NAME

tabs – set tabs on a terminal

SYNOPSIS

tabs [**–n** | **– –***file* | [[**–code**] **–a** | **–a2** | **–c** | **–c2** | **–c3** | **–f** | **–p** | **–s** | **–u**]]
[**+m**[*n*]] [**–T** *type*]
tabs [**–T** *type*] [**+***m*[*n*]] *n1*[,*n2*, . . .]

AVAILABILITY

SUNWcsu

DESCRIPTION

tabs sets the tab stops on the user's terminal according to a tab specification, after clearing any previous settings. The user's terminal must have remotely settable hardware tabs.

OPTIONS

The following options are supported. If a given flag occurs more than once, the last value given takes effect:

–T *type* **tabs** needs to know the type of terminal in order to set tabs and margins. *type* is a name listed in **term**(5). If no **–T** flag is supplied, **tabs** uses the value of the environment variable **TERM**. If the value of **TERM** is **NULL** or **TERM** is not defined in the environment (see **environ**(5)), **tabs** uses **ansi+tabs** as the terminal type to provide a sequence that will work for many terminals.

+m[*n*] The margin argument may be used for some terminals. It causes all tabs to be moved over *n* columns by making column *n***+1** the left margin. If **+m** is given without a value of *n*, the value assumed is **10**. For a TermiNet, the first value in the tab list should be **1**, or the margin will move even further to the right. The normal (leftmost) margin on most terminals is obtained by **+m0**. The margin for most terminals is reset only when the **+m** flag is given explicitly.

Tab Specification

Four types of tab specification are accepted. They are described below: canned, repetitive (–*n*), arbitrary (*n1*,*n2*,...), and file (– –*file*).

If no tab specification is given, the default value is **–8**, that is, UNIX system ''standard'' tabs. The lowest column number is **1**. Note: For **tabs**, column 1 always refers to the leftmost column on a terminal, even one whose column markers begin at 0, for example, the DASI 300, DASI 300s, and DASI 450.

Canned –code

Use one of the codes listed below to select a canned set of tabs. If more than one code is specified, the last code option will be used. The legal codes and their meanings are as follows:

–a **1,10,16,36,72**
Assembler, IBM S/370, first format

–a2 **1,10,16,40,72**
Assembler, IBM S/370, second format

–c **1,8,12,16,20,55**
COBOL, normal format

−c2 **1,6,10,14,49**
COBOL compact format (columns 1-6 omitted). Using this code, the first typed character corresponds to card column 7, one space gets you to column 8, and a tab reaches column 12. Files using this tab setup should include a format specification as follows (see **fspec**(4)):

<:t−c2 m6 s66 d:>

−c3 **1,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,67**
COBOL compact format (columns 1-6 omitted), with more tabs than −**c2.** This is the recommended format for COBOL. The appropriate format specification is (see **fspec**(4)):

<:t−c3 m6 s66 d:>

−f **1,7,11,15,19,23**
FORTRAN

−p **1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61**
PL/I

−s **1,10,55**
SNOBOL

−u **1,12,20,44**
UNIVAC 1100 Assembler

Repetitive

−*n* A *repetitive* specification requests tabs at columns 1+*n*, 1+2**n*, etc., where *n* is a single-digit decimal number. Of particular importance is the value **8**: this represents the UNIX system ''standard'' tab setting, and is the most likely tab setting to be found at a terminal. When −**0** is used, the tab stops are cleared and no new ones are set.

Arbitrary

See **OPERANDS**.

File

−−*file* If the name of a *file* is given, **tabs** reads the first line of the file, searching for a format specification (see **fspec**(4)). If it finds one there, it sets the tab stops according to it, otherwise it sets them as −**8**. This type of specification may be used to make sure that a tabbed file is printed with correct tab settings, and would be used with the **pr** command:

example% tabs −− *file*; **pr** *file*

Tab and margin setting is performed via the standard output.

OPERANDS

The following operand is supported:

n1[,*n2*, . . .] The *arbitrary* format consists of tab-stop values separated by commas or spaces. The tab-stop values must be positive decimal integers in ascending order. Up to 40 numbers are allowed. If any number (except the first one) is preceded by a plus sign, it is taken as an increment to be added to the previous value. Thus, the formats **1**,**10**,**20**,**30**, and **1**,**10**,**+10**,**+10** are considered identical.

EXAMPLES

The command:

example% tabs −a

is an example using −**code** (*canned* specification) to set tabs to the settings required by the IBM assembler: columns 1, 10, 16, 36, 72.

The next command:

example% tabs −8

is an example of using −*n* (*repetitive* specification), where *n* is **8**, causes tabs to be set every eighth position:
1+(1∗8), 1+(2∗8), . . . which evaluate to columns 9, 17, . . .

The command:

example% tabs 1,8,36

is an example of using *n1*,*n2*,. . . (*arbitrary* specification) to set tabs at columns 1, 8, and 36.

The last command:

example% tabs −−$HOME/fspec.list/att4425

is an example of using −−*file* (*file* specification) to indicate that tabs should be set according to the first line of **$HOME/fspec.list/att4425** (see **fspec**(4)).

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **tabs**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

TERM Determine the terminal type. If this variable is unset or null, and if the −**T** option is not specified, terminal type **ansi+tabs** will be used.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

SEE ALSO

expand(1), **newform**(1), **pr**(1), **stty**(1), **tput**(1), **fspec**(4), **terminfo**(4), **environ**(5), **term**(5)

NOTES

There is no consistency among different terminals regarding ways of clearing tabs and setting the left margin.

tabs clears only **20** tabs (on terminals requiring a long sequence), but is willing to set **64**.

The *tabspec* used with the **tabs** command is different from the one used with the **newform** command. For example, **tabs** −**8** sets every eighth position; whereas **newform** −**i**−**8** indicates that tabs are set every eighth position.