

NAME

printcap - printer capability data base

SYNOPSIS

printcap

DESCRIPTION

The **Printcap** function is a simplified version of the `termcap(5)` data base used to describe line printers. The spooling system accesses the **printcap** file every time it is used, allowing dynamic addition and deletion of printers. Each entry in the data base is used to describe one printer. This data base may not be substituted for, as is possible for `termcap(5)`, because it may allow accounting to be bypassed.

The default printer is normally *lp*, though the environment variable `PRINTER` may be used to override this. Each spooling utility supports an option, **-P printer**, to allow explicit naming of a destination printer.

Refer to the *4.3 BSD Line Printer Spooler Manual* for a complete discussion on how to setup the database for a given printer.

CAPABILITIES

Refer to `termcap(5)` for a description of the file layout.

Name	Type	Default	Description
af	str	NULL	name of accounting file
br	num	none	if <i>lp</i> is a tty, set the baud rate (<code>ioctl(2)</code> call)
cf	str	NULL	cifplot data filter
ct	num	120	TCP connection timeout in seconds
df	str	NULL	tex data filter (DVI format)
du	num	1	UID to run daemon as
ff	str	'\f'	string to send for a form feed
fo	bool	false	print a form feed when device is opened
gf	str	NULL	graph data filter (<code>plot(3)</code> format)
hl	bool	false	print the burst header page last
ic	bool	false	driver supports (non standard) <code>ioctl</code> to indent printout
if	str	NULL	name of text filter which does accounting
lf	str	<i>/dev/console</i>	error logging file name
lo	str	<i>lock</i>	name of lock file
lp	str	<i>/dev/lp</i>	device name to open for output, or <i>port@machine</i> to open a TCP socket
mc	num	0	maximum number of copies which can be requested on

			lpr(1), zero = unlimited
ms	str	NULL	if lp is a tty, a comma-separated, stty(1)-like list describing the tty modes
mx	num	0	maximum file size (in BUFSIZ blocks), zero = unlimited
nd	str	NULL	next directory for list of queues (unimplemented)
nf	str	NULL	ditroff data filter (device independent troff)
of	str	NULL	name of output filtering program
pc	num	200	price per foot or page in hundredths of cents
pl	num	66	page length (in lines)
pw	num	132	page width (in characters)
px	num	0	page width in pixels (horizontal)
py	num	0	page length in pixels (vertical)
rc	bool	false	when sending to a remote host, resend copies (see below)
rf	str	NULL	filter for printing FORTRAN style text files
rg	str	NULL	restricted group. Only members of group allowed access
rm	str	NULL	machine name for remote printer
rp	str	<i>lp</i>	remote printer name argument
rs	bool	false	restrict remote users to those with local accounts
rw	bool	false	open the printer device for reading and writing
sb	bool	false	short banner (one line only)
sc	bool	false	suppress multiple copies
sd	str	<i>/var/spool/lpd</i>	spool directory
sf	bool	false	suppress form feeds
sh	bool	false	suppress printing of burst page header
sr	str	NULL	file name to hold statistics of each datafile as it is received
ss	str	NULL	file name to hold statistics of each datafile as it is sent
st	str	<i>status</i>	status file name
tf	str	NULL	troff data filter (cat phototypesetter)
tr	str	NULL	trailer string to print when queue empties
vf	str	NULL	raster image filter

Each two-letter capability has a human-readable alternate name.

Short form	Long form
af	acct.file
br	tty.rate
cf	filt.cifplot
ct	remote.timeout
df	filt.dvi
du	daemon.user

ff	job.formfeed
fo	job.topofform
gf	filt.plot
hl	banner.last
if	filt.input
lf	spool.log
lo	spool.lock
lp	tty.device
mc	max.copies
ms	tty.mode
mx	max.blocks
nf	filt.ditroff
of	filt.output
pc	acct.price
pl	page.length
pw	page.width
px	page.pwidth
py	page.plength
rc	remote.resend_copies
rf	filt.fortran
rg	daemon.restrictgrp
rm	remote.host
rp	remote.queue
rs	daemon.restricted
rw	tty.rw
sb	banner.short
sc	job.no_copies
sd	spool.dir
sf	job.no_formfeed
sh	banner.disable
sr	stat.recv
ss	stat.send
st	spool.status
tf	filt.troff
tr	job.trailer
vf	filt.raster

If the local line printer driver supports indentation, the daemon must understand how to invoke it.

FILTERS

The `lpd(8)` daemon creates a pipeline of *filters* to process files for various printer types. The filters selected depend on the flags passed to `lpr(1)`. The pipeline set up is:

<code>p</code>	<code>pr </code>	regular text + <code>pr(1)</code>
<code>none</code>	<code>if</code>	regular text
<code>c</code>	<code>cf</code>	cifplot
<code>d</code>	<code>df</code>	DVI (<code>tex</code>)
<code>g</code>	<code>gf</code>	<code>plot(3)</code>
<code>n</code>	<code>nf</code>	<code>ditroff</code>
<code>f</code>	<code>rf</code>	Fortran
<code>t</code>	<code>tf</code>	<code>troff</code>
<code>v</code>	<code>vf</code>	raster image

The **if** filter is invoked with arguments:

if [-c] -wwidth -llength -iindent -n login -h host acct-file

The **-c** flag is passed only if the **-l** flag (pass control characters literally) is specified to `lpr(1)`. The *Width* function and *length* specify the page width and length (from **pw** and **pl** respectively) in characters. The **-n** and **-h** parameters specify the login name and host name of the owner of the job respectively. The *Acct-file* function is passed from the **af printcap** entry.

If no **if** is specified, **of** is used instead, with the distinction that **of** is opened only once, while **if** is opened for every individual job. Thus, **if** is better suited to performing accounting. The **of** is only given the *width* and *length* flags.

All other filters are called as:

filter -xwidth -ylength -n login -h host acct-file

where *width* and *length* are represented in pixels, specified by the **px** and **py** entries respectively.

All filters take *stdin* as the file, *stdout* as the printer, may log either to *stderr* or using `syslog(3)`, and must not ignore SIGINT.

REMOTE PRINTING

When printing to a remote printer using **rm**, it is possible to use either **if** or **of**. If both are specified, **of** is ignored. Both filters behave the same except that they are passed different arguments as above. Specifically, the output filter is terminated and restarted for each file transmitted. This is necessary in order to pass the resulting size to the remote `lpd(8)`.

If the **-p** flag was passed to `lpr(1)`, `pr(1)` is not executed locally, but is requested of the remote `lpd(8)`. Any input filtering via **if** will therefore happen before `pr(1)` is executed rather than afterwards.

There are some models of network printers which accept jobs from `lpd(8)`, but they ignore the control file for a job and simply print each data file as it arrives at the printer. One side-effect of this behavior is that the printer will ignore any request for multiple copies as given with the **-#** flag on the `lpr(1)` command. The **rc** entry will cause `lpd(8)` to resend each data file for each copy that the user originally requested. Note that the **rc** entry should only be specified on hosts which send jobs directly to the printer.

If **lp** is specified as *port@machine* (and **rm** is not in use), print data will be sent directly to the given *port* on the given *machine*.

TRANSFER STATISTICS

When a print job is transferred to a remote machine (which might be another unix box, or may be a network printer), it may be useful to keep statistics on each transfer. The **sr** and **ss** options indicate filenames that `lpd` should use to store such statistics. A statistics line is written for each datafile of a job as the file is successfully transferred. The format of the line is the same for both the sending and receiving side of a transfer.

Statistics on datafiles being received would be used on a print server, if you are interested in network performance between a variety of machines which are sending jobs to that print server. The print server could collect statistics on the speed of each print job as it arrived on the server.

Statistics on datafiles being sent might be used as a minimal accounting record, when you want to know who sent which jobs to a remote printer, when they were sent, and how large (in bytes) the files were. This will not give include any idea of how many pages were printed, because there is no standard way to get that information back from a remote (network) printer in this case.

LOGGING

Error messages generated by the line printer programs themselves (that is, the `lpd(8)` and related programs) are logged by `syslog(3)` using the LPR facility. Messages printed on *stderr* of one of the filters are sent to the corresponding **lf** file. The filters may, of course, use `syslogd(8)` themselves.

Error messages sent to the console have a carriage return and a line feed appended to them, rather than just a line feed.

SEE ALSO

`lpq(1)`, `lpr(1)`, `lprm(1)`, `hosts.lpd(5)`, `termcap(5)`, `chkprintcap(8)`, `lpc(8)`, `lpd(8)`, `pac(8)`

4.3 BSD Line Printer Spooler Manual.

HISTORY

The **printcap** file format appeared in 4.2BSD.