NAME

nl — section-aware line numbering

SYNOPSIS

nl [**-p**] [**-d** *\[:*]] [**-**[**hbf**] **a**|**t**|**n**|**p***expr*] [**-l** *consecutive*] [**-n** **rn**|**rz**|**ln**]
[**-s** *separator*] [**-w** *number-width*] [**-v** *first-line*] [**-i** *line-step*] [*file*]...

DESCRIPTION

Numbers matching lines from each *file* (standard input stream if "**-**", the default) to the standard output stream.

The output is divided into pages, each of which has three sections; entry into these sections is controlled through lines consisting exclusively of their respective heading.

Section	Heading	Default numbering
-header	*\:\:\:*	**n**one
-body	*\:\:*	**t**ext (non-empty)
-footer	*\:*	**n**one

The line number is reset at the start of each page (i.e. when entering a higher-or-same section), unless **-p**. Input starts at the body. Headings are written as empty lines.

Within each section, only some lines may be numbered, according to **-hbf**, which must start with:

- **a** all non-empty lines, and every *consecutive*th empty line (so all lines by default),
- **t** non-empty lines,
- **n** no lines, or
- **p***expr* lines matching *expr* (which is a basic regular expression, cf. regex(7)).

If a line is numbered, the number is written aligned according to **-n** to width **-w**:

- **rn** right, with space padding (" 21"),
- **rz** right, with zero padding ("000021"),
- **ln** left, with space padding ("21 "),

followed by the *separator*, followed by the input line, and the current line number is bumped by **-i**.

If it isn't, the line instead is prepended with a (**-w** + length of *separator* in bytes) number of spaces.

OPTIONS

-p, **--no-renumber**
: Don't reset the line number on new pages.

-d, **--section-delimiter**=*\[:*]
: Change the first byte of the delimiter to \; if *:* is specified, change the second thereto. Default to **** and **:**.

-[**hbf**], **--**{**header**|**body**|**footer**}**-numbering**=**a**|**t**|**n**|**p***expr*
: Number only the specified lines in the given section.

-l, **--join-blank**=*consecutive*
: In **a** mode, number the last of *consecutive* consecutive empty lines. Defaults to **1** (every).

-n, **--number-format**=**rn**|**rz**|**ln**
: Line number field alignment.

-s, **--number-separator**=*separator*
: Put *separator* between the line number and data. Defaults to a tab (**0x09**).

-w, **--number-width**=*number-width*
: Width of the line number field. Defaults to **6**.

-v, **--starting-line**=*first-line*
: Initial and reset-to line number. Defaults to **1**.

-i, **--line-increment**=*line-step*
: How much each numbered line increases the number. Defaults to **1**.

EXIT STATUS

1 if a *file* couldn't be opened or the line number overflew. Processing is aborted in the latter case.

EXAMPLES

```
$ cat form
Groceries for February:


        Bananas 3.5kg   $4.51
        Kiwis   2kg     $3.19   Call Siegfried to explain short!
        Bread           $20.21
$ nl form
     1  Groceries for February:


     2          Bananas 3.5kg   $4.51
     3          Kiwis   2kg     $3.19   Call Siegfried to explain short!
     4          Bread           $20.21
$ nl -b p'\$' -w 2 -s : form
   Groceries for February:


 1:     Bananas 3.5kg   $4.51
 2:     Kiwis   2kg     $3.19   Call Siegfried to explain short!
 3:     Bread           $20.21
$ cat form2
::
Groceries for February:
::::
        Bananas 3.5kg   $4.51
        Kiwis   2kg     $3.19   Call Siegfried to explain short!
        Bread           $20.21
::
Planned for March:
::::
        Anise   100g    $5
$ nl -d : -s . -w 2 -n rz form

   Groceries for February:

01.     Bananas 3.5kg   $4.51
02.     Kiwis   2kg     $3.19   Call Siegfried to explain short!
03.     Bread           $20.21

   Planned for March:

01.     Anise   100g    $5
```

SEE ALSO

cat(1) **-nb**.
pr(1) for pagination that isn't just logical.

STANDARDS

Violates IEEE Std 1003.1-2024 ("POSIX.1") for compatibility with the system it tries to describe (AT&T System V Release 4 UNIX); the standard (and most implementations) require the input to be a text file (no embedded null bytes). The "**-**"-as-standard-input-stream behaviour is optional and missing from AT&T System V Release 4 UNIX. Multiple *file*s are an extension, originating from the GNU system.

-l tracks empty lines even across sections, provided all traversed sections are in **a** mode. NetBSD doesn't do this, and it removes heading lines from the output instead of ejecting empty ones.

HISTORY

Appeared fully formed in AT&T System III UNIX as `nl`(1) ("line numbering filter") with a **SYNOPSIS** of

> **nl** [**−h**type] [**−b**type] [**−f**type] [**−v**start#] [**−i**incr] [**−p**] [**−l**num] [**−s**sep] [**−w**width] [**−n**format] file

The only exceptions here are:

- **t** is described as matching "lines with printable text only", but is actually as present-day,
- **-n** is described as present-day, but actually accepts **rn**|**rz**|**l**,
- unnumbered lines are written with number padding plus the separator, and
- in **p***expr*, **$** only matches the end of the line, not the new-line — i.e. **-bp***a$* matches input produced by **printf** `a`, but not **echo** `a`. This makes **$** relatively useless, and new-lines must be matched vebatim instead.

All numeric flags, when not provided with an argument, reset to default. When one is given, it's limited to within [**1**, **8**]. All arguments are required to be welded to their flags.

AT&T System V Release 1 UNIX fixes **-n**, pads unnumbered lines as present-day, and adds **-d**, likewise.

Number parsing is fixed and any non-negative number may be specified. Only **-d** is noted in the manual, which also grows an **EXAMPLES**, consisting of

> The command:
>
> nl −v10 −i10 −d!+ file1 file2
>
> will number files 1 and 2 starting at line number 10 with an increment of ten. The logical page delimiters are !+.

which is notable because, unchangingly, only one *file* may be specified (and, if many are, the final one is used).

AT&T System V Release 2 UNIX fixes the example to only pass one file.

AT&T System V Release 4 UNIX is ported to `<regexpr.h>`, with the effect of yielding present-day semantics: **$** matches the new-line (and end of file if the input is not a text file), a literal new-line never matches.

X/Open Portability Guide Issue 2 ("XPG2") includes AT&T System V Release 3 UNIX `nl`(1) verbatim, but shades the full **SYNOPSIS** UN ("Possibly unsupportable feature."), and adds a "nl [file]" before it. The meaning or purpose of this is unclear.

The Single UNIX Specification ("SUS") re-shades **nl** EX ("Extension.", equivalent to modern-day XSI shading) and removes the no-flag usage. The full usage is missing **-f**, which is unreflected anywhere else. The **CHANGE HISTORY** says "Utility Syntax Guideline support mandated", but instead of the usual disclaimer the **OPTIONS** start with

> The *nl* utility supports the **XBD** specificaiton, **Section 10.2, Utility Syntax Guidelines**, except that the options can be intermingled with the optional *file* operand. Only one file can be named.

with the bit after the comma shaded OB ("Obsolescent."). Notably, while this corresponds to all of the aforementioned systems' behaviour, none of them conform to the USG, and even modern derivatives (such as in the illumos gate) continue to require argument welding.

"Internationalised environment variable support" is also mandated, which corresponds to re-speccing **t** as matching "only lines with text consisting entirely of characters in the current locale's **graph** character classification", which, while "with printable text only" could be read this way, is very obviously not what any implementation has ever done. Similarly, the output format is formalised brokenly — heading lines are to be written out instead of consumed, non-matching lines aren't padded — as present-day.

Version 2 of the Single UNIX Specification ("SUSv2") fixes the **SYNOPSIS** and **t**, rewording it as "only non-empty lines". Naturally, `LC_CTYPE` still mentions **graph** and being used for **t**, leaving it specified in two conflicting ways.

Version 3 of the Single UNIX Specification (“SUSv3”) removes the bit shaded OB, but leaves the bit that says "Only one file can be named", which is unique among the standard inasmuch as that’s already specified in the **SYNOPSIS**.

IEEE Std 1003.1-2008 (“POSIX.1”) allows *file* being "**-**" to mean the standard input stream.