

NAME

awkpretty – prettyprint awk programs

SYNOPSIS

```
awkpretty [ -? ] [ -a ] [ -b ] [ -c nnn ] [ -f awk-program-file ] [ -g ] [ -h ] [ -i nnn ] [ -q ] [ -r ]
[ -s ] [ -v ] [ -w nnn ] [ -x nnn ] [ -y str ] [ -z str ]
[ awk-program-file(s) ] > new-awk-program-file
```

or

```
awkpretty [ --? ] [ --action-column nnn ] [ --action-separator str ] [ --awklex-input ]
[ --blanks-only ] [ --comment-column nnn ] [ --file awk-program-file ]
[ --function-separator str ] [ --globals ] [ --help ] [ --indent nnn ]
[ --quick ] [ --reorder-functions ] [ --standard-comments ]
[ --version ] [ --width nnn ]
[ awk-program-file(s) ] > new-awk-program-file
```

DESCRIPTION

awkpretty is a prettyprinter for **awk**(1) programs.

It uses **awklex**(1) to obtain a lexical token stream derived from a grammatical analysis of the input file(s), and then outputs that stream as an **awk**(1) program with standardized spacing and line breaks, generally improving readability.

OPTIONS

awkpretty accepts two styles of options: the old, concise, but possibly, inscrutable, UNIX hyphen-letter pairs, and longer GNU/POSIX-style options beginning with two hyphens. The GNU/POSIX-style options can be always abbreviated to any unique leading prefix, and the two styles can be used interchangeably.

Options that take a value expect it as a *separate* following command-line argument: use `-w 72`, *not* `-w72`.

`-x nnn`

`--action-column nnn` For the part of the input **awk**(1) program corresponding to *pattern {action}* pairs, where *pattern* may be empty, indent the *{action}* part to begin after column *nnn*.

[Default: 16]

`-y str`

`--action-separator str` Normally, **awkpretty** separates *pattern {action}* sections by two blank lines, to help set them off from one another. This option can be used to provide an alternate separator string, which must consist exclusively of comments and/or whitespace. Newlines in the string must be represented by `\n`, and spaces by `\s`.

[Default: "`\n\n\n`"]

`-a`

`--awklex-input` Input is not **awk** program text, but rather, a lexical token stream from **awklex**(1).

This option is provided for the convenience of other tools that themselves process, or produce, lexical token streams, allowing them to use **awkpretty** as a backend filter to reconstruct **awk** program text.

[Default: Input is **awk** program text]

`-b`

`--blanks-only` Use only blanks to provide needed whitespace in the output.

[Default: use horizontal tabs and blanks, because that is shorter, and also more likely to preserve the intended text alignment when minor edits are made]

-c *nnn*

--comment-column *nnn* Set the comment column to *nnn*. Inline comments will then be lined up to start immediately after that column, unless the line is too long, in which case, they are moved to an otherwise empty following line.

A zero value suppresses columnar alignment: inline comments will be prefixed by only a single tab.

[Default: 40]

-f *awk-program-file*

--file *awk-program-file* Specify the name of the file containing the **awk**(1) program.

Multiple **-f** options can be used, if required; their files are then treated as if logically concatenated in the order that they were specified.

The filename can be a hyphen, **-**, meaning that input should be taken from *stdin*, allowing **awkpretty** to be used in the middle of a command pipeline.

-z *str*

--function-separator *str* Normally, **awkpretty** separates output function definitions by two blank lines, to help set them off from one another. This option can be used to provide an alternate separator string, which must consist exclusively of comments and/or whitespace. Newlines in the string must be represented by **\n**, and spaces by **\s**.

[Default: "\n\n\n"]

-g

--globals

Produce on *stderr* a sorted listing of function names and global variables that they reference, one such pair per line.

Because the **awk** programming language does not have variable declarations, function-local variables are conventionally created by declaring them as additional arguments, unused by callers. Failure to do so can result in a nasty bug when an assignment to a supposedly local variable in a function silently modifies a global variable. This option provides a way to detect such errors.

[Default: no list of global variables]

-?

-h

--?

--help

Print a brief help message, and exit immediately with a success status code.

-i *nnn*

--indent *nnn*

Set the statement level indentation increment to *nnn*. Each logical increase in statement nesting increases leading indentation by that amount.

Values outside the range 0 ... 8 will be reset to the nearer endpoint of that range.

The choice between tabs-and-blanks, or just blanks, for the leading indentation depends on whether **--blanks-only** was specified or not.

[Default: 4]

-q

--quick

Suppress reading of the user initialization file, *\$HOME/awkprettyrc*. Normally, the contents of that file, if it exists, are implicitly inserted at the

beginning the command line, with comments removed and newlines replaced by spaces. Thus, that file can contain any **awkpretty** options defined in this documentation, either one option, or option/value pair, per line, or with multiple options per line. Empty lines, and lines that begin with optional whitespace followed by a sharp (#) are comment lines that are discarded.

If the initialization file contains backslashes, they must be doubled because the text is interpreted by the shell before **awkpretty** sees it.

-r

--reorder-functions

Sort functions into alphabetical order, ignoring lettercase.

In large programs, there is rarely any reason to use any order of function definition other than alphabetical, because that is the only order that other human readers are sure to understand.

[Default: leave functions in their original order]

-s

--standard-comments

Normally, **awkpretty** treats comments in one of two ways: if they begin a line, their indentation is preserved. Otherwise, they are indented to a comment column which is settable by the **--comment-column** option.

However, with this option, **awkpretty** converts comments to a style found useful by decades of practice in LISP programming: three or more leading comment start characters (###) mean that the comment must be flush left; doubled start characters (##) mean that the comment must be aligned at the current statement indentation; a single start character (#) means that the comment should be aligned at the default comment column.

Comments that do not conform to these start character conventions are modified as follows. Comments that are found at beginning of line are untabified, then adjusted to begin with three start characters (###). Comments found indented are untabified, and adjusted to begin with two such characters (##). All other comments (those following non-blank tokens) are adjusted to begin with a single start character (#). The comment is then output according to the conventions described in the previous paragraph.

[Default: leave comments unmodified, though possibly changed in indentation]

-v

--version

Print the program version number, and exit immediately with a success status code.

-w nnn

--width nnn

Set the preferred maximum output line width to *nnn*. Lines with excessively long tokens may extend beyond this limit, since long tokens, such as complete comment strings, are never broken at line boundaries.

A zero value of *nnn* is treated as ‘infinity’, effectively preserving line breaks from the input file.

[Default: 80]

FILES

`$HOME/.awkprettyrc`

User initialization file containing customized **awkpretty** command-line options.

/usr/local/share/lib/awkpretty/awkpretty-1.00/awkpretty.awk

awk(1) program invoked by **awkpretty** to filter the output of **awklex**(1), and produce a prettyprinted program on *stdout*.

SEE ALSO

awk(1), **awklex**(1), **awkunlex**(1), **gawk**(1), **mawk**(1), **nawk**(1).

AUTHOR

Nelson H. F. Beebe

Center for Scientific Computing

University of Utah

Department of Mathematics, 322 INSCC

155 S 1400 E RM 233

Salt Lake City, UT 84112-0090

USA

Email: beebe@math.utah.edu, beebe@acm.org, beebe@ieee.org

WWW URL: <http://www.math.utah.edu/~beebe>

Telephone: +1 801 581 5254

FAX: +1 801 585 1640, +1 801 581 4148