NAME

reform − reformat text file

SYNOPSIS

reform [*tabspec1* [*tabspec2*]] [**+b***n*] [**+e***n*] [**+f**] [**+i***n*] [**+m***n*] [**+p***n*] [**+s**] [**+t***n*]

DESCRIPTION

Reform reads each line of the standard input file, reformats it, and then writes it to the standard output. Various combinations of reformatting operations can be selected, of which the most common involve rearrangement of tab characters. It is often used to trim trailing blanks, truncate lines to a specified length, or prepend blanks to lines.

Reform first scans its arguments, which may be given in any order. It then processes its input file, performing the following actions upon each line, in the order given:

- A line is read from the standard input.
- If **+s** is given, all characters up to the first tab are stripped off and saved for later addition to the end of the line. Presumably, these characters comprise an SCCS SID produced by *get*(I).
- The line is expanded into a tabless form, by replacing tabs with blanks according to the *input* tab specification *tabspec1*.
- If **+p***n* is given, *n* blanks are prepended to the line.
- If **+t***n* is given, the line is truncated to a length of *n* characters.
- All trailing blanks are now removed.
- If **+e***n* is included, the line is extended out with blanks to the length of *n* characters.
- If **+s** is given, the previously-saved SCCS SID is added to the end of the line.
- If **+b***n* is given, the *n* characters at the beginning of the line are converted to blanks, if and only if all of them are either digits or blanks.
- If **+m***n* is included, the line is moved left, i.e., *n* characters are removed from the beginning of the line.
- The line is now contracted by replacing some blanks with tab characters according to the list of tabs indicated by the *output* tab specification *tabspec2,* and is written to the standard output file. Option **+i** controls the method of contraction (see below).

The various arguments accepted by *reform* are as follows:

tabspec1 describes the tab stops assumed for the input file. This tab specification may take on any of the forms described in *tabs*(I). In addition, the operand ‘‘**−−**’’ indicates that the tab specification is to be found in the first line read from the standard input. If no legal tab specification is found there, **−8** is assumed. If *tabspec1* is omitted entirely, ‘‘**−−**’’ is assumed.

tabspec2 describes the tabs assumed for the output file. It is interpreted in the same way as *tabspec1*, except that omission of *tabspec2* causes the value of *tabspec1* to be used for *tabspec2*.

The remaining arguments are all optional and may be used in any combination, although only a few combinations make much sense. Specifying an argument causes an action to be performed, as opposed to the usual default of not performing the action. Some options include numeric values, which also have default values. Option actions are applied to each line in the order described above. Any line length mentioned applies to the length of a line just before the execution of the option described, and the terminating newline is never counted in the line length.

+b*n* causes the first *n* characters of a line to be converted to blanks, if and only if those characters include only blanks and digits. If *n* is omitted, the default value is 6, which is useful in deleting sequence numbers from COBOL programs.

+e*n* causes each line shorter than *n* characters to be extended out with blanks to that length. Omitting *n* implies a default value of 72. This option is useful for those rare cases in which sequence numbers need to be added to an existing unnumbered file. The use of $ in editor regular expressions is more convenient if all lines have equal length, so that the user can issue editor commands such as:

```
s/$/00001000/
```

+f causes a format line to be written to the standard output, preceding any other lines written. See *fspec(V)* for details regarding format specifications. The format line is taken from *tabspec2*, i.e., the line normally appears as follows:

<:t−*tabspec2* **d:>**

If *tabspec2* is of the form **−−***filename* (i.e., an indirect reference to a tab specification in the first line of the named file), then that tab specification line is written to the standard output.

+i*n* controls the technique used to compress interior blanks into tabs. Unless this option is specified, any sequence of 1 or more blanks may be converted to a single tab character if that sequence occurs just before a tab stop. This causes no problems for blanks that occur before the first nonblank character in a line, and it is always possible to convert the result back to an equivalent tabless form. However, occasionally an interior blank (one occurring after the first nonblank) is converted to a tab when this is not intended. For instance, this might occur in any program written in a language utilizing blanks as delimiters. Any single blank might be converted to a tab if it occurred just before a tab stop. Insertion or deletion of characters preceding such a tab may cause it to be interpreted in an unexpected way at a later time. If the **+i** option is used, no string of blanks may be converted to a tab unless there are *n* or more contiguous blanks. The default value is 2. Note that leading blanks are always converted to tabs when possible. **It is recommended that conversion of programs from non-PWB to PWB systems use this option.**

+m*n* causes each line to be moved left *n* characters, with a default value of 6. This can be useful for crunching COBOL programs.

+p*n* causes *n* blanks to be prepended (default of 6 if *n* is omitted). This option is effectively the inverse of **+m***n*, and is often useful for adjusting the position of *nroff(I)* output for terminals lacking both forms tractor positioning and a settable left margin.

+s is used with the **−m** option of *get*(I). The**−m** option causes *g et* to prepend to each generated line the appropriate SCCS SID, followed by a tab. The**+s** option causes *r eform* to remove the SID from the front of the line, save it, then add it later to the end of the line. Because **+e72** is implied by this option, the effect is to produce 80-character card images with SCCS SID in columns 73−80. Up to 8 characters of the SID are shown; if it is longer, the eighth character is replaced by ‘*’ and any characters to the right of it are discarded.

+t*n* causes any line longer than *n* characters to be truncated to that length. If *n* is omitted, the length defaults to 72. Sequence numbers can thus be removed and any blanks immediately preceding them deleted.

The following illustrate typical uses of *reform*. The terms "PWB" and "OBJECT" below refer to UNIX and non−UNIX computer systems, respectively. Each arrow indicates the direction of conversion. The character ‘?’ indicates an arbitrary tab specification; see *tabs(I)* for descriptions of legal specifications.

OBJECT −−−> PWB (i.e., manipulation of RJE output):

Note that files transferred by RJE from OBJECT to PWB materialize with format **−8**.

```
reform −8 ? +t +f <oldfile >newfile   (into arbitrary format)
reform −8 −c +t +b +i <oldfile >newfile (into COBOL)
reform −8 −c3 +t +m +i <oldfile >newfile  (into COBOL, crunched)
```

NOTE: −c3 is the preferred format for COBOL; it uses the least disk space of the COBOL formats.

PWB −−−> OBJECT (i.e., preparation of files for RJE submission):

```
reform ? −8 <oldfile >newfile  (from arbitrary format into−8)
get −p −m sccsfile | reform +s | send ...
```

PWB ONLY (i.e., no involvement with other systems):

```
pr file | reform ? −0 <oldfile   (print on terminal without hardware tabs)
reform ? −0 <oldfile >newfile  (convert file to tabless format)
```

DIAGNOSTICS

All diagnostics are fatal, and the offending line is displayed following the message.
"line too long" a line exceeds 512 characters (in tabless form).
"not SCCS −m" a line does not have at least one tab when **+s** flag is used.
Any of the diagnostics of *tabs(I)* can also appear.

EXIT CODES

0 − normal
1 − any error

SEE ALSO

fspec(V), get(I), nroff(I), send(I), tabs(I)

BUGS

Reform is aware of the meanings of backspaces and escape sequences, so that it can be used as a postprocessor for *nroff.* However, be warned that the **+e, +m, +t** options only count characters, not positions. Anyone using these options on output containing backspaces or halfline motions will probably obtain unexpected results.