

# Usage notes for DVIWIN 2.0

Hippocrates Sendoukas  
University of Southern California  
Dept. of Finance & Bus. Economics  
January 15, 1993

## 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 (which also borrows from previous work). This version is more optimized for smaller processors and exploits some unique characteristics of the operating environment; it requires Windows 3.0 or later and works only on standard or enhanced modes. All program functions can be accessed through standard menus or accelerator keys that are shown after each menu entry. Since the program operation is simple, we will discuss only the functions that may not be immediately obvious. The program builds a bitmap with the current page contents and displays 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.

## Main Menu

### File

This submenu lets you clear the current file, load a new file, print the current file, 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 "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). Note that when you print a document, the driver sends the bitmaps in the current resolution; therefore, you should select the appropriate resolution for your printer before printing a document (this is literally WYSIWYG). Furthermore, some printer drivers for Windows 3.0 have problems with bitmaps larger than 64K bytes; this has been fixed in version 3.1, so it is strongly advisable to upgrade.

### Page

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.

### Size

This submenu lets you select the magnification applied to the document. You can use any valid TeX magnification to accommodate most devices with square pixels (ie., with an aspect ratio of one). There is a keyboard shortcut here not appearing on any menu: the plus sign increases the magnification by half a magstep; the minus sign moves you in the opposite direction.

### Options

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:** 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. The program keeps up to ten fonts open; if more fonts are needed, it closes the least frequently used font. The same method is used when allocating bitmaps for individual characters; if you run out of memory (which is unlikely since Windows has virtual memory), the bitmap of the least frequently used characters will be deallocated to make room for the new characters.

**Remember Position:** The next option determines if the program should preserve your 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.

**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 both mouse buttons (or the middle button) and the program will report the exact position of the cursor.

**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.

**Font Directory:** The next option specifies the base directory for the fonts; this directory has one subdirectory for each magnification and the name of each subdirectory is simply the magnification number. Therefore, if the base directory is `c:\fonts`, the current magnification is 100 and you need the font `cmr10`, the program will try to open the file `c:\fonts\100\cmr10.pk`. If this file is not found, it will try neighboring magnifications. It will also try to find the file `texfonts.sub` in the font base directory; that file lets you specify which font to open if it cannot find the desired font; this substitution can apply either to a single font or to an entire family. Note that the program supports only PK type fonts (and will support any better types in the future). If you have GF or PXL fonts, you can use some standard TeX utilities (`gftopk` or `pxltopk`) to convert them to the PK format. The program supports up to 256 characters per font. You can get a decent set of fonts for a variety of resolutions from the SIMTEL archives. You can also generate them with Metafont and the font sources (from the TeX distribution tape), but this is a very slow process. It takes about three days on an 80386/25 to generate all plain, LaTeX and greek fonts for all magnifications.

**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).

**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.

## Log

This entry 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 program 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.

## Positioning the window

You can move around the current page using the scrollbars or the cursor keys. There is a minor twist however: normal arrow keys produce large movements, while arrow keys in conjunction with the control keys produce very fine movements. The reason for the unconventional setup is that only rarely we need to move the window by one or two pixels. There is also a mouse technique for quick positioning; 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.

## Specials

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

### 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"; 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 worked 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 from 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

### 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 (\), because 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 LaTeX 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 2400x3160 pixels, the DeskJet reports 2400x3130 pixels, and the Apple LaserWriter Plus reports 2394x3231 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:

<b>Printer</b>	<b>Horizontal</b>	<b>Vertical</b>
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.

## **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. 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; I prefer to use the most general technique possible, so the program works on all printers. The program relies heavily on the Windows drivers for screen and printer output; the problem is that these drivers are often buggy: many drivers in Windows 3.0 choked on bitmaps larger than 64K; this appears fixed in Windows 3.1. Even under this version, I found similar bugs in the Paradise video driver for the 640 x 480 x 32K mode.

The program has a rather large appetite for memory: this is not a big issue when previewing a file (because you normally use small resolutions), but it can create problems (lots of swapping to disk) when using larger resolution for printing. One of the problems stems from the 64K limit on memory segments: when the bitmap is larger than 64K, we have to ensure that no raster line crosses the 64K boundary. This condition can be satisfied by setting the horizontal size of the bitmap to the next higher power of 2. This is a stupid restriction and we will have to wait for a true 32-bit system to take care of it. Currently, you will need at least 4 Megabytes of RAM to print on a 300 dpi laser printer without excessive swapping.

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 documentaton for clipmeta.exe
miscwin.dll	Utility routines shared by all executables
ctl3d.dll	More utility routines
commdlg.dll	Common dialog routines (required only for Windows 3.0)
demo.dvi	Demonstration DVI file
demo.tex	Source for demo.dvi
demo.wmf	Graphics file used by demo.dvi

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" must reside on your windows system directory. All other files (except for the printable 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.

## 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.