Copyright (C) 1989, 1992 Aladdin Enterprises. All rights reserved.
Distributed by Free Software Foundation, Inc.

This file is part of Ghostscript.

Ghostscript is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY. No author or distributor accepts responsibility
to anyone for the consequences of using it or for whether it serves an
y
particular purpose or works at all, unless he says so in writing. Ref
er
to the Ghostscript General Public License for full details.

Everyone is granted permission to copy, modify and redistribute
Ghostscript, but only under the conditions described in the Ghostscrip
t
General Public License. A copy of this license is supposed to have be
en
given to you along with Ghostscript so you can know your rights and
responsibilities. It should be in a file named COPYING. Among other
things, the copyright notice and this notice must be preserved on all
copies.

- -
-

This file, use.doc, describes how to use the Ghostscript language
interpreter.

For an overview of Ghostscript and a list of the documentation files,
see
README.

****
******** How to use the Ghostscript interpreter ********
****

The file name of the Ghostscript interpreter is gs.exe (MS-DOS and VMS) or
gs (Unix). To run it, you also need some external initialization file s:

gs_*.ps (gs_2asc.ps, gs_dps1.ps,
gs_fonts.ps, gs_init.ps, gs_lev2.ps, gs_statd.ps)
sym__enc.ps
uglyr.gsf
Fontmap

as well as any other fonts from the Ghostscript distribution (.gsf and .pfa files).

To invoke the interpreter, give the command

gs <filename1> ... <filenameN>

The interpreter will read in the files in sequence and execute them. After doing this, it reads further input from the primary input stream (normally the keyboard). Each line (i.e. characters up to a <return>) is
interpreted separately. To exit from the interpreter, type quit<retur n>.
The interpreter also exits gracefully if it encounters end-of-file. Typing the interrupt character, e.g., control-C, is also safe.

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

You can get a help message by invoking Ghostscript with

gs -h

or

gs -?

This message also lists the available devices. For a little more information, a one-line description of each device appears near the beginning of the file devs.mak.

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
(For compatibility with older versions of Ghostscript, -sOUTPUTFILE= also works.) 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 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).

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 en

ds
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 take
n 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 qu
otes
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 the following form:

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

Then load the defaults into the X server:

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

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.

-rnumber
-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 name
s
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

-dASCIIOUT
disables Ghostscript's normal behavior; instead, writes strings to stdout that would have been displayed or printed. For more informatio n,
see the file gs_2asc.ps.

-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
suppresses 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 bypas s
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 -Z switch only applies if the interpreter was built for a debugging configuration (DEBUG=1 or -DDEBUG selected at compile time).

-A Turn on allocator debugging (gs_malloc and gs_free).

-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 = clipping
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