NAME

notangle, noweave, nountangle − noweb, a literate-programming tool

SYNOPSIS

notangle [**−R**rootname ...] [**−filter** command] [**−L**[format]] [file] ...
nountangle [**−ml**|**−m3**|**−c**|**−c++**|**−awk**|**−tex**|**−f77**|**−f90**|**−lisp**|**−matlab**] [**−R**rootname ...] [**−filter** command] [**−w**width] [file] ...
noweave [options] [file] ...

DESCRIPTION

Noweb is a literate-programming tool like Knuth's *WEB*, only simpler. A *noweb* file contains program source code interleaved with documentation. When *notangle* is given a *noweb* file, it writes the program on standard output. When *noweave* is given a *noweb* file, it reads the *noweb* source and produces, on standard output, *LaTeX*, *TeX*, *troff*, or *HTML* source for typeset documentation. *nountangle* converts a literate program into an ordinary program by turning interleaved documentation into comments. The file name '−' refers to standard input.

FORMAT OF NOWEB FILES

A *noweb* file is a sequence of *chunks*, which may appear in any order. A chunk may contain code or documentation. Documentation chunks begin with a line that starts with an at sign (@) followed by a space or newline. They have no names. Code chunks begin with
<<*chunk name*>>=
on a line by itself. The double left angle bracket (<<) must be in the first column. Chunks are terminated by the beginning of another chunk, or by end of file. If the first line in the file does not mark the beginning of a chunk, it is assumed to be the first line of a documentation chunk.

Documentation chunks contain text that is ignored by *notangle* and copied verbatim to standard output by *noweave* (except for quoted code). *noweave* can work with *LaTeX*, plain *TeX*, *troff* or *HTML*. With plain *TeX*, it inserts a reference to a *TeX* macro package, *nwmac*, which defines commands like **\chapter** and **\section.**

Code chunks contain program source code and references to other code chunks. Several code chunks may have the same name; *notangle* concatenates their definitions to produce a single chunk, just as does *tangle*(1). Code chunk definitions are like macro definitions; *notangle* extracts a program by expanding one chunk (by default, the chunk named <<*>>). The definition of that chunk contains references to other chunks, which are themselves expanded, and so on. *notangle*'s output is readable; it preserves the indentation of expanded chunks with respect to the chunks in which they appear.

Code may be quoted within documentation chunks by placing double square brackets (**[[**...**]]**) around it. These double square brackets are ignored by *notangle,* but they may be used by *noweave* to give the code special typographic treatment, e.g., hypertext links. If quoted code ends with three or more square brackets, *noweave* chooses the rightmost pair, so that, for example, **[[a[i]]]** is parsed correctly. The names of code chunks may appear within quoted code unless that quoted code is itself part of the name of a code chunk.

In code, noweb treats unpaired double left or right angle brackets as literal << and >>. To force any such brackets, even paired brackets or brackets in documentation, to be treated as literal, use a preceding at sign (e.g. **@**<<).

Some programming or formatting languages may require a single **@** sign in the first column. Noweb users may achieve this effect by putting a doubled **@@** in the first column; in this position only, it stands for a single **@** sign.

TANGLING

notangle and *nountangle* accept the same set of options, although some options have effects only on one or the other. The options are:

−R*name*
Expand the <<*name*>> code chunk. The **−R** option can be repeated, in which case each chunk is written to the output. If no **−R** option is given, expand the chunk named <<*>>.

−L*format*

Emit line number indications at chunk boundaries. A line number indication identifies the source of the line that follows it. In *format*, **%F** indicates the name of the source file, **%L** indicates the line number of the source file, **%N** indicates a newline, and **%%** indicates a percent sign. A sign and digit may be inserted between the percent sign and the ‘**L**’, in which case the line number will be adjusted by that amount. If *format* is omitted, the default format is that accepted by the C preprocessor: ‘**#line %L "%F"%N**’. When using the **−L***format* option, *notangle* ensures that all text appears in the same column in input and output. *nountangle* ignores this option.

Common format strings include:

C	**−L’#line %L "%F"%N’**
Sun FORTRAN	**−L’\# %L "%F"%N’**
Icon	**−L’#line %−1L "%F"%N’**
Modula-3	**−L’<*LINE %L "%F" *>%N’**
SML/NJ	**−L’(*#line %L "%F"*)’**

To solve the converse problem, that is, to get noweb to do something sensible with **#line** in its input, see the **sharpline** filter in the examples directory.

−t*k* Copy tabs untouched from input to output, and use tabs for indentation, assuming stops every *k* columns. By default, tabs are expanded to spaces with stops every 8 columns.

−filter *cmd*

Filter the *noweb* source through *cmd* after converting it to tool form and before tangling. *notangle* looks for *cmd* first on the user’s **PATH**, then in **/usr/lib/noweb**. Such filters can be used to add features to *notangle*; for an example see **/usr/lib/noweb/emptydefn**. For experts only.

−markup *parser*

Use *parser* to parse the input file. Enables use of noweb tools on files in other formats; for example, the **numarkup** parser understands *nuweb*(1) format. See *nowebfilters*(7) for more information. For experts only.

−awk | −c | −icn | −icon | −ml | −m3 | −pascal | −f77 | −f90 | −tex

When *nountangle* transforms documentation chunks into comments, use the comment format of the language named. **−c** is the default. *notangle* ignores these options.

−w*n* When *nountangle* transforms documentation chunks into comments, create comments on lines of width *n*. *notangle* ignores this option.

WEAVING

Output from *noweave* can be used in *TeX* documents that **\input nwmac,** in *LaTeX* documents that use the **noweb** package (see *nowebstyle(1)*), and in *HTML* documents to be browsed with *Mosaic(1). Noweave* treats code chunks somewhat like *LaTeX list environments.* If the ‘‘**@** ’’ that terminates a code chunk is followed immediately by text, that text follows the code chunk without a paragraph break. If the rest of the line is blank, *noweave* puts *TeX* into ‘‘vertical mode,’’ and later text starts a fresh, indented paragraph.

No page breaks occur in the middle of code chunks unless necessary to avoid an overfull vbox. The documentation chunk immediately preceding a code chunk appears on the same page as that code chunk unless doing so would violate the previous rule.

Noweave inserts no extra newlines in its *TeX* output, so the line numbers given in *TeX* error messages are the same as those in the input file.

noweave has options that dictate choice of formatter and that support different formatting idioms and tools. Basic options are described here; options related to index and cross-reference information are described in the INDEXING AND CROSS-REFERENCE section.

−latex Emit LaTeX, including wrapper in **article** style with the **noweb** package and page style. (Default)

−tex Emit plain TeX, including wrapper with **nwmac** macros.

−html Emit HTML, using HTML wrapper. The output is uninteresting without **-index** or **-x**. The tags **<nowebchunks>** and **<nowebindex>**, on lines by themselves, produce a list of chunks and an index of identifiers, respectively. If these tags are not present, the list and index are placed at the end of the file.

−latex+html
Assume documentation chunks are LaTeX, but generate HTML for code chunks, suitably marked so conversion with *latex2html(1)* yields reasonable output. A LaTeX wrapper is implied, but can be turned off with **-n**. *Use of this option is* **deprecated;** use **−html** with **−filter l2h** instead.

−troff Emit *troff*(1) markup (with no wrapper). The result should be processed with *noroff*(1). Bug reports for **−troff** to Aharon Robbins **<arnold@gnu.org>.**

−n Don't use any wrapper (header or trailer). This option is useful when *noweave*'s output will be a part of a larger document. See also **−delay.**

−filter *cmd*
Filters the *noweb* source through *cmd* after converting it to tool form and before converting to *TeX*. *noweave* looks for *cmd* first on the user's **PATH,** then in **/usr/lib/noweb.** Such filters can be used to add features to *noweave;* for an example, see **/usr/lib/noweb/noxref.krom.** *Noweave* supports up to four filters; one can get more by shell trickery, for example, **-filter "icon.filter | noidx"**. The **-autodefs**, **-x**, **-index**, and **-indexfrom** options are implemented as filters. Filters are executed with the shell's **eval** command, so *cmd* should be quoted accordingly.

−markup *parser*
Use *parser* to parse the input file. Enables use of noweb tools on files in other formats; for example, the **numarkup** parser understands *nuweb*(1) format. See *nowebfilters*(7) for more information. For experts only.

−option *opt*
Adds **\noweboptions{***opt***}** to the *LaTeX* header. See *nowebstyle(1)* for values of *opt.* Normally useful only with the **−latex** option, but **−option longxref** works black magic with **−html.**

−delay By default, *noweave* puts file-name and other information into the output before the first chunk of the program. **−delay** delays that information until after the first documentation chunk, making act a little bit like the *WEB* "limbo." The option is typically used to enable a user to put a specialized *LaTeX* **\documentclass** command and other preamble material in the first documentation chunk (i.e., *before* the first @ sign). This option also forces trailing cross-referencing information to be emitted just before the final chunk, instead of at the end of the document; the final chunk is expected to contain **\end{document}.** The **−delay** option implies the **−n** option.

−t*k* Expand tabs with stops every *k* columns. (Default is to expand every 8 columns.)

−t Copy tabs to the output.

−v Print the pipeline and RCS info on standard error.

INDEXING AND CROSS-REFERENCE

When used with *LaTeX*, *troff*, or *HTML, noweave* can provide indexing and cross-reference information for chunks and for programming-language identifiers. Identifier definitions may be marked by hand using backticks (‘); the **-filter btdefn** option recognizes these markings. For some languages, defintioins may be found automatically using the **-autodefs** option. This section describes the indexing and cross-reference options; it might well be skipped on first reading.

−x For *LaTeX,* add a page number to each chunk name identifying the location of that chunk's definition, and emit cross-reference information relating definitions and uses. For *HTML*, create hypertext links between uses and definitions of chunks. When **noweave -x** is used with *LaTeX,* the control sequence **\nowebchunks** expands to a sorted list of all code chunks.

−index Build cross-reference information (or hypertext links) for defined identifiers. Definitions are those found in the input files by **−autodefs** *language* or by **−filter**btdefn. Requires *LaTeX* or *HTML.* **−index** implies **−x;** including both will generate strange-looking output. *noweave* does not generate cross-references to identifiers that appear in quoted code (**@[[**...**@]]**), but it does generate hypertext links. When **noweave -index** is used with *LaTeX,* the control sequence **\nowebindex**

expands to an index of identifiers.

−indexfrom *index*

Like **−index,** but the identifiers to be indexed are taken from file *index*. See *noindex(1).*

−autodefs *lang*

Discover identifier definitions automatically. Code in chunks must be in language *lang*. Permissible *lang*s vary but may include **tex** or **icon.** Useless without **−index,** which it must precede.

−showautodefs

Show values of *lang* usable with **-autodefs**.

ERROR MESSAGES

If *notangle* or *noweave* encounters a chunk name within documentation, it assumes that this indicates an error, usually misspelling “<<name>>=”. Other error messages should be self-explanatory.

It is incorrect to refer to a chunk that is never defined, but it is OK for chunks to be defined and not used.

EXAMPLES

If you have trouble digesting this man page, you’re not alone. Here are a few examples to get you started. I’ll assume you have a **foo.nw** file with a C program in chunk **<<foo.c>>** and a header file in chunk **<<foo.h>>**, and that your documentation is marked up using *latex(1)*. I’ll show you how to build things using the most common options.

To rebuild your C source, try

```
notangle −L −Rfoo.c foo.nw > foo.c
```

To rebuild your header file, try

```
notangle −Rfoo.h foo.nw | cpif foo.h
```

There are two compromises here. Omitting **−L** keeps **#line** out of your header file, and using **cpif** prevents the command from rewriting **foo.h** unless the contents have changed. Thus, this is good code to put in a Makefile rule.

To build a printed document, run

```
noweave −autodefs c −index foo.nw > foo.tex
```

If you have your own preamble, containing **\documentclass** and all, you will also need the **−delay** option.

To build a web page, run

```
noweave −filter l2h −autodefs c −index −html foo.nw | htmltoc > foo.html
```

Have fun!

FILES

/usr/lib/noweb/markup	markup preprocessor
/usr/lib/noweb/unmarkup	inverts markup
/usr/lib/noweb/nt	notangle proper
/usr/lib/noweb/finduses	find uses of identifiers for index
/usr/lib/noweb/noidx	generate index and cross-reference info
/usr/lib/noweb/toroff	back end to emit *troff*
/usr/lib/noweb/totex	back end to emit *TeX* or *LaTeX*
/usr/lib/noweb/tohtml	back end to emit HTML
/usr/share/texmf/tex/plain/misc/nwmac.tex	formatting *TeX* macros
/usr/share/texmf/tex/plain/misc/noweb.sty	use in *LaTeX* documents; see *nowebstyle(7)*

SEE ALSO

cpif (1), *nodefs*(1), *noroots*(1), *noweb*(1), *noindex*(1), *noroff* (1), *nowebstyle*(7), *nowebfilters*(7)

BUGS

notangle and *nountangle* fail if names used on the command line contain single quotes.

Ignoring unused chunks can cause problems; if a chunk has multiple definitions and one is misspelled, the misspelled definition is silently ignored. *noroots*(1) can be used to catch this mistake.

The *-L* option of *notangle* puts an implicit initial newline in the format string.

The default *LaTeX* pagestyles don't set the width of the boxes containing headers and footers. Since *noweb* code paragraphs are extra wide, this *LaTeX* bug sometimes results in extra-wide headers and footers. The remedy is to redefine the relevant **ps@*** commands; **ps@noweb** in **noweb.sty** can be used as an example.

latex2html(1) mangles some source files.

noweave has too many options, and this man page is too long.

VERSION

This man page is from *noweb* version 2.11b.

AUTHOR

Norman Ramsey, Harvard University. Internet address **nr@eecs.harvard.edu**.
Noweb home page at **http://www.eecs.harvard.edu/˜nr/noweb**.