NAME

pr — paginate, columnate, and number files

SYNOPSIS

pr [**-c**|**-v**] [**-dpr**] [**-t**|**-T**|**-F**|**-f**] [**-o** *indent*] [**-**{**e**|**i**|**n**}[*tab*][*width*]]...
[**-N** *first-line*] [**-h** *header*] [**-l** *lines*] [**+***first*[**:***last*]] [**-W** *cols*] [*file*]...
pr [**-c**|**-v**] [**-dpr**] [**-t**|**-T**|**-F**|**-f**] [**-o** *indent*] [**-**{**e**|**i**|**n**}[*tab*][*width*]]...
[**-N** *first-line*] [**-h** *header*] [**-l** *lines*] [**+***first*[**:***last*]] **-***columns* [**-a**]
[**-J**|**-**{**w**|**W**} *cols*] [**-s**[*tab*]|**-S**[*separator*]] [*file*]...
pr [**-c**|**-v**] [**-dpr**] [**-t**|**-T**|**-F**|**-f**] [**-o** *indent*] [**-**{**e**|**i**|**n**}[*tab*][*width*]]...
[**-N** *first-line*] [**-h** *header*] [**-l** *lines*] [**+***first*[**:***last*]] **-m** [**-J**|**-**{**w**|**W**} *cols*]
[**-s**[*tab*]|**-S**[*separator*]] [*file*]...

DESCRIPTION

Copies each *file* (standard input stream if "**-**", the default) to the standard output stream, while:

unless **-t**|**-T**,	adding a dated file-name-identifying (unless **-h**) header and ejecting (**72**-line by default) pages by filling them with empty lines and a trailer,
or with **-F**|**-f**,	ejecting the page with a form feed (**0xC**) instead,
if writing to a teletype and **-p** or **-f**,	waiting for a line from the controlling teletype before each (the first) page,
with **+***first*[**:***last*],	writing only the [*first*, *last*] pages of each *file*,
if *columns* > *1*,	filling lines into vertically-aligned columns, vertically or horizontally with **-a**,
and with **-s**,	pasting them with a fixed separator instead
if *columns* > *1* or **-W** and not **-J**,	truncating the printable output (and each column) to the page width (**72** by default),
if **-e**,	expand(1)ing *tab*s in input files to their original-equivalent spaces,
if **-i**,	unexpand(1)ing spaces in the output to *tab*s,
if **-n**,	numbering the lines of each file,
if **-o**,	offsetting each output line by *indent* columns,
if **-c**|**-v**,	escaping non-tab unprintable or invalid input characters, and
if **-d**,	writing an empty line after each real one,

Or, with **-m**, each *file* is its own column, and lines are numbered globally.

Form feeds (**0xC**) in an input line end the current page at the form feed, feed the page, and start the first line of the subsequent page after the form feed. In **-m** mode, this applies to each file independently. **-T** removes form feeds from the input instead.

The heading takes the form of two empty lines each surrounding

```
    2023-05-06 00:27        README Page 4
```

with either a tab (by default) or a space (**-e**, **-i**, or truncating to less than *40* columns) between the fields (but see **STANDARDS**). Not affected by truncation.
If *header* was specified, it takes the central field; otherwise, unless **-m** or reading the standard input stream, it's *file*; otherwise it's blank. Similarly, the date is the start of formatting with **-m**; the time of "opening" if reading the standard input stream; otherwise the modification time (*st_mtim*) of *file*.

Unless **-F**|**-f**, the trailer consists of five empty lines (plus however-many are needed to space to the end of the page). Otherwise, it's a single form feed (**0xC**).

Truncation, width, and column spacing form a very unfortunate hierarchy:

if **-J** and **-S**	truncation off, columns pasted directly with just **-S**;
if **-J** and **-s**	likewise, but with just **-s**;
if **-J**	likewise, but with just a tab (**0x9**);

if **-W** and **-s**	truncate to *cols*, separate columns with **-s**,
if **-W**	truncate to *cols*, align columns;
if one output column	don't truncate;
if **-s** and **-w**	truncate to *cols*, separate columns with **-s**;
if **-s** and not **-w**	truncate to **512** columns, separate columns with **-s**;
if **-w**	truncate to *cols*, align columns;
otherwise	truncate to **72** columns, align columns.

When aligning columns they're padded to nominal width with spaces, then separated by **-s**|**-S** if specified, otherwise a space (**0x20**). Separators for unfilled columns are omitted. The final of **-s**|**-S** applies.

OPTIONS

Note that some of these take optional arguments, and those need to be welded to the flag. Thus, arguments for all short flags are noted explicitly.

+ is used as a flag delimiter, just like **-** usually: this also needs **--**-protection.

-J, **--join-lines**; **-w** *cols*, **--width**=*cols*; **-W** *cols*, **--page-width**=*cols*
-s[*tab*], **--separator**[=*tab*]; **-S**[*separator*], **--sep-string**[=*separator*]
See above. *tab* defaults to a tab (**0x9**); *separator* to empty.

-c, **--show-control-chars**
Write bytes of invalid, incomplete, and non-printable non-tab (**0x9**) characters as **^***X* (cat(1) **-v** form) if smaller than **0x80**, and as *OOO* (octal form) otherwise.

-v, **--show-nonprinting** Like **-c**, but always use the *OOO* form. Overrides **-c**.

-d, **--double-space** Write an empty line after each body line.

-p If the stadard output stream is a teletype write a bell (**0x7**) to the standard error stream, then read (and discard) a line from /dev/tty before the start of each page. No effect otherwise.

-t, **--omit-header** Don't write the heading and trailer, thus leaving the entire length of the page available for *file*s.

-T, **--omit-pagination** **-t** and strip form feeds (**0xC**) from *file*s.

-r, **--no-file-warnings** Don't write errors about failures to open *file*s or /dev/tty to the standard error stream.

-F, **--form-feed** Write a form feed (**0xC**) instead of new-lines to end of page. The trailer is still considered to be **5** lines tall — cf. **-t**.

-f **-F**; also **-p** (sans the bell) but only for the first page written.

-o *indent*, **--indent**=*indent*
Prepend each non-empty output line with *indent* spaces. These do not factor into the line width.

-e[*tab*][*width*], **--expand-tabs**, **--expand-tabs**=[*tab*][*width*] (cannot be empty)
Expand *tab*s in the input *files* to their equivalent number of spaces if spaced every *width* columns. As always in **pr**, non-printable characters are ignored for the width calculations. *tab* may be any non-empty string, the two arguments are split on the first digit, and empty ones are missing. *tab* of tab (**0x9**) and *width* of **8** are the default. Implied if writing multiple columns.

-i[*tab*][*width*], **--output-tabs**, **--output-tabs**=[*tab*][*width*]
Unexpand spaces before *width*-spaced columns in the output to the appropriate number of *tab*s. Tokenisation as above, but in the last form an empty value is taken to be an empty *tab*. Implied if writing multiple columns.

-n[*tab*][*width*], **--number-lines**, **--number-lines**=[*tab*][*width*]
Prepend each input (with **-m**, each output) line with a *width*-space-padded line number in *file* and *tab*. If the number would need more than *width* digits, the final *width* ones are taken. Tokenisation as in **-i**, but the default *width* is **5**.

-N *first-line*, **--first-line-number**=*first-line*
The first line number of each file is *first-line* instead of **1**.

-h *header*, **--header**=*header*
Use *header* as the file-name in the heading instead of *file* (or the empty string if **-m** or reading the standard input stream).

-l *lines*, **--length**=*lines*
Each page has *lines*. Implies **-t** if smaller than heading + footer + one output line would need (≤ **10**). Defaults to **66**.

+*first*[**:***last*], **--pages**=*first*[**:***last*]
Discard pages that fall outside [*first*, *last*] or [*first*, ∞) from each *file*. The **+** form is only accepted once (or not at all if preceded by **--pages**).

-*columns*, **--columns**=*columns*
Columnate each *file* to *columns*. In the **-** form, consecutive decimal digits are accepted (and may be part of a longer flag slug, i.e. **-d10a** works).

-a, **--across** By default, columns are filled down the page. Fill them rightward instead.

-m, **--merge** Each *file* is a single column in a single output file; form feeds (**0xC**) affect each *file* individually. *file*s that couldn't be opened don't take up a column.

SIGNALS

If the standard output stream is a teletype, then

SIGINT Write accumulated errors and terminate by re-raising SIGINT.

otherwise default.

EXIT STATUS

1 if a *file* didn't exist, or truncating and there's too many too-wide columns to guarantee one column for each.

ERRORS

If the standard output stream is a teletype, errors are only written to the standard error stream after formatting finishes. Otherwise, they're written as discovered as usual.

SEE ALSO

Please use any of cat(1), column(1), cut(1), expand(1), fold(1), nl(1), or paste(1) instead of this. They may even do what you want!

pr(1), stat(2), isprint(3), inode(7)

BUGS

Probably should center the heading fields on the page, but the present solution is already too much drip.

STANDARDS

Conforms to IEEE Std 1003.1-2024 ("POSIX.1"). **-f** is shaded XSI. **-cvTNWJS** and **:***last* are extensions, expected to work similarly to the GNU system. Special treatment of form feeds (**0xC**) is an extension originating from Version 5 AT&T UNIX; notably, the GNU system implements it brokenly. Other systems treat them like any other nonprintable. Passing arbitrary strings instead of single characters to **-eins** is an extension.

*file*s must be text files (no embedded NULs, consist of lines). AT&T System V Release 4 UNIX **pr** observes this point strictly (and produces an infinitely-sized file if the latter is violated). Some implementations exit **1** if a *file* is empty and not in **-m** mode — this is allowed, as is just not producing any output for it, which this implementation does.

The date format in the heading is "May 6 00:27 2023" in the **C** (**POSIX**) locale, per standard. The format in other locales is compatible with the GNU system. It's configurable with **-T** on NetBSD.

Strictly, **-d** is required to produce an additional empty line for each input line. This line is skipped if there's one data line on the output page (lest every second page be empty). AT&T System III UNIX always skips the last doubled line for odd page lengths.

Do *not* use **pr** in applications intended to be portable to the GNU system. An unfortunate plethora of odd options is derived therefrom, but that implementation is just not fit for purpose in any capacity.

HISTORY

Research UNIX

Appears in the first edition of the UNIX Programmer's Manual as `pr` (I):

```
    NAME        pr  --  print file
    SYNOPSIS    pr  [ -lcm ] name1 ...
    DIAGNOSTICS -- (files not found are ignored)
    BUGS        none
```

With a five-line trailer and heading format of

```
    Jan  1 00:00:00  /tmp/pp Page 1
```

(note two spaces before the path; the date is just like contemporaneous **date** sans day-of-week), with two empty lines before and *three* after.

The default paper height is **66** lines, as present-day (but, naturally, since the heading is a line taller, there's one fewer lines of text per page). **-l** is equivalent to present-day **-l** *78* "to accommodate legal size paper".

The date is the modification time by default, as present-day in the no-**-m** calling convention, or time of start of formatting with **-c**. **-m** may be used to override a previous **-c**, because flags may be intermingled with *name*s and apply only to the *name*s that follow them.

Messages are rejected à la **mesg n** when writing to a teletype, then reverted (on exit and on (the equivalent of) `SIGINT`).

Version 3 AT&T UNIX sees a **SYNOPSIS** of

```
    pr [-cm] [-h name] [-n] [+n] [file1 ...]
```

reading the standard input stream if no *file*, **-h**, **+***n* and **-***n* as present-day (but **-h** requires *name* to be in the next argument (i.e. forbidding "**-h***head*")). Mingling flags and *file*s is now explicitly documented. Notably, the **BUGS** say

> In multi-column output, non-printing characters other than new-line cause misalignment.

It's unclear if truncation occurs with a single column (it likely doesn't because it doesn't on subsequent systems), or to what width. Indeed, it's not noted that truncation occurs at all. All page length information is removed as well.

Version 4 AT&T UNIX sees a **SYNOPSIS** of

> **pr** [**-h** name] [-*n*] [+*n*] [file ...]

The loss of **-l** is belatedly mourned in the **BUGS** which instead say

> It would be nice to be able to set the number of lines per page.

It is likely here where the origin of the present-day "only the widths of printable characters are accounted for" rule lies.

-cm are unmourned.
Without *file*s, the standard input stream is read with an empty filename in the heading, as present-day.

Version 5 AT&T UNIX sees a **SYNOPSIS** of

> **pr** [**-h** *header*] [**-***n*] [+*n*] [**-w***n*] [**-l***n*] [**-t**] [*name* . . .]

with **-wl** similar to present-day with present-day (**72**, **66**) defaults (and requiring welded arguments). **-t** is as present-day, and the heading length had clearly been fixed to present-day at some point because the heading spacing is as present-day (two lines either side). The date format in the heading is `Mon May 15 04:58` which some may recognise as `ctime`(3) truncated to **16** bytes. For the standard input stream, it's the time of "opening" it.

As undocumented semantics go, if there are fewer **−lines** per page than the heading and trailer (**10** by default or **0** if **−t**) then then the page is silently elongated to **66** lines. Similarly, if *cols* is **0**, it's reset to **72**. If more than **72** or more than *cols columns* are specified, an error is witten to the standard error stream and **pr** exits. The present-day minimum of one column of space between columns is not provided, and lines may happily run right up against each other if wide enough. Indeed, it's almost impossible to have **pr** *not* generate truly unreadable, incorrect, or last-column-untruncated output.

Some non-printable characters *are* considered for their width! The backspace (**0x8**) and escape (**0x1B**) are processed like a backspace (width of **−1**). An equivalent of **−e** and **−i** (but **−i** tabifies at least *three* spaces, not any amount) is always present (thus, effectively, tabs are considered to have their equivalent widths). Additionally, on output, trailing spaces are discarded.

There's a fully-functional present-day **−m** (with a **9**-*file* limit (coincidentally, POSIX specs that as the minimum), and a single "input column" counter, so, naturally **pr −m** <(**printf** 'a\tb') <(**printf** 'a\tbb') <(**printf** 'a\tb') produces spaces between as and bs of length **7** (good), **6**, and **5**, respectively; not passing an explicit *name* segfaults, and empty **read**()s don't stick, so when reading from teletypes the only way to stop reading is by signalling **pr**) and **−s**[*tab*] (taking a single byte and not affecting the truncation width).

Form feeds (**0xC**) are processed as in this implementation by early-ejecting, but only if writing a single column. Otherwise they're normal characters.

In **−***n* mode, lines are fed downward, as present-day, but each column is saturated before the next is started, contrary to this implementation, AT&T System III UNIX, 4.4BSD, &c. Strictly, POSIX only requires that **−t** make **−***n* "use the minimum number of lines to write the output" (i.e. average out column heights, worded specifically in a way to prevent the way this "known historical implementation" is "relatively useless when used with **−t**"), but no implementation actually implements two column-filling algorithms.

Version 6 AT&T UNIX sees a **SYNOPSIS** of
pr [**−h** *header*] [**−***n*] [**+***n*] [**−w***n*] [**−l***n*] [**−t**] [**−s***c*] [**−m**] name . . .

Removing [] around *name* is a formatting error, as are the missing []s around *c*.

Indeed, the only implementation change from Version 5 AT&T UNIX is that the date in the heading gains a space and year, bringing it in line with the current **STANDARDS** format.

Version 7 AT&T UNIX sees a **SYNOPSIS** of
pr [option] ... [file] ...
and fixes **−m** not respecting encountered EOFs. Errors for unopenable files are written to the standard error stream if the standard output stream is not a teletype.

The BSD

3BSD makes **−***0* equivalent to **−***1* instead of SIGFPEing due to division by zero.

4BSD adds **−f**, which is almost as present-day **−F** except it doesn't affect the final page of any file, and it removes the two empty lines preceding the text in the heading (they are still accounted-for as-if they were there, though, so the amount of input text on the page is unchanged).

4.2BSD doesn't take **−***anything* to be **−***columns*-but-no-digits-so-**0** (quietly normalised to **1** since 3BSD), as heretofore, and instead rejects non-digit-only *anything* and writes a warning to the standard error stream for **−***0*.

System V

AT&T System III UNIX sees a **SYNOPSIS** of
pr [options] [files]
and a behavioural (though not implementation) basis of Version 7 AT&T UNIX.

The *options* may be (present-day unless specified otherwise) **−dprtf**, **−o***k*, **−**{**e**|**i**|**n**}*ck* (single-byte *c*), **−h** *header*, **−l***k*, **+***k*, **−***k*, **−a**, **−m** (overrides instead of excluding **−a** and **−***columns*), **−w***k*, **−s***c* (single-byte *c*; the description says that "If the **−s** option is used, lines are not truncated" but this is a documentation error, and, sans **−w**, truncation width defaults to **512**), single or in chunks (incl.

argument-taking flags, like **−s**_*2*t), but are now all parsed before *files* are processed. Unchangedly, non-numeric **+***anything* is the same as **+***0*.

Undocumentedly, flags are case-insensitive. Also undocumentedly, **−x** (as an alias for **−n**) and **−bq** no-ops are "/* retained for historical reasons */", which aren't really attested for. The **−j** no-op is instead "/* ignore GCOS jprint option */". These all possibly relate to the GCOS systems internally used as print spoolers.

files may contain "**-**" for the standard input stream, as present-day, though the date in the heading corresponds to the time of the first written page (respecting **+***k*). The minimum-single-column separation between columns is enforced; instead of swallowing errors, they're with-held until the end of formatting if the standard output stream is a teletype. With **−***columns*, their heights are evened out always.

The time with **−m** or when formatting the standard input stream is the time of formatting the first such page.

With **−d** and three data lines per output page (**−l***3*, **−l***13*), only the first line of each file is written, and the rest are discarded.

−n with **−***k* uses numbers with a global per-output-line counter (similar to with **−m**), but accounts for width as-if the numbers were there. With **−a**, it *does* number each input line as present-day but doesn't include the separator. Numbers are space-padded to the nominal width, but not truncated if longer than nominal.

Input lines are *always* truncated, even when formatting a single column (to **512** columns if **−w** wasn't specified in this case). Empty files are considered errors in non-**−m** mode; these and opening errors are always written to the standard error stream, but they're with-held until the end of formatting or SIGINT if the standard output stream is a teletype. Detexion of too-narrow pages is likewise updated to match the new padding guarandees. The separator for **−n** isn't accounted for in this, however. If **−s** is passed a printable character and the line output undergoes truncation, the *separator* is not written. In addition to the characters already processed specially when accounting for input and output column positions by Version 5 AT&T UNIX, carriage returns (**0xD**) are also processed; non-printable characters are limited to their current isprint(3) understanding instead of being anything from space (**0x20**) up.

AT&T System V Release 1 UNIX adds tab (**0x9**) processing when counting output width (contradicting the isprint(3) policy on input, thus inconsistently overlapping with freshly-separated-out **−e** and **−i**), fixes **−n**, multiple columns implying **−e** and **−i**, and accounts for the **−n** separator in too-narrow-pages detexion as present-day, except that if the separator is a tab (**0x9**) and **−i** was specified or implied (regardless of what its actual *tab* is!) it's counted as-if it were expanded just after the number (i.e. **−n −i** has a total assumed width of **8** per number, always, and **−n −i**_*10* of **10**). Single-column output being truncated is fixed.

AT&T System V Release 3 UNIX grows the **SYNOPSIS** into a proper

> **pr** [[−column] [−**w**width] [−**a**]] [−**e**ck] [−**i**ck] [−**drtfp**] [+page] [−**n**ck] [−**o**offset] [−**l**length] [−**s**separator] [−**h** header] [file ...]
>
> **pr** [[−**m**] [−**w**width]] [−**e**ck] [−**i**ck] [−**drtfp**] [+page] [−**n**ck] [−**o**offset] [−**l**length] [−**s**separator] [−**h** header] file1 file2 ...

tough, contradictorily, makes **−m** with no *file*s no longer segfault. Flags are required to have the correct case. **−m** excludes **−***k* (but only if *k* > **1**, undocumented) and **−a** requires **−***k* with *k* > **1** (oddly, documented). **−xbqj** are removed and **−l***0* resetting to **66** is finally documented, as is **−dl***1* (any odd number) eating the final empty line on each page. Indeed, most of the manual starts to truly specify most behaviour instead of vaguely describing it. Except **−s** which, sans **−w** is said to "prevent tuncation of lines on multicolumn output" but still just defaults to **512** columns.

AT&T System V Release 4 UNIX sees a **SYNOPSIS** of

> **pr** [[**−***columns*] [**−w***columns*] [**−a**]] [**−e***ck*] [**−i***ck*] [**−drtfp**] [**+page**] [**−n***ck*] [**−o***offset*]
> [**−l***length*] [**−s***separator*] [**−h***header*] [**−F**] [*file* ...]

> **pr** [[**-m**] [**-w***columns*]] [**-e***ck*] [**-i***ck*] [**-drtfp**] [**+page**] [**-n***ck*] [**-o***offset*]
> [**-l***length*] [**-s***separator*] [**-h***header*] [**-F**] [*file1 file2 ...*]

with a mismatched **-h**.

-F is described as

> Fold the lines of the input file. When used in multi-column mode (with the **-a** or **-m** options) lines will be folded to fit the current column's width, otherwise they will be folded to fit the current line width (80 columns).

which appears entirely broken in the former (well it works sometimes, presumably by accident), the "current line width" isn't *80 columns*(?!), and the default fold width sans **-w** in this case is **512**. This is also the first implementation that, when it reads a file that doesn't end in a new-line, repeats the last byte ad infinitum.

Additionally, **-f** no longer works on the final page of any file (much like 4BSD), but **-dl***3* &c. is fixed.

Standards

X/Open Portability Guide Issue 2 ("XPG2") includes AT&T System III UNIX (AT&T System V Release 2 UNIX) **pr** verbatim. X/Open Portability Guide Issue 3 ("XPG3") adds localisation for the time in the heading and

> *LC_CTYPE* is used to determine if a character is printable. Non-printable characters are still placed on the standard output, but are not counted for the purposes of column-width and line-length calculations.

which is the present-day wording, but contradicts every extant implementation.

+*k* is shaded PI ("cannot be guaranteed to be consistent" between "all X/Open compliant systems") for no clear reason.

IEEE Std 1003.2 ("POSIX.2") contains a **Synopsis** of

> pr [+*page*] [-*column*] [-adFmrt] [-e[*char*][*gap*]] [-h *header*] [-i[*char*][*gap*]]
> [-l *lines*] [-n[*char*][*width*]] [-o *offset*] [-s[*char*]] [-w *width*] [file . . .]

largely matching the AT&T System V Release 3 UNIX behaviour with X/Open Portability Guide Issue 3 ("XPG3") locales, except that it conforms to the USG with explicit exceptions, so **-ohlw** must work with both welded and separate arguments, and **-ht** *header* is forbidden from working like **-th** *header*; for **-m** the minimum is correctly nine files (one more than the manual states); **-d** is fully-specified and as present-day (cf. **STANDARDS**); **-i** says at least two spaces are unexpanded before a tabstop, not any amount, which is counter; **-p** is missing "since it represents a purely interactive usage." (which is outside of the domain of IEEE Std 1003.2 ("POSIX.2")); most glaringly, **-f** is replaced with a **-F** — it's defined as AT&T System III UNIX **-f** (if it worked as in the manual), but without the pauses. The **Rationale.**, **History of Decisions Made** speaks of **-F** thus:

> Historical implementations of the pr utility have differed in the action taken for the -f option. BSD uses it as described here for the -F option; System V uses it to change trailing <newline>s on each page to a <form-feed> and, if standard output is a TTY device, sends an <alert> to standard error and reads a line from /dev/tty before the first page. Draft 9 incorrectly specified part of the System V behavior, raising several ballot objections. There were strong arguments from both sides of this issue concerning existing practice and additional arguments against the System V -f behavior, on the grounds that it was not a modular design to have the behavior of an option change depending on where output is directed. Therefore, the -f option is not specified and the -F option has been added.

Notably: 4BSD **-f** is *not* present-day **-F** (even discounting the final-page bug), because it also changes how the heading is formatted, and no AT&T UNIX **-f** implementation folds trailing new-lines (**0xA**). Indeed, **-F** is much closer to AT&T System V Release 3 UNIX **-f** (if the standard output stream were never a teletype) than it is to 4BSD **-f**.

The heading format is defined as present-day (cf. **STANDARDS**). Keeping the optional-value flags, multi-digit **-***columns*, and **+**-started **+***page* is attributed to "their heavy usage by existing applications. However, due to interest in the international community, the developers of the standard have agreed to provide an alternative syntax for the next version of this standard that conforms to the spirit of the Utility Syntax Guidelines. This new syntax will be accompanied by the existing syntax, marked as obsolescent.", which has clearly never come to pass. Be it because implementers didn't "develop and promulgate a new syntax for `pr`, perhaps using a different utility name" for inclusion or just because by 1992 no-one really used printing teletypes any-more (indeed, one may say that `pg`(1) *is* that new syntax under a different utility name (though frozen and obsoleted in X/Open Portability Guide Issue 4 ("XPG4")), or, indeed, the-fresh-in-X/Open Portability Guide Issue 4 ("XPG4") `more`(1) is) we may never know.

It's prudent to note that the form-feed-(**0xC**)-on-input behaviour has *never* actually been documented. It's no wonder, then, that it's not specified in either standard.

X/Open Portability Guide Issue 4 ("XPG4") imports IEEE Std 1003.2 ("POSIX.2") **pr** but re-adds **−f** as present-day and **−p** (like present-day but supposed to read until a carriage return (**0xD**) character is read; this is an overinterpretation of the original manual (and, hence, XBD) which says it "rings the terminal bell and waits for a carriage return", which generates a new-line (**0xA**) when pressed), but both shaded EX (equivalent to modern-day XSI).

IEEE Std 1003.1-2001 ("POSIX.1") unshades **−p**.

IEEE Std 1003.1-2008 ("POSIX.1") fixes **−i** to present-day.

IEEE Std 1003.1-2024 ("POSIX.1") fixes the **−p** character and loosens the requirements of empty *file*s outside **−m** mode to present-day (cf. **STANDARDS**); prior, they weren't allowed to be treated specially, so both the Version 1 AT&T UNIX (no output) and AT&T System III UNIX (error) behaviours were illegal.

4.4BSD

Uses a fresh implementation ("`Complies with posix P1003.2/D11`") and manual copied all but verbatim therefrom; the only oddity is the **SYNOPSIS**, reading

> **pr** [*+page*] [**−***column*] [**−adFmrt**] [[**−e**] [*char*] [*gap*]] [**−h** *header*] [[**−i**] [*char*] [*gap*]] [**−l** *lines*] [**−o** *offset*] [[**−s**] [*char*]] [[**−n**] [*char*] [*width*]] [**−w** *width*] [-] [*file ...*]

The **SEE ALSO** is "`cat`(1), `more`(1)".

Note that **−f** is removed and **−F** is indeed per spec (the relic of the heading's two spaces after the date are folded together, though, funnily, the header format string *still* excludes the initial two new-lines (**0xA**), which are **putchar**()ed separately).

There are a scarce few issues here: **−w** is refused instead of ignored with a single column, with **−i** single spaces are also squeezed (you could say this is correct, actually, or just ahead if its time by fifteen years), **−s** with no argument segfaults, and *every byte* is considered to have unity width (not characters, not printable, not their characteristic widths).

And an oddity in parsing by shipping a modified `getopt`(3) as part of the **pr** source supporting an optional argument as "*s***?**" (what we'd now call "*s***::**") and a stand-alone "**#**" modifier allowing ±*digits*, with the **+** non-combinable.

As well as a truly psychotic decision to — rather than using a message like IEEE Std 1003.2 ("POSIX.2") suggests, or just a sane date format like this implementation — use `LC_TIME` *as the* `strftime`(3) *format string*. This is *despite* `LC_TIME` being well-defined as present-day.

NetBSD 1.0 fixes the **−w** error.

NetBSD 1.6 breaks **−i** even more with a "fix" and invents **−T** *timefmt* to replace the `LC_TIME` blunder.

NetBSD 4.0 fixes **−s**.

NetBSD 7.0 finally fixes **−i** except now it's 2015 so the correct fix would've been to revert it.