- -

This file, language.doc, describes the Ghostscript language.

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

The Ghostscript interpreter, except as noted below, is intended to execute properly any source program written in a language defined by reference to the December 1990 printing of the PostScript Language Reference Manual (Second Edition) published by Addison-Wesley (ISBN 0-201-18127-4). The Ghostscript language includes the following elements of the PostScript (TM) language:

- The full PostScript Level 1 language, as also defined in the first edition of the PostScript Language Reference Manual, ISBN 0-201-10174-2, Addison-Wesley, 1985.

- The CMYK color, file system, version 25.0 language, and miscellaneous additions listed in sections A.1.4, A.1.6, A.1.7, and A.1.8 of the Second Edition respectively, including allowing a string operand for the 'status' operator;

- The Display PostScript extensions listed in section A.1.3 of the Second Edition, but excluding the operators listed in section A.1.2. These facilities are only available if the dps feature or the level2 feature was selected at the time that Ghostscript was compiled and linked.

- The composite font extensions listed in section A.1.5 of the Second Edition, but not the ability to handle Type 0 fonts. These facilities are only available if the compfont feature or the level2 feature was selected at the time that Ghostscript was compiled and linked.

- A subset of the other PostScript Level 2 operators and facilities listed in section A.1.1 of the Second Edition, identified below. These facilities are only available if the level2 feature was selected at the time that Ghostscript was compiled and linked.

Ghostscript also includes a number of operators defined below that are not

in the PostScript language.

Stub facilities

The following operators, while provided in the current release, have only a partial or dummy implementation.

Form and pattern operators:
execform

Graphics state operators:
currentblackgeneration, currentcolorscreen,
currenthalftonephase, currentundercolorremoval,
setblackgeneration, setcolorscreen,
sethalftonephase, setundercolorremoval,
currenthalftone, sethalftone

Interpreter parameter operators:
currentsystemparams, currentuserparams
setsystemparams, setucacheparams, setuserparams
ucachestatus

Path construction operators:
ucache

Virtual memory operators:
currentshared, scheck, setshared, setvmthreshold, shareddict,
SharedFontDirectory, vmreclaim

Miscellaneous operators:
serialnumber

Unclassified Level 2 operators

Ghostscript provides the following PostScript Level 2 operators listed in section A.1.1 of the Second Edition and not listed in any of the other A.1 sections.

File operators:
filter (all but DCTEncode/Decode)

Resource operators:
defineresource, findresource, resourceforall, resourcestatus,
undefineresource

Character and font operators:
cshow, xshow, yshow, xyshow
glyphshow
findencoding, rootfont, setcachedevice2
(WMode is supported, but composite fonts are not)

Graphics state operators:
currentcolor, currentcolorspace, setcolor, setcolorspace
(except for Indexed with procedure, true Separation, and Pattern)
currentcolorrendering, setcolorrendering

Miscellaneous operators:

languagelevel

In addition, Ghostscript supports the following Level 2 facilities:

- Use of a dictionary with the image and imagemask operators;

- Use of a string or a file as data source with the image, imagemask, and colorimage operators.

When the Level 2 features are present, Ghostscript also supports the following operator:

<1 or 2> .setlanguagelevel -
Set the current language level to Level 1 or Level 2. When the language level is 1, no Level 2 facilities are provided.

Unimplemented Level 2 facilities

Ghostscript currently does not implement the following Level 2 operators and variables:

currentdevparams, setdevparams
GlobalFontDirectory
makepattern
realtime
currentpagedevice, setpagedevice
startjob

Ghostscript currently does not implement the following Level 2 facilities not enumerated above:

garbage collection
global and local VM
page devices
job control
halftone dictionaries
user names

Ghostscript-specific additions

Miscellaneous

^Z is counted as whitespace.

run can take either a string or a file as its argument. In the former case, it uses findlibfile to open the file (searching directories as needed). In the latter case, it just runs the file, closing it at the end, and trapping errors just as for the string case.

Mathematical operators

<number> arccos <number>
Computes the arc cosine of a number between -1 and 1.

<number> arcsin <number>
Computes the arc sine of a number between -1 and 1.

String operators

<state> <fromString> <toString> type1encrypt <newState> <toSubstring>
Encrypts fromString according to the algorithm for Adobe Type 1 fonts, writing the result into toString. toString must be at least as long as fromString or a rangecheck error occurs. state is the initial state of the encryption algorithm (a 16-bit non-negative integer); newState is the new state of the algorithm.

<state> <fromString> <toString> type1decrypt <newState> <toSubstring>
Decrypts fromString according to the algorithm for Adobe Type 1 fonts, writing the result into toString. Other specifications are as for type1encrypt.

Relational operators

<number|string> <number|string> max <number|string>
Returns the larger of two numbers or strings.

<number|string> <number|string> min <number|string>
Returns the smaller of two numbers or strings.

File operators

<string> findlibfile <foundstring> <file> true or <string> false
Opens the file of the given name for reading. If the file cannot be opened using the supplied name, searches through directories as described in use.doc. If the search fails, findlibfile simply pushes false on the stack and returns, rather than causing an error.

<file> <integer> unread -
Pushes back the last-read character onto the front of the file. If the file is only open for writing, or if the integer argument is not the same as the last character read from the file, causes an ioerror error. May also cause an ioerror if the last operation on the file was not a reading operation.

<file> <device> writeppmfile -
Writes the contents of the device, which must be an image device, onto the file, in Portable PixMap (ppm) format. Does not close the file.

Path operators

<x> <y> <width> <height> rectappend -
<numarray> rectappend -
<numstring> rectappend -
Appends a rectangle or rectangles to the current path, in the same manner as rectfill, rectclip, etc. Only defined if the dps option is selected.

Filters

Ghostscript supports all standard filters except DCTEncode and DCTDecode. Ghostscript does not support the use of a procedure as a data source or sink, only a file or a string. In addition, Ghostscript supports two non-standard filters:

`<file|string> <seed_integer> /eexecDecode filter <file>`
: Creates a filter for decrypting data that has been encrypted using eexec encryption as described in the Adobe Type 1 Font Format documentation. The seed_integer must be 55665 for proper operation.

`<file|string> <hex_boolean> /PFBDecode filter <file>`
: Creates a filter that decodes data in .PFB format, the usual semi-binary representation for Type 1 font files on IBM PC and compatible systems. If hex_boolean is true, binary packets are converted to hex; if false, binary packets are not converted.

Miscellaneous operators

`- currenttime <number>`
: Returns the current value of a continuously-running timer, in minutes. The initial value of this timer is undefined.

`<string> getenv <string> true or false`
: Looks up a name in the shell environment. If the name is found, returns the corresponding value and true; if the name is not found, returns false.

`<name> <array> makeoperator <operator>`
: Constructs and returns a new operator that is actually the given procedure in disguise. The name is only used for printing. The operator has the executable attribute.

`<string> <boolean> setdebug -`
: If the Ghostscript interpreter was built with the DEBUG flag set, sets or resets any subset of the debugging flags normally controlled by -Z in the command line. Has no effect otherwise.

`- oserrno <errno>`
: Returns the error code for the most recent OS error.

`- oserror <string>`
: Returns the error string for the most recent OS error.

Device operators

`<device> copydevice <device>`
: Copies a device.

`<index> getdevice <device>`
: Returns a device from the set of devices known to the system. The first device, which is default, is numbered

0. If the index is out of range, causes a rangecheck error.

<matrix> <width> <height> <palette> makeimagedevice <device>
Makes a new device that accumulates an image in memory. matrix is the initial transformation matrix: it must be orthogonal (i.e., [a 0 0 b x y] or [0 a b 0 x y]). palette is a string of 2^N or 3*2^N elements, specifying how the 2^N possible pixel values will be interpreted. Each element is interpreted as a gray value, or as RGB values, multiplied by 255. For example, if you want a monochrome image for which 0=white and 1=black, the palette should be <ff 00>; if you want a 3-bit deep image with just the primary colors and their complements (ignoring the fact that 3-bit images are not supported), the palette might be <000000 0000ff 00ff00 00ffff ff0000 ff00ff ffff00 ffffff>. At present, the palette must contain exactly 2, 4, 16, or 256 entries, and must contain an entry for black and an entry for white; if it contains any entries that aren't black, white, or gray, it must contain at least the six primary colors (red, green, blue, and their complements cyan, magenta, and yellow); aside from this, its contents are arbitrary.
Alternatively, palette can be null. This is interpreted as 24-bit-per-pixel color, where the four bytes of each pixel are respectively R, G, and B.
Note that one can also make an image device (with the same palette as an existing image device) by copying a device using the copydevice operator.

<device> <index> <string> copyscanlines <substring>
Copies one or more scan lines from an image device into a string, starting at a given scan line in the image. The data is in the same format as for the image operator. Error if the device is not an image device or if the string is too small to hold at least one complete scan line. Always copies an integral number of scan lines.

<device> setdevice -
Sets the current device to the specified device. Also resets the transformation and clipping path to the initial values for the device.

- currentdevice <device>
Gets the current device from the graphics state.

<device> devicename <string>
Gets the name of a device.

<device> <matrix> deviceinitialmatrix <matrix>
Gets the initial matrix of a device, i.e., the one that defaultmatrix would return if the device were the current device.

<device> getdeviceprops <mark> <name1> <value1> ... <namen> <valuen>
Gets all the properties of a device. Currently defined names and values for all devices are:

```
                HWResolution [<float> <float>]
                      X and Y resolution in pixels/inch.
                HWSize [<integer> <integer>]
                      X and Y size in pixels.
                InitialMatrix [<6 floats>]
                      Initial transformation matrix.
                Name <string>
                      Read-only.  The device name.
          For printers, the following are also defined:
                BufferSpace <integer>
                      Buffer space for band lists, if the bitmap
                        is too big to fit in RAM.
                MaxBitmap <integer>
                      Maximum space for a full bitmap in RAM.
                OutputFile <string>
                      () means send to printer directly,
                        otherwise specifies the file name for
                        output; a %d is replaced by the page #;
                        on Unix systems, (|command) writes to a pipe
                PageCount <integer>
                      Read-only.  Counts the number of pages
                        printed on the device.

    <mark> <name1> <value1> ... <namen> <valuen> <device>
        putdeviceprops <device>
          Sets properties of a device.  May cause undefined,
            typecheck, rangecheck, or limitcheck errors.

    - flushpage -
          On displays, flushes any buffered output, so that it
            is guaranteed to show up on the screen; on printers,
            has no effect.

Character operators
-------------------

    <string> .type1addpath -
    <string> <lsbx> <lsby> .type1addpath -
          Adds the description of a character to the current
            path.  The string argument is a scalable
            description encoded in Adobe Type 1 format.  This
            operator, like setcharwidth and setcachedevice, is
            only valid in the context of a show operator.  It
            uses information from the current font, in addition
            to the argument(s).
          The optional lsbx and lsby arguments are left side
            bearing values that override the ones in the
            character outline.

    <font> <char> Type1BuildChar -
          This is not a new operator: rather, it is a name known
            specially to the interpreter.  Whenever the interpreter
            needs to render a character (during a ...show,
            stringwidth, or charpath), it looks up the name
            BuildChar in the font dictionary to find a procedure to
            run.  If it does not find this name, and if the FontType
            is 1, the interpreter instead uses the value (looked up
            on the dictionary stack in the usual way) of the name
            Type1BuildChar.
```

The standard definition of Type1BuildChar is in gs_fonts.ps.
Users should not need to redefine Type1BuildChar, except
perhaps for tracing or debugging.

- -

PostScript is a trademark of Adobe Systems, Incorporated.