NAME

postio – serial interface for PostScript printers

SYNOPSIS

postio **–l** *line* [**–D**] [**–i**] [**–q**] [**–t**] [**–S**] [**–b** *speed*] [**–B** *num*] [**–L** *file*]
[**–P** *string*] [**–R** *num*] [*file* ...]

/usr/lib/lp/postscript/postio

DESCRIPTION

postio sends *files* to the PostScript printer attached to *line.* If no *files* are specified the standard input is sent.

OPTIONS

The first group of *options* should be sufficient for most applications:

–D Enable debug mode. Guarantees that everything read on *line* will be added to the log file (standard error by default).

–q Prevents status queries while *files* are being sent to the printer. When status queries are disabled a dummy message is appended to the log file before each block is transmitted.

–b *speed* Transmit data over *line* at baud rate *speed*. Recognized baud rates are 1200, 2400, 4800, 9600, and 19200. The default *speed* is **9600** baud.

–B *num* Set the internal buffer size for reading and writing *files* to *num* bytes. By default *num* is **2048** bytes.

–l *line* Connect to the printer attached to *line*. In most cases there is no default and **postio** must be able to read and write *line*. If the *line* does not begin with a **/** it may be treated as a Datakit destination.

–L *file* Data received on *line* gets put in *file*. The default log *file* is standard error. Printer or status messages that don't show a change in state are not normally written to *file* but can be forced out using the **–D** option.

–P *string* Send *string* to the printer before any of the input files. The default *string* is simple PostScript code that disables timeouts.

–R *num* Run *postio* as a single process if *num* is 1 or as separate read and write processes if *num* is 2. By default **postio** runs as a single process.

The next two *options* are provided for users who expect to run **postio** on their own. Neither is suitable for use in spooler interface programs:

–i Run the program in interactive mode. Any *files* are sent first and followed by the standard input. Forces separate read and write processes and overrides many other options. To exit interactive mode use your interrupt or quit character. To get a friendly interactive connection with the printer type **executive** on a line by itself.

–t Data received on *line* and not recognized as printer or status information is written to the standard output. Forces separate read and write processes. Convenient if you have a PostScript program that will be returning useful data to the host.

The last option is not generally recommended and should only be used if all else fails to provide a reliable connection:

–S Slow the transmission of data to the printer. Severely limits throughput, runs as a single process, disables the **–q** option, limits the internal buffer size to 1024 bytes, can use an excessive amount of CPU time, and does nothing in interactive mode.

The best performance will usually be obtained by using a large internal buffer (the **–B** option) and by running the program as separate read and write processes (the **–R 2** option). Inability to fork the additional process causes **postio** to continue as a single read/write process. When one process is used, only data sent to the printer is flow controlled.

The *options* are not all mutually exclusive. The **–i** option always wins, selecting its own settings for whatever is needed to run interactive mode, independent of anything else found on the command line. Interactive mode runs as separate read and write processes and few of the other *options* accomplish anything in the presence of the **–i** option. The **–t** option needs a reliable two way connection to the printer and therefore tries to force separate read and write processes. The **–S** option relies on the status query mechanism, so **–q** is disabled and the program runs as a single process.

In most cases **postio** starts by making a connection to *line* and then attempts to force the printer into the IDLE state by sending an appropriate sequence of **ˆT** (status query), **ˆC** (interrupt), and **ˆD** (end of job) characters. When the printer goes IDLE, *files* are transmitted along with an occasional **ˆT** (unless the **–q** option was used). After all the *files* are sent the program waits until it's reasonably sure the job is complete. Printer generated error messages received at any time except while establishing the initial connection (or when running interactive mode) cause **postio** to exit with a non-zero status. In addition to being added to the log file, printer error messages are also echoed to standard error.

EXAMPLES

Run as a single process at 9600 baud and send *file1* and *file2* to the printer attached to **/dev/tty01**:

example% postio –l /dev/tty01 *file1 file2*

Same as above except two processes are used, the internal buffer is set to 4096 bytes, and data returned by the printer gets put in file *log*:

example% postio –R 2 –B 4096 –l/dev/tty01 –L *log file1 file2*

Establish an interactive connection with the printer at Datakit destination *my/printer*:

example% postio –i –l *my/printer*

Send file program to the printer connected to **/dev/tty22**, recover any data in file results, and put log messages in file *log*:

example% postio –t –l /dev/tty22 –L *log program* **>***results*

SEE ALSO

download(1), **dpost**(1), **postdaisy**(1), **postdmd**(1), **postmd**(1), **postprint**(1), **postreverse**(1), **posttek**(1)

DIAGNOSTICS

An exit status of **0** is returned if the files ran successfully. System errors (such as an inability to open the line) set the low order bit in the exit status, while PostScript errors set bit 1. An exit status of **2** usually means the printer detected a PostScript error in the input *files*.

NOTES

The input *files* are handled as a single PostScript job. Sending several different jobs, each with their own internal end of job mark (**^D**) is not guaranteed to work properly. **postio** may quit before all the jobs have completed and could be restarted before the last one finishes.

All the capabilities described above may not be available on every machine or even across the different versions of the UNIX system that are currently supported by the program.

There may be no default *line*, so using the **−l** option is strongly recommended. If omitted, **postio** may attempt to connect to the printer using the standard output. If Datakit is involved, the **−b** option may be ineffective and attempts by **postio** to impose flow control over data in both directions may not work. The **−q** option can help if the printer is connected to RADIAN. The **−S** option is not generally recommended and should be used only if all other attempts to establish a reliable connection fail.