boxes

Section: User Commands (1)
Updated: March 16 2017
[Index | Return to Main Contents (index.html)]

NAME

boxes - text mode box and comment drawing filter

SYNOPSIS

boxes [-hlmrv] [-a format] [-d design] [-f file] [-i indent] [-k bool] [-p pad] [-s size] [-t tabopts] [infile [outfile]]

DESCRIPTION

Boxes is a text filter which can draw any kind of box around its input text. Box design choices range from simple boxes to complex ASCII art. A box can also be removed and repaired, even if it has been badly damaged by editing of the text inside. Since boxes may be open on any side, *boxes* can also be used to create regional comments in any programming language. New box designs of all sorts can easily be added and shared by appending to a free format configuration file.
boxes was originally intended to be used with the *vim(1)* text editor, but it can be tied to any text editor which supports filters, as well as called from the command line as a standalone tool.

OPTIONS

Options offered by *boxes* are the following:

-a *string*

Alignment/positioning of text inside box. This option takes a format string argument which is read from left to right. The format string may not contain whitespace and must consist of one or more of the following components:
hx - horizontal alignment of the input text block inside a potentially larger box. Possible values for *x* are **l** (ell, for left alignment), **c** (center), or **r** (right). This does not affect the justification of text lines within the input text block (use the **j** argument instead).
vx - vertical alignment of the input text block inside a potentially larger box. Possible values for *x* are **t** (for top alignment), **c** (center), or **b** (bottom).
jx - justification of lines within the input text block. Possible values for *x* are **l** (ell, for left justification), **c** (center), or **r** (right). This does not affect the alignment of the input text block itself within the box. Use the **h** and **v** arguments for input text block positioning.

Short hand notations (can be combined with the above arguments):
l (ell) - short for **h**l**v**c**j**l
c - short for **h**c**v**c**j**c
r - short for **h**r**v**c**j**r

The factory default setting for **-a** is **h**l**v**t.

-c *string*

Command line design definition for simple cases. The argument of this option is the definition for the "west" (W) shape. The defined shape must consist of exactly one line, i.e. no multi-line shapes are allowed. The **-c** option is intended as a shortcut for those cases where simple regional comments are to be created, which only need a certain character or sequence of characters to be placed in front of every line. In such cases, it is much more convenient to simply specify **-c** than to do a complete design definition in one's config file, where the only shape defined is the west shape.
This option implies a **-d** and does not access the config file. **-c** may of course be used in conjunction with any of the other options. By default, **-c** is not specified.

-d *string*

Design selection. The one argument of this option is the name of the design to use.

-f *string*

Use alternate config file. The one argument of this option is the name of a valid *boxes* config file, containing new and exciting designs!

-h

Print usage information.

-i *string*

Indentation mode. Possible arguments are "text" (indent text inside of box), "box" (indent box, not text inside of box), or "none" (throw away indentation). Arguments may be abbreviated. The default is to indent the box, but not the text.

-k *bool*

Kill leading/trailing blank lines on removal. The value of *bool* can be specified as on, yes, true, 1, or t, all meaning yes, or off, no, false, 0, or f, which mean no. This is case-insensitive. This option only takes effect in connection with **-r**. If set to yes, leading and trailing blank lines will be removed from the output. If set to no, the entire content of the former box is returned. The default is no, if both the top and the bottom part of the box are open, as is the case with most regional comments. If the box's design defines a top part or a bottom part, the default is yes.

-l

(ell) List designs. Produces a listing of all available box designs in the config file, along with a sample box and information about it's creator. Also checks syntax of the entire config file. If used in connection with **-d**, displays detailed information about the specified design.

-m

Mend box. This removes a (potentially broken) box as with **-r**, and redraws it afterwards. The mended box is drawn according to the options given. This may be important to know when it comes to restoring padding, identation, etc. for the mended box. Implies **-k** false.

-p *string*

Padding. Specify padding in spaces around the input text block for all sides of the box. The argument string may not contain whitespace and must consist of a combination of the following characters, each followed by a number indicating the padding in spaces:
a - (all) give padding for all sides at once
h - (horiz) give padding for both horizontal sides
v - (vertical) give padding for both vertical sides
b - (bottom) give padding for bottom (south) side
l - (left) give padding for left (west) side
t - (top) give padding for top (north) side
r - (right) give padding for right (east) side
Example: **-p a**4**t**2 would define the padding to be 4 characters on all sides, except for the top of the box, where the input text block will be only 2 lines away from the box.
By default, unless specified otherwise in the config file, no padding is used.

-r

Remove box. Removes an existing box instead of drawing it. Which design to use is detected automatically. In order to save time or in case the detection does not decide correctly, combine with **-d** to specify the design. The default is to draw a new box.

-s *width*xheight

Box size. This option specifies the desired box size in units of columns (for width) and lines (for height). If only a single number is given as argument, this number specifies the desired box width. A single number prefixed by 'x' specifies only the box height. The actual resulting box size may vary depending on the individual shape sizes of the chosen design. Also, other command line options may influence the box size (such as **-p**).
By default, the smallest possible box is created around the text.

-t *string*

Tab handling. This option controls how tab characters in the input text are handled. The option string must always begin with a *uint* number indicating the distance between tab stops. It is important that this value be set correctly, or tabulator characters will upset your input text. The correct tab distance value depends on the settings used for the text you are processing. A common value is 8.
Immediately following the tab distance, an optional character can be appended, telling *boxes* how to treat the leading tabs. The following options are available:
e - expand tabs into spaces
k - keep tabs as close to what they were as possible
u - unexpand tabs. This makes *boxes* turn as many spaces as possible into tabs.
In order to maintain backwards compatibility, the **-t** *string* can be just a number. In that case, **e** is assumed for tab handling, which removes all tabs and replaces them with spaces. The factory default for the **-t** option is simply 8, which is just such a case.
For example, you could specify **-t** 4u in order to have your leading tabs unexpanded. In the box content, tabs are always converted into spaces. The tab distance in this example is 4.

-v

Print out current version number.

CONFIGURATION FILES

Boxes will use the configuration file specified on the command line (using **-f**). If no config file is specified on the command line, *boxes* will check for the BOXES environment variable, which may contain a filename to use. If BOXES is not set, *boxes* will try to read $HOME/.boxes and use it as a config file. Failing that, *boxes* will try to read the system-wide config file (see FILES).
The syntax of *boxes* config files is described on the website (see below). They are quite self-explanatory, though.

AVAILABILITY

Boxes is available from its website at http://boxes.thomasjensen.com/ (http://boxes.thomasjensen.com/). The website also features a number of examples illustrating this manual page as well as more in-depth documentation.
Check out *vim(1)* at vim.org (http://www.vim.org/)!

AUTHOR

Boxes was made by Thomas Jensen < boxes at thomasjensen dot com >.
Please see the *boxes* website for the most current email address.

VERSION

This is *boxes* version 1.2.

BUGS

Although it is doing fine in most cases, imho the design autodetector needs some more work.
Should you notice any other unspecified behavior, please tell the author!

ENVIRONMENT

Boxes recognizes the following environment variables:

HOME
: The user's home directory.

BOXES
: Name of *boxes* configuration file, if different from `~/.boxes`.

FILES

$HOME/.boxes
: *boxes* configuration file

/usr/share/boxes
: system-wide configuration file

Index

This document was created by man2html (http://users.actrix.gen.nz/michael/vhman2html.html), using the manual pages.
Time: 22:17:25 GMT, March 16, 2017
Page created April 06, 1999, last touched: 16-Mar-2017.