



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'gs.1' command

\$ man gs.1

GS(1) Ghostscript GS(1)

NAME

gs - Ghostscript (PostScript and PDF language interpreter and pre-
viewer)

SYNOPSIS

gs [options] [files] ... (Unix, VMS)
gswin32c [options] [files] ... (MS Windows)
gswin32 [options] [files] ... (MS Windows 3.1)
gsos2 [options] [files] ... (OS/2)

DESCRIPTION

The gs (gswin32c, gswin32, gsos2) command invokes Ghostscript, an interpreter of Adobe Systems' PostScript(tm) and Portable Document Format (PDF) languages. gs reads "files" in sequence and executes them as Ghostscript programs. After doing this, it reads further input from the standard input stream (normally the keyboard), interpreting each line separately and output to an output device (may be a file or an X11 window preview, see below). The interpreter exits gracefully when it encounters the "quit" command (either in a file or from the keyboard), at end-of-file, or at an interrupt signal (such as Control-C at the keyboard).

The interpreter recognizes many option switches, some of which are described below. Please see the usage documentation for complete information. Switches may appear anywhere in the command line and apply to all

files thereafter. Invoking Ghostscript with the `-h` or `-?` switch produces a message which shows several useful switches, all the devices known to that executable, and the search path for fonts; on Unix it also shows the location of detailed documentation.

Ghostscript may be built to use many different output devices. To see which devices your executable includes, run `"gs -h"`.

Unless you specify a particular device, Ghostscript normally opens the first one of those and directs output to it.

If built with X11 support, often the default device is an X11 window (previewer), else ghostscript will typically use the `bbox` device and print on stdout the dimension of the postscript file.

So if the first one in the list is the one you want to use, just issue the command

```
gs myfile.ps
```

You can also check the set of available devices from within Ghostscript: invoke Ghostscript and type

```
devicenames ==
```

but the first device on the resulting list may not be the default device you determine with `"gs -h"`. To specify `"AbcXyz"` as the initial output device, include the switch

```
-sDEVICE=AbcXyz
```

For example, for output to an Epson printer you might use the command

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

The `"-sDEVICE="` switch must precede the first mention of a file to print, and only the switch's first use has any effect.

Finally, you can specify a default device in the environment variable `GS_DEVICE`. The order of precedence for these alternatives from highest to lowest (Ghostscript uses the device defined highest in the list) is:

Some devices can support different resolutions (densities). To specify the resolution on such a printer, use the `"-r"` switch:

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

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

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

and the highest-density (best output quality) mode with

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

If you select a printer as the output device, Ghostscript also allows you to choose where Ghostscript sends the output -- on Unix systems, usually to a temporary file. To send the output to a file "foo.xyz", use the switch

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

You might want to print each page separately. To do this, send the output to a series of files "foo1.xyz, foo2.xyz, ..." using the "-sOutputFile=" switch with "%d" in a filename template:

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

Each resulting file receives one page of output, and the files are numbered in sequence. "%d" is a printf format specification; you can also use a variant like "%02d".

On Unix and MS Windows systems you can also send output to a pipe. For example, to pipe output to the "lpr" command (which, on many Unix systems, directs it to a printer), use the option

```
-sOutputFile=%pipe%lpr
```

Note that the '%' characters need to be doubled on MS Windows to avoid mangling by the command interpreter.

You can also send output to standard output:

```
-sOutputFile=-
```

or

```
-sOutputFile=%stdout%
```

In this case you must also use the -q switch, to prevent Ghostscript from writing messages to standard output.

To select a specific paper size, use the command line switch

```
-sPAPERSIZE=<paper_size>
```

for instance

```
-sPAPERSIZE=a4
```

or

```
-sPAPERSIZE=legal
```

Most ISO and US paper sizes are recognized. See the usage documentation for a full list, or the definitions in the initialization file "gs_statd.ps".

Ghostscript can do many things other than print or view PostScript and PDF files. For example, if you want to know the bounding box of a PostScript (or EPS) file, Ghostscript provides a special "device" that just prints out this information.

For example, using one of the example files distributed with Ghostscript,

```
gs -sDEVICE=bbbox golfer.ps
```

prints out

```
%%BoundingBox: 0 25 583 732
```

```
%%HiResBoundingBox: 0.808497 25.009496 582.994503 731.809445
```

OPTIONS

-- 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 may contain no 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

-P Makes Ghostscript to look first in the current directory for li?

brary files. By default, Ghostscript no longer looks in the current directory, unless, of course, the first explicitly supplied directory is "." in -I. See also the INITIALIZATION FILES section below, and bundled Use.htm for detailed discussion on search paths and how Ghostscript finds files.

-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) 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. If only one number is given, it is used for both X and Y resolutions.

-ldirectories

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

- This is not really a switch, but indicates to Ghostscript that standard input is coming from a file or a pipe and not interactively from the command line. Ghostscript reads from standard input until it reaches end-of-file, executing it like any other file, and then continues with processing the command line. When the command line has been entirely processed, Ghostscript exits rather than going into its interactive mode.

Note that the normal initialization file "gs_init.ps" makes "system?dict" read-only, so the values of names defined with -D, -d, -S, or -s

cannot be changed (although, of course, they can be superseded by definitions in "userdict" or other dictionaries.)

SPECIAL NAMES

-dNOCACHE

Disables character caching. Useful only for debugging.

-dNOBIND

Disables the "bind" operator. Useful only 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.

-dNOPLATFONTS

Disables the use of fonts supplied by the underlying platform (for instance X Windows). This may be needed if the platform fonts look undesirably different from the scalable fonts.

-dSAFER

Restricts file operations the job can perform. Now the default mode of operation.

-dWRITESYSTEMDICT

Leaves "systemdict" writable. This is necessary when running special utility programs, but is strongly discouraged as it bypasses normal Postscript security measures.

-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.

SAFER MODE

The -dSAFER option restricts file system accesses to those files and directories allowed by the relevant environment variables (such as

GS_LIB) or by the command line parameters (see <https://ghostscript.com/doc/current/Use.htm> for details).

SAFER mode is now the default mode of operation. Thus when running programs that need to open files or set restricted parameters you should pass the -dNOSAFER command line option or its synonym -dDELAYSAFER. Running with NOSAFER/DELAYSAFER (as the same suggests) loosens the security and is thus recommended ONLY for debugging or in VERY controlled workflows, and strongly NOT recommended in any other circumstances.

FILES

The locations of many Ghostscript run-time files are compiled into the executable when it is built. On Unix these are typically based in /usr/local, but this may be different on your system. Under DOS they are typically based in C:\GS, but may be elsewhere, especially if you install Ghostscript with GSview. Run "gs -h" to find the location of Ghostscript documentation on your system, from which you can get more details.

/usr/local/share/ghostscript/#.##/*

Startup files, utilities, and basic font definitions

/usr/local/share/ghostscript/fonts/*

More font definitions

/usr/local/share/ghostscript/#.##/examples/*

Ghostscript demonstration files

/usr/local/share/ghostscript/#.##/doc/*

Diverse document files

INITIALIZATION FILES

When looking for the initialization files "gs_*.ps", the files related to fonts, or the file for the "run" operator, Ghostscript first tries to open the file with the name as given, using the current working directory if no directory is specified. If this fails, and the file name doesn't specify an explicit directory or drive (for instance, doesn't contain "/" on Unix systems or "\" on MS Windows systems), Ghostscript tries directories in this order:

1. the directories specified by the -I switches in the command line

(see below), if any;

2. the directories specified by the GS_LIB environment variable, if any;
3. the directories specified by the GS_LIB_DEFAULT macro in the Ghostscript makefile when the executable was built. When gs is built on Unix, GS_LIB_DEFAULT is usually "/usr/local/share/ghostscript/###:/usr/local/share/ghostscript/fonts" where "###" represents the Ghostscript version number.

Each of these (GS_LIB_DEFAULT, GS_LIB, and -I parameter) may be either a single directory or a list of directories separated by ":".

ENVIRONMENT

GS_OPTIONS

String of options to be processed before the command line options

GS_DEVICE

Used to specify an output device

GS_FONTPATH

Path names used to search for fonts

GS_LIB Path names for initialization files and fonts

TEMP Where temporary files are made

X RESOURCES

Ghostscript, or more properly the X11 display device, 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 is NULL).

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).

useBackingPixmap

Determines whether backing store is to be used for saving display window (default = true).

See the usage document for a more complete list of resources. To set these resources on Unix, put them in a file such as "~/.Xresources" in the following form:

```
Ghostscript*geometry: 612x792-0+0
```

```
Ghostscript*xResolution: 72
```

```
Ghostscript*yResolution: 72
```

Then merge these resources into the X server's resource database:

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

SEE ALSO

The various Ghostscript document files (above), especially Use.htm.

BUGS

See <http://bugs.ghostscript.com/> and the Usenet news group comp.lang.postscript.

VERSION

This document was last revised for Ghostscript version 9.54.0.

AUTHOR

Artifex Software, Inc. are the primary maintainers of Ghostscript.

Russell J. Lang, gsview@ghostgum.com.au, is the author of most of the MS Windows code in Ghostscript.

9.54.0

30 March 2021

GS(1)