally, there is keyboard-only interface for those accustomed to the "more" program. The space bar produces a forward movement by the primary factor, while the Enter key produces a forward movement by the secondary factor; both keys however, move you to the next page if you are at the bottom of the page. The Control-space and Control-Enter (or the "b" and "d" keys for the Unix diehards) do the opposite of the above two commands; that is, they move you backwards. The main advantage of these commands is that they let you go through the entire file using only a single key.

TeX Specials

TeX does not support graphics directly: there is a provision for "\special{ }" commands that are optionally interpreted by DVI drivers, but there is no consensus about the format and the contents of these commands. The driver accepts commands for inserting arbitrary graphics in your document; all unrecognized special commands are ignored.

1. Graphics Files

We deal with various graphics files using two techniques: the first approach is to use standard Windows metafiles; these files provide several advantages: they can be scaled to various resolutions, can contain almost any graphics command, are reasonably fast and compact, and are supported by most Windows applications and the system itself. There is one complication with standard metafiles though: most applications do not produce disk files in this format; they do however use this format for transfers to the clipboard. Therefore, we can capture the contents of the clipboard and save them to a metafile, which can later be used in our TeX documents. This task is done by the utility "clipmeta.exe"; its operation is very simple: you just run it, and it prompts you for the filename to be used for the metafile. I tested this technique with Excel, PowerPoint, Word for Windows, 1-2-3 for Windows, Toolbook and Paintbrush; it works perfectly in all cases.

There is also another method for importing graphics, which was developed for PageMaker and is also supported by PowerPoint, Word for Windows, Toolbook and other programs. An application can use a dynamic link library, which is responsible for importing the specified file; the application can then simply display the imported graphic. The dynamic link library is called a graphics filter, and the above mentioned applications come with a collection of such filters for various formats; this approach eliminates the need for an intermediate metafile. The DVI driver can use any of these filters, and therefore, if you have any of the above mentioned applications (for the filters), you can import any file, provided that you have the appropriate filter. The DVI driver looks in "win.ini" for the entries under the section "[MS Graphic Import Filters]" (this is the same method used by Microsoft programs). The entries in this section should be in the form "description=driver,extension", where "description" is an arbitrary string describing the supported format, "driver" is the full pathname of the graphics filter and "extension" is the extension used by the format. I tested some of the filters that come with Word for Windows, and they appear to work properly, except that the "eps" filter does not import Postscript files; instead, it imports TIFF or Metafile pictures embedded in the Postscript file.

There is one more thing that you should be aware of: Aldus developed an extended metafile format (called a "placeable" metafile), which circumvents some limitations of standard metafiles. Unfortunately, they chose the same extension (wmf) as the one for standard metafiles, and this can be the source of some confusion. If you try to import a file with this extension, the DVI driver first treats it as a standard metafile; if this fails, then it looks for the appropriate filter. If the imported graphic contains text, you will get much better results by using the scalable fonts introduced in Windows 3.1.

2. Special commands

The general form of the special command is

```
\special{action filename, x-paper-size y-paper-size}
```

where:

"filename" is the name of the graphics file (without extension). If for example you want to read the file "fig.wmf", you will use only the word "fig". Note that the filename must be followed by a comma. If you need to import graphics files from other drives or directories, you can use full pathnames (without an extension), but you should use forward slashes (/) instead of the customary backslashes (\); otherwise, TeX will complain about undefined commands.

"x-paper-size" and "y-paper-size" are the desired width and height on paper. Both sizes consist of a positive number followed by a standard TeX dimension; valid dimensions are:

String	Name	Conversion
in	inch	1in = 2.54cm
bp	big point	72bp = 1in
pt	point	72.27pt = 1in
pc	pica	1pc = 12pt
sp	scaled point	65536sp = 1pt
dd	didot point	1157dd = 1238pt
cc	cicero	1cc = 12dd
cm	centimeter	2.54cm = 1in
mm	millimeter	10mm = 1cm

"action" is a string indicating how the driver should insert the graphics file in the document; if it is "center", the driver puts the specified file (using its "natural" dimensions) at the center of the indicated rectangle. If the natural dimensions of the graphic are larger than the indicated rectangle, the graphic will overflow into the adjacent text. If the action string is "isoscale", the driver will scale the graphic, so it just fits in the indicated rectangle, but it will use the same scale on both axes, so the aspect ratio remains the same and there is no distortion. If the action string is "anisoscale", the driver will scale the graphic to fit in the indicated rectangle without regard for the aspect ratio.

Suppose that you want to insert the metafile "fig.wmf" at a given point in the document, you want to scale it so that it takes 4.8 inches horizontally and 3.6 inches vertically, and you do not care about the aspect ratio. This can be easily done by the command: `\special{anisoscale fig, 4.8in 3.6in}`". You will also need to leave some empty space for the graph (after the special command). A sample macro might be:

```
\def\myfigure#1#2{
  \begin{figure}[ht]
  \caption{#2}
  \special{anisoscale #1, \the\hsize 3.6in}
  \vspace{3.6in}
  \end{figure}}
```

This macro takes two parameters: the filename and the caption of the figure. It uses the `\hsize` value to adjust for varying text widths. This is just a sample macro; you can generalize it, or you can customize it more for particular types of graphics files. Whenever the driver imports a graphics file, it produces an entry in the log file with the name of the graphics file and its natural dimensions. This information can be useful in determining the amount of space to leave on the document.

As mentioned above, the placeable metafiles provide information about the size of the image, while the standard ones do not. Therefore, it would be impossible for the DVI driver to place such metafiles correctly: the only possible action would be to "anisoscale", ie., to just fit the graphic into the space that you specified. There is a special procedure to help you in such cases: when you take the metafile from the clipboard, the program "clipmeta" tells you the dimensions of the metafile; write them on a piece of paper. In your TeX document, use the command:

```
\special{action filename, x-paper-size y-paper-size x-fig-size y-fig-size}
```

instead of

```
\special{action filename, x-paper-size y-paper-size}.
```

where "x-fig-size" and "y-fig-size" are the dimensions reported by "clipmeta". This should give you the necessary functionality. If you have the graphics filters to use the placeable metafiles (from Word, Pagemaker, Toolbook, etc.), use them by all means and forget about this procedure. If, however, you do not have these filters, you can achieve the same results

by the above trick.

Printer Alignment

If a printer has a resolution of 300 dots per inch and can print on standard "Letter" paper, you would expect to be able to use 2550 (8.5 x 300) pixels horizontally, and 3300 (11 x 300) pixels vertically. Most printers cannot print on the entire page, and Windows reports a smaller number of available pixels. For example, the HP LaserJet II reports 2400 x 3160 pixels, the DeskJet reports 2400 x 3130 pixels, and the Apple LaserWriter Plus reports 2394 x 3231 pixels. This is not a big problem, since any document has more than adequate margins; if however you want to position the text accurately, you need to know how the missing pixels are divided among the four sides of the paper; unfortunately, Windows does not provide any help on this subject, and we may need to find it out on our own. The DVI driver assumes that the missing pixels are symmetrically distributed; for example, on a LaserJet II, it assumes that there are 75 missing pixels on both vertical sides, and 70 missing pixels on both horizontal sides. If this assumption is wrong, you can set the missing pixels manually; this is done by using the "Print Setup" entry in the "File" submenu and selecting the "Align" button on the setup dialog. At that point, you have to enter the missing pixels on the left and top sides (printer origin). Your selections are stored in "win.ini" under the section "[Printer Origin]". When using the manual option, you may need to experiment in order to find the correct number of pixels. This can be easily done by processing the plain TeX file:

```
\nopagenumbers
\parindent = 0pt
\ vbox {
  \hrule height 0.01pt
  \vrule width 0.01pt height \vsize
  \hfill
  \vrule width 0.01pt height \vsize
  \hrule height 0.01pt}
\end
```

which produces an empty box containing the entire printable area by plain TeX. If the alignment is correct, the upper left corner of the box should be at 1 inch left and 1 inch down (relative to the upper left corner of the paper), while the lower right corner should be at 7.5 inches left and 9.9 inches down. I run alignment tests on several printers, and settled on the following values:

Printer	Horizontal	Vertical
HP DeskJet	75	60
HP LaserJet II	65	74
HP LaserJet III	75	75

Keep in mind that many printers are not very accurate in positioning the paper (eg. the LaserJet III), so you should be very careful if you need pinpoint accuracy.

Interaction with TeX

The program tries to simplify the usual "edit-compile-view" cycles; whenever it loses the input focus (this happens when you minimize it or switch to another program), it closes the current DVI file but remembers its position within that file. When it receives the input focus, it checks for any changes in the DVI file; if there are no changes, it continues from the same point. If on the other hand the file has changed (presumably by TeX), it reloads the file and positions you at the same place as before; it also tries to use as many of the old fonts as possible in order to minimize the "refreshing" time. This approach lets you switch from dviwin to the editor (or TeX) and back easily with no need for complicated interprocess mechanisms (so you can use it with any version of TeX). The only thing that you should not do is switch to dviwin while TeX is still running, because the DVI file is not yet valid; the program will complain if this happens.

Caveats

The DVI driver prints a document simply by sending a bitmap to the Print Manager. This technique is a bit simplistic, but it works on any printer supported by Windows and it is guaranteed that the printer output will be identical to the screen output. The only possible optimization is to avoid sending long sequences of whitespace (it already does this). There are much more efficient methods for printing a DVI file to some printers (eg., downloading fonts to most laser printers), but these methods do not work on all printers (on the other hand, my method is optimal for dot-matrix or inkjet printers); I prefer to use the most general technique possible, so the program does not depend on any particular printer. If you use a Postscript printer and are not satisfied with this approach, please use `dvips` which is a fine program (`dviwin` is still good for previewing); I cannot afford to tailor the program to every possible printer; this is supposedly the job of the operating system. I can promise you however to work on changes that will benefit the majority of the program's users.

The program relies on the Windows drivers for screen and printer output; the problem is that these drivers are often buggy: many video drivers choke on bitmaps larger than 64K; I have seen this problem with Trident, Paradise and Diamond drivers, but there are probably many more buggy ones; Trident adapters produce random GP faults if the bitmap is larger than 32K; other adapters get confused and display the first 64K bytes repeatedly. The current version of the program circumvents the common bugs, so it should work properly on all systems. I have gone to great pains to ensure that the program is very reasonable on the demands to the video driver; if you still encounter any display problems, I would humbly suggest that the video driver is hopeless and you better complain to the manufacturer.

The program's appetite for memory has been dramatically reduced; virtually no swapping should be necessary even when creating 300dpi bitmaps on a machine with 4M of RAM. I have managed to go up to 228dpi on a machine with only 2M of RAM on Standard mode (ie., with no virtual memory). If however you want to use the driver at 600dpi (or above), be prepared to have at least 6M of RAM or lots of time for swapping; the problem is that a 600dpi full page bitmap takes over 4M of RAM and there is no obvious way to decrease this requirement.

You can run multiple instances of the program with no ill effects; the only thing that you should keep in mind is that the program saves its configuration upon exiting. Therefore, if you run two instances of the program, the configuration of the last one will be retained for the future.

The driver has proven very useful to me, and I hope that it serves you equally well. I have tested and debugged the program extensively, but I cannot afford to make any guarantees; anybody who uses it assumes all risks. On the other hand, if you find any bug or have any suggestion for improvements, I will be more than happy to hear it and I will do my best to fix it. You can contact me via e-mail at "sendouk@scf.usc.edu", or regular mail at "3230 Overland Ave. #201, Los Angeles, CA 90034".

Packing List

Make sure that you have all the relevant files:

Filename	Description
dviwin.exe	DVI driver
dviwin.hlp	Help file for dviwin.exe
dviwin.wri	Printable documentation for dviwin.exe
wbr.exe	Text file browser
wbr.hlp	Help file for wbr.exe
wbr.wri	Printable documentation for wbr.exe
clipmeta.exe	Utility for exporting metafiles from the clipboard
clipmeta.wri	Printable documentation for clipmeta.exe
miscwin.dll	Utility routines shared by all executables
ctl3d.dll	3D dialog routines
demo.dvi	Demonstration DVI file
demo.tex	Source for demo.dvi
demo.wmf	Graphics file used by demo.dvi
dviwin2.hlp	Help file for dviwin.exe with large fonts
wbr2.hlp	Help file for wbr.exe with large fonts
helpme.wri	Answers to common questions and problems
dviwin.sub	Sample font substitution file
whats.new	List of changes between releases

The only required files for the DVI driver are "dviwin.exe", "miscwin.dll" and "ctl3d.dll". You can avoid the browser entirely by using another viewer or editor. If you intend to use graphics in your documents, I would strongly recommend the utility "clipmeta.exe". The file "ctl3d.dll" is distributed by Microsoft which requires it to reside on your windows system directory (typically c:\windows\system). All other files except for the documentation must reside either in a directory specified by the PATH environment variable, or the base directory of Windows. Of course, you will also need the PK font files as described earlier in this document. If you use a high resolution adapter (1024x768 or higher) you may find the standard fonts in the help files a bit hard to read; the files dviwin2.hlp and wbr2.hlp are identical to dviwin.hlp and wbr.hlp respectively, except that they use larger fonts for easier viewing. To use them, just rename them to dviwin.hlp and wbr.hlp.

Acknowledgements

All improvements in the new version are due to the kind users around the world who identified many missing features and incompatibilities with various hardware/software. I am indebted to all your suggestions and I would like to especially thank Mike Reid and Leonid Pryadko for their numerous ideas, productive discussions and extensive testing; of course, I remain fully responsible for any remaining bugs or omissions.

Licensing Agreement

The author of this software grants to any individual or non-commercial organization the right to use and to make an unlimited number of copies of this software. You may not decompile, disassemble, reverse engineer, or modify the software. This includes, but is not limited to modifying/changing any icons, menus, or displays associated with the software. This software cannot be sold without written authorization from the author. This restriction is not intended to apply to connect time charges, or flat rate connection/ download fees for electronic bulletin board services. The author of this program accepts no responsibility for damages resulting from the use of this software and makes no warranty or representation, either express or implied, including but not limited to, any implied warranty of merchantability or fitness for a particular purpose. This software is provided as is, and you, its user, assume all risks when using it.

The File submenu

This submenu lets you load a new file, close the current file, print the current file, configure your printer or exit the program. At the end of the submenu, you will also see the last four files that you viewed; you can open any of them simply by clicking on the desired filename; the program also remembers the position within each file, and will place you there automatically. The "Print" entry leads to a dialog box that lets you select the pages to be printed, and setup the printer (you can also select the active printer if you have installed multiple drivers). You can also select the resolution at which the document is sent to the printer; it defaults to the "Auto" value which is as close as possible to the printer's horizontal resolution, but you can easily override it. When exiting the program through the "Exit (Save Parm.s.)" command, all parameters are saved for future reference; You can use the "Quit" command if you want to exit the program without saving the current parameters.

The Page submenu

This submenu lets you move around the current page, or to other pages. You can also use the standard cursor keys to duplicate most of these functions. The "Goto" entry lets you specify the page that you want to display; a negative number specifies a page relative to the end of the document. That is, a minus one displays the last page of the document; a minus two shows the page before that, and so on. You can go to the next or previous page of the document by pressing the "PgDn" or "PgUp" keys or by clicking the right mouse button on the vertical scrollbar; a click on the upper half of the scrollbar is equivalent to a Page Up command, while a click on the lower half is equivalent to a Page Down command. The next four commands move you to each corner of the page; if you press them once, they move you to the respective margin; a second press moves you to the physical corner of the paper.

The Size submenu

This submenu lets you select the magnification applied to the document. You can use practically any magnification to accommodate most devices with square pixels (ie., with an aspect ratio of one). The last two entries in the submenu are initially empty: you can put there any desirable magnification to accommodate your screen or printer; this is done through the "Custom Sizes" entry in the "Options" submenu. There is a keyboard shortcut here not appearing on any menu: the plus key (in the numeric keypad) moves you one step up in the Size table; the minus key moves you in the opposite direction. This shortcut is available only for the standard resolutions; since the custom resolutions appear at the end of the table, they are not numerically ordered, so the shortcut is disabled to prevent any mishaps. Keep in mind that higher magnifications require more memory. For machines with 4M of RAM, the program works comfortably (ie., without significant swapping) with resolutions up to 400dpi on Letter or A4 paper sizes; you will need more memory for higher resolutions or larger paper sizes.

The Options submenu

This submenu lets you set several options about the operation of the program. All options (as well as window size and position) are saved in your system's configuration file (win.ini), so you do not need to set them repeatedly.

Trace Fonts

Rulers

Keep horizontal position

Keep vertical position

Warning beeps

Save parms. by default

Default margins

Apply margins

Font directory

Log viewer

Custom sizes

Font cache

Scroll factors

Magnifying Glass

Page Size

Trace Fonts

This option controls the tracing of fonts; the program creates a log file where it stores various messages. If this option is selected, the program will print in the log file all font activities. If you encounter any font problems (such as an empty screen), turn on this option before opening the dvi file, and look at the log file after the loading; the messages in the log file should help locate the problem. The program keeps up to seventeen fonts open; if more fonts are needed, it closes the least frequently used font. If the program cannot find a font even though that font exists and the path is correct, verify that you have a "FILES=50" (or higher) statement in your config.sys file.

Rulers

If this option is turned on, the cursor turns into a crosshair, two rulers appear, and the current cursor position is reflected on both rulers. This is very useful if you need to align text precisely. If you need more precise information about text positioning, you can press the left mouse button in conjunction with the control key, and the program will report the exact position of the cursor.

Keep Horizontal Position

The next option determines if the program should preserve your horizontal position when you change pages. This is useful if the displayed document is wider than the screen, and you want to align the window in order to see as much text as possible.

Keep Vertical Position

This option determines if the program should preserve your vertical position when you change pages. This is useful if the displayed document is taller than the screen, and you want to align the window in order to see as much text as possible.

Warning Beeps

If this option is turned on, the program will beep whenever you try to do something illegal (eg., trying to go past the last page of the document); otherwise, it will be quiet.

Save parms. by default

This option controls the meaning of the default exit (double click on the top-left window corner); if it is turned on, the program will save the parameters; otherwise, it is equivalent to the "Quit" command.

Default Margins

The next entry lets you set the default initial position when changing pages; TeX documents have both margins set to one inch; therefore, if you set the margins to one inch, you will maximize the initial viewing area. You can always move around the page (using the scrollbars or the cursor keys), but it is better to begin from a convenient location.

Apply Margins

You may find it inconvenient to specify the numerical values of the margins. If you select this command, the program will set the default margins to the current ones. In this way, all you need to do is position the document properly and select this command.

Font Directory

The next option specifies the structure of the font directory. We normally create a base directory (eg., c:\fonts) with a subdirectory for each resolution. Some programs (including dviwin 2.0) require the name of each subdirectory to be just the resolution number, while others require different names. A simple way to resolve the problem is to specify a directory template in order to find the fonts. The template consists of the base directory followed by a subdirectory containing the special characters "\$r" which will be replaced by the required resolution when the program is looking for a particular font. If for example your base directory is "c:\fonts" and the subdirectories are simply the resolution numbers, you should set the font directory to "c:\fonts\\$r". If on the other hand the subdirectories are of the form "100dpi", "121dpi", etc., you should set the font directory to "c:\fonts\\$rdpi". This method should help resolve conflicts among various DVI drivers. Whenever you change the font directory, the program tries to ensure that you have specified the \$r component in the template. If the specification is not correct, the program will not be able to find any PK fonts, so nothing will be displayed if you rely on such fonts. Similarly, you will get a warning if you specify \$r more than once.

The specification of the base directory is different from the first version of the program; if upgrading, make sure that you add the suffix \r to your base directory.

The program can also use font libraries (fli files) produced by emTeX. For each font that it opens, it will look first in all font libraries in the base directory and if it does not find the required font, it will search for PK files using the normal template. Note however that the font libraries distributed with emTeX consist of printer fonts only; you will either need to add screen fonts to these libraries, or create new font libraries with the screen fonts that you intend to use.

In some cases you may want to keep fonts under different base directories; this can be useful if you are on a network; in that case you may want to have often used fonts on a local hard disk for faster access, while keeping rarely used fonts on a network drive to conserve some disk space. This can be easily done by specifying multiple templates separated by semicolons (this is identical to the format of the PATH statement under DOS). If for example you keep the heavily used fonts under the "c:\fonts" directory and you have some rarely used fonts under the "m:\tex\pkfonts", you can specify "c:\fonts\r;m:\tex\pkfonts\r" in the template.

The program first searches for a font in all font libraries; then it looks for PK files. If it still cannot find the font, it will try neighboring resolutions. It will also try to find the file "dviwin.sub" in the font base directory; that file lets you specify which font to open if it cannot find the desired font; the format of this file is quite simple. A sample line might be:

```
cmbx10 58 -> cmbx6 100
```

This line instructs the program to use the font cmbx6.pk at 100 dpi if it cannot find the font cmbx10.pk at 58 dpi. You can also substitute an entire font family with another. If for example you do not want to store the cmdunh10 family, you can specify:

```
cmdunh10 -> cmr10
```

This instructs the driver to use the font cmr10 (at the appropriate resolution) if it cannot find the font cmdunh10.

You can also use comments in the font substitution file: everything after a percent mark (%) is ignored. Finally note that font substitutions may let you view a document even when you don't have the appropriate fonts, but the results will not be visually pleasing. It is a good idea to check the log file for any font substitutions if you ever see anything wrong about the fonts in your document.

The specification of the font substitution file is different from the first release of the program: in that version, the font substitution file was called "texfonts.sub" and the resolution was specified in TeX units for compatibility with the Beebe drivers. The problem is that the TeX

units are somewhat confusing since we normally deal with dpi. To avoid any further confusion, I renamed the file to "dviwin.sub" so that there are no clashes with any other programs expecting to use the file "texfonts.sub" with the old conventions. Inside this distribution, you will find a sample substitution file with comments showing the equivalent substitutions under the old conventions. I apologize for this incompatibility, but the other method was simply too confusing. The sample substitution file also shows an easy method to handle fonts with long filenames (eg., lcircle10 and lcirclew10 which are longer than the 8-character limit under DOS).

The program does not support GF or PXL fonts; if you have such fonts, you can use some standard TeX utilities (gftopk or pxtopk) to convert them to the PK format. The program supports up to 256 characters per font. You can get a decent set of PK fonts for a variety of resolutions from the SIMTEL archives (files "dvivga2.zip" through "dvivga8.zip" in the directory pd1:<msdos.tex>).

In general, you will need the base resolution and up to four magsteps for each device that you want to use; if for example you use a 640x480 mode, you will probably want to set the base resolution to 100dpi. Therefore, you will need the fonts at 100dpi, 110dpi (magstephalf), 121dpi (magstep1), 145dpi (magstep2), 174dpi (magstep3) and 208dpi (magstep4). If you use an 800x600 mode you will probably set the base resolution to 121dpi so you will need the fonts at 121, 132, 145, 174, 208 and 250dpi. If you use a 1024x768 mode, you will probably set the base resolution at 145dpi, so you will need the appropriate fonts (according to the above pattern). If you want to use the program for printing, you will also need a similar series beginning at the printer's resolution.

Log Viewer

This option lets you specify the program that will be used to view the log file when you use the "Log" command from the main menu. You can specify any browser or editor that you like; you can also use a small browser that I wrote (wbr.exe); it is quite convenient, can use any font installed in your system and can search for regular expressions. The browser is executed asynchronously (that is, you can switch between the browser and the DVI driver at will).

Custom Sizes

The table in the "Size" submenu lets you use most common resolutions, but it cannot account for every possible device in the market. Therefore, the program lets you specify up to two custom resolutions for your particular devices. Any positive numbers are acceptable, but make sure that you have the appropriate fonts for these resolutions. You can delete a custom size by specifying a zero value.

Font cache

When closing a file or loading another document, the program does not discard the old fonts; it keeps them in a cache, so the next time that it needs the same font, it can take it from there without accessing the disk; this can improve the speed significantly, but it takes some memory. For this reason, you can control the size of the cache which can range from zero up to fifty; a value of 20 should be enough for most documents. When setting the size of the cache, you can also see the number of cached fonts; this information should help you determine an appropriate size for the cache.

Scroll factors

The program lets you select the amount by which the window will scroll when you click on the scrollbar or press the arrow keys. The first scrolling factor applies to the left mouse button on the scrollbar (or the normal arrow keys). The second scrolling factor applies to the left mouse button on the small arrows at the end of the scrollbars (or the shift-arrow keys). These factors are expressed in percentages of the window dimensions. For example, a value of 80% for the first scroll factor means that when you click on the vertical scrollbar or press the up or down arrow, the window will scroll by 80% of its height (ie., there will be a 20% overlap). If you click on the horizontal scrollbar, the window will scroll by 80% of its width.

Magnifying Glass

The program lets you magnify a small area of the page for easier inspection by clicking the middle mouse button (or the left+right buttons). The magnification is done through simple bitmap expansion for performance purposes. The problem is that the expanded bitmap looks quite ugly; therefore, the program lets you smooth the resulting bitmap. This entry leads to a dialog box where you can select the degree of smoothing; this is accomplished by a 3x3 convolution (low-pass filtering) over the magnified bitmap. The magnifying glass can be dismissed by pressing any key or clicking any mouse button on its surface.

There is also another option called "Dynamic update" in this dialog; if you turn it on, then the contents of the magnifying glass are updated whenever you move it. If this option is turned off, the magnifying glass contents are not updated when you move it (they are updated when you click the left mouse button). This behavior is useful if you want to compare the close-up view to the normal view (otherwise, the normal view would be covered by the magnifying glass).

Page Size

This is another submenu where you can select the page size; it defaults to the standard "Letter" size (8.5 x 11 inches), but you can use most popular paper sizes including the metric ones (eg., A4). If the selected paper size is metric, the rulers are subdivided in centimeters; otherwise, they are subdivided in inches. Similarly, the margins and positions are specified in millimeters units if the paper size is metric.

Viewing the log file

The "Log" entry in the main menu displays all messages generated from processing the current DVI file. I took a rather unusual approach to implement this feature: the program always writes the messages on a temporary file (its name is related to the process ID, so there is no conflict between multiple copies of the DVI driver running simultaneously). When we select the "Log" entry, the driver runs a browsing program asynchronously and instructs it to read the log file. The advantages of this approach are that it was much easier to code, the browsing program is better than an embedded browser, and we can switch between the two programs at will, since we run it asynchronously. The only disadvantage is that the browser displays only the log file that was current at the time that it started; therefore, new messages will not appear automatically. If however you select the "Log" entry again, the DVI driver will start the browser again with the latest version of the log file. The browser defaults to "wbr.exe", but you can use any browser or editor that you like, by changing the "Log Viewer" entry in the "Options" submenu.

Hippocrates A. Sendoukas

Internet address: sendouk@scf.usc.edu

Mail address: 3230 Overland Ave. #201, L.A., CA 90034, USA

