Overview

GSview for Windows is a graphical interface for MS-Windows **Ghostscript**. Ghostscript is an interpreter for the PostScript page description language used by laser printers. For documents following the Adobe PostScript Document Structuring Conventions, GSview allows selected pages to be viewed or printed. GSview should be used with Ghostscript 3.53 or later. GSview was inspired by Tim Theisen's X11 Ghostview program.

Installation

It is recommended that you use the installation program for installing GSview.   For OS/2, use os2setup.exe.   For MS-Windows,   use winsetup.exe.   See the file README.GV for more details.

If you wish to install GSview manually, see the Manual Installation topic.

Uninstalling GSview and Ghostscript
Manual Installation
Ghostscript Installation
Include Path
Fonts

Uninstalling GSview and Ghostscript

To uninstall GSview, remove all the files in the **gsview** directory. To uninstall Ghostscript, remove all the files in the **gs3.53** directory.

Also remove
```
c:\windows\gsview.ini
c:\windows\gsview32.ini
c:\windows\gswin.ini
c:\windows\gswin32.ini
```
from the appropriate system directory.

Remove the **GS Tools** group from the Program Manager.

If you know how to edit the registry, remove the following keys:
```
psfile
.eps
.ps
```

Manual Installation

It is recommended that you use the installation program for installing GSview. The following instructions describe how to install GSview without using the installation program.

First you need to install Ghostscript. It is recommended that Ghostscript 3.53 or later be used. (If you wish to use 16-bit Ghostscript 2.6.1, see the __Ghostscript Version__ topic. If using the 32-bit versions of GSview or Ghostscript, you must use Ghostscript 3.12 or later.)

Make a directory d:\gsview.

Next install GSview for Windows by copying gsview.exe, gsview32.exe, gsview.hlp and gsv16spl.exe to the **d:\gsview** directory.

Start GSview then select __Options__ | **Configure Ghostscript**.

Enter the correct executable path for Ghostscript. For example
    `c:\gs3.53\gswin.exe`

Enter the correct Ghostscript include path into the **Ghostscript Include Path** field. This include path must include the directories where the Ghostscript library files (gs_*.ps and Fontmap) and the Ghostscript fonts (*.gsf) are located. For example:
    `c:\gs3.53;c:\gs3.53\fonts`
Do NOT put a **-I** before the include path.

Leave the **Other Options** field empty.

If you do not get the **Ghostscript EXE** field correct, GSview will not be able to load Ghostscript. If you do not get the **Ghostscript Include Path** correct, Ghostscript will immediately exit with an error code.

Under Windows 95 or NT, the 32 bit version of GSview (gsview32.exe) requires the 32 bit version of Ghostscript (gswin32.exe), and the 16 bit version of GSview (gsview.exe) requires the 16 bit version of Ghostscript (gswin.exe). Under Windows 95 or NT you need a version of Ghostscript later than 3.12. Under Windows 3.1 with Win32s, the 16 and 32 bit versions can be intermixed.

Set the environment variable TEMP to point to a directory for temporary files. For example:
    `SET TEMP=c:\temp`
The directory must exist and must be writeable.

If you have some Type 1 fonts on your system, it may be possible to tell Ghostscript to use them. See **Fontmap.os2** and **Fontmap.atm** supplied with Ghostscript for examples. See the __Fonts__ topic.

If you have problems, try reading the help topic __Common Problems__.

Ghostscript Installation

The following describes how to install Ghostscript, not GSview. The primary documentation on installing Ghostscript is found in the Ghostscript file use.doc.

Aladdin Ghostscript for the PC is available on the Internet from
```
ftp://ftp.cs.wisc.edu/ghost/aladdin/gsNNNxxx.zip
```
Where NNN is the version number.   For Ghostscript 3.53, the files required to run Ghostscript are:
```
gs353ini.zip
```
plus one or more of
```
gs353dos.zip   (MS-DOS 386 EXE)
gs353os2.zip   (OS/2 2.0 or later EXE)
gs353win.zip   (MS-Windows 3.1 16bit EXE)
gs353w32.zip   (MS-Windows 3.1/Win32s or MS-Windows 95 or NT EXE)
```
If you do not already have a set of Type 1 fonts, you will also need
```
gs353fn1.zip
```

The following assumes that Ghostscript is to be installed on drive d:. Unzip the INI zip file and the required EXE zip files from the root directory.   This should put the files into the directory d:\gs3.53.

If you also need the fonts, change to the d:\gs3.53 directory and then unzip gs353fn1.zip.   This will put font files into .\fonts. To run Ghostscript, set the GS_LIB environment variable as follows
```
 SET GS_LIB=d:\gs3.53;d:\gs3.53\fonts
```
then start the appropriate EXE.   To run Ghostscript without using the environment variable, use the following command
```
 d:\gs3.53\gs -Id:\gs3.53;d:\gs3.53\fonts
```
where the EXE name needs to be replaced with the appropriate name (gs, gs386, gswin, gswin32 or gsos2).

If you already have some Type 1 fonts, there are two ways to tell Ghostscript about these fonts: editing/replacing the Fontmap file, or the GS_FONTPATH environment variable.   See the Ghostscript use.doc and the sample Fontmap files for details.

See the Include Path and Fonts topics for more details.

Include Path

Ghostscript needs to read some initialisation files during startup, and needs to read font files before drawing text. When used on a PC, it is normal to tell Ghostscript where to find these files using the Ghostscript **-I** command line switch. Alternatively, the **GS_LIB** environment variable can be used. See below for details for Unix or VMS.

The **-I** switch or **GS_LIB** environment variable contains a list of   directories to search for the required initialisation files. On a PC, it is normal to put the initialisation files in the same directory as the Ghostscript program.   If Ghostscript is in c:\gs, then **-Ic:\gs** would be used as a command line option to Ghostscript.

Ghostscript also needs to find font files.   The list of fonts known to Ghostscript is contained in the Fontmap file.   The directories which contain these fonts should be added to the **-I** switch. If the Ghostscript *.gsf fonts are in c:\gs\fonts, and some other *.pfb fonts are in c:\psfonts, then **-Ic:\gs;c:\gs\fonts;c:\psfonts** would be used as a command line option to Ghostscript on a PC.

Under Unix or VMS with Ghostview, the include path should have been configured when Ghostscript was compiled. If not,   and Ghostview is being used, use the **GS_LIB** environment variable instead of **-I**. The Ghostscript include files are generally not put in the same directory as the executable under Unix. Instead the executable might be in
```
 /usr/local/bin
```
while the include files are in
```
 /usr/local/lib/ghostscript/gs3.53
```
and the fonts in
```
 /usr/local/lib/ghostscript/fonts
```
Consequently, the **GS_LIB** environment variable would need to be set to
```
 /usr/local/lib/ghostscript/gs3.53:/usr/local/lib/ghostscript/fonts
```
It is to be hoped that these were set as the default when Ghostscript was compiled, and that it is unnecessary to use **GS_LIB**.

For more information, see the following topic.

File searching

File searching

(from Ghostscript use.doc)

When looking for the initialisation files (gs_*.ps), the files related to fonts (Fontmap, *.pfa, *.pfb, *.gsf), or the file for the 'run' operator, Ghostscript first checks whether the file name specifies an explicit directory or drive (i.e., doesn't begin with '/' on Unix systems; doesn't contain a ':' or begin with a '/' or '\' on MS-DOS systems; doesn't contain a ':' or a square bracket on VMS systems).   If it does, Ghostscript simply tries to open the file using the given name.   Otherwise, Ghostscript will try directories in the following order:

1. The directory/ies specified by the -I switch(es) in the command line (see below), if any;

2. The directory/ies specified by the GS_LIB environment variable, if any;

3. The directory/ies specified by the GS_LIB_DEFAULT macro in the Ghostscript makefile, if any.

Each of these (GS_LIB_DEFAULT, GS_LIB, and -I parameter) may be either a single directory, or a list of directories separated by a character appropriate for the operating system (':' on Unix systems, ';' on VMS systems, ';' on MS-DOS systems). Ghostscript 3.12 use.doc states that:

    Note that Ghostscript does *not* attempt to open the file
    in the current directory.   This is consistent with common
    practice on Unix, but it is different from the usual
    practice on MS-DOS.

This is incorrect.   Ghostscript does search in the current directory.

Fonts

Ghostscript locates fonts in two ways:

1. Those listed explicitly in <u>Fontmap</u>

2. Font files found in directories listed in the <u>GS_FONTPATH</u> environment variable.

Those listed in the <u>Fontmap</u> file should be locatable on the Ghostscript include path, or should have fully qualified paths.   Note that you must use / or \\ and must not use \ within a directory name.

Ghostscript comes with a number of free fonts, most with a .gsf extension.   These are ordinary Type 1 fonts (although they cannot be used with Adobe Type Manager).   The Ghostscript fonts can be broken into three main classes:

1. Type 1 outline fonts with hinting (most *.pfa).   These produce good quality output.   The standard Fontmap file uses these in preference to the other font types.   There are relatively few of these fonts.

2. Type 1 outline fonts that have been created from bitmap fonts (p*.gsf, z*.gsf, etc.).   These produce poor quality output. Avoid them if at all possible. A full set of the common 35 PostScript fonts is available.

3. Type 1 stroked fonts created from the public domain Hershey fonts (h*.gsf, h*.pfa).   These produce moderate quality output, but do not correspond to standard PostScript fonts.

If you have some other Type 1 outline fonts (*.pfa, *.pfb), it is possible to use these with Ghostscript. Whether or not you are allowed to do this depends on the font licence.   If you do use these fonts with Ghostscript, it is **your** responsibility to make sure that the font licence permits this use. To use the other Type 1 fonts, you will need to replace or edit the <u>Fontmap</u> file.   The reason for using these other Type 1 fonts is that they are almost certainly of better quality than the Type 1 bitmap derived fonts supplied with Ghostscript.   However, good quality fonts are not usually free.

<u>Fontmap</u>
<u>GS_FONTPATH</u>
<u>Platform Fonts</u>

Fontmap

The **Fontmap** file tells Ghostscript what fonts are available and where to find them. Each line in **Fontmap** may be one of the following:

1. A comment.   These are lines that start with a %
```
   % fontmap aka Fontmap - standard font catalog for Ghostscript.
```

2. A font name and file name.   For example
```
   /Courier                (ncrr.pfa)      ;
   /Courier-Italic         (ncrri.pfa)     ;
```
The first of these lines says that the font name **Courier** is to be found in the file **ncrr.pfa**. The file **ncrr.pfa** must contain a font named Courier. If the name is not Courier then a font alias must be used. This is described later. Note that the file name is a PostScript string and so \ has a   special meaning.   If **ncrr.pfa** was in the c:\gs\fonts directory, you would need to   have **c:\gs\fonts** as one of the directories listed in the include path,   or you would need to use the file name (c:/gs/fonts/ncrr.pfa) or (c:\\gs\\fonts\\ ncrr.pfa).   Using / is preferred to \\.

3. A font alias. For example
```
   /Courier-Oblique             /Courier-Italic          ;
```
This tells Ghostscript that if Courier-Oblique is requested, Courier-Italic is to be used instead. The standard Fontmap file uses aliases to replace poor quality bitmap-derived fonts with good quality hinted outline fonts. Ghostscript has a bitmap derived version of the Helvetica font in the file phvr.gsf.   The Fontmap file could instruct Ghostscript to use this with the following line:
```
   /Helvetica                   (phvr.gsf)      ;
```
However, the NimbusSansL-Regular font looks like Helvetica and is a better quality font, so the default Fontmap file actually uses
```
   /Helvetica                   /NimbusSansL-Regular    ;
   /NimbusSansL-Regular         (n019003l.gsf)  ;
```

A description of the required formatting for each line is near the top of each **Fontmap** file.

Ghostscript comes with a number of alternate **Fontmap** files. These include
```
 Fontmap        The standard fontmap file
 Fontmap.ATB    For Adobe Type Basics (65 Type 1 fonts)
 Fontmap.ATM    For Adobe Type Manager (13 fonts)
 Fontmap.OS2    For Type 1 fonts shipped with OS/2 (13 fonts)
 Fontmap.OSF    For DEC OSF/1 systems
 Fontmap.Ult    For DEC Ultrix systems
 Fontmap.VMS    For VAX/VMS systems with DECwindows/Motif
```
If you want to use one of the alternate Fontmap files, the procedure is as follows:

1. Copy Fontmap to Fontmap.old

2. Copy Fontmap.??? (your selected Fontmap file) to Fontmap

3. Add the directory that contains the Type 1 fonts to the include path (**-I** or **GS_LIB**)

GS_FONTPATH

(From Ghostscript use.doc)

When Ghostscript starts up, it also looks at the **GS_FONTPATH** environment variable, which is also a list of directories.   It goes to those directories and looks for all files that appear to contain PostScript fonts; it then effectively adds all those files and fonts to its internal copy of the Fontmap (the catalog of fonts and the files that contain them).   If you are using one of the following types of computer, you may wish to set **GS_FONTPATH** to the indicated value so that Ghostscript will automatically acquire all the installed Type 1 fonts:

```
System type             GS_FONTPATH
-----------             -----------
AIX                     /usr/lpp/DPS/fonts/outlines
NeXT                    /NextLibrary/Fonts/outline
OSF/1                   /usr/lib/X11/fonts/Type1Adobe
Silicon Graphics        /usr/lib/DPS/outline/base
Sun (Solaris 2.3)       /usr/openwin/lib/X11/fonts/Type1/outline
Ultrix                  /usr/lib/DPS/outline/decwin
```

See also Fontmap.

Platform Fonts

Platform fonts are described in Ghostscript fonts.doc.

The bitmap derived fonts used by Ghostscript are of poor quality.   The computer hosting Ghostscript may have the same fonts in better   quality versions, either as scalable fonts (e.g. Type 1 or TrueType) or as bitmaps. To improve the display of documents, Ghostscript can use these **platform fonts** instead of using the low quality fonts.

This can be illustrated with an example.   The default /Helvetica-Bold font is phvb.gsf, an outline font derived from a bitmap.   When the /Helvetica-Bold font is requested, phvb.gsf is read.   When a character is to be rendered to the display, Ghostscript instead asks MS-Windows for the Helvetica-Bold font at the appropriate size.   MS-Windows then draws the requested character from the TrueType Arial Bold font, and Ghostscript puts it on the display.   The resulting output is of better quality than the /Helvetica-Bold bitmap derived font.

In another example, the same request for /Helvetica-Bold under   Unix/X11 might instead display characters using a prebuilt bitmap font if one is available in the requested size.

There are some limitations to using **Platform fonts**.

1. Platform fonts are only used for upright (Portrait) characters.   Rotated characters will be displayed using the original PostScript font.

2. A PostScript version of the required font must be available. This is needed to obtain character bounding box information, encoding vector for character sets, and drawing rotated characters.

3. Platform fonts may only used for a limited range of sizes. For example, MS-Windows only uses platform fonts for 6 to 36 point fonts.

4. MS-Windows lies about the available font sizes.   Ghostscript asks for a particular font size and MS-Windows returns a font that it claims is the same size.   However MS-Windows may instead return a font of a different size that it thinks will look better.   Often it looks worse because the intercharacter spacing is out of proportion to the character size.   If this happens, platform fonts can be disabled by adding -dNOPLATFONTS to the Ghostscript Command Line.

5. Platform fonts will only be used for the display.   Output to printer devices will continue to use the PostScript font.

 MS-Windows Ghostscript has a fixed alias table for fonts. In the table below, the name on the left is the name of the   PostScript font, and the name on the right is the name that Ghostscript will try if MS-Windows doesn't know the PostScript   name.

```
Courier         Courier New
Helvetica       Arial
Helvetica       Helv
Times           Times New Roman
Times           Tms Rmn
```

Platform fonts are not supported under OS/2.

Not yet written: Unix/X11 Ghostscript may have an ability to use Xresources to specify font aliases.   If this is the case, then the method should be described here.

Document Structuring Conventions

Adobe has defined a set of extended comment conventions that provide additional information about the page structure and resource requirements of a PostScript file. If a file contains these Document Structuring Convention (DSC) comments, GSview can display pages in random order using Goto Page and display pages in reverse order using Previous Page.   Selected pages can be extracted to another file or printed.

If a file does not contain DSC comments, GSview can only display the pages in the original order.

DSC conforming files start with the comment line:
```
 %!PS-Adobe-3.0
```
where the number 3.0 may change and is the DSC version number. Some programs write PostScript files with a control-D as the first character of the file, followed by the comment line mentioned above. GSview will correctly report that these files are not DSC conforming, but will still display them with page selection features available. Complain to the author of the program that produced the PostScript file. To make the file DSC conforming, edit it to remove the control-D character.

DSC conforming files contain lines such as:
```
 %%Pages: 24
 %%Page: 1 1
```
These lines tell GSview how many pages a document contains and where they start.   GSview uses this information to select individual pages.

Encapsulated PostScript Files (EPSF) are single page documents that contain a subset of the **DSC** comments and PostScript commands. EPS files start with the comment line:
```
 %!PS-Adobe-3.0 EPSF-3.0
```
EPS files are commonly used for inclusion in other documents and for this reason require the bounding box comment:
```
   %%BoundingBox: llx lly urx ury
```
where llx, lly, urx and ury are integers giving the x and y coordinates of the lower left and upper right corners of a bounding box which encloses all marks made on the page.

Some EPS files contain a preview of the PostScript document. This preview can be a Windows Metafile, a TIFF file, or an Interchange preview (EPSI format). For the Windows Metafile or TIFF file preview, the EPS file under DOS contains a binary header which specifies the location and lengths of the preview and PostScript language sections of the EPS file. For the Interchange format, the preview is contained in DSC comments starting with
```
 %%BeginPreview: width height depth lines
```
An EPS file with a preview can be created from an EPS file without a preview using Add EPS Preview.

Opening a Document

The **Open** command on the **File** menu opens a file and displays the first page.

If the file contains DSC comments, pages can be selected using Next Page, Previous Page and Goto Page.

If the file does not contain DSC comments, Previous Page and Goto Page will not work.   Another file should not be selected until a last page of the file has been displayed.

When a file is open, GSview will display the document filename, the current page (if available) and while the cursor is over the image, the location of the cursor in coordinates specified by Options | Units.   The coordinate can be PostScript points (1/72"), millimetres or inches. The cursor location is useful for calculating bounding boxes.

The **Select File** command is similar to **Open** but it does not display the document. This command is useful for opening a document prior to printing it.

The **Save As** command saves a copy of the current document. This is useful if GSview is being used as a PostScript viewer by another application and you wish to save the currently displayed file.

Extract allows a range of pages to be copied from the current document to a new document.   For example, ten pages can be extracted from the middle of the current document and written to another file, which will later be sent to a printer.

The **Close** command closes the currently open document. This should be used before the current file is changed by another program. If you do not do this and GSview detects that the file length or date have changed, it will close Ghostscript and rescan the document.

See also Print.

PS to EPS

PS to EPS

In general, it is not possible to convert a PostScript file to EPS. However, many single page PostScript files can be converted to EPS by changing the first line of the file to
` %!PS-Adobe-3.0 EPSF-3.0`
and then adding or fixing up the **%%BoundingBox** comment.

When used incorrectly, the PS to EPS command can produce PostScript files with incorrect DSC comments.   Such a document will cause problems when you try to include it inside another document.

To convert a PostScript file to EPS, the original file **must** be a **single page** document.   If the document contains DSC comments and is multi page, extract the desired page with File | Extract. If the document does not contain DSC comments, you will need to edit the file by hand to extract the desired page.

EPS documents **must not** use any of the following operators:
```
banddevice    clear          cleardictstack  copypage
erasepage     exitserver     framedevice     grestoreall
initclip      initgraphics   initmatrix      quit
renderbands   setglobal      setpagedevice   setpageparams
setshared     startjob       letter          note
legal         a3             a4              a5
```

The following operators should be used with care:
```
nulldevice    setgstate      sethalftone     setmatrix
setscreen     settransfer    setcolortransfer
```

It is **your** responsibility to make sure that the above requirements are met.

To test if a document contains any of the above operators, select Options | EPS Warn and then Open the desired document. After the page has been displayed, look in the Ghostscript text window. If any of the above operators have been used you should see lines like:
` Warning: EPS files must not use ...`
If you find these warnings then do **not** use **PS to EPS**. Remember to turn off EPS Warn afterwards.

A document must be displayed before **PS to EPS** is used. The document must contain a **showpage**. This is required so that the bounding box can be measured.

For documents without DSC comments, **PS to EPS** allows a bounding box to be specified, then writes out an EPS file consisting of an EPS wrapper around the original document.

For documents with DSC comments, **PS to EPS** will change the first line of the file to
` %!PS-Adobe-3.0 EPSF-3.0`
then allows the **%%BoundingBox** comment to be changed or added.

For EPS documents, **PS to EPS** allows the **%%BoundingBox** comment to be changed.

PS to EPS does not clip the document to the **%%BoundingBox**. To do so would require changing the PostScript code itself. **PS to EPS** only changes the DSC comments.

PS to EPS does not add a preview to a document. If you want a preview you add it with Edit | Add EPS Preview after first creating an EPS file with a correct **%%BoundingBox**.

See also Add EPS Preview, Extract, EPS Warn.

Page Selection

View | **Next Page** or the **+** button moves to the next page of a   document. This works even if the document does not contain <u>DSC</u> comments.

View | **Previous Page** or the **-** button moves to the previous page.

View | **Redisplay** redisplays the current page.

View | **Goto Page** or the **pointing hand** button shows a dialog box   which allows selection of the next page number to display.   The **Select Page** dialog box shows page labels since these are likely   to be more useful than a sequential page number.

The **Previous Page**, **Redisplay** and **Goto Page** commands work only if the document contains <u>DSC</u> comments.

Zoom

To enlarge a displayed feature, position the cross-hair mouse pointer over the feature then press the right mouse button. The window will swap from normal display resolution to zoom resolution and the status line will have the word **Zoomed** appended to it. The zoomed feature will be in the centre of the window. To cancel **Zoom**, press the right mouse button again or select any command that redraws the page (e.g. Redisplay, Next Page). By default the zoom resolution is 300 dots per inch but this can be changed with the Media | Zoom Resolution command.

Zoom will only work for DSC conforming documents.

Document Information

A brief information area at the top of the window is used by GSview to display the document filename, the current page number and label (if available) and while the cursor is over the image, the location of the cursor in coordinates specified by Options | Units. The cursor location is useful for calculating bounding boxes.

The **Info** command on the **File** menu shows a dialog box with the following information about the DSC comments in the current document.

File is the full pathname to the document.

Type is **DSC**, **EPS**, **No DSC comments** or 'Ignoring DSC Comments'. EPS is an Encapsulated PostScript File - a single page document that contains a subset of the DSC comments and PostScript commands. **EPS** files are commonly used for inclusion in other documents. **Ignoring DSC Comments** is displayed if Options | Ignore DSC is selected.

Title is a text title that can be used when printing banner pages and for routing or recognising documents.

Date is the time the document was created.

BoundingBox specifies a box that encloses all the marks painted on the page.   The four integer values are the coordinates of the lower left and upper right corners of the bounding box in default user coordinates (1/72 inch).

Orientation is either **Portrait** or **Landscape**.

Default Media gives the media name followed by the width and height of that media in default user coordinates (1/72 inch).

Page Order is either **Ascending**, **Descending** or **Special**

Pages is the total number of pages in the document.

Page gives the page label and page number.

Bitmap is the size of the display bitmap in pixels which may be useful if you are copying the displayed image to the clipboard.

Printing

The **Print** command on the **File** menu allows printing of the document using Ghostscript. A **Printer Setup** dialog box allows selection of the Ghostscript printer driver and resolution, the page range and the Spooler output.

All pages, individual pages or any combination may be printed. The **All**, **Odd** and **Even** buttons provide quick selection of pages.

The **mswinpr2** printer driver uses the windows printer drivers and should work with any printer with raster capabilities. Printer resolution cannot be selected from within GSview; use the Control Panel instead.

The **mswinprn** printer driver is an older version of **mswinpr2** and may not still work.   This driver is very slow and should be considered a last resort if no other Ghostscript printer driver is available.

With all other printer drivers, Ghostscript sends the output direct to the printer, without passing through a Windows printer driver. If you have trouble printing you may have to **Print To File** and then **Print File** or use the DOS command **COPY /B FILENAME PRN**.

The Win32 versions of GSview and Ghostscript can't send output directly to a printer port so an attempt is made to pass the output unchanged through a Windows printer driver.   This does not work with some Windows printer drivers.   Use 'Print To File' if you have problems.

This list of available devices and resolutions is stored in the [Devices] section of gsview.ini (or gsview32.ini for gsview32.exe). The default list of devices and resolutions is taken from the standard distribution version of Ghostscript 3.33 and may not be complete.

To print a document without displaying it, open the document using Select File.

Print To File is similar to the **Print** command except that Ghostscript will write the output to a file instead of sending it to a printer.

If you want to produce a bitmap, some useful drivers are **bmpmono**, **bmp16**, **bmp16m** and **bmp256**.

Print File sends a file to a local port, bypassing the Windows printer drivers. This is useful for sending a document to a PostScript printer, or for sending an output file produced by Ghostscript to a printer.

Spooler
Properties

Spooler

The **Select Printer Port** or **Select Printer** dialog box prompts you for the name of a printer queue or port to which output should be sent. This queue or port will be saved and will be the default selection next time.

Under Windows 3.1, a port name is used.   Output will be spooled for this port.

Under Windows 3.1 / Win32s, a port name is used.   Output will be written directly to this port.   The spooler is not used.

Under Windows 95 or NT, a printer queue name is used. At present this must be a local queue.

See also Print

Properties

The **Printer Setup** dialog box has a **Properties** button which allows some printer drivers to configure extra properties and a page offset to be specified.

A page offset can be specified for each printer.   This will only work if Ghostscript Version is set to 3.51 or later. The page offset is useful for correcting a mismatch between the page origin of a Ghostscript printer device and a particular printer. Increasing the X value will translate the image towards the right. Increasing the Y value will translate the image downwards.

Properties are typically used to set BitsPerPixel for a colour   printer or other types of colour or density correction.

Properties are specific to a particular printer.   Changing the value of the **BitsPerPixel** property on one printer does not   change it for any other printer.

When you press the **OK** button in the **Properties** dialog box, the current settings are written to the gsview.ini (or gsview32.ini for gsview32.exe) file.

No **Properties** are predefined in GSview.   The Edit and **New** buttons allow you to modify available **Properties** for that printer. See the Edit Properties topic for more details.

Edit Properties

Edit Properties

Not every printer supports the use of optional <u>Properties</u>.   To find out which printers support <u>Properties</u> and which <u>Properties</u> are recognised by each printer, read the Ghostscript file devices.doc or look at the Ghostscript source code.

There are two ways to add or edit <u>Properties</u>.

The first method uses the <u>Edit</u> or **New** button on the <u>Properties</u>   dialog box.

Each property must be either a number or a string.   Number properties are equivalent to the Ghostscript **-d** command line option.   String   properties are equivalent to the Ghostscript **-s** command line option. Each property consists of a **Name** and **Value**.   These are used as   **-dNAME=VALUE** or **-sNAME=VALUE**. The **Value** is chosen from the comma separated list of **Values** entered into the **Edit Properties** dialog box.   Spaces must not be embedded in the **Name** or **Values**.

To delete a property, select it on the <u>Properties</u> dialog box, then press <u>Edit</u>, then press the **Delete** button on the **Edit Properties** dialog box.

The second method is to manually edit the GSview INI file.

For each printer, you must add two sections to the gsview.ini (or gsview32.ini for gsview32.exe) file. The following example shows how to add property information for the cdjcolor driver. First add a section which gives the current values. This section, after the first character is removed, gives the options that will appear in the **Property** list box. The first character is **s** for string or **d** for number.

```
 [cdjcolor]
 dBitsPerPixel=24
 dDepletion=1
 dShingling=2
 dBlackCorrect=4
```

Next add a section which gives the values to display in the **Value** list box.

```
 [cdjcolor values]
 dBitsPerPixel=1,3,8,16,24
 dDepletion=1,2,3
 dShingling=0,1,2
 dBlackCorrect=0,1,2,3,4,5,6,7,8,9
```
GSview will also add the value **[Not defined]** to the listbox.

When GSview prints a file, it will give Ghostscript the contents of the [cdjcolor] section of gsview.ini (or gsview32.ini for gsview32.exe) as follows:

```
 -dBitsPerPixel=24 -dDepletion=1 -dShingling=2 -dBlackCorrect=3
```
If the value of a property is **[Not defined]**, that property will not be sent to Ghostscript.

Some entries for the cdj family of drivers are supplied in the property.ini file that comes with GSview.

Text Extract and Find

In general, extracting text from a PostScript document is not a trivial operation.   Words may be broken. Text may be encoded. Ligatures may be used (e.g replacing 'fi' with a single character). There may be no relationship between the location of a word in the PostScript file and its location on the page.

However, it is common for PostScript documents to contain text in the same order as it appears on the page, and for it to be given in PostScript strings, surrounded by parentheses.   Complete lines may be given in one string, or one word per string.   For this sort of document, extracting text can be done with reasonable success.

<u>Edit</u> | **Text Extract** will extract text contained in strings from specified pages and write it to a text file. Line breaks in this text file correspond to lines in the document. Spaces in the text file correspond to spaces within strings, or to separate strings.   A more effective method of extracting text is to use ps2ascii.ps supplied with Ghostscript.

<u>Edit</u> | **Find** will search for text and display the first page that contains the text.   **Find** asks for a search text and a range of pages in which to search. The preceeding comments about extracting text from a PostScript document should be noted.   **Find** first extracts text from the document, then searches it ignoring all spaces in both the document and the search text.   Case is ignored when searching. Consequently the search text **these** would match both **These** and **The serial**.   No information is given about where the word is located on a given page because this information is not available without a complete PostScript interpreter.

<u>Edit</u> | **Find Next** will continue the search from the next page.

Clipboard

The GSview window can be copied to the Clipboard as a bitmap by selecting **Copy** from the **Edit** menu. The bitmap may be a Device Independent Bitmap or it may be a Device Dependent Bitmap, depending on how Ghostscript was compiled. The default for Ghostscript 3.53 is a Device Independent Bitmap (BMP format).

An alternative way to get a bitmap output from Ghostscript is to use one of the BMP drivers.   See Print.

Paste To copies a Device Independent Bitmap from the Clipboard (if available) to a BMP file.

Convert Bitmap converts between a Device Independent Bitmap and a Device Dependent Bitmap. If the clipboard contains a Device Independent Bitmap (BMP format), this is converted to a Device Dependent Bitmap and added to the clipboard. If the clipboard does not contain a colour palette, one is created from the Device Independent Bitmap and added to the clipboard. This option is present because some applications (notably Windows Paintbrush) won't recognise a Device Independent Bitmap in the clipboard.

Add EPS Preview takes a bitmap from the clipboard and uses it to add a preview to an EPS file. **Add EPS Preview** can create a DOS EPS file with a Windows Metafile or TIFF preview, or an EPSI file with an Interchange preview. To use the **Add EPS Preview** command the following steps must be followed.

1. Make sure the document has a correct bounding box. A bounding box can be added or changed using File | PS to EPS.

2. Select Orientation | Portrait.

3. Select Options | EPS Clip. This will cause Ghostscript to use a display window the size of the bounding box instead of the page size.

4. Select Media | Resolution and set a suitable resolution for the preview.   If the resolution is too high the bitmap may not fit in the clipboard, or will make the EPS file excessively large.

5. Open an EPS file that does not contain a preview.

6. Select **Edit** | **Copy**.   This tells Ghostscript to copy the display bitmap to the Clipboard.   This allows GSview to access the bitmap in the next step.

7. Select **Edit** | **Add EPS Preview**, then the preview format, then the new EPS filename.   GSview will write a new file containing the original PostScript EPS file and a preview created from the bitmap in the clipboard. The available preview formats are **Interchange**, **TIFF 4**, **TIFF 5** and **Windows Metafile**. If adding an Interchange preview, the document must have an **%%EndComments** line, otherwise GSview may put the preview in the wrong place. A TIFF 5 preview is a Class B image with no compression as described in Appendix G of the TIFF 5.0 memorandum.   A TIFF 4 preview is almost identical to the TIFF 5 preview, but avoids using tags which are not described in the TIFF 4 specification. WordPerfect 5.1 requires a TIFF 4 preview.

8. Reset Orientation | Portrait, Options | EPS Clip and Media | Resolution to their previous values.

To extract the PostScript or Preview section from a DOS EPS file, use File | Select File followed by **Edit** | **Extract EPS** then **PostScript** or **Preview**.

See also PS to EPS.

Options

The **Options** menu has the following selections:

Ghostscript Command
Ghostscript Version
Sounds
Units
Save Settings
Safer
Save Last Directory
Button Bar
Quick Open
Auto Redisplay
EPS Clip
EPS Warn
Ignore DSC

Ghostscript Command

The **Ghostscript Command** option allows selection of the command to use when executing Ghostscript. A default will be constructed using the GSview path and will look like:

```
gswin -Ic:\gs;c:\gs\fonts
```

GSview will first attempt to find Ghostscript in the GSview directory and then will try the PATH. If GSview still cannot find Ghostscript, use this option to set the full pathname to Ghostscript. If you wish to specify that Ghostscript should look for its initialisation files in a different place, modify the -I option. Try to keep the command length short, otherwise GSview will have trouble printing files with Ghostscript. Only the Ghostscript EXE and include path may be specified - other items will confuse GSview.

Ghostscript Version

GSview is best used with Ghostscript version 3.53 or later.

The 16-bit version of GSview can be used with the 16-bit Ghostscript 2.6.1 by setting this option to **Ghostscript 2.6.1**.

The default is **Ghostscript 3.51 or later**.

See Common Problems for a description of what happens if you set the **Ghostscript Version** to the wrong setting.

Sounds

The **Sounds** option assigns sounds to various events. For each event the sound can be set to **None**, a **Speaker Beep** or a **Wave** file.

You must have a sound driver loaded before using Wave files. Wave file sounds are not available under MS-Windows 3.0.

The events are:

Output Page: the PostScript showpage operator was executed.

No Page: an invalid page was selected. For example, pressing **Prev** while on the first page of a document with DSC comments.

No Number: a command required page numbering and the document did not have page numbering. For example, pressing Goto Page when viewing a document without DSC comments.

Not Open: a command required a document to be open and this was not the case. For example, pressing Goto Page when no document is open.

Error: many types of errors.

Timeout: no response from Ghostscript within a timeout period. For example, display snowflak.ps on a PC with a 286-12 CPU.

Start: GSview opened.

Exit: GSview closed.

The defaults are for **No Page** and **Error** to be a **Speaker Beep** and all other events to be **None**.

Units

The **Units** option sets the units used to display the cursor location on the status bar.   Available units are PostScript points (**pt** = 1/72"), millimetres (**mm**) and inches (**in**). The default is **pt**.

Save Settings

The **Save Settings Now** option saves the GSview window position, window size, last used printer, last directory, Sounds, Units, Save Last Directory, User Defined, Ghostscript Command, Button Bar, Quick Open, Auto Redisplay, EPS Clip, EPS Warn, Ignore DSC, Depth, Orientation, Media, Resolution and Zoom Resolution options to the initialisation file gsview.ini (or gsview32.ini for gsview32.exe) in the Windows system   directory. GSview reads this file during startup.

When the **Save Settings on Exit** option is checked, GSview will automatically save the above settings when you quit GSview.

Safer

When the **Safer** option is **checked**, GSview will give Ghostscript the **-dSAFER** flag, which disables the deletefile and renamefile operators, and the ability to open files in any mode other than read-only. This is the default.

When the **Safer** option is **unchecked** Ghostscript can change files.

Save Last Directory

When the **Save Last Directory** option is **checked**, GSview will save the current directory when you quit GSview.   When GSview is started next, this will be made the current directory. This is the default.

When **Save Last Directory** option is **unchecked**, the current directory when GSview is started will be the directory where GSview is located, or the working directory specified by the Program Manager.

Button Bar

When the **Button Bar** option is **checked**, GSview will display a Button Bar down the left side of the window.   This is the default. The Button Bar contains the following items in order from top to bottom:

File | Open

File | Print

File | Info

Help | **Contents**

View | Goto Page

View | Next Page

View | Previous Page

Go forward 5 pages

Go back 5 pages

Increase resolution by 1.2

Decrease resolution by 1/1.2

Edit | Find

Edit | Find Next

If using the increase/decrease resolution buttons, Auto Redisplay should be set.   Instead of using these buttons, it is also possible to use the Media | Resolution command. When the **Button Bar** option is **unchecked**, GSview will not display the Button Bar.

Quick Open

When the **Quick Open** option is **checked**, GSview will not reload Ghostscript before every document, making opening of documents quicker.

GSview tries to preserve the Ghostscript state between documents, but a document may still leave the Ghostscript interpreter in an unusual state or cause an error.   If an error occurs, Ghostscript will close. Error messages are displayed in the Ghostscript window. Select Redisplay to reopen the document.

If **Quick Open** is **unchecked**, GSview will close Ghostscript and restart it before each new document or whenever the page orientation, resolution or size is changed.

Auto Redisplay

When the **Auto Redisplay** option is **checked**, GSview will redisplay <u>DSC</u> documents when the <u>Orientation</u>, <u>Resolution</u>, <u>Depth</u> or <u>Media</u> are changed. This is the default.

If **Auto Redisplay** is **unchecked**, the <u>View</u> | <u>Redisplay</u> command must be used to redisplay a document after changing the <u>Orientation</u>, <u>Resolution</u>, <u>Depth</u> or <u>Media</u>.

EPS Clip

When the **EPS Clip** option is **checked**, GSview will clip the display bitmap to the bounding box of an EPS file instead of using the page size specified on the Media menu.   This is useful when adding a bitmap preview to an EPS file.

If **EPS Clip** is **unchecked**, GSview will use the page size specified on the Media menu for EPS files. This is the default.

EPS Clip does not alter the original document, it only affects how much of the document is displayed by GSview. **EPS Clip** will only work in Portrait orientation.

See also Clipboard | Add EPS Preview

EPS Warn

When the **EPS Warn** option is **checked**, GSview will write a prolog to Ghostscript when each file is opened.   This prolog will produce warning messages in the Ghostscript text window if any PostScript operators that should not be used in <u>EPS</u> files are used.   An example warning message is:
 `Warning: EPS files must not use /initgraphics`
EPS Warn is not infallible.   It is possible to access restricted operators without **EPS Warn** producing a warning.

The default for **EPS Warn** is **unchecked**.

See also <u>PS to EPS</u>.

Ignore DSC

Some documents incorrectly claim to conform to the Adobe Document Structuring Conventions. Attempting to display one of these bogus documents will probably leave GSview horribly confused and unable to display the document.   If **Ignore DSC** is **checked**, GSview will treat the document as if it does not contain DSC comments and will only display the pages in the original order.

The default for **Ignore DSC** is **unchecked**.

Page Orientation

The **Portrait**, **Landscape**, **Upside-down** and **Seascape** (reverse Landscape) commands on the **Orientation** Menu select the page orientation used by the display. **Landscape** implies a clockwise rotation of the paper by 90 degrees. **Seascape** implies an anti-clockwise rotation of the paper by 90 degrees. These orientation options only affect the display and do not affect the print commands. If a DSC page orientation comment is found, the orientation will be selected automatically.

When the **Swap Landscape** option is **checked**, GSview swaps the meaning of Landscape and Seascape. Most of the Landscape documents that I have encountered require a 90 clockwise rotation of the paper to view. However, there is no standard and some documents need to be rotated the other way. The **Swap Landscape** button allows GSview to automatically rotate the document the right way in response to the **%%Orientation** comment in the PostScript file.

See also Page Size and Display Resolution.

Page Size and Display Resolution

The **Resolution** command on the **Media** menu selects the display resolution in dots per inch. The default for a VGA display is 96 dots per inch.

For DSC conforming files, pressing the right mouse button will zoom into the page at what is usually printer resolution. Pressing the right mouse button a second time will zoom back out to normal display resolution. The **Zoom Resolution** command on the **Media** menu sets the zoom resolution in dots per inch.

The **Depth** sub menu on the **Media** menu selects the display depth in bits per pixels.

The **Media** menu also allows selection of page size. Available page sizes are:
```
Letter           8.5 x 11    inch
Tabloid         11   x 17    inch
Ledger          17   x 11    inch
Legal            8.5 x 14    inch
Statement        5.5 x  8.5 inch
Executive        7.5 x 10    inch
A3              297 x 420    mm
A4              210 x 297    mm
A5              148 x 210    mm
B4              257 x 364    mm
B5              182 x 257    mm
Folio            8.5 x 13    inch
Quarto           8.5 x 10.8 inch
10x14           10   x 14    inch
```

A user defined size can be specified in PostScript points (1/72 inch) with the **User Defined** command. A size of 480x360 points at 96 dpi will give an image size of 640x480 pixels.

If a DSC media comment is found, the page type will be selected automatically.   If the media specification is not one of the above page types, the **User Defined** size will be set.

Keys

Following are the key assignments for GSview.

O, **o**   Open and display a file.   (<u>File</u> | <u>Open</u>)

C, **c**   Close file.   (<u>File</u> | <u>Close</u>)

N, **n**, **+**   Next Page.   (<u>View</u> | <u>Next Page</u>)

V, **v**, **-**   Previous Page.   (<u>View</u> | <u>Previous Page</u>)

G, **g**   Goto Page.   (<u>View</u> | <u>Goto Page</u>)

I, **i**   File information.   (<u>File</u> | <u>Info</u>)

R, **r**   Redisplay page.   (<u>View</u> | <u>Redisplay</u>)

S, **s**   Select file: open but don't display.   (<u>File</u> | <u>Select File</u>)

A, **a**   Save As. (<u>File</u> | <u>Save As</u>)

P, **p**   Print all or some pages to a printer.   (<u>File</u> | <u>Print</u>)

F, **f**   Print all or some pages to a File.   (<u>File</u> | <u>Print To File</u>)

E, **e**   Extract some pages to another File.   (<u>File</u> | <u>Extract</u>)

F1   Help.   (**Help** | **Contents**)

Ctrl+C, **Insert**   Copy displayed bitmap to clipboard.   (<u>Edit</u> | <u>Copy</u>)

Up   Scroll up 16 pixels.

Down   Scroll down 16 pixels.

Left   Scroll left one screen.

Right   Scroll right one screen.

Page Up   Scroll up one screen (window height).

Page Down   Scroll down one screen.

Home   Scroll to top of page.

End   Scroll to bottom of page.

Running GSview from the File Manager

If you used the GSview installation program, then the File Manager will already know about PostScript files.   If you didn't use the GSview installation program, read on.

To run GSview when a PostScript file is double clicked in the Windows 3.1 File Manager, the following sequence must be followed   to teach File Manager about PostScript files.

From the **Program Manager**, run the Registration Info Editor using **File** | **Run...** then type **regedit**. From the **Registration Info Editor** select **Edit** | **Add File Type...** then enter the following fields:
```
Identifier = psfile
Filetype = PostScript
Action = Open
Command = gsview %1
Uses DDE = unchecked
Action = Print
Command = gsview /p %1
Uses DDE = unchecked
```
Then press **OK**.

From the **File Manager**, select **File** | **Associate** then enter the following fields:
```
Files with Extension = ps
Associate With = PostScript (gsview)
```
Then press **OK**.

Put gsview.exe and gsview.hlp in a directory on your PATH. That's it!   Now when you double click on a PostScript file, the **File Manager** will run GSview.   When you drop a PostScript file on the **Print Manager**, GSview will print the file.

For Windows 95, the configuration is instead made using Windows Explorer.

Start **Windows Explorer**.   Select **View** | Options .   Select the **File Types** tab.   Select the **New Type** button.   Enter the following fields:
```
Description of type = PostScript
Associated Extension = PS EPS
```
Press the **New** button, then enter
```
Action = open
Application used to perform action = c:\gsview\gsview32.exe
```
Press the **OK** button

Press the **New** button, then enter
```
Action = print
Application used to perform action = c:\gsview\gsview32.exe /p
```
Press the **OK** button

Press the **Close** button.

Press the **Close** button.

Command line options

Usage:
```
gsview [/D] filename
gsview [/D] /F filename
gsview [/D] /P filename
gsview [/D] /S[port] filename
```

To start GSview and display filename.ps use:
```
gsview filename.ps
```
To start GSview and print filename.ps using Ghostscript (<u>File</u> | <u>Print</u>) use:
```
gsview /P filename.ps
```
To start GSview and print filename.ps to a file using Ghostscript (<u>File</u> | <u>Print To File</u>) use:
```
gsview /F filename.ps
```
To start GSview and spool filename.ps for printing directly to a printer (<u>File</u> | <u>Print File</u>) use:
```
gsview /S filename.ps
```
To start GSview and spool filename.ps for printing directly to printer port LPT3: use:
```
gsview /SLPT3: filename.ps
```

To start GSview in debug mode use:
```
gsview /D
```
In debug mode GSview will **not** remove its temporary files. This is to allow inspection of these files after GSview has finished.

GSview ignores the case of option: /p is the same as /P.

World Wide Web

The World Wide Web home page for Ghostscript, Ghostview and GSview   is at
`http://www.cs.wisc.edu/~ghost/index.html`

GSview can be used as a PostScript file viewer for several OS/2 and MS-Windows Web browsers.   See the GSview home page for details.

Copyright

The **About** menu item shows the GSview copyright message and GSview version number.

```
GSVIEW.EXE - A Ghostscript graphical interface
Copyright (C) 1993-1996, Russell Lang.  All rights reserved.
Portions Copyright (C) 1994, Timothy O. Theisen.  All rights reserved.
```

This file is part of GSview.

This program is distributed with NO WARRANTY OF ANY KIND.   No author or distributor accepts any responsibility for the consequences of using it, or for whether it serves any particular purpose or works at all, unless he or she says so in writing.   Refer to the GSview Free Public Licence (the "Licence") for full details.

Every copy of GSview must include a copy of the Licence, normally in a plain ASCII text file named LICENCE.   The Licence grants you the right to copy, modify and redistribute GSview, but only under certain conditions described in the Licence.   Among other things, the Licence requires that the copyright notice and this notice be preserved on all copies.

```
   Author: Russell Lang
 Internet: rjl@aladdin.com
```

Common Problems

Problem: Can't run gswin...

GSview requires Ghostscript for Windows (gswin.exe). This error message usually occurs if you don't have Ghostscript for Windows, or if GSview can't find Ghostscript for Windows.

From the GSview menu select <u>Options</u> | <u>Ghostscript Command</u> and enter the correct executable path and include path for Ghostscript. For example:
```
 c:\gs\gswin.exe -Ic:\gs
```

You must set the include path for Ghostscript using either the **-I** command or the **GS_LIB** environment variable.

Make sure you don't already have Windows Ghostscript running. Only one copy of Windows Ghostscript can be running at a time. Only one copy of GSview can be running at a time.

If you can't get GSview to run Ghostscript correctly, make sure you can run Ghostscript on its own.

Problem: Ghostscript starts and then immediately exits with error code 1.

Read the error message in the Ghostscript window before pressing OK in the message box.

Ghostscript probably couldn't find its initialisation files. Set the Ghostscript include path by adding **-Idirectory** to the command in <u>Options</u> | <u>Ghostscript Command</u>, where **directory** includes the name of the directory that contains the Ghostscript initialisation files, Alternatively, set the environment variable GS_LIB.

Read the file use.doc that comes with Ghostscript.

If you can't get GSview to run Ghostscript correctly, make sure you can run Ghostscript on its own.

Alternatively, you may be using an incompatible version of Ghostscript, or you may have the <u>Options</u> | <u>Ghostscript Version</u> set incorrectly. This version of GSview should be used with Ghostscript version **3.12** or later. If you are using an incompatible version of Ghostscript or you have <u>Options</u> | <u>Ghostscript Version</u> set incorrectly, the Ghostscript text window will show one of the following error messages
```
 undefined in ear
 undefined in XXclear
```
where **XX** are a pair of garbage characters.

Problem: Message Box says **Incompatible Windows Ghostscript**

You probably have an earlier version of Ghostscript. Get Ghostscript version 3.12 or later.

Problem: GSview says 'Drawing...' or 'Printing...' and shows an hourglass cursor.   When the cursor is moved off the GSview window it changes to a normal arrow.

The above state is normal while GSview is waiting for Ghostscript to do something.   It may be that the document is very complex and Ghostscript is just taking a long time.   In this state it is safe to resize or scroll the window, but do not select any menu commands.

If this is not the case then GSview probably got no response from Ghostscript. Open the Ghostscript text window and look at any error messages. Close Ghostscript using the Ghostscript system menu. If GSview is still waiting, press the GSview Open button (even though the cursor is still an hourglass). Press the 'Yes' button on the 'gsview is busy' message box. GSview should then revert to the non-waiting mode.

Problem: Message Box says **Imitation pipe handle is zero**

If you are running Windows 95 or NT, this is probably caused by a mismatch between GSview and Ghostscript. Make sure you are running both GSview32 and gswin32.

Problem: GSview displays a "Pipe error" message box.

GSview could not open a temporary file. GSview requires a temporary file for **piping** commands to Ghostscript. This temporary file is created in the directory given by the environment variable TEMP, or if that is not set, in the current directory. **Pipe error** will result when

1. TEMP is not set and the current directory is read-only. For example, the current directory is on a CD-ROM or a network.

Solution: Set TEMP environment variable in AUTOEXEC.BAT.

2. TEMP is set but does not point to a valid read-write directory.

Solution: Change TEMP to point to a directory that exists and is read-write.

3. Disk is full.

Problem: Ghostscript under Win32s won't print anything.

Under Win32s, the 16 bit spooler functions are not available and the 32 bit spooler functions are not implemented. That is, Win32s is the worst of both worlds. The printer output code in Aladdin Ghostscript 3.51 for Win32s attempts to write directly to a port. This works on some computers, but fails on others. If it doesn't work on your computer, you have two options:

1. Print To File, then Print File. GSview under Win32s can write to the 16 bit spooler, using a 16 bit program.

2. Try the mswinpr2 device.

Aladdin Ghostscript 3.53 starts the 16 bit program gs16spl.exe to spool printer output. It does not suffer from the problem described above.

Problem: GSview says that a multipage PostScript file produced by MS-Windows contains 0 pages and will only show the first page.

This is because the document does not have correct DSC comments. From the Control Panel, select **Printers**, **Options...**, then in the **Print to** group box click on the **Printer** radio button. You cannot use the **Print To Encapsulated PostScript File** for printing multipage files. The correct method is to connect the printer to **FILE:**. In addition, from the Control Panel select **Printers**, **Options...**, **Advanced** and then check **Conform to Adobe Document Structuring Convention**.

The DSC comment **%%Pages: 0** means that the document does not produce any pages. That is, the PostScript **showpage** operator is not used. If you find a PostScript document that has multiple pages and contains the **%%Pages: 0** comment, change the first line from **%!PS-Adobe-** to **%!**. GSview will then ignore the DSC comments and allow you to view all pages, but only in the original order. Complain to the author of the program that produced that PostScript file.

Some PostScript printer drivers include code that is specific to a particular printer. The PostScript output from these drivers may be unportable and may not display in GSview. If you are having this problem, try using a reasonably generic PostScript driver such as **Apple LaserWriter II NT** for PostScript

level 2 printers, or **Apple LaserWriter Plus** for PostScript level 1 printers.

Problem: PostScript files produced by MS-Windows start with a Control-D.

Since this occurs even when the PostScript printer **Conform to Document Structuring Convention** checkbox is checked, this must be considered a bug in the MS-Windows PostScript printer driver. The bug fix is documented in the MS-Windows PRINTERS.WRI file. Edit the win.ini file and search for the PostScript printer section. There may be more than one.   In each of these sections add **CTRLD=0** as shown below.

```
[Apple LaserWriter II NT,FILE]
CTRLD=0
```

Problem: PostScript files produced by Word for Windows 6.0 cause a "Missing %%Pages comment" message box.

Congratulations.   You have just found a mistake in the DSC comments when Word included an EPS file. Word should have surrounded the included EPS file with the lines
```
%%BeginDocument: filename.eps
%%EndDocument
```
Because Word didn't do this, GSview can't tell how many pages are in the document and where they are located.

Please complain to Microsoft.   There is a problem in the EPSIMP.FLT filter version 2.01 which Microsoft needs to fix.

In the interim, you have two solutions:

1. Select <u>Options</u> | <u>Ignore DSC</u>

2. Edit the PostScript file to correct the DSC comments. Search the PostScript file for all lines containing
```
%MSEPS Preamble
```
From each of these lines, search forward for the start of the included EPS file which should start with a line like
```
%%PS-Adobe-3.0 EPSF-3.0
```
Above these lines add the line
```
%%BeginDocument: AddedByHand
```

Then search for all lines containing
```
%MSEPS Trailer
```
Above these lines add the line
```
%%EndDocument
```

GSview should then be able to display the file correctly.

Problem: Ghostscript always outputs to a printer instead of the GSview window.

You must not use the GS_DEVICE environment variable.

Problem: Ghostscript displays a message box "pipe overflow".

The document may contain incorrect DSC comments.

You must wait until Ghostscript is not busy before attempting to display another page.   Exit and then restart GSview.

Whats new

```
----------------------
 Version 1.4 1996-01-02
 ----------------------
 All:
  File was being opened twice if file was changed.
  Should be used with Aladdin Ghostscript 3.53.
 Win16:
  Prevented GPF when running out of memory.
  Reduced stack size and pipe buffer size to allow larger documents
  to be displayed.  If only numeric page labels are used, 500 page
  documents should be OK.
 PM:
  Requires emx 0.9b or later.
  PDF viewing was broken when "Configure Ghostscript" was added.



 ---------------------------
 Version 1.33 beta 1995-11-30
 ---------------------------
 All:
  Added PageOffset for printers.  See the Properties dialog box.
  Add 'Edit Properties' dialog box to allow user to add, edit or delete
  properties for a printer without having to hand edit the INI file.
 Win:
  Installation program now creates "GS Tools" program group and
  adds File Manager associations for .ps and .eps to the registry.



 ---------------------------
 Version 1.32 beta 1995-11-23
 ---------------------------
 All:
  Installation programs added.
 Win32s:
  Uses new gsv16spl.exe to write to the 16-bit spooler from Win32s.
 PM:
  Fixed printing of non DSC documents (was broken when printer
  dialog boxes were combined).



 ---------------------------
 Version 1.31 beta 1995-11-03
 ---------------------------
 This is an interim release, intended to fix some known problems
 and add a few new features, but does not include the full DLL
 implementation needed to work reliably under Windows 95/NT,
 or paper size handling using level 2 setpagedevice.
 It is intended for use with Aladdin Ghostscript 3.51.
 All:
```

```
   Still doesn't work with landscape PostScript that sets the
   clipping path.
   Combined printer dialog boxes.
   Allow user supplied preview file in creation of DOS EPS file for both
   GSview "Edit | Add EPS Preview | User Supplied Preview" and EPSTOOL.
   Added warning if EPS BoundingBox was entirely off the page.
   Replace label of "%%Pages: () 1" with page number.
   Changed "Ghostscript Command" into "Configure Ghostscript" which
   has separate "Ghostscript EXE" and "Ghostscript Include Path".
   Changed TIFF code to work on big-endian architectures, to allow
   EPSTOOL to run under Solaris.
   Directories src, gsgrab and epstool are now in ZIP files, since
   most people won't want them unzipped.
PM:
 Uses print queues instead of writing to ports.
 It is now possible to use a Ghostscript path with embedded spaces.
 Allow user to specify drawing method GpiDrawBits or WinDrawBitmap
 to dodge display driver bugs.
 Recover when last directory is invalid (usually no floppy in drive).
Win:
 Added Ghostscript 3.51 to "Options | Ghostscript Version".  Selecting
 GS 3.51 puts quotes around filenames in response files to allow embedded
 spaces.  This won't work with GS 3.33.
Win32:
 Uses print queues under Win95 and WinNT.
```

Internals

The preferred method for driving Ghostscript would be to start it as a child process and then to send it PostScript code through a pipe. Windows 3.1 supports neither child processes nor pipes.

GSview interacts with Ghostscript by sending messages between the GSview and Ghostscript windows. GSview creates an imitation pipe to Ghostscript by writing into a shareable global memory block (or memory mapped file under Windows 95 or NT) and then passing the handle to Ghostscript.

GSview starts Ghostscript for displaying using
```
 gswin -dBitsPerPixel=x -dSAFER -rXDPIxYDPI -gWIDTHxHEIGHT -sGSVIEW=xxxx -
```
where xxxx is the handle to the GSview window. Ghostscript then tells GSview the handle to the text window by sending a message WM_USER with wParam HWND_TEXT=0 to the GSview window. Instead of creating another window for the image, Ghostscript creates a child window of the GSview window for displaying the bitmap. Ghostscript tells GSview the handle to this child image window by sending a message WM_USER with wParam HWND_IMGCHILD=1.

Ghostscript also sends WM_USER messages to GSview with wParam GSWIN_CLOSE=2 when it is exiting, SYNC_OUTPUT=3 when the image needs to be redrawn (win_sync_output), OUTPUT_PAGE=4 when a page is to be output (win_output_page), SCROLL_POSITION=5 when the window is scrolled, PIPE_DATA=6 for passing the handle to a global memory block (in lParam) for the imitation pipe, BEGIN=7 when "-1 false .outputpage" is executed and END=8 when "-2 false .outputpage" is executed.

When Ghostscript sends the OUTPUT_PAGE command it waits until it receives a NEXT_PAGE command from GSview.

GSview sends WM_USER messages to the Ghostscript Image window with wParam NEXT_PAGE=10 when it is time to move to the next page and COPY_CLIPBOARD=11 when the bitmap should be copied to the clipboard. GSview sends WM_USER messages to the Ghostscript Text window with wParam PIPE_REQUEST=12 when it wants more pipe data.

GSview uses temporary files of the name gsviewXX.XXX in the directory given by the TEMP environment variable. Keep TEMP short to avoid the gswin command line exceeding 128 characters.

GSview starts Ghostscript for printing using
```
 gswin -sGSVIEW=xxxx @optfile
```
optfile contains
```
 -dNOPAUSE
 -dSAFER
 -sDEVICE=devname
 -rXDPIxYDPI
 -gWIDTHxHEIGHT
 -sOutputFile=outfilename
 filename.ps
 quit.ps
```