NAME

cawf, nroff − C version of the nroff-like, Amazingly Workable (text) Formatter

SYNOPSIS

cawf [−**c***config*] [−**d***device*] [−**e**] [−**f***font*] [−**h**] [−**m***acros*] [*file* ...]

DESCRIPTION

Cawf formats the text from the input *file*(s) (standard input if none) in an approximation of *nroff*. It comes closest to duplicating *nroff*'s **man** or **ms** macro package styles. It has some limited support for *nroff*'s **me** macros.

OPTIONS

Options must precede file names.

−**c***config*

defines an alternate path to the device configuration file. Normally the device configuration file is found in *device.cf* in the *cawf* library (see the **FILES** section).

The device configuration file contains device character strings for selecting fonts and the bold or italic type faces. See the **DEVICES** section for more information.

−**d***device*

specifies the name of the output device. There are three built−in devices − ANSI, NONE and NORMAL − and other devices may be defined in the device configuration file. See the **DEVICES** section for more information.

The NORMAL device is the default.

−**e** directs *cawf* to issue an eject (FF or ^L) after the last page.

−**f***font* specifies the one font for the device, declared with the −**d***device* option, that is to be used for the entire document. *Font* must match a font associated with the device's stanza in the device configuration file. See the **DEVICES** section for more information.

No *font* may be specified for the built−in devices ANSI, NONE or NORMAL.

−**h** requests a help display.

−**m***acro* specifies the macro file to be used. The standard *cawf* distribution supplies macro files to support ''−man'', ''−me'' or ''−ms''. *Cawf* finds a macro file by constructing its name from 'm', *acro* and **.mac** − e. g., −**m***an* is converted to **man.mac**. The default directory for macro files is defined when *cawf* is compiled; it's *C:\SYS\LIB\CAWF* in the MS−DOS environment; */usr/lib/cawf* in the UNIX environment.

file ... are the names of files containing *nroff* source text.

NROFF COMPATIBILITY

Cawf accepts the following raw *nroff* requests:

.\"	.ad	.bp	.br	.ce	.de	.di	.ds
.el	.fi	.fl	.ft	.i0	.ie	.if	.in
.it	.lg	.li	.ll	.ls	.na	.ne	.nf
.nr	.ns	.pl	.po	.ps	.rm	.rn	.rr
.rs	.so	.sp	.ta	.ti	.tm	.tr	

and the following in-text codes:

\$	\%	*	\"	\c	\f	\h	\k
\n	\s	\w					

plus the full list of *nroff*/*troff* special characters in the original V7 *troff* manual.

Many restrictions are present; the behavior in general is a subset of *nroff*'s. Of particular note are the following:

- The fully supported nroff request control character is the period. There is limited support for the non−break, acute accent control character.

- Point sizes do not exist; **.ps** is ignored.
- Special vertical spacing − the **.vs** request included − is ignored.
- Conditionals cover only the numeric comparisons >, =, <, >= and <= on **\n(.$**; string comparisons between a macro parameter and a literal; **n** (always true); and **t** (always false). Only single line input is accepted from conditionals; multi−line input − e.g., \(*anything*\) − is not supported.
- The handling of strings is generally primitive.
- Horizontal motion via **\h** must be supplied with a number register interpolation and must be positive - e. g., **\w\n(NN**, where the value in NN is >= 0.
- The **\k** function is reliable only after TAB characters, so it is useful only for measuring table positions.
- The **.di** request only turns output on and off − any macro name is ignored.
- Expressions - e. g., **.sp** - are reasonably general, but the **|**, **&**, and **:** operators do not exist, there must be white space between the end of the *nroff* function and the beginning of the expression, and **\w** requires that quote (') be used as the delimiters. **\w** counts the characters inside the quotes and scales the result in ens, so that, for example, \w'\(bu' equals 4n, and \w'\(bu'/1n equals 4.
- The only acceptable count for the **.it** request is one, and it is effective only with **man**, **me** or **ms** macros.
- The default scaling factor is 'v' for the **.ne**, **.sp**, and **.pl** raw *nroff* requests; it is 'u' for **.nr**; and 'n' for **.in**, **.ll**, **.ls**, **.po**, **.ta** and **.ti**. (A different scaling factor may be specified with a trailing character.)
- Some obsolete or meaningless requests − **.i0**, **.lg** and **.li** − are silently ignored.

White space at the beginning of lines, and embedded white space within lines is dealt with properly. Sentence terminators at ends of lines are understood to imply extra space afterward in filled lines. Tabs are implemented crudely and not exactly, although usually they work as expected. Hyphenation is done only at explicit hyphens, em-dashes, and *nroff* discretionary hyphens. By default bold and italic characters are emulated with backspacing and overprinting, but the **−d** and **−f** options, combined with the contents of the device configuration file, may be used to generate special codes for bold and italic characters. (See the **DEVICES** section for more information.)

MAN MACROS

The **man** macro set replicates the full V7 manual macros, plus a few semi-random oddballs. The full list is:

.AT	.B	.BI	.BR	.BY	.DE	.DT	.HP
.I	.IB	.IP	.IR	.IX	.LP	.NB	.P
.PD	.PP	.RB	.RE	.RI	.RS	.SH	.SM
.SS	.TH	.TP	.UC				

.BY and **.NB** each take a single string argument (respectively, an indication of authorship and a note about the status of the manual page) and arrange to place it in the page footer. **.AT** and **.IX** do nothing.

ME MACROS

The **me** macro subset has been derived from the *cawf* **ms** macros by Chet Creider <creider@csd.uwo.ca>. It includes:

.(l	.(q	.)l	.)q	.b	.bu	.i	.ip
.lp	.np	.pp	.r	.sh	.sm	.u	.uh

The .(l C and .(l L options are supported. In addition, the .AB, .AE, .AI, .AU, .DA, .ND, .TL and .UX macros have been retained from the **ms** set, and the .XP macro has been borrowed from the Berkeley additions to the **ms** macro set.

MS MACROS

The **ms** macro set is a substantial subset of the V7 manuscript macros. The macros are:

.AB	.AE	.AI	.AU	.B	.CD	.DA	.DE
.DS	.I	.ID	.IP	.LD	.LG	.LP	.ND

.NH	.NL	.PP	.QE	.QP	.QS	.R	.RE
.RP	.RS	.SH	.SM	.TL	.TP	.UL	.UX

Size changes are recognized but ignored, as are **.RP** and **.ND**. **.UL** just prints its argument in italics. **.DS**/**.DE** does not do a keep, nor do any of the other macros that normally imply keeps.

The **DY** string variable is available. The **PD**, **PI**, and **LL** number registers exist and can be changed.

HEADERS AND FOOTERS

Cawf allows the placement of text into the five line header and footer sections from the **LH**, **CH**, **RF**, **LF**, **CF**, and **RF** string variables, via the control of the **.^b** request:

.^b fh 1 enables header string placement on the first page
.^b fh 0 disables header string placement on the first page
.^b HF 1 enables header/footer string placement
.^b HF 0 disables header/footer string placement

There are appropriate **.^b** requests in the distribution **man**, **me** and **ms** macro files. (The **me** and **ms** macro files use another **.^b** request, **.^b NH**, to enable numbered header processing.)

OUTPUT

The default output format supported by *cawf*, in its distributed form, is that appropriate to a dumb terminal, using overprinting for italics (via underlining) and bold. The *nroff* special characters are printed as some vague approximation (it's sometimes extremely vague) to their correct appearance.

One part of *cawf*'s knowledge of the output device, related to the formation of characters, is established by a device file, which is read before the user's input. The search for it begins in *cawf*'s library directory, under the name *term*.**dev** (where *term* is the value of the TERM environment variable). Failing to find that, *cawf* searches for **dumb.dev**. (See the **FILES** section for a description of the path to *cawf*'s library directory.) The device file uses special internal requests to set up resolution, special characters and more normal *nroff* functions to set up page length, etc.

Cawf has limited support for fonts special forms of bold and italic characters. It is provided through the **−c** *config*, **−d***device* and **−f***font* options. See the **DEVICES** section for more information.

Note the distinction between the device and the output device configuration files. The device file typically defines characters and constant output parameters. The output device configuration file defines font and type face codes. It is usually not necessary to define a separate device file for each device represented in the output device configuration file − the *dumb.dev* device file will suffice for almost all representations.

DEVICES

Cawf supports primitive output device configuration for font and type face control. One font may be selected for the entire document by directing *cawf* to issue a font selection control character string at the beginning of the document, and control character strings may be selected for switching between the bold, italic and Roman type faces.

The **−c** *config*, **−d***device* and **−f***font* options direct the font and type face selections.

The **−d***device* option specifies the name of the device. *Cawf* has three built−in devices − ANSI, NONE and NORMAL. When the ANSI device is selected, *cawf* issues the ANSI shadow mode control codes, ''ESC [7 m'', to represent the bold face; the ANSI underscore control codes, ''ESC [4 m'', to represent the italic face; and the ANSI control codes, ''ESC [0 m'', to represent the ROMAN face. No **−f***font* specification is permitted with the ANSI device.

When the NONE device is selected, *cawf* uses no special output codes to represent the type faces. No **−f***font* specification is permitted with the ANSI device.

The NORMAL output device is the default. When it's selected, *cawf* overprints each bold character two times, using three issuances of each bold character, separated by backspace characters; it issues an underscore and backspace before each italic character. No **−f***font* specification is permitted with the ANSI device. The *bsfilt*(1) filter may be used to further process the backspace codes output for a NORMAL device.

All other devices named in the **−d***device* option must be represented by a stanza in the device configuration file. The device configuration file is usually contained in *device.cf* in *cawf's* library directory (see the **FILES** section for more information). An alternate device configuration file path may be specified with the **−c***config* option.

The **DEVICE CONFIGURATION FILE** section describes the organization of the device configuration file. It is easy to add devices to the *device.cf* supplied in the *cawf* distribution.

The **−f***font* option may be used with the **−d***device* option, when the appropriate stanza in the device configuration file contains an entry for the named *font*. The **DEVICE CONFIGURATION FILE** section describes how fonts are defined in device configuration file stanzas.

DEVICE CONFIGURATION FILE

The device configuration file defines the special character codes necessary to direct output devices to select fonts and to produce bold, italic and Roman type faces.

The configuration file is usually found in *device.cf* in *cawf*'s library directory (see the **FILES** section for more information). It is organized into two main parts − comments and device stanzas. Comments are any lines that begin with the pound sign ('#') character. They are informational only and *cawf* ignores them. *Cawf* also ignores empty lines, so they may be used as vertical white space.

Stanzas name devices and define their font and type face control strings. A stanza begins with the name of the device, starting at the beginning of a line and occupying the entire line. The body of the stanza, defining fonts and type faces, is formed of lines beginning with white space (a TAB or space characters) that directly follow the device name.

Individual lines of the stanza body contain a key character, followed by a equal sign, followed by the font name (if a font key) and the output device control codes. *Cawf* issues the font control codes once, at the beginning of output, so only one font may be selected. The type face control codes are issued at each change of type face.

The key characters are:

b	for bold
f	for font definition
i	for italic
r	for Roman

The 'b', 'i' and 'r' key codes are followed by an equal sign ('=') and their control code definition. The 'f' key code is followed by an equal sign ('='), the font name, another equal sign and the font control code definition.

Control code definitions may contain any printable ASCII characters. Non−printable characters may be encoded in octal notation with the '\nnn' form or in hexadecimal with the '\xnn' form. The special code, '\E' (or '\e') represents the ESC control character (\033 or \x1b).

Here's a sample showing the definition for the HP LaserJet III. The stanza name is ''lj3''. All its non−printable characters are ESCs; the first is coded in octal form; the second with '\E'; the rest, in hexadecimal form. TAB is used as the leading white space character for the stanza body lines.

```
# HP LaserJet III

lj3
	b=\033(s7B
	i=\E(s1S
	r=\x1b(s0B\x1b(s0S
	f=c10=&l0O(8U(s0p12h10v0s0b3T
	f=c12ibm=&l0O(10U(s0p10.00h12.0v0s0b3T
	f=lg12=&l0O(8U(s12h12v0s0b6T
```

The distribution *device.cf* file defines the following devices and fonts.

epson dot matrix printer in Epson FX-86e/FX-800 mode
Bold: Double-strike
Fonts: none

ibmppds IBM Personal Printer Data Stream (PPDS) protocol
Bold: Double-strike
Italic: Underline
Fonts: none

kxp1124 Panasonic KX−P1124 dot matrix printer in PGM mode
Bold: Emphasized
Fonts:

c10	10 Characters Per Inch (CPI) Courier
c12	12 CPI Courier
bps10	10 CPI Bold PS
bps12	12 CPI Bold PS
p10	10 CPI Prestige
p12	12 CPI Prestige
s10	10 CPI Script
s12	12 CPI Script
ss10	10 CPI Sans Serif
ss12	12 CPI Sans Serif

kxp1180 Panasonic KX−P1180 dot matrix printer in PGM mode
Bold: Emphasized
Fonts:

c10	10 Characters Per Inch (CPI) Courier
c12	12 CPI Courier
bps10	10 CPI Bold PS
bps12	12 CPI Bold PS
p10	10 CPI Prestige
p12	12 CPI Prestige
ss10	10 CPI Sans Serif
ss12	12 CPI Sans Serif

lj3 HP LaserJet III
Fonts:

c10	10 point, 12 Characters Per Inch (CPI) Courier
c12ibm	12 point, 10 CPI Courier, IBM−PC Symbol Set
lg12	12 point, 12 CPI Letter Gothic

vgamono VGA monochrome monitor for MS−DOS
(ANSI.SYS driver required for MS−DOS)
Italic: Reverse-video
Fonts: none

FILES

Cawf resource files are located in the *cawf* library directory − *C:\SYS\LIB\CAWF*, the MS−DOS environment default; or */usr/lib/cawf*, the UNIX environment default. These defaults can be overridden by the CAWFLIB environment variable, or changed in the cawflib.h header file.

common	common device-independent initialization
device.cf	output device configurations
*.dev	device-specific initialization
m*.mac	macro package files

DIAGNOSTICS

Unlike *nroff*, *cawf* complains whenever it sees unknown requests. All diagnostics appear on the standard error file.

HISTORY

Vic Abell of Purdue University <abe@cc.purdue.edu> derived *cawf* from *awf*, ''the Amazingly Workable (text) Formatter,'' written by Henry Spencer of the University of Toronto. The Toronto work was a supplement to the C News project. The Purdue effort was aimed at producing a C language version that would run on small systems, particularly MS−DOS ones. The adaptation of the **me** macros was done by Chet Creider <creider@csd.uwo.ca>. Chet also contributed ideas for device, font and type face support.

The MS−DOS version of *cawf* has been compiled with version 2.5 of Microsoft's Quick-C compiler. It runs under the Mortis Kern Systems Toolkit KornShell, *ksh*(1), and COMMAND.COM.

BUGS

Nroff and troff mavens will have many complaints. Some may even represent bugs and not deliberate omissions.

Watch out for scaling factors - especially on requests like **w**.

The overprinting required to create bold and italicized characters is tiresome on a slow printer. The *bsfilt*(1) post−filter from this distribution may be used to alleviate that nuisance by managing the backspacing codes from *cawf*'s NORMAL device output.

The printing of bold and italic characters is sometimes better handled by special printer codes. Use *cawf*'s −**c** *config*, −**d***device* and −**f***font* options to produce special font and device output control codes.

Cawf has a small amount of built-in code for the **man**, **me** and **ms** macro packages, but none for any others.

The stacking for the **.so** request is limited.

SEE ALSO

bsfilt(1), colcrt(1), man(7), me(7), ms(7) and nroff(1).