NAME

nl - line numbering filter

SYNOPSIS

nl [**-p**] [**-b** *type*] [**-d** *delim*] [**-f** *type*] [**-h** *type*] [**-i** *incr*] [**-l** *num*] [**-n** *format*] [**-s** *sep*] [**-v** *startnum*] [**-w** *width*] [*file*]

DESCRIPTION

The **nl** utility reads lines from the named *file*, applies a configurable line numbering filter operation, and writes the result to the standard output. If *file* is a single dash ('**-**') or absent, **nl** reads from the standard input.

The **nl** utility treats the text it reads in terms of logical pages. Unless specified otherwise, line numbering is reset at the start of each logical page. A logical page consists of a header, a body and a footer section; empty sections are valid. Different line numbering options are independently available for header, body and footer sections.

The starts of logical page sections are signalled by input lines containing nothing but one of the following sequences of delimiter characters:

Line	Start of
\:\:\:	header
\:\:	body
\:	footer

If the input does not contain any logical page section signalling directives, the text being read is assumed to consist of a single logical page body.

The following options are available:

-b *type*   Specify the logical page body lines to be numbered. Recognized *type* arguments are:

        a   Number all lines.

        t   Number only non-empty lines.

        n   No line numbering.

        p*expr* Number only those lines that contain the basic regular expression specified by *expr*.

The default *type* for logical page body lines is **t**.

-d *delim*   Specify the delimiter characters used to indicate the start of a logical page section in the input file. At most two characters may be specified; if only one character is specified, the first character is replaced and the second character remains unchanged. The default *delim* characters are "\:".

-f *type*   Specify the same as **-b** *type* except for logical page footer lines. The default *type* for logical page footer lines is **n**.

-h *type*   Specify the same as **-b** *type* except for logical page header lines. The default *type* for logical page header lines is **n**.

-i *incr*   Specify the increment value used to number logical page lines. The default *incr* value is 1.

-l *num*   If numbering of all lines is specified for the current logical section using the corresponding **-b a**, **-f a** or **-h a** option, specify the number of adjacent blank lines to be considered as one. For example, **-l** 2 results in only the second adjacent blank line being numbered. The default *num* value is 1.

-n *format*   Specify the line numbering output format. Recognized *format* arguments are:
                   ln   Left justified.
                   rn   Right justified, leading zeros suppressed.
                   rz   Right justified, leading zeros kept.

                   The default *format* is **rn**.

-p   Specify that line numbering should not be restarted at logical page delimiters.

-s *sep*   Specify the characters used in separating the line number and the corresponding text line. The default *sep* setting is a single tab character.

-v *startnum* Specify the initial value used to number logical page lines; see also the description of the **-p** option. The default *startnum* value is 1.

-w *width*   Specify the number of characters to be occupied by the line number; in case the *width* is insufficient to hold the line number, it will be truncated to its *width* least significant digits. The default *width* is 6.

ENVIRONMENT

The LANG, LC_ALL, LC_CTYPE and LC_COLLATE environment variables affect the execution of **nl** as described in environ(7).

EXIT STATUS

The **nl** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Number all non-blank lines:

```
$ echo -e "This is\n\n\na simple text" | nl
    1  This is


    2  a simple text
```

Number all lines including blank ones, with right justified line numbers with leading zeroes, starting at 2, with increment of 2 and a custom multi-character separator:

```
$ echo -e "This\nis\nan\n\n\nexample" | nl -ba -n rz -i2 -s "->" -v2
000002->This
000004->is
000006->an
000008->
000010->
000012->example
```

Number lines matching regular expression for an *i* followed by either *m* or *n*

```
$ echo -e "This is\na simple text\nwith multiple\nlines" | nl -bp'i[mn]'
       This is
    1  a simple text
       with multiple
    2  lines
```

SEE ALSO

jot(1), pr(1)

STANDARDS

The **nl** utility conforms to IEEE Std 1003.1-2001 ("POSIX.1").

HISTORY

The **nl** utility first appeared in AT&T System III UNIX.

BUGS

Input lines are limited to LINE_MAX (2048) bytes in length.