

NAME

column – columnate lists

SYNOPSIS

column [options] [*file* ...]

DESCRIPTION

The **column** utility formats its input into multiple columns. It supports three modes:

fill columns before rows

This is the default mode (required for backwards compatibility).

fill rows before columns

This mode is enabled with the **-x**, **--fillrows** option.

create a table

Determine the number of columns the input contains and create a table. This mode is enabled with the **-t**, **--table** option. Output is aligned to the terminal width in interactive mode and 80 columns in non-interactive mode (see **--output-width** for more details). Custom formatting can be applied by using various **--table-*** options.

Input is taken from *file*, or otherwise from standard input. Empty lines are ignored and all invalid multibyte sequences are encoded with the x<hex> convention.

OPTIONS

The argument *columns* for **--table-*** options is a comma separated list of user supplied names, defined with **--table-column** *name1,name2,...*, *indices of columns, as they appear in the input, beginning with 1, or names, defined by a --table-columns attribute. It's possible to mix names and indices. The special placeholder '0' (e.g. -R0) may be used to specify all columns and '-1' (e.g. -R -1) to specify the last visible column. It's possible to use ranges like '1-5' when addressing columns by indices.*

-J, --json

Use JSON output format to print the table. The option **--table-columns** is required and the option **--table-name** is recommended.

-c, --output-width *width*

Output is formatted to a width specified as a number of characters. The original name of this option is **--columns**; this name is deprecated since v2.30. Note that input longer than width is not truncated by default. The default is the terminal width and 80 columns in non-interactive mode. The column headers are never truncated.

The placeholder "unlimited" (or 0) can be used to prevent restricting output width. This is recommended for example when redirecting output to a file.

-d, --table-noheaders

Omit printing the header. This option allows the use of user supplied column names on the command line, but keeps the header hidden when printing the table.

-o, --output-separator *string*

Column delimiter for table output (default is two spaces).

-s, --separator *separators*

Possible input item delimiters (default is whitespace).

-S, --use-spaces *number*

When not in table mode, use whitespaces instead of tabulators to align the columns. This option

specifies the minimum number of whitespaces that separate two columns.

-t, --table

*Determine the number of columns the input contains and create a table. Columns are by default delimited with whitespace, or with characters supplied using the **--output-separator** option. Table output is useful for pretty-printing.*

-C, --table-column attributes

*Define a column with a comma separated list of column attributes. This option can be used more than once, every use defines a single column. Attributes replace some of **--table-** options. For example, **--table-column name=FOO,right** defines a column where text is aligned to right. The option is mutually exclusive to **--table-columns**.*

Supported attributes are:

name=string

Column name.

trunc

*Truncate column text when necessary. The same as **--table-truncate**.*

right

*Right align text. The same as **--table-right**.*

width=number

*Column width. It's used only as a hint. To force it, specify the **strictwidth** attribute as well.*

strictwidth

*Strictly follow column **width=** setting.*

noextreme

*Ignore unusually long cell width. See **--table-noextreme** for more details.*

wrap

*Allow using a multi-line cell for long text if necessary. See **--table-wrap** for more details.*

hide

*Don't print the column. See **--table-hide** for more details.*

json=type

Define column type for JSON output. Supported types are string, number and boolean.

-N, --table-columns names

*Specify column names with a comma separated list. The names are used for the table header and column addressing in option arguments. See also **--table-column**.*

-l, --table-columns-limit number

Specify maximum number of input columns. The last column will contain all remaining line data if the limit is smaller than the number of the columns in the input data.

-R, --table-right columns

Right align text in specified columns.

-T, --table-truncate columns

Specify columns where text can be truncated when necessary, otherwise very long table entries may be printed on multiple lines.

-E, --table-noextreme *columns*

Specify columns where is possible to ignore unusually long (longer than average) cells when calculate column width. The option has impact to the width calculation and table formatting, but the printed text is not affected.

The option is used for the last visible column by default.

-e, --table-header-repeat

Print header line for each page.

-W, --table-wrap *columns*

Specify columns where multi-line cells can be used for long text.

-H, --table-hide *columns*

*Don't print specified columns. The special placeholder '-' may be used to hide all unnamed columns (see **--table-columns**).*

-O, --table-order *columns*

Specify the output column order.

-n, --table-name *name*

Specify the table name used for JSON output. The default is "table".

-m, --table-maxout

Fill all available space on output.

-L, --keep-empty-lines

*Preserve whitespace-only lines in the input. The default is to ignore all empty lines. This option's original name was **--table-empty-lines**, but has since been deprecated because it gives the false impression that the option only applies to table mode.*

-r, --tree *column*

Specify the column to use for a tree-like output. Note that the circular dependencies and other anomalies in child and parent relation are silently ignored.

-i, --tree-id *column*

Specify the column that contains each line's unique child IDs for a child-parent relation.

-p, --tree-parent *column*

Specify the column that contains each line's parent IDs for a child-parent relation.

-x, --fillrows

Fill rows before filling columns.

-h, --help

Display help text and exit.

-V, --version

Display version and exit.

ENVIRONMENT

The environment variable **COLUMNS** is used to determine the size of the screen if no other information is available.

HISTORY

The **column** command appeared in 4.3BSD-Reno.

BUGS

Version 2.23 changed the **-s** option to be non-greedy, for example:

```
printf "a:b:c\n1::3\n" | column -t -s ':'
```

Old output:

```
a  b  c
1  3
```

New output (since util-linux 2.23):

```
a  b  c
1      3
```

Historical versions of this tool indicated that "rows are filled before columns" by default, and that the **-x** option reverses this. This wording did not reflect the actual behavior, and it has since been corrected (see above). Other implementations of **column** may continue to use the older documentation, but the behavior should be identical in any case.

EXAMPLES

Print fstab with a header line and align numbers to the right:

```
sed 's/#.*//' /etc/fstab | column --table --table-columns SOURCE,TARGET,TYPE,OPTIONS
```

Print fstab and hide unnamed columns:

```
sed 's/#.*//' /etc/fstab | column --table --table-columns SOURCE,TARGET,TYPE --table
```

Print a tree:

```
echo -e '1 0 A\n2 1 AA\n3 1 AB\n4 2 AAA\n5 2 AAB' | column --tree-id 1 --tree-parent
1  0  A
2  1  | -AA
4  2  | | -AAA
5  2  | ' -AAB
3  1  ' -AB
```

SEE ALSO

colrm(1), **ls(1)**, **paste(1)**, **sort(1)**

REPORTING BUGS

For bug reports, use the issue tracker <<https://github.com/util-linux/util-linux/issues>>.

AVAILABILITY

The **column** command is part of the util-linux package which can be downloaded from Linux Kernel Archive <<https://www.kernel.org/pub/linux/utils/util-linux/>>.