August 16, 1992

NAME

gs - PostScript previewer

SYNOPSIS

gs [switches] [file1.ps file2.ps ...]

DESCRIPTION

Ghostscript is an interpreter for Adobe Systems' PostScript (tm) language. Gs reads each given file in sequence and displays or prints it as a Ghostscript file. It then interprets commands from the primary input stream (normally the keyboard) until an end of file character (^D or CTRL-D) is typed. Typing `quit' at the Ghostscript prompt also terminates Ghostscript execution. Typing the interrupt character, e.g., **control-C**, also terminates Ghostscript execution.

You can get a help message by invoking Ghostscript with

```
     gs -h
```

or

```
     gs -?
```

Choosing the output device

Ghostscript may be built with multiple output devices. Ghostscript normally opens the first one and directs output to it. To use device xyz as the initial output device, include the switch

```
     -sDEVICE=xyz
```

in the command line. Note that this switch must precede the first .ps file, and only its first invocation has any effect. For example, for printer output in a normal configuration that includes an Epson printer driver, you might use the shell command

```
    gs -sDEVICE=epson myfile.ps
```

instead of just

```
    gs myfile.ps
```

Alternatively, you can type

```
    (epson) selectdevice

    (myfile.ps) run
```

All output then goes to the printer instead of the display until further notice. You can switch devices at any time by using the selectdevice procedure, e.g.,

```
    (vga) selectdevice
```

or

```
    (epson) selectdevice
```

As yet a third alternative, you can define an environment variable GS_DEVICE as the desired default device name. The order of precedence for these alternatives, highest to lowest, is:

```
               selectdevice
               (command line)
               GS_DEVICE
               (first device in build list)
```

To select the density on a printer, use

```
     gs -sDEVICE=<device> -r<xres>x<yres>
```

For example, on a 9-pin Epson-compatible printer, you can get the lowest-density (fastest) mode with

```
     gs -sDEVICE=epson -r60x72
```

and the highest-density mode with

```
     gs -sDEVICE=epson -r240x72.
```

On a 24-pin printer, the lowest density is

```
     gs -sDEVICE=epson -r60x60
```

and the highest-density 24-pin mode is

```
     gs -sDEVICE=epson -r360x180
```

If you select a printer as the output device, Ghostscript also allows you to control where the device sends its output. Normally, output goes directly to the printer (PRN) on MS-DOS systems, and to a scratch file on Unix or VMS systems. To send the output to a series of files foo1.xyz, foo2.xyz, ..., use the switch

```
     -sOutputFile=foo%d.xyz
```

The %d is a printf format specification; you can use other formats like %02d. Each file will receive one page of output. Alternatively, to send the output to a single file foo.xyz, with all the pages concatenated, use the switch

```
     -sOutputFile=foo.xyz
```

On Unix systems, you can send the output directly to a pipe. For example, to pipe the output to the command `lpr' (which, on many Unix systems, is the command that spools output for a printer), use the switch

```
     -sOutputFile=|lpr
```

To find out what devices are available, type

```
    devicenames ==
```

after starting up Ghostscript. Alternatively you can use the -h or -? switch in the command line, as described above.

Device configuration

Ghostscript is normally configured to expect U.S. letter paper, although there is a way to make A4 paper the default for certain printers at compilation time (see devs.mak for details). To select a different paper size as the default, use the switch

```
     -sPAPERSIZE=a_known_paper_size
```

e.g.,

```
     -sPAPERSIZE=a4
```

or

```
     -sPAPERSIZE=legal
```

You can use any paper size listed in the table at the beginning of gs_statd.ps. (Individual documents can also specify a paper size, which will take precedence over the one specified on the command line.)

Printing on a Hewlett-Packard DeskJet or LaserJet at full resolution (300 DPI) requires a printer with at least 1.5 Mb of memory. 150 DPI printing requires only .5 Mb. You can select 150 DPI printing with the command line switch

```
     -r150
```

On MS-DOS systems using the Borland compiler, if Ghostscript gives you a 'limitcheck in setdevice' error, it may mean Ghostscript's standard buffer size wasn't large enough. Likewise, if Ghostscript gives you a 'VMerror in setdevice' error, it means the buffer size was too large. You can use the -dBufferSpace= switch to set the buffer size to a different value, e.g.,

```
     -dBufferSpace=50000
```

The default value is 25000; the smallest value Ghostscript accepts is 10000; the largest valid value is 65000.

File searching

When looking for the initialization files (gs_*.ps), the files related to fonts, or the file for the 'run' operator, Ghostscript first tries opening the file with the name as given (i.e., using the current working directory if none is specified). If this fails, and the file name doesn't specify 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), Ghostscript will try directories in the following order:

- The directory/ies specified by the -I switch(es) in the command line (see below), if any;
- The directory/ies specified by the GS_LIB environment variable, if any;
- The directories specified by the GS_LIB_DEFAULT macro, which is typically set to "/usr/lib/ghostscript:/usr/lib/ghostscript/fonts" at compilation time in the Ghostscript makefile.

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 and MS-DOS systems).

VMS-specific notes

On VMS systems, the last character of each "directory" name indicates what sort of entity the "directory" references. If the "directory" name ends with a colon, it is taken as referring to a logical device, e.g.:

```
        $ DEFINE GHOSTSCRIPT_DEVICE DUA1:[GHOSTSCRIPT_14]
        $ DEFINE GS_LIB GHOSTSCRIPT_DEVICE:
```

If the "directory" name ends with a closing square bracket, it is taken as referring to a real directory, e.g.:

```
        $ DEFINE GS_LIB DUA1:[GHOSTSCRIPT]
```

To run Ghostscript with switches, you must type a command like

```
$ gs "-dNODISPLAY"
```

because the C run time library will convert the command parameters/arguments to lowercase unless you enclose them in double quotes which preserves the case.

If you are on an X Windows display (for which gs is built), you can do

```
$ set display/create/node="domain-name"/transport=tcpip
```

For example,

```
$ set display/create/node="doof.city.com"/transport=tcpip
```

and then run Ghostscript

```
$ gs
```

If you write printer output to a file and then want to print the file later, use the "/PASSALL" qualifier to the PRINT command.

MS-DOS notes

If you are running Ghostscript on a MS-DOS machine with a display that is not EGA/VGA compatible, you must use the Borland compiler. You must build Ghostscript with the BGI driver as the default, and you will need the appropriate .BGI file from the Borland Turbo C library. (Ghostscript includes the EGA/VGA driver in the executable.)

If you are using the BGI driver, two additional environment variables become relevant:

BGIPATH
: defines the directory where Ghostscript will look for the appropriate BGI driver. If BGIPATH is not defined, Ghostscript will look in the directory defined as BGIDIR in the makefile. In either case, if no driver is found in the designated directory, Ghostscript will look in the current directory.

BGIUSER
: a string of the form nn.dname, where nn is a hexadecimal number giving a display mode and dname is the name of a file containing a user-supplied BGI driver. If BGIUSER is defined and the BGI device is selected, Ghostscript will supply nn as the display mode and will obtain the driver from the file named dname.

Some applications, such as Microsoft Word, require a prologue in front of the PostScript files they output. In the case of Word, this is one of the *.ini files included with the Word distribution. Other applications may require other prologues. These may be specified on the Ghostscript command line, e.g.,

```
     gs prologue.ini myfile.ps
```

X Windows resources

Ghostscript looks for the following resources under the program name ``Ghostscript'':

borderWidth
the border width in pixels (default = 1)

borderColor
the name of the border color (default = black)

geometry
the window size and placement, WxH+X+Y (default = ???)

xResolution
the number of x pixels per inch default is computed from WidthOfScreen and WidthMMOfScreen

yResolution
the number of y pixels per inch default is computed from HeightOfScreen and HeightMMOfScreen

To set these resources, put them in a file (such as ~/.Xdefaults in X11R3 or ~/.Xresources in X11R4 or X11R5) in the following form:

```
    Ghostscript*geometry: 612x792-0+0
    Ghostscript*xResolution: 72
    Ghostscript*yResolution: 72
```

Then load the defaults into the X server:

```
% xrdb -merge ~/.Xresources
```

OPTIONS

The interpreter recognizes several switches described below, which may appear anywhere in the command line and apply to all files thereafter.

Normal switches

@filename

Causes Ghostscript to read filename and treat its contents the same as the command line. (This is intended primarily for getting around DOS' 128-character limit on the length of a command line.) Switches or file names in the file may be separated by any amount of white space (space, tab, line break); there is no limit on the size of the file.

-- filename arg1 ...
Takes the next argument as a file name as usual, but takes all remaining arguments (even if they have the syntactic form of switches) and defines the name ARGUMENTS in userdict (not systemdict) as an array of those strings, *before* running the file. When Ghostscript finishes executing the file, it exits back to the shell.

-Dname=token

-dname=token
Define a name in systemdict with the given definition. The token must be exactly one token (as defined by the `token' operator) and must not contain any whitespace.

-Dname

-dname
Define a name in systemdict with value=null.

-Sname=string

-sname=string
Define a name in systemdict with a given string as value. This is different from -d. For example,

```
        -dname=35
```

is equivalent to the program fragment

```
        /name 35 def
```

whereas

```
        -sname=35
```

is equivalent to

```
        /name (35) def
```

-q Quiet startup -- suppress normal startup messages, and also do the equivalent of -dQUIET.

-gnumber1xnumber2

Equivalent to -dDEVICEWIDTH=number1 and -dDEVICEHEIGHT=number2. This is for the benefit of devices (such as X11 windows and VESA displays) that require (or allow) width and height to be specified.

-rnumber1xnumber2

Equivalent to -dDEVICEXRESOLUTION=number1 and -dDEVICEYRESOLUTION=number2. This is for the benefit of devices (such as printers) that support multiple X and Y resolutions.

-Idirectories

Adds the designated list of directories at the head of the search path for library files.

- This is not really a switch. It indicates to Ghostscript that the standard input is coming from a file or a pipe. Ghostscript reads from stdin until reaching end-of-file, executing it like any other file, and then continues processing the command line. At the end of the command line, Ghostscript exits rather than going into its interactive mode.

Note that gs_init.ps makes systemdict read-only, so the values of names defined with -D/d/S/s cannot be changed (although, of course, they can be superseded by definitions in userdict or other dictionaries.)

Special names

-dDISKFONTS

causes individual character outlines to be loaded from the disk the first time they are encountered. (Normally Ghostscript loads all the character outlines when it loads a font.) This may allow loading more fonts into RAM, at the expense of slower rendering.

-dNOBIND

disables the `bind' operator. Only useful for debugging.

-dNOCACHE

disables character caching. Only useful for debugging.

-dNODISPLAY

suppress the normal initialization of the output device. This may be useful when debugging.

-dNOPAUSE
disables the prompt and pause at the end of each page. This may be desirable for applications where another program is 'driving' Ghostscript.

-dSAFER
disables the deletefile and renamefile operators, and the ability to open files in any mode other than read-only. This may be desirable for spoolers or other sensitive environments.

-dWRITESYSTEMDICT
leaves systemdict writable. This is necessary when running special utility programs such as font2c and pcharstr, which must bypass normal PostScript access protection.

-sDEVICE=device
selects an alternate initial output device, as described above.

-sOutputFile=filename
selects an alternate output file (or pipe) for the initial output device, as described above.

Debugging switches

The -T and -Z switch only applies if the interpreter was built for a debugging configuration (this is usually not the case for Ghostscript executables installed for public use).

-A Turn on allocator debugging.

-e Turn on tracing of error returns from operators.

-E Abort when any operator returns with an error.

-Mn Force the interpreter's allocator to acquire additional memory in units of nK, rather than the default (currently 20K on MS-DOS systems, 50K on Unix). n is a positive decimal integer (not exceeding 63 on MS-DOS systems).

-Zxxx

Turn on debugging printout. Each of the xxx characters selects an option: if the string is empty, all options are selected. Case is significant.

1 = type 1 font interpreter (type1addpath)
2 = curve subdivider/rasterizer
a = allocator (large blocks only)
A = allocator (all calls)
b = bitmap image processor
B = bitmap images, detail
c = color/halftone mapper
d = dictionary put/undef
f = fill algorithm (summary)
F = fill algorithm (detail)
g = gsave/grestore[all]
h = halftone renderer
i = interpreter, just names
I = interpreter, everything
k = character cache
K = character cache, every access
l = command lists, bands
L = command lists, everything
m = makefont and font cache
n = name lookup (new names only)
o = outliner (stroke)
p = path tracer
q = rectangle fill
r = arc renderer
s = scanner
t = tiling algorithm
u = undo saver (for save/restore)
U = undo saver, more detail
v = rectangle fill
V = device-level output
w = compression encoder/decoder
x = transformations
z = trapezoid fill

EXAMPLES

```
spam% gs
 Initializing... done.
 Ghostscript 2.5.2 (9/19/92)
   Copyright (C) 1990, 1992 Aladdin Enterprises, Menlo Park, CA.
   All rights reserved.
 Distributed by Free Software Foundation, Inc.
 Ghostscript comes with NO WARRANTY: see the file LICENSE for details.
 GS> 100 100 moveto 200 400 lineto stroke
 GS> erasepage
 GS> (/usr/um/lib/ghostscript/tiger.ps) run
```

```
>>showpage, press <return> to continue<<
GS> quit
```

Unix NOTES

At most installations, the only available display device is x11, so you must be running X11 windows to use ghostscript unless you use the command line option -dNODISPLAY. There is also a script called 'gsnd' which is just 'gs -DNODISPLAY -q $*'.

At most installations, Ghostscript was not built for a debugging configuration therefore the command line option -Zxxx is not available. This speeds up the interpreter.

Ghostscript works with color postscript files, postscript files created by Tom Rokicki's dvips, M-PLOT, Mathematica, Tell-a-Graf, etc. You can, for example, preview multiple pages of a postscript file created by Latex/dvips/psfig and you will be able to see any included postscript figure. If the postscript file adheres to Adobe's Document Structuring Convention, like the ones generated by Latex/dvips, then you are advised to use **ghostview** to preview your postscript document for it provides a far more friendly and graphical user interface.

Ghostscript also has the ability to convert a postscript file to a PPM file. Once you have a file in PPM format you can display or manipulate it in many ways. You can, for example, use 'xv' or other tools to convert the PPM file to a variety of formats, like GIF, TIFF, PICT, RLE, XPM, GPR (only available on the apollos, through the program /progs/bit2bit/bin/ppm2gpr) etc.

Example:

```
spam% gs
Initializing... done.
Ghostscript 2.5.2 (9/19/92)
  Copyright (C) 1990, 1992 Aladdin Enterprises, Menlo Park, CA.
  All rights reserved.
Distributed by Free Software Foundation, Inc.
Ghostscript comes with NO WARRANTY: see the file LICENSE for details.
GS> (pstoppm.ps) run
Usage: (file) ppmNrun
   converts file.ps to file.ppm (single page),
     or file.1ppm, file.2ppm, ... (multi page).
   N is # of bits per pixel (1, 8, or 24).
Examples: (golfer) ppm1run   ..or..   (escher) ppm8run
Optional commands you can give first:
   horiz_DPI vert_DPI ppmsetdensity
   (dirname/) ppmsetprefix
   page_num ppmsetfirstpagenumber
GS> 100 120 ppmsetdensity
GS> (/tmp/) ppmsetprefix
GS> (tiger) ppm24run
Writing /tmp/tiger.ppm
GS> quit

spam% ppmtogif < /tmp/tiger.ppm > /tmp/tiger.gif
ppmtogif: computing colormap...
ppmtogif: 39 colors found

spam% ls -l tiger.ps /tmp/tiger.ppm /tmp/tiger.gif
-rw-r--r--   1 oliveria    41796 Apr 21 18:44 /tmp/tiger.gif
-rw-r--r--   1 oliveria  3366054 Apr 21 18:36 /tmp/tiger.ppm
-rw-r--r--   1 oliveria    78519 Apr 21 17:39 tiger.ps
```

Notice that ppm24run creates huge files.

IslandDraw version 3.0 or greater is also able to read (select 'Convert...' under the 'File' menu) single-page postscript files. It can also save the postscript file in a variety of formats.

SYSTEM FILES

gs_init.ps Ghostscript reads this automatically when it starts up. It contains definitions of many standard procedures and initialization for a wide variety of things.

gs_fonts.ps Font initialization. gs_init.ps reads this in. It initializes Ghostscript's font machinery and provides some utility procedures that work with fonts.

gs_statd.ps gs_init.ps reads this in. It creates a dummy statusdict and some other environmental odds and ends for the benefit of P*stScr*pt files that really want to be printed on a LaserWriter.

gs_2asc.ps This utility file helps extract the ASCII text from PostScript source files. It redefines many operators. For more information, see the comments in the file.

gs_dps1.ps gs_init.ps reads this in if the dps feature is included in the configuration. It provides support for various Display PostScript and Level 2 features.

sym__enc.ps the Symbol encoding, loaded only if referenced.

ART

chess.ps a black-and-white chessboard.

golfer.ps a gray-scale picture of a stylishly dressed woman swinging a golf club.

escher.ps a colored version of a hexagonally symmetric Escher drawing of interlocking butterflies. Can be printed on monochrome devices, with somewhat less dramatic results.

cheq.ps a chessboard "font" used by chess.ps (obtained from the Adobe file server).

snowflak.ps a rectangular grid of intricate colored snowflakes. (Renders very slowly.)

colorcir.ps a set of nested ellipses made up of colored bars.

tiger.ps a dramatic colored picture of a tiger's head.

UTILITIES

bdftops.ps a utility for converting BDF fonts to outline form.

decrypt.ps a utility for decrypting the eexec section of a font.

gslp.ps a utility for doing "line printing" of plain text files.

impath.ps a utility for reconstructing outlines from bitmap images, used by bdftops.

landscap.ps a file that you can put in front of your own files to get them rendered in landscape mode.

pstoppm.ps a utility for rendering PostScript files onto PPM (bitmap) files.

ps2image.ps a utility for converting an arbitrary PostScript file into a PostScript file consisting of just bitmaps, one per page.

wrfont.ps a utility for writing out an unprotected Type 1 font, such as the standard Ghostscript fonts.

ODDS and ENDS

empty.ps an empty file.

lines.ps a test program for line joins and caps.

pcharstr.ps a program to print out the CharStrings and Subrs in a Type 1 font.

ppath.ps a couple of utilities for printing out the current path, for debugging.

prfont.ps a program to print a font catalog.

quit.ps a file containing just the word "quit".

traceop.ps a memory usage monitor for debugging.

type1ops.ps the Type 1 font format opcodes.

unprot.ps a program to disable access checking.

FILES

Fontmap A catalog of fonts known to **gs**. Lists the GhostScript name, corresponding font file name, font encoding and font identification number for each font that **gs** knows.

/usr/um/lib/ghostscript
Initialization files, utilities, sample postscript files

/usr/um/lib/ghostscript/fonts
Ghostscript and Hershey fonts (*.gsf files)

KNOWN PROBLEMS

The interactive interpreter requires that every statement fit on a line, i.e., you can't have an unpaired (or {.

On a MS-DOS system, interrupting Ghostscript by typing ^C doesn't restore the display mode.

The Ghostscript language does not include the following operators of the PostScript language:

resetfile
banddevice, renderbands

The following are not implemented completely:

%statementedit (file name): interpreted as equivalent to %lineedit.

Most of the new color operators, particularly those that support the CMYK color model, are implemented as Ghostscript language procedures, and they essentially emulate CMYK using RGB.

The following operators that expect arrays won't accept packed arrays:

```
definefont (Subrs (type 1 fonts))
setdash (dash pattern)
setcachedevice (bounding box)
makeimagedevice (palette)
```

The file operator only recognizes modes r and w, not the newer modes r+, w+, a, and a+.

Floating point exceptions terminate Ghostscript, rather than producing a rangecheck error.

Some access checks aren't implemented.

copypage does nothing in the MS-DOS implementation, and merely synchronizes the X connection in the Unix implementation. showpage is a procedure that does a copypage and then beeps the bell and waits for the user to hit a key. (copypage does do the right thing for printers.)

strokepath produces incorrect output for dashed lines.

The implementation only looks at the PaintType of the font when doing show, not when doing charpath. Because of this, stroked fonts don't work correctly with charpath.

arcto gives an error for negative radii.

Changing the contents of the Encoding array or the Metrics dictionary of a font dynamically doesn't produce the expected result (may have no effect) if character caching is enabled.

Halftone patterns "flip over" at the 50% coverage point, producing anomalous visual effects on some color devices.

We have not been able to test 2-, 4-, and 16-bit memory devices as thoroughly as 1-, 8-, 24-, and 32-bit devices; please report any problems.

Opening more than one window device at the same time doesn't work. This is the case for both X Windows and Microsoft Windows.

restore doesn't properly undo currentgstate.

copy doesn't handle gstates.

BUGS

See the netnews group ``gnu.ghostscript.bug'' for more information.

SEE ALSO

The PostScript Language Reference Manual, Adobe Systems Incorporated, Second Edition, Addison-Wesley Publishing Company, ISBN 0-201-18127-4, 1990

X(1), ghostview(1), lpr(1), dvips(1), ppm(5), xv(1)