Common questions about dviwin

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

This document attempts to answer the most common questions that I received about dviwin; please read this file about any problems, and if you cannot find the answer here, do not hesitate to contact me. My e-mail address is "sendouk@scf.usc.edu"

When I open a dvi file I get only an empty page. Why does this happen?

This indicates that the program cannot find any fonts; it usually happens if you did not install any fonts in your hard disk or the base directory specification is incorrect. If you are not sure about setting up the fonts, please read the manual. If you select the "Log" entry in the main menu, you will see the log file where the program indicates which fonts it cannot find. It is always a good idea to check the log file whenever something goes wrong. If you want to know exactly which font files the program is attempting to use, check the "Trace Fonts" entry in the "Options" submenu *before* opening the dvi file. *If you are upgrading from version 2.0, make sure that you change the font directory specification*. If for example your base directory is "c:\tex\fonts", you should change it to "c:\tex\fonts\$r". I am sorry for this incompatibility, but the new method is much better.

The program finds the fonts, but the letters are too big and on top of each other.

This happens when you try to preview at a screen resolution (eg., 100dpi), but you only have fonts for the printer (eg., 300dpi). If dviwin cannot find the correct fonts at 100dpi, it tries all neighboring resolutions until it finds a match. If the matching font is much larger than the requested font, you will end up with overlapping characters, because the positioning is still based at 100dpi. You can find a decent set of screen fonts in the files "dvivga2.zip" through "dvivga8.zip" in the directory pd1:<msdos.tex> of wsmr.simtel-20.army.mil, or the directory pub/msdos/tex of oak.oakland.edu. Unlike the first release of the program, newer versions look only for fonts within three magsteps of the original font resolution, so if you preview at normal screen resolutions but you have only printer fonts, nothing will be displayed.

I use a resolution of 100dpi for previewing and my documents display fine. When I try to print however at 300dpi, there are many characters missing.

This is probably an indication that you don't have the appropriate fonts at 300dpi. The dvivga fonts are reasonable for screen resolutions, but they are far from complete for printer resolutions; the easiest way to get the required fonts is to do an anonymous login at the CTAN archives (ftp.shsu.edu, ftp.tex.ac.uk or ftp.uni-stuttgart.de) and switch to the directory systems/msdos/emtex/lj_fonts. This directory contains several subdirectories with FLI files (EmTeX font libraries). Just get all these files and put them in your base directory; dviwin will use them automatically. When you do this, make sure that you get rid of any dvivga fonts at resolutions of 300dpi or above, to avoid wasting space on duplicate fonts.

I am still unclear about the font setup. Can you give an example?

My base directory is "c:\usr\texfonts". Under this directory I have subdirectories called "70", "76", "84", "92", "100" and so on. Each of these subdirectories contains a full set of PK fonts for the particular resolution. Therefore, when dviwin looks for cmr10 at 100dpi, it should look for the file "c:\usr\texfonts\100\cmr10.pk"; this is accomplished by telling dviwin that the font directory is "c:\usr\texfonts\$r". The same specification will also work if you want to use "fli" files in the base directory (c:\usr\texfonts).

I am using another dvi driver and the names of the subdirectories are c:\tex\pkfonts\100dpi, c:\tex\pkfonts\110dpi, etc. How can I instruct dviwin to use these directories?

Specify the font directory as "c:\tex\pkfonts\$rdpi" and everything will be fine.

The program cannot find the fonts lcircle10 and lcirclew10. Why does this happen?

These names exceed the DOS limit of 8 characters. If you try to create such files under DOS, the names will be truncated to "lcircle1" and "lcirclew" respectively. Dviwin will find these files in the PK format, but you may encounter problems if you store them in a font library without any name remapping. The easiest way around this problem is to add the following lines to the font substitution file:

```
lcircle10       ->      lcircle1
lcirclew10      ->      lcirclew
```

If you use the dvivga fonts from the simtel archives, you will find that they have renamed these two fonts to "circle10" and "circlew1" respectively. In that case, change the two substitution lines as following:

```
lcircle10       ->      circle10
lcirclew10      ->      circlew1
```

The file "dviwin.sub" already contains the last two substitutions. so you don't need to change anything if you use the dvivga fonts.

The program cannot find some fonts even though they are installed on my disk.

If the font directories are specified correctly, this may indicate that there are not enough file handles in the system. Make sure that you use a value of at least 50 for the FILES statement in your config.sys file.

I have some EmTeX font libraries in the base directory, but dviwin does not use them.

Older versions of EmTeX used another format for the font libraries; this format is now obsolete, and dviwin supports only the new format. If you have such libraries, you can update them to the new format by using a recent version of the "fontlib" program which is distributed with EmTeX.

I have EmTeX font libraries for the Epson 9-pin printers; these font libraries use the new format, but dviwin still does not use them.

The fonts in these libraries have a horizontal resolution of 240dpi and a vertical resolution of 216dpi. This corresponds to the highest possible resolution using 9-pin printers, but the standard

Windows drivers do not support this resolution. You can get good results by generating 240x144 dpi fonts (with metafont), adding this resolution to dviwin and setting the printing resolution (in the Print dialog) to "Auto". If you create FLI files out of the PK fonts, I would strongly recommend that you add the comment "0.6" (the aspect ratio) to the FLI files; this will prevent any confusion with other FLI files having the same horizontal resolution. I have already generated several font libraries containing all fonts for plain TeX, LaTeX (but no SliTeX) and some utility fonts used by Knuth (logo* and manfnt). These libraries contain the proper comment indicating the aspect ratio to dviwin; you can find them in the directory "systems/msdos/emtex/fx_med_fonts" at the CTAN archives (ftp.shsu.edu, ftp.tex.ac.uk or ftp.uni-stuttgart.de) under the names "fx_med_*.fli".

I have a 24-pin printer; I sometimes want to use the 360x180dpi mode, while other times I want to use the 360x360dpi mode. Is this possible with dviwin?

Dviwin can easily handle this case, provided that you have the correct fonts and the font specification is unambigous. If you use PK files, I would suggest that you add the directories "360x180", "360x360", etc. under your base directory, and you add the suffix ";XXXX\$xx$y" to the directory specification (where XXXX is your base directory). If you prefer to use FLI files, then you can simply put all of them in the font base directory, but make sure that you add the comment "0.5" (the aspect ratio) to *all* 360x180 libraries, and the comment "1.0" to *all* 360x360 libraries. If you don't do this, dviwin will not be able to distinguish between the two sets of fonts (because the FLI format does not specify the vertical resolution explicitly), and the output will definitely be messed up.

I have fonts for a 300dpi printer as well as for a 24-pin printer. The printing output is often incorrect. What is the problem?

This may indicate that dviwin is confused about the vertical resolution of your fonts. When using the 300dpi printer, magstep1 corresponds to 360dpi, which is equal to magstep0 for the high (360x360) or medium (360x180) resolutions of the 24-pin printer. If the 24-pin fonts are for the medium resolution, dviwin may be confusing these fonts to the 360x360 dpi fonts used by the 300dpi printer at magstep1. The best solution for this problem is to follow the instructions in the previous question in order to eliminate any ambiguity in the font selection.

I have installed the "share.exe" program and I am getting sharing violations. Why does this happen?

The basic idea behind share.exe is to arbitrate file access among different processes. Unfortunately, the implementation is not that great and sometimes it complains even if two or more processes try to read the same file (this is wrong: it should complain only if a program tries to read a file while another is writing it). You can placate it by setting the attributes of the files in question to "Read Only".

I have PK fonts at 118dpi and I don't want to waste space by installing new fonts at similar resolutions. Is there a way to use my fonts?

The first version of dviwin could use only the resolutions appearing in the "Resolution" submenu. The new version lets you add two custom resolutions to the submenu, so you can use any fonts that you like. You can select the entry "Custom Resolutions" from the "Options" submenu, and you will see a dialog box where you will specify the resolutions that you want. As soon as you do

this, you can select any of these resolutions from the "Resolution" submenu.

Why do you use these particular font resolutions? they do not follow the standard geometric progression exactly. What if I want to use the proper progression?

I decided to use these fonts in order to remain compatible with other drivers from the Beebe family, and because the appropriate fonts are already at simtel, so anybody can download them. It would be an enormous waste of space and bandwidth if I required everybody to obtain new fonts. The progression is sometimes off by 1dpi, but this does not really affect the viewing or printing quality. There is a workaround though: if you use the custom sizes, the program does not make any assumptions about your fonts, so it uses the standard geometric progression; suppose for example that you use the *built-in* resolution of 100dpi: if you need to load a font at magstep1, the program will look for the font at 121dpi (instead of 120dpi) for compatibility purposes. If on the other hand you use a *custom* size of 100dpi, the program will look for 120dpi when it needs a font at magstep1. This method should give you the most flexibility. By the way, it appears that the resolutions in the Beebe drivers are generated by applying negative magsteps to metafont using a base resolution of 300dpi; I would speculate that the main reason for this choice is to have a smooth transition between screen and printer fonts.

The startup time seems a bit longer than before. Why does this happen?

When you start the program, it looks in the base directory for font libraries. If it finds any, it makes a list of all fonts in all libraries. This information is used when searching for fonts for your documents; in essence, you spend about one extra second upon startup but you save more time, because there will be less disk accesses during the processing of your documents. If you don't use any font libraries, the startup time should be a bit shorter because it does not allocate as much memory as before.

Why do you open the font libraries more than once?

The program uses a separate file handle for each font even though they may be in the same font library. This lets it exploit the file buffering better and speeds up the font reading somewhat.

The printer output is extremely small. What is the problem?

The problem was that dviwin would send the bitmap to the printer at the *current* resolution. Therefore, you needed to switch to the appropriate resolution for the printer before issuing the "Print" command; this was done for flexibility purposes. Since however most people are annoyed by it, I decided to use another method: the "Print" dialog contains a combo-box that lets you select the resolution used for the printer. This value defaults to "Auto" which means that the program will use a resolution which is as close as possible to the printer's resolution. This will work even when you change the default printer; it is also independent from the previewing resolution. Therefore, you will not need to do anything under normal circumstances. If however you want to produce magnified or reduced output, you can still do it with minimal fuss.

When I try to print on a laser printer, the output is weird: the text comes out at the proper size, but it does not fill the page and there are some extraneous characters.

This is almost certainly due to insufficient memory in the laser printer. You will need about 1M of RAM at 300dpi and 4M of RAM at 600dpi. This behavior is typical of HP printers; Apple printers reset and print a test page when the memory limits are exceeded. Keep in mind that two-up printing requires more printer memory; you will usually be able to print normal pages on a laser printer equipped with only 512K; if however you try to print two reduced pages on such a printer, you will probably exceed its memory limits. The best solution is to add some memory to your printer; if this is not practical, you can try to select a lower printing resolution from the Printer Setup dialog; *if you do this, make sure that you set the resolution to "Auto" in the Print dialog*.

I selected a lower resolution in my printer driver, but dviwin produced a GP fault.

This was a bug in dviwin 2.5 and it has been fixed; it would often happen with the LaserJet and DeskJet drivers if you selected 150 or 75dpi resolution. Still those drivers are somewhat puzzling: even when you use the lower resolutions, they report the printer resolution as 300dpi and they require dviwin to send them the page at 300dpi; then they reduce the bitmap internally. This approach does work, but it has some disadvantages: dviwin will need to compose the page at 300dpi which requires much more memory than the lower resolutions; furthermore, the resolution reduction does not produce great results: you could get better output by producing the lower resolution fonts with Metafont. The only advantage that I can think of this method is that you only need a single set of fonts at 300dpi instead of three sets.

The program is too slow when printing. Are there any improvements in this version?

The lack of speed was due to two problems: the first one was that the first version of the program was using too much memory and Windows would swap excessively if you had 4M of RAM or less. The memory requirements of newer releases have been drastically reduced and it can handle even 400dpi resolutions without significant swapping on 4M machines (you will need at least 6M of RAM for 600dpi printing). The second problem is that the program sends a bitmap to the printer; this method can be slow on high resolution printers; I made some optimizations, but the bottleneck is the interface of the computer to the printer. I run some tests on a 386/25 machine with 4M of RAM; when pressing the PgDn key at 300dpi, dviwin 2.0 needed 15 seconds to display the next page; newer versions need 2.5 seconds; this difference is due only to the reduced memory requirements and the lack of swapping. When however you compare the printing speed between the two versions, the difference is much smaller, because of the computer-printer bottleneck. Note that PostScript printers are particularly hostile to bitmaps; if you use a LaserJet or compatible printer with a PostScript cartridge, you will get *much* faster output by using the printer in its native mode.

I understand the logic behind the printing method, but I still need to use a PostScript printer. Is there anything I can do to increase the speed?

The new version of dviwin lets you attach another dvi driver when printing; therefore, you will get the best results from your PostScript printer if you ask dviwin to call a dvi driver tailored for PS printers. You can do this easily by setting the "External Printer" entry in the "Options" submenu and checking the option "Enable external print" from the Print dialog. If the former is non-empty and the latter is checked, then dviwin will execute the command that you specify. The dviwin documentation describes the templates for several dvi drivers. You can also see (and edit) the printing command right before it is executed if you check the option "Preview external print" in the Print dialog. There is one thing that you should be keep in mind if you use the "share" program: when dviwin executes an external print command, it minimizes itself and closes the dvi file but not

the PK or FLI files; therefore, the other dvi driver can read the dvi file, but you will get sharing violations for the font files (because of a deficiency in share). You can rectify the situation easily by setting the attributes of all font files to "Read Only"; if however you try to restore the dviwin window before the external driver has finished, then you will get another sharing violation about the dvi file and dviwin will complain that it cannot access that file. There is no easy solution around this problem; you either have to wait until the external driver has finished, or avoid using share.

I am running another dvi driver through the External Print option, but it complains that it cannot find many fonts even though they are installed, and the path specification is correct.

The system is probably running out of file handles. You can rectify this easily by specifying a high enough value in the "FILES=..." statement in your config.sys file; I would suggest a value of at least 50, but you may need to increase it further if you run many programs concurrently and each program accesses many files..

Is there a way to print two TeX pages on a single physical page?

The easiest way to accomplish this task is to use the program "dvidvi" (or dvi2dvi) in conjunction with the driver. There are two common page orderings: the first one is to print pages [1,2] on the first physical page, [3,4] on the second, [5,6] on the third and so on. This can be used for program listings. The second useful page ordering is to shuffle the pages for making a booklet; if there are N pages in the document (and N is even), you will want pages [N,1] on the first physical page, [2,N-1] on the second, [N-2,3] on the third and so on. Then you can just put staples in the middle, fold the paper and the booklet will be ready. In both situations you will also want to print the odd physical pages in one pass and the even ones in a second pass; in this way, you can re-insert the paper from the first pass to the printer to get double sided printing (keep in mind that some printers are not really good at this, so you may need to do it with a copier). Suppose now that the input file is "xyz.dvi" and you want to use the first ordering (for program listings). You can issue the commands:

```
dvidvi  4:0,1(8.5in,0in)  xyz.dvi  xyz1.dvi
dvidvi  4:2,3(8.5in,0in)  xyz.dvi  xyz2.dvi
```

These commands will produce the files "xyz1.dvi" and "xyz2.dvi" which contain the reordered pages. If on the other hand you want to use the second ordering (for booklets), you can issue the commands:

```
dvidvi  4:-3,0(8.5in,0in)  xyz.dvi  xyz1.dvi
dvidvi  4:-1,2(8.5in,0in)  xyz.dvi  xyz2.dvi
```

You will again get the files "xyz1.dvi" and "xyz2.dvi" but the page ordering will be different. For A4 paper, use the dimension "210mm" instead of "8.5in" in all the above commands. Now start dviwin and select a Ledger page size (17in by 11in); for European paper, select the A3 landscape size (420mm by 297mm). There are two ways to proceed from here: if the printer can handle such a large paper size, just print it at the appropriate resolution and reduce the output with a photocopier; this will give you great results because the effective resolution will be higher. If however the printer cannot handle such paper, we can reduce the output with dviwin; from the "Print" dialog, select a resolution between 65% and 75% of the actual printer resolution. In the printer "Setup" dialog, select Landscape printing, and in the Alignment dialog select automatic

page positioning. If for example you use a laser printer capable of handling only Letter or A4 paper at 300dpi, you will probably use a resolution of 208dpi (you can go to 228dpi if you use a wide margin format such as LaTeX). This arrangement will let you print the file in the desired manner.

Up to this point, we assumed that the original dvi file was intended for normal printing; this has the advantage that you do not need to change anything in the TeX file. If however you are willing to modify the TeX file for this printing method, you can make the text a bit narrower and taller to fit as much of the page as possible; you can also experiment with the parameters of dvidvi. You can find dvidvi in the directory pub/os2/all/unix/tex of ftp-os2.nmsu.edu; dvi2dvi can be found in the directory pd1:<msdos.tex> of wsmr-simtel20.army.mil or pub/msdos/tex of oak.oakland.edu.

I used dviwin to print on my dot-matrix printer, but the output is ugly.

Previous versions of dviwin supported only devices with square pixels; many dot-matrix printers have an aspect ratio which is not equal to one, so the printing quality was marginal. The new version supports devices with any aspect ratio. Just select the "Custom Resolutions" entry from the "Options" submenu, and enter the appropriate horizontal and vertical resolutions. If you set the printing resolution to "Auto", dviwin will select the correct resolution and fonts when printing. By the way, you can apply the same method even for the screen in the unlikely case that your video mode has non-square pixels.

I am still a bit unclear about the graphics capabilities. Can you explain them better?

The graphics capabilities are not that complicated. We have two ways to import graphics: the first one is without any graphics filters, and the second one with graphics filters. The first case works only with "standard" metafiles. The second case works with "placeable" metafiles, or whatever graphics files are supported by your filter.

Consider the first case (without filters). You make a graph for example with Excel, copy the graph to the clipboard and then run the "clipmeta" program to save the metafile [make sure that you specify a standard (or plain) metafile format in the "Save As..." dialog]. Let's say that you saved the graph in the file "abc.wmf". Now put the commands

```
\special{anisoscale abc, x-size y-size}
\vskip y-size
```

in your TeX file. x-size and y-size are the desired dimensions of the graph (look at the demo.tex file). When you view or print the file, the graphic should be fine.

Consider now the second case (with the filters). We will do the same example as above, but with two different graphics files. Let's say that you want to import a "placeable" metafile and a TIFF file. By the way, even though TIFF files are standard in desktop publishing, they are not particularly good, because they cannot be scaled very well (the same applies to any bitmap format). Metafiles on the other hand are great because they are small, can contain most drawing or graphics commands and can be scaled very well (with the proper methods). Anyway, let's talk about the technique for using the two graphics files. The first and most important requirement is that you have the appropriate filters. If for example you have Word for Windows, you will need the files "wmfimp.flt" and "tiffimp.flt". You will also need an entry in your win.ini file as:

```
[MS Graphic Import Filters]
```

```
Windows Metafile(.WMF)=c:\binw\filters\wmfimp.flt,WMF
Tagged Image Format(.TIF)=c:\binw\filters\tiffimp.flt,TIF
```

These lines instruct dviwin to use the filter "wmfimp.flt" for files with a "WMF" extension, and the filter "tiffimp.flt" for files with a "TIF" extension. I put them in the directory "c:\binw\filters" but you can put them anywhere you like. Word for Windows adds this entry in "win.ini" upon installation, so you may already have it.

Once this step is done, just put the \special{} and \vskip{} commands in your TeX file, and everything should work out fine. The beauty of this setup is that it is extensible: if tomorrow we decide to use a new graphics format, we only need a new filter, and all applications will benefit. If you have Word for Windows, Toolbook, Powerpoint or Pagemaker, the filters are already in your hard disk.

The capability for various graphics files is pretty nice: most of the time however you can use plain metafiles with no loss in functionality (provided that you use a Windows program to generate the graphs). Therefore, you do not really need the filters for everyday work. I just thought that this capability would be nice.

I tried to import an HPGL or a PIC file using the filter supplied with Word for Windows 2.0, but nothing is displayed. What is the problem?

This is a bug in the graphics filter: it is supposed to set the window limits when importing the graphics file, but it doesn't do it, and Windows draws the entire graphics file on a single pixel. Word for Windows knows about this problem and fixes it silently. I did the same thing in the new release, and these files now display properly. Keep in mind that some of these filters are not that great: the PCX filter for example cannot import a file containing more than 256 colors, and even then, the current video mode must match the video mode used when creating the file.

I tried to import a PostScript file but I can only see an empty rectangle.

The problem is that the EPS filter does not really import a graphics file. It just creates an empty rectangle to indicate the location of the graphic; the actual translation of the PostScript file is delegated to the translator inside the PostScript printer. Therefore, there is currently no easy way of importing such a file.

So that particular filter is not useful; is there any way to include PostScript graphics?

The most logical way to accomplish this would be to have a filter that interprets a PS file and creates a metafile. This would benefit not only dviwin, but any program that imports graphics. Therefore, we would need the entire functionality of a PS interpreter (eg., Ghostscript) in a filter; unfortunately, PS is much more complicated than a normal graphics format (Ghostscript is still incomplete even after 5 years of development by many people), so I am rather pessimistic about the availability of such a filter (especially at a low price).

When I insert a metafile in my document, the text inside the graphics file looks ugly. Is there a way around this problem?

This usually happens if you use a non-scalable font in the metafile. Try using the "Times New

Roman" font instead. Microsoft also sells a larger collection of TrueType fonts; if you have it, you can get pretty good results by using the "Century Schoolbook" font which blends better with the cmr fonts.

I am trying to include a color graphics file in my document and I have a color printer, but dviwin converts the graphic to black and white. Why?

The program uses a monochrome bitmap for the rasterization of the document; therefore, Windows converts any colors to black and white automatically. The reason for this behavior is that a monochrome bitmap takes much less memory and can be displayed faster than a color bitmap. I plan to support color graphics (but only for the \special{} commands) in a future release of the program.

Why don't you distribute the graphics filters with dviwin?

The graphics filters are commercial programs; I would have to license them from their authors to distribute them with dviwin, and that is clearly infeasible for a free program.

I want to include some bitmapped graphics in my document; dviwin does import them, but they look very ugly. Is there a way to improve their appearance?

There are two basic problems with bitmapped graphics: scaling and color assignment.

1. Bitmapped graphics cannot be scaled easily; if you instruct dviwin to scale such graphics, dviwin passes this request to Windows which does not do a good job in scaling the graphic. The best way to print a bitmap is to print it at its natural size (ie., each pixel in the bitmap should correspond to a pixel in the printing device). This can be achieved by computing the graphic dimensions at the intended printing resolution and supplying these values to dviwin in the \special command; suppose for example that you want to print the file "abc.wmf" which is a 640x480 bitmap on a 300dpi printer; the printed dimensions will be 640/300 = 2.1333in by 480/300=1.6in. You can then give the command:

   \special{center abc, \the\hsize 2.5in 2.1333in 1.6in}

   which will print the bitmap without any scaling. This implies of course that the dimensions specified in the \special command are completely device-dependent: if you want to print the same document on a dot-matrix printer with different resolution, you will have to compute the required dimensions again. It is not a particularly appealing method, but you will get the highest possible quality from your output device. A natural consequence of the above statement is that the screen output of dviwin may not look good if the selected resolution is not equal to the printing resolution; it will look just fine if you select the printing resolution in conjunction with a zoom factor.

   The new version of dviwin lets you specify the graphic's dimensions in pixels. Using the bitmap from the above example, we can specify the special command as:

   \special{center abc, \the\hsize 2.5in 640px 480px}

   and dviwin will take care of all conversions. You only need to leave enough vertical space in your document.

   Note that we specified the bitmap's dimensions in the \special command; this is *absolutely necessary even when using a graphics filter*, because clipmeta does not know the resolution of the device on which you want to print the bitmap, so it cannot provide any reasonable dimensions.

2. The second issue with bitmapped data is how to represent colors on a monochrome printer. Windows does this conversion for you, but the results are usually horrendous because it uses a simple algorithm to assign some colors to black and other colors to white. You can achieve *much* better results if you use a dedicated painting program to reduce the colors *before* you produce the metafile. For this operation, I would recommend Paint Shop Pro which is a very nice shareware program and you can find it in the directory pub/pc/win3/desktop at ftp.cica.indiana.edu under the name pspro102.zip; it can reduce the colors using several dithering algorithms and you will get great results on most printing devices. Some printers have a similar mechanism for reducing the colors, but their method is unually inflexible, and you will waste lots of paper until you get the correct dithering; the above described method will work with all printers and you can easily see the exact output before you print.

If you follow the above steps, you will usually find that the printed bitmap is too small (at least if the printer has a sufficiently high resolution). The solution to this problem is to scale the bitmap with a painting program *before* you produce the metafile; any decent program should be able to handle scaling by an integer factor without any geometric distortions. If you are also doing a color reduction, make sure that the scaling is done *before* the dithering; in this way, the dithering algorithm has more pixels over which to distribute the error, and you will get excellent results. If on the other hand you do the scaling after the dithering, you will end up with large, ugly pixels. I understand that all these steps sound a bit cumbersome (although you can do them quicker than I can describe them), but it is not trivial to handle bitmaps properly; I think that the proposed method yields better results than most other methods.

I am getting GP faults while importing a large graphics file. What is the problem?

Many video drivers do not handle large bitmaps well and produce GP faults. These faults are not due to dviwin; they will occur with any program that asks the driver to display the bitmap. The new version of clipmeta takes care of this problem by splitting large bitmaps into smaller pieces; this is done automatically, so you don't need to worry about it; the only thing that you may want to consider is that some programs export bitmaps as well as metafiles which simply contain the bitmap (PaintBrush is such an example). Clipmeta will then export the metafile, and it will not be able to split it into pieces, because it does not examine the metafile contents; therefore, you will end up with a big bitmap that may cause GP faults. If on the other hand there is only a bitmap in the clipboard, clipmeta will split it into pieces before creating the metafile. Therefore, if you are trying to export bitmapped data, make sure that you get the actual bitmap into the clipboard instead of a metafile; clipmeta will take care of the rest. If you use PaintBrush, make sure that the entry "Omit Picture Format" in the "Options" submenu is checked. Please read the clipmeta documentation for more info about that program.

I am trying to include a graphic in my document, but I only get a solid black rectangle.

Some programs cannot export a Device Independent Bitmap (DIB); they only export a Device Dependent Bitmap (DDB); this will work on a 16-color mode (because the palette is fixed) or on HiColor and TrueColor modes (because there is no palette). On a 256-color mode however, the results are disastrous: the bitmap colors depend on the palette that was current at the time that the bitmap was displayed, but it is most likely that these particular colors will not be available when you use the bitmap in your document. The result is that the colors will be completely wrong and you often end up with a solid black picture. That's one of the reasons for my recommendations against using DDBs. The only way out of this problem currently is to use an application that sends a DIB to the clipboard.

How does the refresh capability work?

When you switch to another program, dviwin closes the dvi file but keeps the time and date information of that file. When you switch back to dviwin, it checks the time and date information of the dvi file, and if it is different, then it reloads the file but keeps you at the same position as before. It also tries to re-use as many of the old fonts as possible, so the reloading should be much faster than the initial loading. This approach is extremely useful if you wish to integrate the editing and previewing; suppose that you edit a file, run TeX and start the previewer. If you see something wrong, you only need to switch to the editor and rerun TeX; as soon as you switch to the previewer, it will load the new version of the file automatically (and quickly). You can also automate the process even further if you use a good editor and a batch language for Windows.

Another advantage of this approach is that it does not depend on DDE or other Windows intricacies, so it will work with any version of TeX.

What are the advantages of the font cache? how large should it be?

Windows is quite slow in accessing the disk (much slower than DOS); this is a big disadvantage for dviwin which needs to access the dvi file as well as the fonts. Therefore, I decided to keep old fonts just in case we may need them again in another document. When you close a document (either explicitly or by loading another one), the old fonts are not discarded; they are kept in the font cache. If the new document needs any of the fonts in the cache, it will take them from there, so the loading will be much faster. You obviously need to set a limit in the cache size (otherwise you will run out of memory), but this limit depends on the type of your documents as well as the available memory. A cache size of 10 should be a good initial value; then you can observe the number of fonts in the cache from the "Cache Size" entry in the "Options" submenu; a relatively good heuristic is to adjust the cache size so that it is usually almost full. You will need of course to take the amount of available RAM into consideration. The program clears the cache completely when printing; at that point, you probably need all the RAM that you can get.

I start dviwin and load a document. The cache size is 10 but I see no fonts in the cache. How is this possible?

Only *currently unused* fonts are in the cache. If you start dviwin and load a file, it will read all the fonts required by that file, but all of the fonts are "in use". Therefore, the cache will be empty. If you load a second file (or a revised version of the same file) which does not use all the fonts from the first file, then you will see some fonts inside the font cache.

The display looks fuzzy when I zoom. Why does this happen? is there a way to improve the readability of fonts?

When you use high resolution fonts in conjunction with zooming, the program uses a method called "anti-aliasing" to produce the reduced characters. These characters are rather fuzzy because of the color interpolation. You can improve the readability of the text by adjusting a color interpolation parameter from the "Options" submenu.

What is the color interpolation parameter?

When you use a zoom factor greater than one, dviwin needs to use several colors between the foreground and background colors in a visual sense. The naive approach is to use a linear interpolation between the two colors. To use for example black letters on a white background, one may try to use a 25% gray, a 50% gray and a 75% gray. This works, but it ignores two fundamental nonlinearities: the human eye has a logarithmic response to changes in intensity, and the actual brightness of a pixel (as measured by a photometer) is not necessarily a linear function of the voltage applied to the CRT. One usually deals with these phenomena by applying "gamma correction" which is simply a power function. The color interpolation parameter is a monotonic transformation of the gamma correction parameter generalized for any combination of colors. We use a monotonic transformation simply for convenience in the selection of values. The transformation is set so that a zero value implies a linear function (ie., gamma=1); as the transformation parameter goes to +100, gamma moves towards positive infinity; as the transformation parameter approaches -100, gamma goes to zero.

Why is the close-up view so coarse?

When the zoom factor is greater than one, the magnifying glass displays the "unzoomed" text which should look just fine. When however the zoom factor is equal to one, the close-up view is obtained by taking the actual bitmap and doubling the bits in both axes; therefore, it is not surprising that it is ugly. The alternative is to open new fonts at higher resolutions and typeset the page again, but this will take enough time to be unusable.

You mention that there is a limit of 17 open font files. Why? Are there any other limits?

The C compiler imposes a limit of 20 open files for buffered I/O; apart from the fonts, dviwin opens three files: the dvi file, the log file and the font substitution file. Therefore, there are 17 files available. We can use a higher limit with unbuffered I/O, but the net speed effect is negative. There are a few other limits too:

1. The maximum number of Windows fonts inside a *single* metafile is 20.
2. The maximum custom resolution is 3000dpi (this is just a sanity check).
3. The unpacked raster for a single character cannot exceed 64K bytes.
4. The maximum number of pages in the dvi file is 16384.
5. The maximum size of the TeX stack is 2048.
6. The maximum size of the parameters of a \special{} command is 32K bytes.

The last four limits come from the 16-bit architecture of Windows 3.x (ie., there are no preallocated arrays). I think that all these limits are sufficiently high as not to be binding.

Is there a way to search for text inside dviwin?

There is currently no way to do this: I am planning to add this capability in the future, but it will accomodate only simple strings; you will not be able to search for "{\bf test}" or "n^2" because this would require a full TeX parser which is beyond the scope of a dvi driver.

Can I use virtual fonts with dviwin?

The current release cannot use these fonts. I plan to support them in the future.

Is there a way to use TrueType or PostScript fonts in my TeX documents?

I have some ideas about using any Windows font (TrueType, ATM or even the bitmapped fonts) without using any PK files and I plan to work on it for the next version. The main problem is how to produce a TFM file; I would appreciate if any reader can show me how to do this. Once the TFM files are in place, you will only need a few macros to set up your document. Keep in mind however, that these fonts will not be as common as the cmr family, so you will lose some portability. The basic premise of such fonts is that they save you some disk space by eliminating the need for many PK fonts, but you will still need the standard PK fonts if you want to read or print documents written by other people.

Why do you support only Windows 3.1 instead of 3.0 as well?

The first release (2.0) had a problem when running under Windows 3.0. I am using a Microsoft-supplied library (ctl3d.dll) which was supposed to work under Windows 3.0. The problem is that

this library did not actually work under the older version; my fault was that I did not test it myself. Eventually, I received a new version of the library that works properly under Windows 3.0, so this problem has been solved. However, I found a more serious problem with Windows 3.0: when you import a graphics file through the \special{} command, the driver allocates a bitmap and asks Windows to draw the graphics file into the bitmap. The problem is that the bitmap will often be larger than 64K bytes and Windows does not display the graphics file properly (it does not report any errors either). There are some ways around this bug, but they take time, and I think that it is unfair to penalize all users for this bug in Windows 3.0; given the amount of bugs that exist in Windows 3.0, the best policy is to upgrade.