

knoweb.sty

A L^AT_EX package for noweb

Joseph S. Riel joer@san.rr.com

5 September 2015

Contents

| | | |
|----------|---|----------|
| 1 | User Guide | 2 |
| 1.1 | Introduction | 2 |
| 1.2 | Options | 3 |
| 1.2.1 | Code Font | 3 |
| 1.2.2 | Marginalia | 3 |
| 1.2.3 | Chunk Labeling | 3 |
| 1.2.4 | Chunk Name Breaking | 4 |
| 1.2.5 | Code Breaking | 4 |
| 1.2.6 | Cross References | 4 |
| 1.2.7 | Identifiers | 4 |
| 1.2.8 | Indices | 4 |
| 1.2.9 | Languages | 5 |
| 1.3 | Commands | 5 |
| 1.3.1 | Indices | 5 |
| 1.4 | Style Hooks | 5 |
| 2 | Code | 6 |
| 2.1 | Chunk Names | 8 |
| 2.2 | Margin Tags | 9 |
| 2.3 | Adjusting Placement of Code on the Page | 10 |
| 2.4 | Page-breaking strategy | 11 |
| 2.5 | Environments for Setting Code | 11 |
| 2.5.1 | Code Font Size | 12 |
| 2.5.2 | Vertical Spacing | 13 |
| 2.5.3 | Code Breaking | 15 |
| 2.6 | Comments in Code | 16 |
| 2.7 | The noweb Page Style | 17 |
| 2.8 | Chunk Cross-Reference | 18 |
| 2.8.1 | Macros for the Definition Line | 19 |
| 2.8.2 | Package Options | 20 |

| | | |
|--------|--|-----------|
| 2.9 | Page Ranges | 22 |
| 2.10 | Sub-page References | 28 |
| 2.10.1 | <code>\subpageref</code> | 28 |
| 2.10.2 | <code>\subpagepair</code> | 29 |
| 2.10.3 | <code>\sublabel</code> | 30 |
| 2.10.4 | <code>\newsublabel</code> | 31 |
| 2.10.5 | Assign Sub-page Reference Options | 34 |
| 2.10.6 | Converting Integers to Strings | 34 |
| 2.11 | WEB-like Chunk Numbering | 37 |
| 2.12 | Indexing (Identifier Cross-Reference) Support | 38 |
| 2.12.1 | Tracking Definitions and Uses | 38 |
| 2.12.2 | Subscripted Identifiers | 39 |
| 2.12.3 | Writing Lists with Commas and Conjunction | 41 |
| 2.12.4 | Improved Index Code | 41 |
| 2.12.5 | Supporting a Mini-Index at the End of each Chunk | 43 |
| 2.12.6 | Support for Chunk and Identifier Indices | 45 |
| 2.13 | Support for Hypertext | 51 |
| 2.14 | Support for Hypertext Translation to HTML | 52 |
| 2.15 | Support for Prettyprinting | 52 |
| 2.16 | Language-Dependent Macros | 53 |
| 2.16.1 | Support for English | 54 |
| 2.16.2 | Support for Portuguese | 56 |
| 2.16.3 | Support for Spanish | 57 |
| 2.16.4 | Support for French | 59 |
| 2.16.5 | Support for German | 61 |
| | Chunks | 63 |
| | Index | 66 |
| | References | 68 |

1 User Guide

1.1 Introduction

This document describes the \LaTeX package `knoweb.sty`. It is based on `noweb.sty` [3], much of the code and comments are copied verbatim. There are, however, some significant differences:

\TeX Support for plain \TeX has been removed.

Options The `\noweboptions` macro has been removed; package options are passed as standard \LaTeX package options. Two new options are `spanish` and `twocolumn`. One drawback of removing `\noweboptions` is that the `shift` filter for `noweave` used it.

Line Size The dimension macros `\codehsize` and `\nwdefspace` have been removed. The purpose of `\codehsize`, so far as I could ascertain, was to prevent overflow box warnings, however, it was not clear that it did so. The purpose of `\nwdefspace` was to align the chunk cross-references on the definition line; this is better done by using `\linewidth`.

Margin Tags The macro `\nwthemargintag` has been removed, its functionality has been incorporated into `\nwmargin`. The macro `\nwmargin` was removed; `\marginparsep` now sets the space between a margin tag and the text body. This was done so that the option `twocolumn` could be added to the package; when it is active margin tags are set using `\marginpar`, modified by the `mparhack.sty` package. Tags in the second column are set in the right margin.

1.2 Options

Package options to knoweb are passed in the standard L^AT_EX fashion, as a list of comma-separated fields enclosed in square brackets to the `\usepackage` command. For example

```
\usepackage[footnotesizecode,longchunks,spanish]{knoweb}
```

passes the options `footnotesizecode`, `longchunks`, and `spanish` to knoweb.

1.2.1 Code Font

Options are provided to set the font-family and font-size of the typset code. Each standard L^AT_EX font-size setting declaration, `\tiny` to `\Huge`, has a corresponding option, with code appended. Thus the option `footnotesizecode` set the font size of the typeset code to `\footnotesize`. The default is `normalsizecode`.

1.2.2 Marginalia

Tags can be inserted into the margins indicating the page number or sub-page number of code chunks. Several options affect the style and placement of these tags.

The options `margin` and `nomargin` enable and disable the printing of these tags. The default is `margin`.

Normally margin tags are printed in the left margin, in-line with a chunk definition. However, if the `twocolumn` option is given and the document is in two-column mode, the tags for chunks that start in the second column are printed in the right margin. The `twocolumn` option does not need to be explicitly passed to knoweb; the more usual method is to pass to `\documentclass`—L^AT_EX's package option mechanism passes it to knoweb.

1.2.3 Chunk Labeling

Several options are provided for labeling code chunks.

alphasubpage: label code chunk definitions with the field `\pagenumber\subpage`, where `\subpage` is a lower-case character, starting at 'a' and incremented for each new code chunk definition on a page. This is the default.

numsubpage Similar to **alphasubpage** but $\langle subpage \rangle$ is an integer starting from 1 and separated from $\langle pagenumber \rangle$ by a decimal.

nosubpage Turn off the subpage designator.

webnumbering Sequentially number code chunks through the document.

shortstrings Used with **alphasubpage**; limits the $\langle subpage \rangle$ designator to a single character ('a' through 'z'). This is the default.

longstrings Used with **alphasubpage** to extend the $\langle subpage \rangle$ designator to multiple characters. Only needed for those rare documents with more than 26 chunk definitions on a single page.

1.2.4 Chunk Name Breaking

The option **donotbreakchunknames** prevents breaking chunk names in the documentation across lines. The default behaviour is to permit such breaks.

1.2.5 Code Breaking

The option **breakcode** permits breaking code to use less vertical white space. The parameter **\nwbreakcodespace** controls the amount of white space that may be left on a page, see §1.4.

1.2.6 Cross References

Code chunk definitions can include cross-references to other code chunks. Three mutually exclusive options are provided:

shortxref: Insert tags in the definition line of a chunk assignment. This is the default.

longxref: Insert tags in small paragraphs, at the end of a chunk, as in Knuth.

noxref: Insert no cross references.

1.2.7 Identifiers

Subscripts or hyperlinks can be used with indexed symbols to point to their definitions. The options **subscriptidents** and **nosubscriptidents** turn on and off the subscripting. The options **hyperidents** and **nohyperidents** turn on and off hyperlinks.

1.2.8 Indices

The **longchunks** option adds indices after each chunk name in the chunk index (produced by **\nowebchunks**) indicating where it is defined and where it is used.

The **externalindex** option modifies the operation of **\nowebindex** to insert an external index, one created by **noindex**.

1.2.9 Languages

Support for inserting text appropriate text for a specified language in the printed chunk cross-references is provided by the language options `english`, `american`, `french`, `frenchb`, `german`, `ngerman`, `spanish`, and `portuges`. The default is `english`. The options `american` and `english` are identical, as are `frenchb` and `french`, and `german` and `ngerman`.

1.3 Commands

1.3.1 Indices

The command `\nowebchunks` creates an index of the chunks. If the `longchunks` package option is selected, each entry in the index includes links to the definition and uses of the chunk. A preamble describing these links is also printed, unless the starred version of the command, `\nowebchunks*`, is used.

The command `\nowebindex` creates an index of the identifiers. Each entry in the index includes links to the definition and uses of the identifier. A preamble describing these links is also printed, unless the starred version of the command, `\nowebindex*`, is used. The package option `externalindex` changes the operation to use an external index, see 1.2.8.

1.4 Style Hooks

`\nwcodeindent`: length parameter that sets the indentation of the code.

Use `\setlength{\nwcodeindent}{\langle new value \rangle}` to reassign it.

`\nwcodecommentsep`: length parameter that sets the vertical separation between the end of a code chunk and any following annotation.

Use `\setlength{\nwcodecommentsep}{\langle new value \rangle}` to reassign it.

`\nwcodepenalty`: counter that sets the penalty for page breaking between lines of a code chunk. The default is `\@highpenalty`.

Use `\nwcodepenalty=\langle new value \rangle` to reassign it.

`\nwtagstyle`: macro that sets the font in which tags are displayed.

Use `\renewcommand{\nwtagstyle}{\langle new definition \rangle}` to reassign it.

`\nwbreakcodespace`: Used with the package option `breakcode`, see §???. It controls the amount of vertical space left on a page when a code chunk is broken. The default is `0.2in`.

Use `\setlength{\nwbreakcodespace}{\langle new definition \rangle}` to reassign it.

2 Code

6a $\langle \text{knoweb.sty 6a} \rangle \equiv$ 6d
 $\langle \text{identification 6b} \rangle$
 $\langle \text{initial code 6c} \rangle$
 $\langle \text{declaration of options 9a} \rangle$
 $\langle \text{execution of options 9e} \rangle$
`\ProcessOptions\relax`

6b $\langle \text{identification 6b} \rangle \equiv$ (6a)
`% knoweb.sty -- LaTeX support for noweb`
`% DO NOT read or edit this file. Use knoweb.nw instead.`
`\ProvidesPackage{knoweb}`
`\NeedsTeXFormat{LaTeX2e}`

The $\langle \text{initial code 6c} \rangle$ consists of control sequences that must be defined before options are either declared or executed.

6c $\langle \text{initial code 6c} \rangle \equiv$ (6a) 9b
`\RequirePackage{calc}`

Most code is set in an environment in which `\setupcode` has been executed. In this environment, only `\`, `{`, and `}` have their usual categories; every other character represents itself. Appropriate `\chardefs` ensure that the special characters can be escaped with a backslash.

6d $\langle \text{knoweb.sty 6a} \rangle + \equiv$ <6a 6e
`\chardef\other=12`
`\newcommand{\setupcode}{%`
`\chardef\=\`\`
`\chardef\{=\`\{`
`\chardef\}=\`\}`
`\catcode'\$=\other`
`\catcode'\&=\other`
`\catcode'\#=\other`
`\catcode'\%=\other`
`\catcode'\~= \other`
`\catcode'_=\other`
`\catcode'\^=\other`
`\catcode'\\"=\other %fixes problem with german.sty`
`\obeyspaces\Tt`
`}`

Defines:

`\setupcode`, used in chunks 12a, 14a, and 15d.

Uses `\Tt` 10c.

`\nwendquote` is called after quoted code. It resets the spacefactor. There is no corresponding `\nwbeginquote`.

6e $\langle \text{knoweb.sty 6a} \rangle + \equiv$ <6d 7a
`\newcommand{\nwendquote}{\relax\ifhmode\spacefactor=\@m \fi}`

Defines:

`\nwendquote`, never used.

`\eatline` is used to consume newlines that should be ignored, for example, the newlines at the end of ‘@ %def *<identifiers>*’ lines. It is inserted by `noweave`.

```
7a <knoweb.sty 6a>+≡ <6e 7b>
\newcommand\eatline[1]{%
{\catcode'\^M=\active
\gdef\eatline#1^M{\relax}}}
```

Defines:

`\eatline`, never used.

`\nwnewline` is inserted by `noweave` in multiline code (multiple lines enclosed by ‘[[’ and ‘]]’) in a documentation chunk.

```
7b <knoweb.sty 6a>+≡ <7a 7c>
\newcommand{\nwnewline}{\ifvmode\else\hfil\break\leavevmode\hbox{}\fi}
```

Defines:

`\nwnewline`, never used.

Within a code environment, it may be necessary to restore the category codes in order to set a module (chunk) name. This hack doesn’t properly restore “ for use in `german.sty`.

```
7c <knoweb.sty 6a>+≡ <7b 7d>
\newcommand{\setupmodname}{%
\catcode'\$=3
\catcode'\&=4
\catcode'\#=6
\catcode'\%=14
\catcode'\~=13
\catcode'\_ =8
\catcode'\^ =7
\catcode'\ =10
\catcode'\^M=5
\let\{=\nwlbrace
\let\}=\nwrbrace
%bad news — don’t know what catcode to give "
\Rm}
```

Uses `\Rm 10c`.

Let active space equal to control space. The assignment is moved to the beginning of the document so that it does not interfere with the `verbatim` package. See [2, p. 381].

```
7d <knoweb.sty 6a>+≡ <7c 8a>
{\obeyspaces\AtBeginDocument{\global\let =\ }}
```

2.1 Chunk Names

noweave brackets uses of chunk names with `\LA` and `\RA`, which handle the angle brackets, font, and environment.

```
8a <knoweb.sty 6a>+≡ <7d 8b>
    \newcommand{\LA}{\begingroup\nw@chunknamebox\bgroup\setupmodname\It$\langle$}
    \newcommand{\RA}{\l/$\rangle$\egroup\endgroup}
```

Defines:

`\LA`, used in chunks 8c and 47b.

`\RA`, used in chunks 8c and 47b.

Uses `\It` 10c and `\nw@chunknamebox` 9a.

`\nw@equivbox` and `\nw@plusequivbox` are used to set the ‘ \equiv ’ and ‘ $+ \equiv$ ’ that open a chunk definition or its continuation.

```
8b <knoweb.sty 6a>+≡ <8a 8c>
    \newsavebox{\nw@equivbox}
    \savebox{\nw@equivbox}{\equiv$}
    \newsavebox{\nw@plusequivbox}
    \savebox{\nw@plusequivbox}{\mathord{+}\mathord{\equiv}$}
```

Defines:

`\nw@equivbox`, used in chunk 8c.

`\nw@plusequivbox`, used in chunk 8c.

noweave brackets definitions of chunk names with `\moddef` and either `\endmoddef` or `\plusendmoddef`.

```
8c <knoweb.sty 6a>+≡ <8b 10a>
    \newenvironment{moddef}
    {\leavevmode
    \kern-\nwcodeindent
    \LA
    }
    {\RA
    \ifmmode\equiv
    \else\unhcopy\nw@equivbox
    \fi
    }
    \newcommand{\plusendmoddef}{%
    \RA
    \ifmmode\mathord{+}\mathord{\equiv}
    \else\unhcopy\nw@plusequivbox
    \fi
    }
```

Defines:

`\plusendmoddef`, never used.

`moddef`, never used.

Uses `\LA` 8a, `\nw@equivbox` 8b, `\nw@plusequivbox` 8b, `\nwcodeindent` 10d, and `\RA` 8a.

By default, chunk names may be broken across lines. Declare an option for changing this.

9a *<declaration of options 9a>+≡* (6a) 9c>
`\DeclareOption{donotbreakchunknames}`
`{\let\nw@chunknamebox=\mbox}`

Defines:

`\nw@chunknamebox`, used in chunks 8a, 9b, and 12a.
`donotbreakchunknames`, never used.

Allocate `\nw@chunknamebox` and assign its default so that chunk names may be broken across lines.

9b *<initial code 6c>+≡* (6a) <6c 11b>
`\newcommand{\nw@chunknamebox}{}%`
`\let\nw@chunknamebox=\relax`

Uses `\nw@chunknamebox` 9a.

2.2 Margin Tags

Within a code environment, margin tags are used to mark sub-page numbers in the margins, separated by `\marginparsep`.¹ The margin tag normally goes in the left column, adjacent to the chunk definition; however, if `twocolumn` mode is used, the margin tag for a definition in the second column is put into the right margin. A long chunk name may cause the cross-references to interfere with this margin tag.

9c *<declaration of options 9a>+≡* (6a) <9a 9d>
`\DeclareOption{nomargintag}{\let\nwmargintag=\@gobble}`
`\DeclareOption{margintag}{%`
`\newcommand{\nwmargintag}[1]{%`
`\leavevmode`
`\kern-\nwcodeindent`
`\llap{#1\kern\marginparsep}%`
`\kern\nwcodeindent}}`

Defines:

`\nwmargintag`, used in chunk 9d.
`margintag`, used in chunk 9e.
`nomargintag`, used in chunk 34a.

Uses `\nwcodeindent` 10d.

9d *<declaration of options 9a>+≡* (6a) <9c 11a>
`\DeclareOption{twocolumn}{%`
`\def\nwmargintag{\leavevmode\marginpar}}`

Defines:

`twocolumn`, never used.

Uses `\nwmargintag` 9c.

9e *<execution of options 9e>≡* (6a) 12c>
`\ExecuteOptions{margintag}`

Uses `margintag` 9c.

¹This package originally used `\nwmargin glue` to set the spacing between the tag and the text. With the addition of `twocolumn` option, which uses `\marginpar` to typeset the tags, it was simpler to use `\marginparsep`.

10a `<knoweb.sty 6a>+≡` <8c 10b>
`\if@twocolumn\RequirePackage{mparhack}\fi`

`\nwtagstyle` determines the style in which tags are displayed.

10b `<knoweb.sty 6a>+≡` <10a 10c>
`\newcommand{\nwtagstyle}{\footnotesize\Rm}`

Defines:

`\nwtagstyle`, used in chunks 19b and 47b.

Uses `\Rm 10c`.

We have to be careful with font-changing in the presence of different font-selection schemes. In the L^AT_EX New Font Selection Scheme something like `\it\tt` will attempt to use an italic typewriter font. Thus we define new commands like `\Tt` which will work with both the Plain and old and new L^AT_EX schemes. A problem with these definitions arises with NFSS: in math mode they do not work unless the `oldfont backwards-compatibility` option is in effect. For the moment, you can get round this by using `\mbox`.

If you wanted code set in a different font, you could re-define `\Tt`. [L^AT_EX2e actually behaves like OFSS, but the extra `\reset@font` does no harm.]

10c `<knoweb.sty 6a>+≡` <10b 10d>

```
%here is support for the new-style (capitalized) font-changing commands
%thanks to Dave Love
\newcommand{\Rm}{\normalfont\rm}
\newcommand{\It}{\normalfont\it}
\newcommand{\Tt}{\normalfont\tt}
\newcommand{\Bf}{\normalfont\bf}
```

Defines:

`\Bf`, never used.

`\It`, used in chunk 8a.

`\Rm`, used in chunks 7c, 10b, 17, and 18a.

`\Tt`, used in chunks 6d, 17, 19a, 43a, 44b, and 49b.

2.3 Adjusting Placement of Code on the Page

10d `<knoweb.sty 6a>+≡` <10c 12a>

```
\newlength{\nwcodeindent}
\setlength{\nwcodeindent}{10pt}
```

Defines:

`\nwcodeindent`, used in chunks 8c, 9c, 13c, 18a, and 44b.

Style
hook

Style
hook

`\noweboptions{shift}` is used to shift the whole page left to make room for wide code lines. It may be emitted by nowave `-shift`, or it might be given by a user. JR: The original code only shifted even-numbered (verso) pages when two-sided printing was in effect. I see no reason for that, so this implementation always shifts both even and odd pages.

11a \langle declaration of options 9a $\rangle + \equiv$ (6a) \langle 9d 12b \rangle

```

\DeclareOption{shift}{%
  \setlength{\dimen@}{-0.8in}
  \addtolength{\evensidemargin}{\dimen@}
  \addtolength{\oddsidemargin}{\dimen@}
}
\DeclareOption{noshift}{\relax}

```

Defines:

- noshift, never used.
- shift, never used.

2.4 Page-breaking strategy

We want to insert penalties aiming for:

1. No page breaks in the middle of a code chunk unless necessary to avoid an overfull vbox;
2. Documentation immediately preceding a code chunk should appear on the same page as that code chunk unless doing so would violate rule 1.

`\filbreak` is useful for this sort of thing (see *The T_EXbook*) and is used to encourage breaks at the right places between chunks. Appropriate penalties are inserted elsewhere, between code lines in particular.

2.5 Environments for Setting Code

`\nwbegincode` and `\nwendcode` are used by nowave to bracket code chunks. The `webcode` environment is intended for users who want to paste nowave output into papers.

The definition of `\nwbegincode` is based on the verbatim implementation in `verbatim.sty`.

11b \langle initial code 6c $\rangle + \equiv$ (6a) \langle 9b 14d \rangle

```

\newcommand{\nwbegincode}[1]{%
  \begingroup
  \langle \nwbegincode separation and penalties 13b \rangle
  \@begincode }
\newcommand{\nwendcode}{%
  \endtrivlist \endgroup \filbreak}% keeps code on 1 page
\newenvironment{webcode}{%
  \@begincode
}{%
  \endtrivlist}

```

Defines:

- `\nwbegincode`, never used.
- `\nwendcode`, used in chunk 15b.
- `webcode`, never used.

This is just common code between `\nwbegincode` and `webcode`.

12a `<knoweb.sty 6a>+≡` `<10d 13a>`

```

\def\@begincode{%
  <\trivlist cliché (à la verbatim) 13c>
  <\obeylines setup 13d>
  <zap ligatures, fix spaces 14a>
  \noweb size
  \setupcode
  \let\nw@chunknamebox=\mbox}

```

Uses `\nw@chunknamebox 9a` and `\setupcode 6d`.

2.5.1 Code Font Size

The command `\noweb size` governs the size at which code is set. Slitex users might try

```
\newcommand{\noweb size}{\normal size\baselineskip=20pt \parskip=5pt }
```

to avoid code lines that are too far apart.

12b `<declaration of options 9a>+≡` `(6a) <11a 15b>`

```

\DeclareOption{tinycode}{\let\noweb size=\tiny}
\DeclareOption{footnotesizecode}{\let\noweb size=\footnotesize}
\DeclareOption{scriptsizecode}{\let\noweb size=\scriptsize}
\DeclareOption{smallcode}{\let\noweb size=\small}
\DeclareOption{normalsizecode}{\let\noweb size=\normal size}
\DeclareOption{largecode}{\let\noweb size=\large}
\DeclareOption{Largecode}{\let\noweb size=\Large}
\DeclareOption{LARGEcode}{\let\noweb size=\LARGE}
\DeclareOption{hugecode}{\let\noweb size=\huge}
\DeclareOption{Hugecode}{\let\noweb size=\Huge}

```

Defines:

- `footnotesizecode`, never used.
- `Hugecode`, never used.
- `hugecode`, never used.
- `LARGEcode`, never used.
- `Largecode`, never used.
- `largecode`, never used.
- `normalsizecode`, used in chunk 12c.
- `scriptsizecode`, never used.
- `smallcode`, never used.
- `tinycode`, never used.

12c `<execution of options 9e>+≡` `(6a) <9e 21b>`

```

\ExecuteOptions{normalsizecode}

```

Uses `normalsizecode 12b`.

2.5.2 Vertical Spacing

`\nwcodetopsep` is the glue placed before code chunks.

13a `\knoweb.sty 6a` + \equiv <12a 13e>

```
\newlength{\nwcodetopsep}
\setlength{\nwcodetopsep}{3pt plus 1.2pt minus 1pt}
```

Defines:

`\nwcodetopsep`, used in chunk 13b.

13b `\nwbegincode separation and penalties 13b` \equiv (11b)

```
\setlength{\topsep}{\nwcodetopsep}
\@beginparpenalty=\@highpenalty
\@endparpenalty=-\@highpenalty
```

Defines:

`\@beginparpenalty`, never used.

`\@endparpenalty`, never used.

Uses `\nwcodetopsep` 13a.

The `\trivlist` cliché isn't quite a cliché because we adjust `\leftskip` for indentation by `\nwcodeindent`.

13c `\trivlist cliché (à la verbatim) 13c` \equiv (12a)

```
\trivlist \item[]%
\setlength{\leftskip}{\@totalleftmargin+\nwcodeindent}%
\setlength{\rightskip}{\z@}%
\parskip\z@ \parindent\z@ \parfillskip\@flushglue
```

Uses `\nwcodeindent` 10d.

The penalty inserted between verbatim lines would normally be `\interlinepenalty`, but we want to prohibit breaks there.

Note the bug lurking somewhere in this code, as reported by Steven Ooms:

I have some lay-out problems in the documentation chunks. When using the (La)TeX commands `\hline` or `\vtop` the right margin is always extended far beyond the page margin after the first code chunk has been typeset. I'm still looking for the exact cause of it, but to me it seems that LaTeX supposes for those commands that the line width for the documentation chunk is as large as that for code chunks, which isn't true in reality.

13d `\obeylines setup 13d` \equiv (12a)

```
\@@par
\def\par{\leavevmode\null \@@par \penalty\nwcodepenalty}%
\obeylines
```

Uses `\nwcodepenalty` 13e.

`\nwcodepenalty` is the penalty for breaking between lines in a code chunk. If you set it to 10000, code will never be broken across pages. I guess this should be settable in `\noweboptions`.

13e `\knoweb.sty 6a` + \equiv <13a 14c>

```
\newcount\nwcodepenalty \nwcodepenalty=\@highpenalty
```

Defines:

`\nwcodepenalty`, used in chunks 13d and 18.

Style
hook

The cursing-chunk accounts for the addition of a mess of characters to those reset by `\@noligs` in L^AT_EX2e.

14a `<zap ligatures, fix spaces 14a>≡` (12a)
`\@noligs <make all those damn active characters “other” 14b>`
`\setupcode \frenchspacing \@vobeyspaces`

Uses `\setupcode 6d`.

We cannot make the backquote, “”, `\other` because then we get ligatures. Why Knuth put these ligatures in the `tt` font I wish I knew. But we step on all the others.

14b `<make all those damn active characters “other” 14b>≡` (14a)
`\ifx\verbatim@nolig@list\undefined\else`
`\let\do=\nw@makeother \verbatim@nolig@list \do@noligs\‘`
`\fi`

Uses `\nw@makeother 14c`.

14c `<knoweb.sty 6a>+≡` <13e 15a>
`\def\nw@makeother#1{\catcode‘#1=12 }`

Defines:

`\nw@makeother`, used in chunks 14b and 38b.

noweave uses `\nwbegindocs{nnn}` and `\nwenddocs` to bracket documentation chunks. If a documentation chunk does not continue the current paragraph, noweave inserts `\nwdocspar`, which uses `\filbreak` in an attempt to keep the documentation chunk on the same page as the code chunk that follows it. (The code chunk will have another `\filbreak` after it—see `\nwbegincode`.) `\nwbegindocs` does not start a new paragraph if the previous chunk did not end one, i.e. did not cause T_EX to enter vmode; if it does start a new paragraph, it is only indented by the use of `\nwdocspar`.

14d `<initial code 6c>+≡` (6a) <11b 19a>
`\newcommand{\nwbegindocs}[1]{\ifvmode\noindent\fi}`
`\newcommand{\nwenddocs}{\relax}`
`\newcommand{\nwdocspar}{}`
`\let\nwdocspar=\filbreak`

Defines:

`\nwbegindocs`, never used.

`\nwdocspar`, used in chunk 15b.

`\nwenddocs`, never used.

2.5.3 Code Breaking

Some people do not like the vertical white space that noweb leaves at the bottom of pages. The style option `breakcode` permits breaking code to use less white space. The parameter `\nwbreakcodespace` controls the amount of white space left.

15a `<knoweb.sty 6a>+≡` <14c 15c>

```

\newcommand\nw@semifilbreak[1]{%
  \vskip0pt plus#1
  \penalty-200
  \vskip0pt plus -#1}
\newlength{\nwbreakcodespace}
\setlength{\nwbreakcodespace}{0.2in}

```

Defines:

- `\nw@semifilbreak`, used in chunk **15b**.
- `\nwbreakcodespace`, never used.

15b `<declaration of options 9a>+≡` (6a) <12b 20b>

```

\DeclareOption{breakcode}{%
  \renewcommand{\nwdocspar}{\nw@semifilbreak{0.2in}}%
  \renewcommand{\nwendcode}{\endtrivlist\endgroup}%
}

```

Defines:

- `breakcode`, never used.

Uses `\nw@semifilbreak` **15a**, `\nwdocspar` **14d**, and `\nwendcode` **11b**.

The page-breaking strategy implies ragged bottom pages, so we should turn it on in general (this is relevant for the report style):

15c `<knoweb.sty 6a>+≡` <15a 15d>

```

\raggedbottom

```

noweave doesn't bracket quoted code with `\code` and `\edoc` any more. It probably should do something nifty, just to make T_EX hackers happy, but it doesn't.

15d `<knoweb.sty 6a>+≡` <15c 16a>

```

\newcommand{\code}{%
  \leavevmode
  \begingroup
    \setupcode
    \@vobeyspaces
    \obeylines}
\newcommand{\edoc}{%
  \endgroup}

```

Defines:

- `\code`, never used.
- `\edoc`, never used.

Uses `\setupcode` **6d**.

2.6 Comments in Code

By default, comments in a code chunk are typeset in the same font as the code, a fixed-width typewriter font. There are applications where it is useful to typeset comments in code; to do this effectively, a noweb filter must be used to identify the comments to be typeset and insert the appropriate markup. This section assigns a \LaTeX environment and commands that can be used by such a filter.

Assign an environment, `nwtypesetcomment` that can be used to typeset a comment in a code chunk. Redefine `\par` to call `\nw@commentpar`. Call `\nwtypesetcommentfont` to set the font; this can be customized by the user. Reassign the category codes of most of the special characters so that they behave as usual.

```
16a <knoweb.sty 6a>+≡ <15d 16b>
  \newenvironment{nwtypesetcomment}
  {\def\par{\nw@commentpar}%
   \nwtypesetcommentfont
   \catcode'\$=3
   \catcode'\&=4
   \catcode'\^M=5
   \catcode'\#=6
   \catcode'\^=7
   \catcode'\_ =8
   \catcode'\ =10
   \catcode'\^I=10
   \catcode'\~=13
   \catcode'\%=14
   \let\{=\nwlbrace
   \let\}=\nwrbrace
   \let\\=\nwbackslash
  }
  {}}
```

Defines:

`nwtypesetcomment`, never used.

Uses `\nw@commentpar` 16c and `\nwtypesetcommentfont` 16b.

Assign a command that sets the font used in an inline comment. The default assignment is to use the roman family.

```
16b <knoweb.sty 6a>+≡ <16a 16c>
  \newcommand{\nwtypesetcommentfont}{\rmfamily}
```

Defines:

`\nwtypesetcommentfont`, used in chunk 16a.

```
16c <knoweb.sty 6a>+≡ <16b 17>
  \newcommand{\nw@commentpar}{}
  \let\nw@commentpar=\par
```

Defines:

`\nw@commentpar`, used in chunk 16a.

2.7 The noweb Page Style

Headers contain file name, date, and page number. noweb emits `\nwfilename{name}` for each new file. In the noweb page style, new files cause page breaks; otherwise they are ignored.

```
17 <knoweb.sty 6a>+≡ <16c 18a>
  \newlength{\@original@textwidth}
  \newcommand{\ps@noweb}{%
    \setlength{\@original@textwidth}{\textwidth}%
    \let\@mkboth=\@gobbletwo
    \let\@oddfoot=\relax
    \let\@evenfoot=\relax
    \if@twoside
      \def\@evenhead{\makebox[\@original@textwidth]{%
        \Rm \makebox[5ex][l]{\thepage}{\Tt\leftmark}\hfill\today}}%
      \def\@oddhead{\makebox[\@original@textwidth]{%
        \Rm \today\hfill{\Tt\leftmark}\makebox[5ex][r]{\thepage}}}%
    \else
      \def\@oddhead{\makebox[\@original@textwidth]{%
        \Rm \today\hfill{\Tt\leftmark}\makebox[5ex][r]{\thepage}}}%
      \let\@evenhead=\@oddhead
    \fi
    \let\chaptermark=\@gobble
    \let\sectionmark=\@gobble
    \let\subsectionmark=\@gobble
    \let\subsubsectionmark=\@gobble
    \let\paragraphmark=\@gobble
    \let\subparagraphmark=\@gobble
    \def\nwfilename{%
      \begingroup
        \let\do\@makeother\dospecials
        \catcode'\{=1
        \catcode'\}=2
        \nw@filename}%
    \def\nw@filename##1{%
      \endgroup
      \markboth{##1}{##1}%
      \let\nw@filename=\nw@laterfilename}%
  }
  \def\nw@laterfilename#1{\endgroup\clearpage\markboth{#1}{#1}}
  \let\nwfilename=\@gobble
```

Defines:

- \nw@laterfilename, never used.
- \nwfilename, never used.
- \ps@noweb, never used.

Uses \Rm 10c and \Tt 10c.

2.8 Chunk Cross-Reference

The macros `\nwalsodefined`, `\nwused`, and `\nwnotused` are emitted by the noweb cross-referencers. If unused chunks are output chunks, a filter can slip in `\let\nwnotused=\nwoutput`. The style uses `\nwcodecomment` for all annotations that follow code chunks. Fiddling with it can change the appearance of the output. Note that `\nwcodecomment` is used after `\nwbegincode`, with `\obeylines` in effect. Because linebreaking can occur here, we need to change the `\interlinepenalty`. A little vertical space (`\nwcodecommentsep`) appears before the first comment.

We firkled with `\rightskip` in `\nwbegincode` above; now we want to reset it so that paragraphs are the normal width (`\textwidth`, possibly less `\nwcodeindent`) and set ragged right. This is done as usual by making `\rightskip` naturally zero but stretchable.

18a `<knoweb.sty 6a>+≡` <17 20a>

```

\newcommand{\nwcodecomment}[1]{%
  @@par\penalty\nwcodepenalty
  <add \nwcodecommentsep if this is the first \nwcodecomment 18b>%
  \hspace{-\nwcodeindent}{%
    \setlength{\rightskip}{\z@ plus1in}%
    \interlinepenalty\nwcodepenalty
    \let\\\relax\footnotesize\Rm #1\@@par\penalty\nwcodepenalty}}

```

Defines:

`\nwcodecomment`, used in chunks 19a, 43, and 44.

Uses `\nwcodeindent 10d`, `\nwcodepenalty 13e`, and `\Rm 10c`.

18b `<add \nwcodecommentsep if this is the first \nwcodecomment 18b>≡` (18a)

```

\if@firstnwcodecomment
  \vskip\nwcodecommentsep
  \penalty\nwcodepenalty
  \@firstnwcodecommentfalse
\fi

```

Uses `\@firstnwcodecommentfalse 21d`, `\if@firstnwcodecomment 21d`, `\nwcodecommentsep 21d`, and `\nwcodepenalty 13e`.

This stuff is used at the end of a chunk by the `longxref` package option.

19a `<initial code 6c>+≡` (6a) <14d 19b>

```

\newcommand\nw@alsodefined[1]{%
  \nwcodecomment{\nw@langdepdef\space\nw@langdepin\space\nw@chunkcommachunketc{#1}.}}
\newcommand\nw@used[1]{%
  \nwcodecomment{\nw@langdepcud\space\nw@langdepin\space\nw@chunkcommachunketc{#1}.}}
\newcommand\nw@notused[1]{%
  \nwcodecomment{\nw@langdeprtc.}}
\newcommand\nw@output[1]{%
  \nwcodecomment{\nw@langdepcwf\space{\Tt\nw@stripstar#1*\stripped}.}}
\def\nw@stripstar#1#2\stripped{#1}

```

Defines:

`\nw@alsodefined`, used in chunk 20c.
`\nw@notused`, used in chunk 20c.
`\nw@stripstar`, never used.
`\nw@used`, used in chunk 20c.
`\nw@output`, never used.

Uses `\nw@chunkcommachunketc` 22a, `\nw@langdepcud` 54a, `\nw@langdepcwf` 54a, `\nw@langdepdef` 54a, `\nw@langdepin` 54a, `\nw@langdeprtc` 54a, `\nwcodecomment` 18a, and `\Tt` 10c.

2.8.1 Macros for the Definition Line

When the `shortxref` package option is enabled, additional information, in the form of tags, is added to the definition line of a chunk. The T_EX file generated by `noweave` calls, in order, the following macros with appropriate arguments:

`\nw@startdeflinemarkup`: push the tags to the right margin, ensure there is a minimal space between the chunk name and the tags, and begin a group in which the font style is set with `\nwtagstyle`.

`\nw@usesondefline`: prints a list, enclosed in parentheses, of tags to the chunks that use this chunk. Its single argument is a list of keys, $\{\langle key_1 \rangle \langle key_2 \rangle \dots \langle key_n \rangle\}$, each key corresponds to the argument of a `\sublabel` command.

`\nw@prevnextdefs`: print tags to the previous and next chunks of the same name.

`\nw@enddeflinemarkup`: end the group started by `\nw@startdeflinemarkup`.

19b `<initial code 6c>+≡` (6a) <19a 21c>

```

\newcommand{\nw@startdeflinemarkup}{%
  \nobreak\hskip .5em plus 1fill\nobreak\beginngroup\nwtagstyle}
\newcommand{\nw@usesondefline}[1]{(\nw@chunkspacechunketc{#1})}
\newcommand{\nw@prevnextdefs}[2]{%
  \ifx\relax#1\else\nobreakspace\nobreakspace\nwprevdefptr{#1}\fi
  \ifx\relax#2\else\nobreakspace\nobreakspace\nwnextdefptr{#2}\fi}
\newcommand{\nw@enddeflinemarkup}{\endgroup}

```

Defines:

`\nw@enddeflinemarkup`, used in chunk 20b.
`\nw@prevnextdefs`, used in chunk 20b.
`\nw@startdeflinemarkup`, used in chunk 20b.
`\nw@usesondefline`, used in chunk 20b.

Uses `\nw@chunkspacechunketc` 27d, `\wnextdefptr` 20a, `\nwprevdefptr` 20a, and `\nwtagstyle` 10b.

Print the tags to the left and right chunks; left- and right-pointing triangles are added to the tags.

20a *<knoweb.sty 6a>+≡* *<18a 21d>*

```

\newcommand{\nwprevdefptr}[1]{%
  \mbox{$\mathord{\triangleleft}\,\,\mathord{\mbox{\subpageref{#1}}}$}}
\newcommand{\nwnextdefptr}[1]{%
  \mbox{$\mathord{\mbox{\subpageref{#1}}}\,\,\mathord{\triangleright}$}}

```

Defines:

- \nwnextdefptr, used in chunk 19b.
- \nwprevdefptr, used in chunk 19b.

Uses \subpageref 28g.

2.8.2 Package Options

Declare the package options used to select display style for chunk cross-references.

The **shortxref** option uses symbols on the definition line.

20b *<declaration of options 9a>+≡* *(6a) <15b 20c>*

```

\DeclareOption{shortxref}{%
  \let\nwalsodefined=\@gobble
  \let\nwused=\@gobble
  \let\nwnotused=\@gobble
  \let\nwprevnextdefs=\nw@prevnextdefs
  \let\nwusesondefline=\nw@usesondefline
  \let\nwstartdeflinemarkup=\nw@startdeflinemarkup
  \let\nwenddeflinemarkup=\nw@enddeflinemarkup
}

```

Defines:

- shortxref, used in chunk 21b.

Uses \nw@enddeflinemarkup 19b, \nw@prevnextdefs 19b, \nw@startdeflinemarkup 19b, \nw@usesondefline 19b, \nwalsodefined 21c, \nwenddeflinemarkup 21c, \nwnotused 21c, \nwprevnextdefs 21c, \nwstartdeflinemarkup 21c, \nwused 21c, and \nwusesondefline 21c.

The **longxref** option uses small paragraphs after each chunk, as in Knuth.

20c *<declaration of options 9a>+≡* *(6a) <20b 21a>*

```

\DeclareOption{longxref}{%
  \let\nwalsodefined=\nw@alsodefined
  \let\nwused=\nw@used
  \let\nwnotused=\nw@notused
  \let\nwprevnextdefs=\@gobbletwo
  \let\nwusesondefline=\@gobble
  \let\nwstartdeflinemarkup=\relax
  \let\nwenddeflinemarkup=\relax
}

```

Defines:

- longxref, never used.

Uses \nw@alsodefined 19a, \nw@notused 19a, \nw@used 19a, \nwalsodefined 21c, \nwenddeflinemarkup 21c, \nwnotused 21c, \nwprevnextdefs 21c, \nwstartdeflinemarkup 21c, \nwused 21c, and \nwusesondefline 21c.

The `noxref` option uses no chunk cross-references.

21a *<declaration of options 9a>+≡* (6a) <20c 34a>
`\DeclareOption{noxref}{%`
`\let\nwalsodefined=\@gobble`
`\let\nwused=\@gobble`
`\let\nwnotused=\@gobble`
`\let\nwprevnextdefs=\@gobbletwo`
`\let\nwusesondefline=\@gobble`
`\let\nwstartdeflinemarkup=\relax`
`\let\nwenddeflinemarkup=\relax`
`}`

Defines:

`noxref`, never used.

Uses `\nwalsodefined 21c`, `\nwenddeflinemarkup 21c`, `\wnotused 21c`, `\nwprevnextdefs 21c`, `\nwstartdeflinemarkup 21c`, `\nwused 21c`, and `\nwusesondefline 21c`.

The default is `shortxref`

21b *<execution of options 9e>+≡* (6a) <12c 34b>
`\ExecuteOptions{shortxref}`
 Uses `shortxref 20b`.

Allocate the commands. These are redefined by the package options.

21c *<initial code 6c>+≡* (6a) <19b 28a>
`\newcommand\nwalsodefined{}`
`\newcommand\nwused{}`
`\newcommand\nwnotused{}`
`\newcommand\nwprevnextdefs{}`
`\newcommand\nwusesondefline{}`
`\newcommand\nwstartdeflinemarkup{}`
`\newcommand\nwenddeflinemarkup{}`

Defines:

`\nwalsodefined`, used in chunks 20 and 21a.

`\nwenddeflinemarkup`, used in chunks 20 and 21a.

`\wnotused`, used in chunks 20 and 21a.

`\nwprevnextdefs`, used in chunks 20 and 21a.

`\nwstartdeflinemarkup`, used in chunks 20 and 21a.

`\nwused`, used in chunks 20 and 21a.

`\nwusesondefline`, used in chunks 20 and 21a.

21d *<knoweb.sty 6a>+≡* <20a 22a>
`\newlength\nwcodecommentsep`
`\setlength{\nwcodecommentsep}{3pt plus 1pt minus 1pt}`
`\newif@iffirstnwcodecomment\@firstnwcodecommenttrue`

Defines:

`\@firstnwcodecommentfalse`, used in chunk 18b.

`\@firstnwcodecommenttrue`, never used.

`\iffirstnwcodecomment`, used in chunk 18b.

`\nwcodecommentsep`, used in chunk 18b.

2.9 Page Ranges

The goal is to combine sub-page numbers in a way that makes sense. Multiple sub-pages of one page become that page, and individual pages are combined into ranges. (A range may be only one page.)

Does this, indeed, make sense? More to the point, is it useful? For source code with several chunks per page (such as this), a range of pages is too imprecise; better to list just the chunks that contain the referenced item. This can be achieved by using the `webnumbering` option; however, it would be useful to have this capability in the `alphasubpage` and `numsubpage` options.

Given a list of keys,

$\langle\langle key_1 \rangle\rangle \langle\langle key_2 \rangle\rangle \cdots \langle\langle key_n \rangle\rangle$,

convert the keys to ranges and then

22a $\langle knoweb.sty 6a \rangle + \equiv \quad \quad \quad \langle 21d \ 23a \rangle$

```
\newcommand\nw@chunkcommachunketc[1]{%list of keys
  \initialize \nw@pages and \nw@pagecount 22b
  \def\##1{\nw@firstpagel{##1}%
    \let\=\nw@nextpagel}%
  #1%
  \append range to range list 24e}%
  \insert “chunk(s)” 22c
  \def\##1{\nw@hyperpagenum##1}%
  \nw@commafy{\nw@pages}}
```

Defines:

`\nw@chunkcommachunketc`, used in chunks 19a and 44b.

Uses `\nw@commafy` 41b, `\nw@firstpagel` 27a, `\nw@nextpagel` 27b, and `\nw@pages` 28a.

22b $\langle initialize \ \nw@pages \text{ and } \nw@pagecount \ 22b \rangle \equiv \quad \quad \quad (22a \ 23a \ 27d)$

```
\gdef\nw@pages{}%
\nw@pagecount=\z@
```

Uses `\nw@pagecount` 27a and `\nw@pages` 28a.

22c $\langle insert \ “chunk(s)” \ 22c \rangle \equiv \quad \quad \quad (22a \ 23a)$

```
\ifnum\nw@pagecount=\@ne
  \nw@langdepchk
\else
  \nw@langdepchks
\fi
\nobreakspace
```

Uses `\nw@langdepchk` 54a, `\nw@langdepchks` 54a, and `\nw@pagecount` 27a.

This is similar to `\nw@chunkcommachunketc`, however,

23a `\knoweb.sty 6a` +≡ <22a 23b>

```
\newcommand{\subpages}[1]{%\list of \subpage \page\}
\initialize \nw@pages and \nw@pagecount 22b\}
\def\##1{\edef\@tempa{\noexpand\nw@firstpage##1}\@tempa
\def\####1{\edef\@tempa{\noexpand\nw@nextpage####1}\@tempa}}%
#1%
\append range to range list 24e\}
\insert "chunk(s)" 22c\}
\def\##1{\@firstoftwo##1}%
\nw@commafy{\nw@pages}}
```

Defines:

`\subpages`, never used.

Uses `\nw@commafy 41b` and `\nw@pages 28a`.

23b `\knoweb.sty 6a` +≡ <23a 23c>

```
\newcount\nw@lopage % lo
\newcount\nw@hipage % hi
\newcount\nw@losub % subpage of lo
\newcount\nw@hisub % subpage of hi
```

Defines:

`\nw@hipage`, used in chunks 24–6.

`\nw@hisub`, used in chunks 24a and 25a.

`\nw@lopage`, used in chunks 23–6.

`\nw@losub`, used in chunks 23–5.

Insert nothing. Assign the counters `\nw@lopage` and `\nw@losub`. Assign `\nw@hipage` to be one greater than `\nw@lopage`. Increment `\nw@pagecount`.

23c `\knoweb.sty 6a` +≡ <23b 24a>

```
\newcommand\nw@firstpage[3]{%\subpage \page \key\}
\nw@lopage=#2 \nw@losub=#1
\def\nw@loxreftag{#3}%
\advance\nw@pagecount by \@ne
\(\nw@hipage ← \nw@lopage + 1 24c\}
}
```

Defines:

`\nw@firstpage`, used in chunks 24b and 27a.

`\nw@loxreftag`, used in chunk 25a.

Uses `\nw@lopage 23b`, `\nw@losub 23b`, and `\nw@pagecount 27a`.

Insert nothing.

24a $\langle \text{knoweb.sty 6a} \rangle + \equiv$ $\langle 23c \ 26d \rangle$

```
\newcommand\nw@nextpage[3]{%\subpage\page\key}
\ifnum\nw@hipage=#2
\advance\nw@hipage by \@ne
\advance\nw@pagecount by \@ne
\nw@hisub=#1
\def\nw@hixreftag{#3}%
\else
\ifnum#2<\nw@lopage
\new range starting with #2 24b)%
\else
\ifnum#2>\nw@hipage
\new range starting with #2 24b)%
\else
\nw@losub=\z@ \nw@hisub=\z@
\fi\fi\fi
}
```

Defines:

\nw@hixreftag, used in chunk 25a.

\nw@nextpage, used in chunk 27b.

Uses \nw@hipage 23b, \nw@hisub 23b, \nw@lopage 23b, \nw@losub 23b, and \nw@pagecount 27a.

24b $\langle \text{new range starting with \#2 24b} \rangle \equiv$ (24a)

```
\append range to range list 24e)
\nw@firstpage{#1}{#2}{#3}
```

Uses \nw@firstpage 23c.

24c $\langle \nw@hipage \leftarrow \nw@lopage + 1 \ 24c \rangle \equiv$ (23c)

```
\nw@hipage=\nw@lopage\advance\nw@hipage by\@ne
```

Uses \nw@hipage 23b and \nw@lopage 23b.

24d $\langle \nw@count@ \leftarrow \nw@hipage - 1 \ 24d \rangle \equiv$

```
\count@=\nw@hipage\advance\count@ by\m@ne
```

Uses \nw@hipage 23b.

Append a chunk range to \nw@pages. With \nw@pages equal to

$\{\langle range_1 \rangle\}\{\langle key_1 \rangle\} \cdots \{\langle range_n \rangle\}\{\langle key_n \rangle\}$,

the result is

$\{\langle range_1 \rangle\}\{\langle key_1 \rangle\} \cdots \{\langle range_n \rangle\}\{\langle key_n \rangle\} \{\langle range_{n+1} \rangle\}\{\langle key_{n+1} \rangle\}$.

24e $\langle \text{append range to range list 24e} \rangle \equiv$ (22–4 27d)

```
\set \@tempa to page range(s), marked with \ 25a)
```

```
\edef\@tempa{\noexpand\nw@ixappend\noexpand\nw@pages{\@tempa}}%
```

```
\@tempa
```

Uses \nw@ixappend 42a and \nw@pages 28a.

Assign \@tempa to one of

- (a) $\{\langle chunk id \rangle\}\{\langle key \rangle\}$,
- (b) $\{\langle chunk id_1 \rangle\}\{\langle key_1 \rangle\}\{\langle chunk id_2 \rangle\}\{\langle key_2 \rangle\}$,
- (c) $\{\langle chunk id_1 \rangle - \langle chunk id_2 \rangle\}\{\}$.

25a $\langle set \@tempa to page range(s), marked with \ 25a \rangle \equiv$ (24e)

```

\advance\nw@hipage by \m@ne
\count@ ← \nw@hipage – \nw@lopage 25b
\ifcase\count@%same page
\edef\@tempa{%
\noexpand\noexpand\noexpand\%
{\nwthepagenum{\number\nw@losub}{\number\nw@lopage}}%
{\nw@loxreftag}}%
\or%next page
\edef\@tempa{%
\noexpand\noexpand\noexpand\%
{\nwthepagenum{\number\nw@losub}{\number\nw@lopage}}
{\nw@loxreftag}}%
\noexpand\noexpand\noexpand\%
{\nwthepagenum{\number\nw@hisub}{\number\nw@hipage}}
{\nw@hixreftag}}%
\else
\use simple rules from Chicago style manual 25c)%
\fi
Uses \nw@hipage 23b, \nw@hisub 23b, \nw@hixreftag 24a, \nw@lopage 23b, \nw@losub 23b,
\nw@loxreftag 23c, and \nwthepagenum 34a 34c.
```

25b $\langle \count@ ← \nw@hipage – \nw@lopage 25b \rangle \equiv$ (25a)

```

\count@=\nw@hipage
\advance\count@ by-\nw@lopage
Uses \nw@hipage 23b and \nw@lopage 23b.
```

The Chicago Manual of Style [1, §8.69–70] specifies two systems for abbreviating inclusive numbers. The original version of this package, noweb.sty, used a modified version of the first system. Here we implement the second (simpler) system. From the manual: “the second number of the range includes only the changed part of the first number:

| | | | |
|--------|---------|---------|------------|
| 3–10 | 600–13 | 1002–6 | 1496–504 |
| 71–2 | 1100–23 | 321–5 | 14325–8 |
| 96–117 | 107–8 | 415–532 | 11564–78 |
| 100–4 | 505–17 | 1536–42 | 13729–803” |

Modify \nw@hipage to correspond to the Chicago style manual, then set \@tempa accordingly.

25c $\langle use simple rules from Chicago style manual 25c \rangle \equiv$ (25a)

```

\chicago: \ell ← \nw@lopage, h ← \nw@hipage, 10k ← 1 26a
\chicago: find 10k such that  $\lfloor \ell / 10^k \rfloor = \lfloor h / 10^k \rfloor$  26b
\chicago: \nw@hipage ← h –  $\lfloor h / 10^k \rfloor 10^k$  26c
\define \@tempa to be  $\{\nw@lopage - \nw@hipage\}\}$  26e
```

26a $\langle \text{chicago: } \ell \leftarrow \backslash\text{nw@lopage}, h \leftarrow \backslash\text{nw@hipage}, 10^k \leftarrow 1 \text{ 26a} \rangle \equiv$ (25c)
 $\backslash\text{nw@lo} = \backslash\text{nw@lopage}$
 $\backslash\text{nw@hi} = \backslash\text{nw@hipage}$
 $\backslash\text{nw@pwrten} = \backslash@ne$

Uses $\backslash\text{nw@hi}$ 26d, $\backslash\text{nw@hipage}$ 23b, $\backslash\text{nw@lo}$ 26d, $\backslash\text{nw@lopage}$ 23b, and $\backslash\text{nw@pwrten}$ 26d.

26b $\langle \text{chicago: find } 10^k \text{ such that } \lfloor \ell/10^k \rfloor = \lfloor h/10^k \rfloor \text{ 26b} \rangle \equiv$ (25c)
 $\backslash\text{loop}$
 $\backslash\text{divide}\backslash\text{nw@lo}$ by10
 $\backslash\text{divide}\backslash\text{nw@hi}$ by10
 $\backslash\text{multiply}\backslash\text{nw@pwrten}$ by10
 $\backslash\text{ifnum}\backslash\text{nw@lo} = \backslash\text{nw@hi} \% \text{exit loop}$
 $\backslash\text{else}$
 $\backslash\text{repeat}$

Uses $\backslash\text{nw@hi}$ 26d, $\backslash\text{nw@lo}$ 26d, and $\backslash\text{nw@pwrten}$ 26d.

26c $\langle \text{chicago: } \backslash\text{nw@hipage} \leftarrow h - \lfloor h/10^k \rfloor 10^k \text{ 26c} \rangle \equiv$ (25c)
 $\backslash\text{multiply}\backslash\text{nw@hi}$ by $\backslash\text{nw@pwrten}$
 $\backslash\text{advance}\backslash\text{nw@hipage}$ by $-\backslash\text{nw@hi}$

Uses $\backslash\text{nw@hi}$ 26d, $\backslash\text{nw@hipage}$ 23b, and $\backslash\text{nw@pwrten}$ 26d.

Allocate counters for computing page ranges. Initialize $\backslash\text{nw@pwrten}$ to one, which is its value when entering the group in chunk 25c.

26d $\langle \text{knoweb.sty 6a} \rangle + \equiv$ $\langle 24a \text{ 26f} \rangle$
 $\backslash\text{newcount}\backslash\text{nw@lo}$
 $\backslash\text{newcount}\backslash\text{nw@hi}$
 $\backslash\text{newcount}\backslash\text{nw@pwrten}$

Defines:

$\backslash\text{nw@hi}$, used in chunk 26.
 $\backslash\text{nw@lo}$, used in chunk 26.
 $\backslash\text{nw@pwrten}$, used in chunk 26.

26e $\langle \text{define } \backslash@tempa \text{ to be } \{\backslash\text{nw@lopage} - \backslash\text{nw@hipage}\} \text{ 26e} \rangle \equiv$ (25c)
 $\backslash\text{edef}\backslash@tempa\{\%$
 $\backslash\text{noexpand}\backslash\text{noexpand}\backslash\text{noexpand}\backslash\%$
 $\{\{\backslash\text{number}\backslash\text{nw@lopage} - \backslash\text{number}\backslash\text{nw@hipage}\}\%$
 $\{\}\}$

Uses $\backslash\text{nw@hipage}$ 23b and $\backslash\text{nw@lopage}$ 23b.

26f $\langle \text{knoweb.sty 6a} \rangle + \equiv$ $\langle 26d \text{ 27a} \rangle$
 $\backslash\text{newcount}\backslash\text{nw@pagetemp}$

Defines:

$\backslash\text{nw@pagetemp}$, never used.

Assign the macro `\@tempa`

27a `\knoweb.sty 6a` +≡ <26f 27b>

```
\newcommand\nw@firstpagel[1]{%\langle key\rangle
\@ifundefined{r@#1}
{\warn of undefined reference to #1 and add page ?? 28b}
{\edef\@tempa{\noexpand\nw@firstpage\subpagepair{#1}{#1}}%
\@tempa}}
\newcount\nw@pagecount
```

Defines:

`\nw@firstpagel`, used in chunks 22a and 27d.

`\nw@pagecount`, used in chunks 22–4.

Uses `\nw@firstpage` 23c and `\subpagepair` 29b.

This is identical to `\nw@firstpagel` except that it calls `\nw@nextpage` rather than `\nw@firstpage`.

27b `\knoweb.sty 6a` +≡ <27a 27c>

```
\newcommand\nw@nextpagel[1]{%\langle key\rangle
\@ifundefined{r@#1}
{\warn of undefined reference to #1 and add page ?? 28b}
{\edef\@tempa{\noexpand\nw@nextpage\subpagepair{#1}{#1}}%
\@tempa}}
```

Defines:

`\nw@nextpagel`, used in chunks 22a and 27d.

Uses `\nw@nextpage` 24a and `\subpagepair` 29b.

Print `\langle identifier \rangle` and make it an active hyperlink to `\langle link \rangle`.

27c `\knoweb.sty 6a` +≡ <27b 27d>

```
\newcommand\nw@hyperpagenum[2]{%\langle identifier\rangle\langle link\rangle
\nwhyperreference{#2}{#1}}
```

Defines:

`\nw@hyperpagenum`, never used.

Uses `\nwhyperreference` 51b.

Given a list of keys, return the corresponding list of chunk identifiers, separated by spaces.

27d `\knoweb.sty 6a` +≡ <27c 28g>

```
\newcommand\nw@chunkspacechunketc[1]{%\{\langle key_1\rangle\cdots\langle key_n\rangle\}
\initialize\nw@pages and \nw@pagecount 22b}
\def\##1{\nw@firstpagel{##1}\let\=\nw@nextpagel}%
#1%
\append range to range list 24e}
\def\##1{\nw@hyperpagenum##1\let\=\nw@pagenumslrest}%
\nw@pages}
\newcommand\nw@pagenumslrest[1]{\nobreakspace\nw@hyperpagenum#1}
```

Defines:

`\nw@chunkspacechunketc`, used in chunk 19b.

`\nw@pagenumslrest`, never used.

Uses `\nw@firstpagel` 27a, `\nw@nextpagel` 27b, and `\nw@pages` 28a.

Allocate a command used to record a list of something.

28a $\langle \text{initial code 6c} \rangle + \equiv$ (6a) $\langle 21c \ 30b \rangle$
`\newcommand{\nw@pages}{}`

Defines:

`\nw@pages`, used in chunks 22–4, 27d, and 28b.

28b $\langle \text{warn of undefined reference to \#1 and add page ?? 28b} \rangle \equiv$ (27)
 $\langle \text{warn of undefined reference to \#1 28c} \rangle \%$
`\nw@ixappend\nw@pages{\{\bf ??\}}`

Uses `\nw@ixappend` 42a and `\nw@pages` 28a.

28c $\langle \text{warn of undefined reference to \#1 28c} \rangle \equiv$ (28b)
`\@warning{Reference ‘\#1’ on page \thepage{} undefined}`

2.10 Sub-page References

This is the wonderful code that Dave Love provided to make page references like 7a, 7b, and so on.

This code provides a mechanism for defining ‘page sub-references’ using `\sublabel{foo}` referenced with `\subpageref{foo}`. Sub-references will be numbered like 28d, 28e, 28f unless there is only one on the page, in which case the letter is dropped.

To be able to use `\subpageref` we must define the label with `\sublabel`, used like label. (Using `\ref` with a label defined by `\sublabel` will produce the sub-reference number, by the way, and `\pageref` works as expected.) Note that `\subpageref` is robust and `\ref` and `\pageref` are redefined to be robust also, as they will be in future L^AT_EX releases. Incidentally, these expand to the relevant text plus `\null`—you might want to strip this off, e.g. for sorting lists.

2.10.1 `\subpageref`

Given $\langle \text{key} \rangle$, print the identifier ($\langle \text{page no.} \rangle \langle \text{subpage} \rangle$ or $\langle \text{chunk no.} \rangle$) associated with $\langle \text{key} \rangle$, from a previous `\sublabel` call, and, if hyperlinking is enabled, insert a hyperlink to the reference.

28g $\langle \text{knoweb.sty 6a} \rangle + \equiv$ $\langle 27d \ 29a \rangle$
`\newcommand{\subpageref}[1]{\% \langle \text{key} \rangle`
`\nwhyperreference{\#1}{\nw@genericref\nw@subpageref{\#1}}}`

Defines:

`\subpageref`, used in chunks 20a, 39c, 43a, 47b, and 49d.

Uses `\nw@genericref` 29d, `\nw@subpageref` 29a, and `\nwhyperreference` 51b.

Insert the identifier to a chunk into the text. If there is but one chunk on a page (or `webnumbering` is enabled), the `\on<page no.>` is undefined, so just `<page no.>` is inserted; otherwise use the macro `\nwthepagenum` to convert the two parameters into the desired format.

The two parameters are delimited by `\`, which is inserted by `\nw@g@nericref`.

29a `<knoweb.sty 6a>+≡` <28g 29b>

```
\def\nw@subpageref#1#2\\{%\<sub-page> \<page no.>
  \@ifundefined{2on#2}
    {#2}
    {\nwthepagenum{#1}{#2}}}
```

Defines:

`\nw@subpageref`, used in chunk 28g.

Uses `\nwthepagenum` 34a 34c.

2.10.2 `\subpagepair`

Given `<key>`, return `{\<sub-page>}{\<page no.>}`. If `<key>` has not been allocated (by a call to `\sublabel`), return `{0}{0}`.

29b `<knoweb.sty 6a>+≡` <29a 29c>

```
\newcommand{\subpagepair}[1]{%\<key>
  \@ifundefined{r@#1}%
    {0}{0}
    {\nw@genericref\nw@subpagepair{#1}}}
```

Defines:

`\subpagepair`, used in chunk 27.

Uses `\nw@genericref` 29d and `\nw@subpagepair` 29c.

Return `{\<sub-page>}{\<page no.>}`. If the referenced page has just one chunk, then set the `<sub-page>` field to zero. The parameters are delimited by `\`, which is inserted by `\nw@g@nericref`.

29c `<knoweb.sty 6a>+≡` <29b 29d>

```
\def\nw@subpagepair#1#2\\{%\<sub-page> \<page>
  \@ifundefined{2on#2}
    {0}{#2}
    {#1}{#2}}
```

Defines:

`\nw@subpagepair`, used in chunk 29b.

The `<action>` parameter is either the macro `\nw@subpageref` or `\nw@subpagepair`.

29d `<knoweb.sty 6a>+≡` <29c 30a>

```
\newcommand\nw@genericref[2]{%\<action> \<key>
  \expandafter\nw@g@nericref\csize r@#2\endcsize#1{#2}}
```

Defines:

`\nw@genericref`, used in chunks 28g and 29b.

Uses `\nw@g@nericref` 30a.

30a `<knoweb.sty 6a>+≡` `<29d 30c>`

```

\newcommand\nw@g@nericref[3]{%\control sequence\action\key}
\ifx#1\relax
\ref{#3}%trigger the standard ‘undefined ref’ mechanisms (i.e. barf)
\else
\expandafter#2#1\\%
\fi}

```

Defines:

`\nw@g@nericref`, used in chunk 29d.

2.10.3 `\sublabel`

`\sublabel` is like the `\label` command, except that it writes `\newsublabel` rather than `\newlabel` to the .aux file. For hyper-referencing, all labels must be hypertext anchors, for which we use `\nwblindhyperanchor`. This command is emitted by noweave.

30b `<initial code 6c>+≡` `(6a) <28a 34c>`

```

\newcommand{\sublabel}[1]{%\key}
\nwblindhyperanchor{#1}%
\@bsphack
\if@filesw
{\let\thepage=\relax
\def\protect{\noexpand\noexpand\noexpand}%
\edef\@tempa{\write\@auxout{\string
\newsublabel{#1}{\thepage}}}%
\expandafter\@tempa
\if@nobreak\ifvmode\nobreak\fi\fi
\fi
\@esphack
}

```

Defines:

`\sublabel`, used in chunk 37b.

Uses `\newsublabel` 31a and `\nwblindhyperanchor` 51b.

`\nosublabel` creates a label with a sub-page part of 0. Does noweave emit this command?

30c `<knoweb.sty 6a>+≡` `<30a 31a>`

```

\newcommand{\nosublabel}[1]{%
\@bsphack
\if@filesw
{\let\thepage\relax
\def\protect{\noexpand\noexpand\noexpand}%
\edef\@tempa{\write\@auxout{\string
\newlabel{#1}{0}{\thepage}}}%
\expandafter\@tempa
\if@nobreak\ifvmode\nobreak\fi\fi
\fi
\@esphack}

```

Defines:

`\nosublabel`, never used.

2.10.4 \newsublabel

This is the macro that does the important work. Its first argument is the $\langle key \rangle$, its second is $\{\langle ref value \rangle \langle page no. \rangle\}$. The field $\langle ref value \rangle$ appears to be always empty. Note that $\langle page no. \rangle$ is $\langle chunk number \rangle$ when the package option **webnumbering** is in effect.

31a $\langle knoweb.sty 6a \rangle + \equiv$ $\langle 30c 32d \rangle$

```

\newcommand\newsublabel[2]{% $\langle key \rangle$   $\{\langle ref value \rangle \langle page no. \rangle\}$ 
   $\langle save \langle page no. \rangle$  into  $\nw@thispage$  31b $\rangle$ 
   $\langle update \nw@subpage$  and  $\nw@lastpage$  31c $\rangle$ 
   $\langle if$  two sublabels on page, assign  $\nw@thispage$  31d $\rangle$ 
   $\langle add \langle key \rangle$  to  $\pending@sublabels$  list 32a $\rangle$ 
   $\langle call \newlabel$  for each  $\langle key \rangle$  in  $\pending@sublabels$  32b $\rangle$ 
   $\langle clear \pending@sublabels$  32c $\rangle$ 
}
```

Defines:

$\nw@thispage$, used in chunks 30b and 37a.

31b $\langle save \langle page no. \rangle$ into $\nw@thispage$ 31b $\rangle \equiv$ (31a)

```

\edef\nw@thispage{\@cdr#2\@nil}%
```

Defines:

$\nw@thispage$, used in chunks 31 and 32b.

Check whether $\nw@thispage$ has changed from the value of $\nw@lastpage$, which was saved by a previous \newsublabel (or is \relax if this is the first one). If the page is the same, increment the counter, $\nw@subpage$, that records the number of sub-labels on a page; otherwise reset it and redefine $\nw@lastpage$.

31c $\langle update \nw@subpage$ and $\nw@lastpage$ 31c $\rangle \equiv$ (31a)

```

\ifx\nw@thispage\nw@lastpage
  \advance\nw@subpage by\@ne
\else
  \nw@subpage=\@ne
  \edef\nw@lastpage{\nw@thispage}%
\fi
```

Uses $\nw@lastpage$ 33d, $\nw@subpage$ 33d, and $\nw@thispage$ 31b.

31d $\langle if$ two sublabels on page, assign $\nw@thispage$ 31d $\rangle \equiv$ (31a)

```

\ifnum\nw@subpage=\tw@
  \global\@namedef{2on\nw@thispage}{}%
\fi
```

Uses $\nw@subpage$ 33d and $\nw@thispage$ 31b.

Write a normal `\newlabel` with the sub-reference as the normal reference value in the second argument. Unfortunately, with hypertext support, the second argument of `\newlabel` gets complicated. It is either

- $\langle ref\ value\rangle\langle page\ no.\rangle$ when normal L^AT_EX is running, or
- $\langle ref\ value\rangle\langle page\ no.\rangle\langle text\rangle\langle hyper\ category\rangle\langle URL\rangle$ when the `nameref` package is running.

Unify these two things by producing $\langle ref\ value\rangle\langle page\ no.\rangle\backslash nw@labeltrailers$.

There may be pending labels in support of `\nextchunklabel`, as defined in chunk 33a. Because we want to define all of the “pending sublabels” in exactly the same way, we do something a bit odd—we make the current label a pending label as well.

32a $\langle add\ \langle key\rangle\ to\ \backslash pending@sublabels\ list\ 32a\rangle\equiv$ (31a)
 $\backslash pendingsublabel\{\#1\}\%$

Uses `\pendingsublabel` 33b.

Finally, for each key in `\pending@sublabels` call `\newlabel`, assigning the macro $\backslash r@{\langle key\rangle}$ to expand to $\{\backslash nw@subpage\}\{\backslash nw@thispage\}$.

32b $\langle call\ \backslash newlabel\ for\ each\ \langle key\rangle\ in\ \backslash pending@sublabels\ 32b\rangle\equiv$ (31a)
 $\backslash edef\@tempa\##1\{\backslash noexpand\backslash newlabel\{\##1\}\%$
 $\{\{\backslash number\backslash nw@subpage\}\{\backslash nw@thispage\}\backslash nw@labeltrailers\}\}\%$
 $\backslash pending@sublabels$

Uses `\nw@labeltrailers` 32d, `\nw@subpage` 33d, `\nw@thispage` 31b, and `\pending@sublabels` 33c.

32c $\langle clear\ \backslash pending@sublabels\ 32c\rangle\equiv$ (31a)
 $\backslash def\backslash pending@sublabels\{\}\%$
 Uses `\pending@sublabels` 33c.

The assignment of `\nw@labeltrailers` depends on whether the package `nameref` is loaded, which is determined by using `\ifpackageloaded`. The assignment must occur before the macro `\subname` is expanded (it calls `\nw@labeltrailers`) and `\subname` is in the aux file, which is read (the first time) before the contents of `\AtBeginDocument` are expanded.

To handle this, and permit a call to `\subname` in the body of the document, when the contents of `\AtBeginDocument` are expanded, `\nw@labeltrailers` is assigned **and** its definition is written to the aux file. In subsequent passes through L^AT_EX, the definition of `\nw@labeltrailers` is available to the `\subname` macros in the aux file.

32d $\langle knoweb.sty\ 6a\rangle+\equiv$ <31a 33a>
 $\backslash AtBeginDocument\{\%$
 $\backslash xdef\backslash nw@labeltrailers\{\@ifpackageloaded\{nameref\}\{\}\{\}\}\}\%$
 $\backslash write\@mainaux\{\string\gdef\string\backslash nw@labeltrailers\{\backslash nw@labeltrailers\}\}\}$

Defines:

`\nw@labeltrailers`, used in chunk 32b.

Now we keep track of those pending guys. The goal here is to save them up until they are all equivalent to the label on the next chunk. We have to control expansion so chunks like 32a (32a) can be labelled twice.

33a `\knoweb.sty 6a` +≡ <32d 33b>

```

\newcommand\nextchunklabel[1]{%\<key>
  \nwblindhyperanchor{#1}%
  \@bsphack
  \if@files
  {\let\thepage=\relax
   \edef\@tempa{\write\@auxout{\string\pendingsublabel{#1}}}%
   \expandafter}\@tempa
   \if@nbreak\ifvmode\nobreak\fi\fi
  \fi
  \@esphack
}
```

Defines:

`\nextchunklabel`, never used.

Uses `\nwblindhyperanchor` 51b and `\pendingsublabel` 33b.

Insert `\@tempa{\<key>}` at the front of the list `\pending@sublabels`.

33b `\knoweb.sty 6a` +≡ <33a 33c>

```

\newcommand\pendingsublabel[1]{%\<key>
  \def\@tempa{\noexpand\@tempa}%
  \edef\pending@sublabels{\noexpand\@tempa{#1}\pending@sublabels}}
```

Defines:

`\pendingsublabel`, used in chunks 32a and 33a.

Uses `\pending@sublabels` 33c.

Define the initial expansion of `\pending@sublabels`. Subsequent calls to `\pendingsublabel` redefines it to a list `\@tempa{\<keyn>}... \@tempa{\<key1>}`.

33c `\knoweb.sty 6a` +≡ <33b 33d>

```

\newcommand\pending@sublabels{}
```

Defines:

`\pending@sublabels`, used in chunks 32 and 33b.

We need to define these.

33d `\knoweb.sty 6a` +≡ <33c 35a>

```

\newcommand\nw@lastpage{\relax}
\newcount\nw@subpage
```

Defines:

`\nw@lastpage`, used in chunk 31c.

`\nw@subpage`, used in chunks 31 and 32b.

2.10.5 Assign Sub-page Reference Options

34a $\langle \text{declaration of options 9a} \rangle + \equiv$ (6a) $\langle 21a \ 36c \rangle$

```

\DeclareOption{alphasubpage}{%\sub-page\page}
\renewcommand\nwthepagenum[2]{#2\ifnum#1=\z@ \else\nw@int@to@str{#1}\fi}}
\DeclareOption{numsubpage}{%
\renewcommand\nwthepagenum[2]{#2\ifnum#1=\z@ \else.\@arabic{#1}\fi}}
\DeclareOption{nosubpage}{%
\renewcommand\nwthepagenum[2]{#2}%
\ExecuteOptions{nomargintag}}

```

Defines:

`\nwthepagenum`, used in chunks 25a and 29a.

`alphasubpage`, used in chunk 34b.

`nosubpage`, never used.

`numsubpage`, never used.

Uses `\nw@int@to@str` 36c and `nomargintag` 9c.

The default is `alphasubpage`.

34b $\langle \text{execution of options 9e} \rangle + \equiv$ (6a) $\langle 21b \ 36d \rangle$

```

\ExecuteOptions{alphasubpage}

```

Uses `alphasubpage` 34a.

Dummy assignment to `\nwthepagenum`. This is reassigned during option processing.

34c $\langle \text{initial code 6c} \rangle + \equiv$ (6a) $\langle 30b \ 39c \rangle$

```

\newcommand\nwthepagenum[2]{}

```

Defines:

`\nwthepagenum`, used in chunks 25a and 29a.

2.10.6 Converting Integers to Strings

To uniquely label the sub-page references, we need a macro that converts integers into strings, specifically, into strings of lower-case letters. The \LaTeX macro `\@alph` converts the natural numbers from 1 to 26 into the characters ‘a’ to ‘z’. This should suffice for the majority of applications.

In rare instances there may be more than 26 chunks on a page. In such a case, we need a sub-page lettering scheme that maps integers to multiple character strings. The chosen scheme is shown in table 1.

| n | 1 | 2 | ... | 26 | $26+1$ | ... | 26^2+26 | 26^2+26+1 | ... | 26^3+26^2+26 |
|-------------|---|---|-----|----|--------|-----|-----------|-------------|-----|----------------|
| $\alpha(n)$ | a | b | ... | z | aa | ... | zz | aaa | ... | zzz |

Table 1: Long integers to strings

The conversion requires a bit of thought because it is *not* an ordinary conversion of integer to string. The problem is that the meaning of the letters depends on the position; the letter a acts like a zero in some positions or a one in others.

If we write the recurrence $B_k = B_{k-1} + 26^k$, with $B_0 = 0$, we can use a string of k letters to represent numbers between B_{k-1} and B_k . Within that string, ‘a’ is 0, ‘b’ is 1, ... ‘z’ is 25 and we use standard integer-conversion methods to encode $n - B_{k-1}$.

The first loop finds B^k and sets n to $\langle n \rangle - 1 - B_{k-1}$, where $\langle n \rangle$ is the parameter. The second does the usual string conversion. The entire macro body is enclosed in braces so that it can be used with `\loop` without picking up the wrong `\repeat`. This macro handles integers up to 321,272,406, which corresponds to ‘zzzzzz’; it is limited by the range of \TeX ’s count registers.

35a `\knoweb.sty 6a`)+≡ <33d 37a>

```
\newcommand\nw@longalph[1]{\% $\langle n \rangle$ 
   $\langle$ assign temporary count registers 35b $\rangle$ 
   $\langle n \leftarrow \langle n \rangle - 1; B^k \leftarrow 26$  35c $\rangle$ 
  \loop $\langle$ until  $B^k > n$  35d $\rangle$ 
     $\langle n \leftarrow n - B^k; B^k \leftarrow 26 B^k$  35e $\rangle$ 
  \repeat
  \loop $\langle$ while  $B^k > 1$  35f $\rangle$ 
     $\langle B^k \leftarrow \lfloor B^k/26 \rfloor; d \leftarrow \lfloor n/B^k \rfloor$  35g $\rangle$ 
     $\langle$ write character corresponding to  $d$  36a $\rangle$ 
     $\langle n \leftarrow n - d B^k$  36b $\rangle$ 
  \repeat
}
```

Defines:

`\nw@longalph`, used in chunk 36c.

35b `\langle assign temporary count registers 35b \rangle` ≡ (35a)

```
\newcount\n
\let\d=\@tempcnta
\let\Btok=\@tempcntb
```

35c `$\langle n \leftarrow \langle n \rangle - 1; B^k \leftarrow 26$ 35c \rangle` ≡ (35a)

```
\n=#1\advance\n by\m@ne\Btok=26
```

35d `\langle until $B^k > n$ 35d \rangle` ≡ (35a)

```
\ifnum\Btok>\n\else
```

35e `$\langle n \leftarrow n - B^k; B^k \leftarrow 26 B^k$ 35e \rangle` ≡ (35a)

```
\advance\n by -\Btok
\multiply\Btok by 26
```

35f `\langle while $B^k > 1$ 35f \rangle` ≡ (35a)

```
\ifnum\Btok>\@ne
```

35g `$\langle B^k \leftarrow \lfloor B^k/26 \rfloor; d \leftarrow \lfloor n/B^k \rfloor$ 35g \rangle` ≡ (35a)

```
\divide\Btok by 26
\d=\n
\divide\d by \Btok
```

The value of d is an integer from 0 to 25. To print the corresponding lower-case letter, increment d and use `\@alph`. This is done in a group so that the external value of d is not changed.²

36a *<write character corresponding to d 36a>*≡ (35a)
`{\advance\d by\@ne\@alph{\d}}%`

36b *< $n \leftarrow n - dB^k$ 36b>*≡ (35a)
`\multiply\d by \Btok`
`\advance\n by -\d`

Assign Integer to String Conversion Options

36c *<declaration of options 9a>+≡* (6a) <34a 37b>
`\DeclareOption{shortstrings}{\let\nw@int@to@str=\@alph}`
`\DeclareOption{longstrings}{\let\nw@int@to@str=\nw@longalph}`

Defines:

`\nw@int@to@str`, used in chunk 34a.

`longstrings`, never used.

`shortstrings`, used in chunk 36d.

Uses `\nw@longalph` 35a.

Set the default to `shortstrings`.

36d *<execution of options 9e>+≡* (6a) <34b 41a>
`\ExecuteOptions{shortstrings}`

Uses `shortstrings` 36c.

²It may be faster to simply decrement d after printing rather than grouping; however, this better expresses the intent.

2.11 WEB-like Chunk Numbering

To get the effect of WEB-like chunk numbers redefine `\sublabel` to use a counter instead of the current page number. Because the numbers are all distinct, no sub-page number is ever used.

37a `<knoweb.sty 6a>+≡` `<35a 38a>`

```

\newcount\nw@chunkcount
\nw@chunkcount=\@ne
\newcommand{\nw@weblabel}[1]{%\langle key\rangle
\nwblindhyperanchor{#1}%
\@bsphack
\if@filesw{%
\let\thepage\relax
\def\protect{\noexpand\noexpand\noexpand}%
\edef\@tempa{\write\@auxout{\string
\newsublabel{#1}{\number\nw@chunkcount}}}%
\expandafter}%
\@tempa
\global\advance\nw@chunkcount by \@ne
\if@nobreak\ifvmode\nobreak\fi\fi
\fi
\@esphack}

```

Defines:

`\nw@weblabel`, used in chunk 37b.

Uses `\newsublabel` 31a and `\nwblindhyperanchor` 51b.

37b `<declaration of options 9a>+≡` `(6a) <36c 40a>`

```

\DeclareOption{webnumbering}{%
\renewcommand\sublabel{\nw@weblabel}}

```

Defines:

`webnumbering`, never used.

Uses `\nw@weblabel` 37a and `\sublabel` 30b.

2.12 Indexing (Identifier Cross-Reference) Support

2.12.1 Tracking Definitions and Uses

All index definitions and uses are associated with a label defined with `\sublabel` or `\nosublabel`. Either the label is the `\sublabel` of the code chunk in which the definition or use appears, or it is a `\nosublabel` appearing in the middle of a documentation chunk. The commands `\nwindexdefn` and `\nwindexuse` are inserted into the generated T_EX file by `noweave`.

38a `<knoweb.sty 6a>+≡` <37a 39a>

```
\newcommand{\nwindexdefn}{%
  <remove ligatures, make actives \other 38b>
  \@nwindexdefn}
\newcommand{\@nwindexdefn}[3]{%<printable id> <id label> <key>
  \nw@auxix{\protect\nwixd}{#2}{#3}}
\newcommand{\nwindexuse}{%
  <remove ligatures, make actives \other 38b>
  \@nwindexuse}
\newcommand{\@nwindexuse}[3]{%<printable id> <id label> <key>
  \nw@auxix{\protect\nwixu}{#2}{#3}}
```

Defines:

```
\@nwindexdefn, never used.
\@nwindexuse, never used.
\nwindexdefn, never used.
\nwindexuse, never used.
```

Uses `\nw@auxix` 39a, `\nwixd` 42b 49d, and `\nwixu` 42b 49d.

38b `<remove ligatures, make actives \other 38b>≡` (38a 43b)

```
\@noligs
\ifx\verbatim@nolig@list\undefined
\else
  \let\do=\nw@makeother
  \verbatim@nolig@list
\fi
```

Uses `\nw@makeother` 14c.

Write the string ‘`\nwixadd{<marker>}{<id label>}{<supage label>}`’ to the auxiliary file. The field `<marker>` is either ‘`\nwixd`’ or ‘`\nwixu`’, corresponding to the definition or use of an indexed item, respectively.

39a `<knoweb.sty 6a>+≡` <38a 39b>

```
\newcommand{\nw@auxix}[3]{%<marker> <id label> <key>
  \@bsphack
  \if@files
  {\let\nwixd=\relax
   \let\nwixu=\relax
   \def\protect{\noexpand\noexpand\noexpand}%
   \edef\@tempa{\write\@auxout{\string\nwixadd{#1}{#2}{#3}}}%
   \expandafter\@tempa
   \if@nbreak\ifvmode\nobreak\fi\fi
  \fi
  \@esphack}
```

Defines:

`\nw@auxix`, used in chunk 38a.

Uses `\nwixadd` 39b, `\nwixd` 42b 49d, and `\nwixu` 42b 49d.

If the macro `\nwixl@<id label>` is undefined, define it so that its expansion is `<marker>{<key>}`. Otherwise, reassign it, appending `<marker>{<key>}` to its current expansion.

39b `<knoweb.sty 6a>+≡` <39a 41b>

```
\newcommand{\nwixadd}[3]{%<marker> <id label> <key>
  \@ifundefined{nwixl@#2}%
  {\global\@namedef{nwixl@#2}{#1{#3}}}%
  {\expandafter\nw@ixappend\csname nwixl@#2\endcsname{#1{#3}}}}
```

Defines:

`\nwixadd`, used in chunk 39a.

Uses `\nw@ixappend` 42a.

2.12.2 Subscripted Identifiers

We use either explicit subscripts or hyperlinks to point identifiers to their definitions. Each command takes two arguments, `<identifer>` and `<link>`.

39c `<initial code 6c>+≡` (6a) <34c 45b>

```
\newcommand{\nw@subscriptident}[2]{\mbox{$\mbox{#1}_{\mathrm{\subpageref{#2}}}$}}
\newcommand{\nw@nosubscriptident}[2]{#1}
\newcommand{\nw@hyperident}[2]{\leavevmode\nwhyperreference{#2}{#1}}
```

Defines:

`\nw@hyperident`, used in chunk 40.

`\nw@nosubscriptident`, used in chunk 40.

`\nw@subscriptident`, used in chunk 40.

Uses `\nwhyperreference` 51b and `\subpageref` 28g.

We can use subscripts, hyperlinks, or nothing on all identifiers. Noweave inserts the macros `\nwlinkedidentq` and `\nwlinkedidentc` into the \TeX file.

40a *<declaration of options 9a>+≡* (6a) <37b 40b>

```

\DeclareOption{subscriptidents}{%
  \let\nwlinkedidentq=\nw@subscriptident
  \let\nwlinkedidentc=\nw@subscriptident
}
\DeclareOption{nosubscriptidents}{%
  \let\nwlinkedidentq=\nw@nosubscriptident
  \let\nwlinkedidentc=\nw@nosubscriptident
}
\DeclareOption{hyperidents}{%
  \let\nwlinkedidentq=\nw@hyperident
  \let\nwlinkedidentc=\nw@hyperident
}
\DeclareOption{nohyperidents}{%
  \let\nwlinkedidentq=\nw@nosubscriptident
  \let\nwlinkedidentc=\nw@nosubscriptident
}

```

Defines:

`\nwlinkedidentc`, never used.
`\nwlinkedidentq`, used in chunk 40b.
`hyperidents`, used in chunk 41a.
`nohyperidents`, never used.
`nosubscriptidents`, never used.
`subscriptidents`, never used.

Uses `\nw@hyperident 39c`, `\nw@nosubscriptident 39c`, and `\nw@subscriptident 39c`.

We can change only identifiers appearing in quoted code.

40b *<declaration of options 9a>+≡* (6a) <40a 44c>

```

\DeclareOption{subscriptquotedidents}{%
  \let\nwlinkedidentq=\nw@subscriptident
}
\DeclareOption{nosubscriptquotedidents}{%
  \let\nwlinkedidentq=\nw@nosubscriptident
}
\DeclareOption{hyperquotedidents}{%
  \let\nwlinkedidentq=\nw@hyperident
}
\DeclareOption{nohyperquotedidents}{%
  \let\nwlinkedidentq=\nw@nosubscriptident
}

```

Defines:

`hyperquotedidents`, never used.
`nohyperquotedidents`, never used.
`nosubscriptquotedidents`, never used.
`subscriptquotedidents`, never used.

Uses `\nw@hyperident 39c`, `\nw@nosubscriptident 39c`, `\nw@subscriptident 39c`, and `\nwlinkedidentq 40a`.

The default is to hyperlink everything.

41a $\langle \text{execution of options 9e} \rangle + \equiv$ (6a) $\langle 36d \ 44d \rangle$
 $\backslash \text{ExecuteOptions}\{\text{hyperidents}\}$
 Uses hyperidents 40a.

2.12.3 Writing Lists with Commas and Conjunction

Given a list of items, $\backslash\langle item_1 \rangle \backslash\langle item_2 \rangle \dots \backslash\langle item_{n-1} \rangle \backslash\langle item_n \rangle$, return a list with the items separated by commas, spaces, and an appropriate conjunction:

$$\backslash\langle item_1 \rangle, \backslash\langle item_2 \rangle, \dots, \backslash\langle item_{n-1} \rangle, \langle and \rangle \sim \backslash\langle item_n \rangle.$$

The conjunction, $\langle and \rangle$, depends on the selected language. For $n \geq 3$ a comma is inserted after each item but the last and a tie is inserted between the conjunction and the final item.

41b $\langle \text{knoweb.sty 6a} \rangle + \equiv$ $\langle 39b \ 42a \rangle$
 $\backslash \text{newcommand}\backslash \text{nw@commafy}[1]\{ \%$
 $\quad \{ \backslash \text{nw@ixlistcount}\{ \#1 \} \%$
 $\quad \quad \backslash \text{count@} = \backslash \text{nw@ixcounter} \quad \quad \% \backslash \text{count@} \leftarrow \langle \text{number of items in list} \rangle$
 $\quad \quad \backslash \text{let}\backslash @\text{comma@each} = \backslash \%$
 $\quad \quad \backslash \text{ifcase}\backslash \text{count@}$
 $\quad \quad \quad \backslash \text{let}\backslash = \backslash @\text{comma@each} \quad \quad \% \langle \text{number of items in list} \rangle = 0$
 $\quad \quad \backslash \text{or}$
 $\quad \quad \quad \backslash \text{let}\backslash = \backslash @\text{comma@each} \quad \quad \% \langle \text{number of items in list} \rangle = 1$
 $\quad \quad \backslash \text{or}$
 $\quad \quad \quad \backslash \text{def}\backslash \{ \backslash \text{def}\backslash \{ \backslash \text{space}\backslash \text{nw@langdepand}\backslash \text{space}\backslash @\text{comma@each} \} \%$
 $\quad \quad \quad \quad \backslash @\text{comma@each} \} \%$
 $\quad \quad \backslash \text{else} \quad \quad \% \langle \text{number of items in list} \rangle \geq 3$
 $\quad \quad \quad \backslash \text{def}\backslash \{ \backslash \text{def}\backslash \{ , \backslash \text{space} \%$
 $\quad \quad \quad \quad \quad \backslash \text{advance}\backslash \text{count@} \text{ by } \backslash \text{m@ne}$
 $\quad \quad \quad \quad \quad \backslash \text{ifnum}\backslash \text{count@} = \backslash @\text{ne}$
 $\quad \quad \quad \quad \quad \quad \backslash \text{nw@langdepand}\backslash \text{nobreakspace}$
 $\quad \quad \quad \quad \quad \backslash \text{fi}$
 $\quad \quad \quad \quad \quad \quad \backslash @\text{comma@each} \} \%$
 $\quad \quad \quad \quad \quad \backslash @\text{comma@each} \} \%$
 $\quad \quad \backslash \text{fi}$
 $\quad \quad \#1 \} \}$

Defines:

$\backslash \text{nw@commafy}$, used in chunks 22a, 23a, and 43b.

Uses $\backslash \text{nw@ixcounter}$ 42c, $\backslash \text{nw@ixlistcount}$ 42c, and $\backslash \text{nw@langdepand}$ 54a.

2.12.4 Improved Index Code

There are two kinds of lists. One kind is a generic list in which elements are preceded by $\backslash\langle$. If the elements are index elements, they are $\langle \text{printable identifier} \rangle \langle \text{label} \rangle$ pairs. The other kind is a list of sub-page labels, in which each element is preceded by either $\backslash \text{nwixd}$ or $\backslash \text{nwixu}$.

Append the tokens to $\langle list \rangle$.

42a $\langle knoweb.sty 6a \rangle + \equiv$ $\langle 41b 42b \rangle$

```
\newcommand\nw@ixappend[2]{% $\langle list \rangle$  \marker $\langle element \rangle$ 
  {\toks0=\expandafter{#1}%
  \def\@tempa{#2}%
  \toks2=\expandafter{\@tempa}%
  \xdef#1{\the\toks0 \the\toks2 }}}}
```

Defines:

$\nw@ixappend$, used in chunks 24e, 28b, 39b, and 50d.

The reference list for an identifier labeled $\langle id \rangle$ is always called $\nwixl@ \langle id \rangle$. Most applications will work with reference lists by applying $\backslash\backslash$ either to the defs or to the uses.

42b $\langle knoweb.sty 6a \rangle + \equiv$ $\langle 42a 42c \rangle$

```
\newcommand\nw@ixuses[1]{% $\langle label \rangle$ 
  \def\nwixu{\backslash}%
  \let\nwixd=\@gobble
  \@nameuse{nwixl@#1}}
\newcommand\nw@ixdefs[1]{% $\langle label \rangle$ 
  \def\nwixd{\backslash}%
  \let\nwixu=\@gobble
  \@nameuse{nwixl@#1}}}
```

Defines:

$\nw@ixdefs$, used in chunks 42d and 43a.

$\nw@ixuses$, used in chunks 42d and 44b.

\nwixd , used in chunks 38a and 39a.

\nwixu , used in chunks 38a and 39a.

Counting Some applications count uses to see whether there is any need to display information. Count the number of items in a list. Assign the global counter $\nw@ixcounter$ the value.

42c $\langle knoweb.sty 6a \rangle + \equiv$ $\langle 42b 42d \rangle$

```
\newcommand\nw@ixlistcount[1]{% $\langle list with items prefixed with '\backslash' \rangle$ 
  {\count@=\z@
  \def\backslash##1{\advance\count@ by\@ne}%
  #1%
  \global\nw@ixcounter=\count@}}
\newcount\nw@ixcounter
```

Defines:

$\nw@ixcounter$, used in chunks 41b, 43, and 44.

$\nw@ixlistcount$, used in chunks 41b, 42d, and 44a.

Assign macros for counting the number of definitions and uses of an identifier.

42d $\langle knoweb.sty 6a \rangle + \equiv$ $\langle 42c 43a \rangle$

```
\newcommand\nw@ixdefcount[1]{\nw@ixlistcount{\nw@ixdefs{#1}}}
\newcommand\nw@ixusecount[1]{\nw@ixlistcount{\nw@ixuses{#1}}}
```

Defines:

$\nw@ixdefcount$, never used.

$\nw@ixusecount$, used in chunks 43c and 44b.

Uses $\nw@ixdefs$ 42b, $\nw@ixlistcount$ 42c, and $\nw@ixuses$ 42b.

2.12.5 Supporting a Mini-Index at the End of each Chunk

When displaying identifiers used, show the identifier and its definitions.

43a `<knoweb.sty 6a>+≡` <42d 43b>

```
\newcommand\nw@ixid@defs[1]{%\index pair}
  {\Tt \@car#1\@nil}%
  \def\##1{\nw@ixdefs@space\subpageref{##1}\nw@ixdefs{\@cdr#1\@nil}}}
```

Defines:

`\nw@ixid@defs`, used in chunk 43b.

Uses `\nw@ixdefs` 42b, `\nw@ixdefs@space` 44c, `\subpageref` 28g, and `\Tt` 10c.

noweave inserts the calls to `\nwidentuses` into the T_EX file. Its argument is a list of index pairs, each index pair has the form `{\printable id}{\id label}`.

43b `<knoweb.sty 6a>+≡` <43a 43c>

```
\newcommand{\nwidentuses}{%
  <remove ligatures, make actives \other 38b>
  \@nwidentuses}
\newcommand{\@nwidentuses}[1]{%\<list of index pairs>
  \nwcodecomment{%
    \nw@langdepuss\space
    \let\=\nw@ixid@defs
    \nw@commafy{#1}.}}
```

Defines:

`\@nwidentuses`, never used.

`\nwidentuses`, used in chunk 45d.

Uses `\nw@commafy` 41b, `\nw@ixid@defs` 43a, `\nw@langdepuss` 54a, and `\nwcodecomment` 18a.

The definitions section is a bit more complex, because it is omitted if none of the identifiers defined is ever used.

Set the global counter `\nw@ixcounter` to the number of used identifiers in the argument.

43c `<knoweb.sty 6a>+≡` <43b 44a>

```
\newcommand\nw@ixtotaluses[1]{%\<list of index pairs>
  {\count@=\z@
  \def\##1{%\<index pair>
    \nw@ixusecount{\@cdr##1\@nil}%
    \advance\count@ by\nw@ixcounter}%
  #1\global\nw@ixcounter=\count@ }}
```

Defines:

`\nw@ixtotaluses`, used in chunk 44a.

Uses `\nw@ixcounter` 42c and `\nw@ixusecount` 42d.

Display the symbols defined by a chunk.

44a *<knoweb.sty 6a>+≡* <43c 44b>

```
\newcommand{\nwidentdefs}[1]{%<list of index pairs>
  \ifnw@hideunuseddefs
    \nw@ixtotaluses{#1}%
  \else\nw@ixlistcount{#1}%
  \fi
  \ifnum\nw@ixcounter>\z@
    \nwcodecomment{\nw@langdepdfs:}%
    {\def\\##1{\nw@ixid@uses##1}#1}%
  \fi}
```

Defines:

\nwidentdefs, used in chunk 45d.

Uses \ifnw@hideunuseddefs 45b, \nw@ixcounter 42c, \nw@ixlistcount 42c, \nw@ixtotaluses 43c, \nw@langdepdfs 54a, and \nwcodecomment 18a.

44b *<knoweb.sty 6a>+≡* <44a 46a>

```
\newcommand\nw@ixid@uses[2]{%<ident> <label>
  \nw@ixusecount{#2}%
  {\addtolength{\leftskip}{\nwcodeindent}}%
  \ifnum\nw@ixcounter>\z@
    \nwcodecomment{{\Tt #1},\space
      \nw@langdepusd\space
      \nw@langdepin\space\nw@chunkcommachunketc{\nw@ixuses{#2}}.}%
  \else
    \ifnw@hideunuseddefs
      \else
        \nwcodecomment{{\Tt #1},\space\nw@langdepnvu.}%
      \fi
    \fi}}
```

Defines:

\nw@ixid@uses, never used.

Uses \ifnw@hideunuseddefs 45b, \nw@chunkcommachunketc 22a, \nw@ixcounter 42c, \nw@ixusecount 42d, \nw@ixuses 42b, \nw@langdepin 54a, \nw@langdepnvu 54a, \nw@langdepusd 54a, \nwcodecomment 18a, \nwcodeindent 10d, and \Tt 10c.

Package Options

44c *<declaration of options 9a>+≡* (6a) <40b 45c>

```
\DeclareOption{breakdefs}{\def\nw@ixdefs@space{\penalty200\ }}
\DeclareOption{nobreakdefs}{\def\nw@ixdefs@space{\nobreakspace}}
```

Defines:

\nw@ixdefs@space, used in chunk 43a.

breakdefs, used in chunks 44d and 45a.

nobreakdefs, used in chunk 45a.

44d *<execution of options 9e>+≡* (6a) <41a 53>

```
\ExecuteOptions{breakdefs}
```

Uses breakdefs 44c.

- 45a** *<undocumented – man page: \noweboptions 45a>*≡
`.TP`
`.B breakdefs, nobreakdefs`
`.BR breakdefs ,`
 which is the default,
 permits long lists of definitions to be broken in identifier cross-reference.
 Useful if identifier cross-reference produces lots of overfull hboxes.
`.B nobreakdefs`
 is the old behavior, which should never be needed.
 Uses `breakdefs 44c` and `nobreakdefs 44c`.
- 45b** *<initial code 6c>*+≡ (6a) <39c 46b>
`\newif\ifnw@hideunuseddefs\nw@hideunuseddefsfalse`
 Defines:
`\ifnw@hideunuseddefs`, used in chunk 44.
`\nw@hideunuseddefsfalse`, never used.
`\nw@hideunuseddefstrue`, used in chunk 45c.
- 45c** *<declaration of options 9a>*+≡ (6a) <44c 45d>
`\DeclareOption{hideunuseddefs}{\nw@hideunuseddefstrue}`
 Defines:
`hideunuseddefs`, never used.
 Uses `\nw@hideunuseddefstrue 45b`.
- 45d** *<declaration of options 9a>*+≡ (6a) <45c 47d>
`\DeclareOption{noidentxref}{%`
`\let\nwidetdefs=\@gobble`
`\let\nwidetuses=\@gobble}`
 Defines:
`noidentxref`, never used.
 Uses `\nwidentdefs 44a` and `\nwidentuses 43b`.

2.12.6 Support for Chunk and Identifier Indices

Assign control sequences for typesetting chunk and identifier indices.

Chunk Index Assign the user-command, `\nowebchunks`, that typesets the chunk index. The command prints a language-dependent preamble, `\nw@langdeppre`, that explains the distinction between the underlined and normal indices then it typsets the index. The starred version of `\nowebchunks` omits the preamble.

Check for the starred version, set the flag `\nw@chunks preamble` appropriately, then call `\nw@chunks` (which may have been reassigned by the option `externalindex`).

```
46a <knoweb.sty 6a>+≡ <44b 47a>
\newcommand\nowebchunks{%
  \@ifstar
    {\nw@chunks preamble false\nw@chunks}
    {\nw@chunks preamble true\nw@chunks}}
\newif\ifnw@chunks preamble
```

Defines:

`\ifnw@chunks preamble`, used in chunk 47a.

`\nowebchunks`, used in chunk 46b.

Uses `\nw@chunks` 46b.

Internal Chunk Index Assign the macro that typesets the internal built chunk index. The macro `\nwixs@c` is the list of chunks; if it is not defined, issue a warning and do nothing.

```
46b <initial code 6c>+≡ (6a) <45b 46c>
\newcommand\nw@chunks{%
  \@ifundefined{nwixs@c}
    {\@warning{There are no \string\nowebchunks}}
    {\begin{thenowebchunks}\nwixs@c\end{thenowebchunks}}}
```

Defines:

`\nw@chunks`, used in chunks 46a and 51a.

Uses `\nowebchunks` 46a and `thenowebchunks` 47a.

External Chunk Index Assign the macro that typesets an external chunk index, that is, the contents of the file `\jobname.nwi`.

```
46c <initial code 6c>+≡ (6a) <46b 47c>
\newcommand\nowebchunks@external{%
  {\let\nwixadds@i=\@gobble
  \def\nwixadds@c##1{\nw@onechunk##1}%
  \assign\nwixaddsx 50a}%
  \begin{thenowebchunks}\@input{\jobname.nwi}\end{thenowebchunks}}
\@namedef{r@nw@notdef}{\@ifundefined{nw@langdepnvd}}}
```

Defines:

`\nowebchunks@external`, used in chunk 51a.

`\r@nw@notdef`, never used.

Uses `\nw@langdepnvd` 54a and `thenowebchunks` 47a.

The Chunks Index Environment Print a language appropriate preamble, `\nw@langdeppre`, unless the `longchunks` option was not selected or the starred version of `\nowebchunks` was used. Then setup the paragraph indentation. *Why is 10pt used? Should this be 1em?*

47a `<knoweb.sty 6a>+≡` <46a 47b>

```
\newenvironment{thenowebchunks}
{\ifnw@chunks preamble
 \ifnw@longchunks
 \nw@langdeppre\par\vspace{1ex}%
 \fi\fi
 <set index environment paragraph spacing 49c>
 \def\##1{\nw@onechunk ##1}}
{}
```

Defines:

`thenowebchunks`, used in chunk 46.

Uses `\ifnw@chunks preamble` 46a, `\ifnw@longchunks` 47c, `\nw@langdeppre` 54a, and `\nw@onechunk` 47b.

Typeset an indexed chunk as ‘`<<name> <label> [<list>]`’, where `<name>` is the name of the chunk, typeset in the normal font; `<label>` is the chunk label, typeset with `\tagstyle`; and the optional `<list>` is a list of tags pointing to the definition and uses of the chunk, it is typeset only if the `longchunks` package option is enabled. *It might be useful to add a `\chunknamestyle` macro to permit the user to modify the font used in the chunk name. This could be used everywhere, or restricted to the index.*

47b `<knoweb.sty 6a>+≡` <47a 48a>

```
\newcommand\nw@onechunk[3]{%<name> <label of first definition> <list with \nwixd, \nwixu>
 \@ifundefined{r@#2}{-}{%
 \indent\LA #1\nobreakspace{\nwtagstyle\subpageref{#2}}\RA
 \ifnw@longchunks\nobreakspace{\nw@underlinedefs{#3}}\fi
 \par}}
```

Defines:

`\nw@onechunk`, used in chunk 47a.

Uses `\ifnw@longchunks` 47c, `\LA` 8a, `\nw@underlinedefs` 49d, `\nwtagstyle` 10b, `\RA` 8a, and `\subpageref` 28g.

Chunk Package Option Assign the package option `longchunks`, used to add a list of tags of the definition and uses of a chunk in the chunk index.

47c `<initial code 6c>+≡` (6a) <46c 48b>

```
\newif\ifnw@longchunks
\nw@longchunksfalse
```

Defines:

`\ifnw@longchunks`, used in chunk 47.

47d `<declaration of options 9a>+≡` (6a) <45d 51a>

```
\DeclareOption{longchunks}{\nw@longchunkstrue}
```

Defines:

`longchunks`, never used.

Identifier Index Assign the user-command, `\nowebindex`, that typesets the identifier index. The command prints a language-dependent preamble, `\nw@langdeppre`, that explains the distinction between the underlined and normal indices then it typesets the index. The starred version of `\nowebindex` omits the preamble.

Check for the starred version, set the flag `\nw@indexpreamble` appropriately, then call `\nw@index` (which may have been reassigned by the option `externalindex`).

```
48a <knoweb.sty 6a>+≡ <47b 49a>
  \newcommand\nowebindex{%
    \@ifstar
      {\nw@indexpreamblefalse\nw@index}
      {\nw@indexpreambletrue\nw@index}}
  \newif\ifnw@indexpreamble
Defines:
  \ifnw@indexpreamble, used in chunk 49a.
  \nowebindex, used in chunk 48b.
Uses \nw@index 48b.
```

Internal Identifier Index Assign the macro that typesets the internally built identifier index. The macro `\nwixs@i` is the list of identifiers; if it is not defined, issue a warning and do nothing.

```
48b <initial code 6c>+≡ (6a) <47c 48c>
  \newcommand\nw@index{%
    \@ifundefined{nwixs@i}
      {\@warning{The \string\nowebindex{} is empty}}
      {\begin{thenowebindex}\nwixs@i\end{thenowebindex}}}
```

Defines:
`\nw@index`, used in chunks 48a and 51a.
 Uses `\nowebindex` 48a and `thenowebindex` 49a.

External Identifier Index Assign the macro that typesets an external identifier index, that is, the contents of the file `\jobname.nwi`.

```
48c <initial code 6c>+≡ (6a) <48b 50b>
  \newcommand\nowebindex@external{%
    {\let\nwixadds@c=\@gobble
     \def\nwixadds@i##1{\nw@indexline##1}%
     \assign\nwixaddsx 50a}%
    \begin{thenowebindex}\@input{\jobname.nwi}\end{thenowebindex}}}
```

Defines:
`\nowebindex@external`, used in chunk 51a.
 Uses `thenowebindex` 49a.

The Identifier Index Environment Print a language appropriate preamble, `\nw@langdeppre`, unless the starred version of `\nowebindex` was used. Then setup the paragraph indentation.

49a `\knoweb.sty 6a` +≡ <48a 49b>

```

\newenvironment{thenowebindex}
{
  \ifnw@indexpreamble
    \nw@langdeppre\par\vspace{1ex}%
  \fi
  \set index environment paragraph spacing 49c
  \def\##1{\nw@indexline ##1}}
{}

```

Defines:

`thenowebindex`, used in chunk 48.

Uses `\ifnw@indexpreamble` 48a, `\nw@indexline` 49b, and `\nw@langdeppre` 54a.

Typeset an indexed identifier as ‘`<name>: <list>`’, where `<name>` is the name of the identifier, typeset in a typewriter font and `<list>` is a list of tags pointing to the definition and uses of the identifier.

49b `\knoweb.sty 6a` +≡ <49a 49d>

```

\newcommand\nw@indexline[2]{%
  \indent{\Tt #1}:\space{\nw@underlinedefs\@nameuse{nwixl@#2}}%
  \par}

```

Defines:

`\nw@indexline`, used in chunk 49a.

Uses `\nw@underlinedefs` 49d and `\Tt` 10c.

Common Index Macros Why is 10pt used? Should this be 1em?

49c `\set index environment paragraph spacing 49c` ≡ (47a 49a)

```

\vspace{3pt}
\setlength{\parskip}{\z@}%
\setlength{\parindent}{-10pt}%
\addtolength{\leftskip}{10pt}%
\addtolength{\rightskip}{\z@ plus10pt}%
\@afterindenttrue

```

49d `\knoweb.sty 6a` +≡ <49b 50c>

```

\newcommand\nw@underlinedefs{%<list with \nwixd and \nwixu>
  \let\\=\relax
  \def\nw@comma{,\space}%
  \def\nwixd##1{\\\underline{\subpageref{##1}}\let\\\nw@comma}%
  \def\nwixu##1{\\\subpageref{##1}\let\\\nw@comma}}

```

Defines:

`\nw@underlinedefs`, used in chunks 47b and 49b.

`\nwixd`, used in chunks 38a and 39a.

`\nwixu`, used in chunks 38a and 39a.

Uses `\subpageref` 28g.

50a `<assign \nwixaddsx 50a>≡` (46c 48c)
`\def\nwixaddsx##1##2{\@nameuse{\nwixadds@##1}{##2}}%`

Defines:

`\nwixaddsx`, used in chunk 51a.

Allocate and assign the default value of `\nwixaddsx`.

50b `<initial code 6c>+≡` (6a) <48c 50d>
`\newcommand{\nwixaddsx}{}`
`\let\nwixaddsx=\@gobbletwo`

Defines:

`\nwixaddsx`, used in chunk 51a.

noweave inserts strings `'\nwixlogsorted{<type>}{<data>}'` into the T_EX file. When expanded these strings create the lists of indexed chunks and identifiers. The field `<type>` is either 'c', indicating a chunk, or 'i', indicating an identifier. The expansion of `\nwixlogsorted` writes the string `'\nwixadds{<type>}{<data>}'` to the auxiliary file.

50c `<knoweb.sty 6a>+≡` <49d 51b>
`\newcommand\nwixlogsorted[2]{%<type> <data>`
`\@bsphack`
`\if@filesw`
`\toks0={#2}%`
`\immediate\write\@auxout{\string\nwixadds{#1}{\the\toks0}}%`
`\if@nobreak\ifvmode\nobreak\fi\fi`
`\fi`
`\@esphack}`

Defines:

`\nwixlogsorted`, never used.

Uses `\nwixadds` 50d.

Append an element to the chunk list, `\nwixs@c`, or the identifier list, `\nwixs@i`, depending whether the first argument, `<type>`, is 'c' or 'i', respectively. If the list does not exist, create it.

50d `<initial code 6c>+≡` (6a) <50b>
`\newcommand\nwixadds[2]{%<type> <data>`
`\@ifundefined{\nwixs@#1}%`
`{\global\@namedef{\nwixs@#1}{\{\{#2\}\}}%`
`{\expandafter\nw@ixappend\csname nwixs@#1\endcsname{\{\{#2\}\}}}`

Defines:

`\nwixadds`, used in chunks 50c and 51a.

Uses `\nw@ixappend` 42a.

If an external index is used, we need a .nwi file, `\nwixadds` is to be ignored, and we use `\nwixaddsx`.

51a `<declaration of options 9a>+≡` (6a) `<47d 54a>`

```
\DeclareOption{externalindex}{%
  \let\nwixaddsx=\nwixadds
  \let\nwixadds=\@gobbletwo
  \let\nw@index=\nowebindex@external
  \let\nw@chunks=\nowebchunks@external}
```

Defines:

`externalindex`, never used.

Uses `\nowebchunks@external` 46c, `\nowebindex@external` 48c, `\nw@chunks` 46b, `\nw@index` 48b, `\nwixadds` 50d, and `\nwixaddsx` 50a 50b.

2.13 Support for Hypertext

There are two sets of support for hypertext. Balasubramanian Narasimhan wrote initial support for `hyper.sty`. Norman Ramsey wrote support for `hyperref.sty` (May 1998). The macros `\nwblindhyperanchor` and `\nwhyperreference` are assigned to handle the appropriate package.

The `\nwblindhyperanchor` macro takes a single argument, `<key>`, and inserts a hyperlink anchor at the location it is expanded. If no hypertext package has been loaded, it gobbles its argument.

The `\nwhyperreference` macro has two arguments, `<key>` and `<identifier>`. It inserts a hyperlink consisting of the text `<identifier>` at the location it is expanded and pointing to the anchor `<key>`. If no hypertext package is loaded, it gobbles its first argument, `<key>`, and prints `<identifier>`.

51b `<knoweb.sty 6a>+≡` `<50c 52a>`

```
\newcommand\nwblindhyperanchor{\@gobble}
\newcommand\nwhyperreference{\@gobble}
\AtBeginDocument{%
  \@ifpackageloaded{hyperref}
  {\gdef\nwblindhyperanchor#1{\hypertarget{noweb.#1}{\relax}}}%
  {\gdef\nwhyperreference#1#2{\hyperlink{noweb.#1}{#2}}}
  {}
  \@ifpackageloaded{hyper}
  {\gdef\nwblindhyperanchor{\blindhyperanchor}%
   \gdef\nwhyperreference{\hyperreference}}
  {}}}
```

Defines:

`\nwblindhyperanchor`, used in chunks 30b, 33a, and 37a.

`\nwhyperreference`, used in chunks 27c, 28g, and 39c.

2.14 Support for Hypertext Translation to HTML

52a *<knoweb.sty 6a>+≡* <51b 52b>

```
\newcommand\nwanchorto{%
  \begingroup\let\do\@makeother\dospecials
  \catcode'\{=1 \catcode'\}=2 \nw@anchorto}
\newcommand{\nw@anchorto}[1]{\endgroup\def\nw@next{#1}\nw@anchortofin}
\newcommand{\nw@anchortofin}[1]{#1\footnote{See URL \texttt{\nw@next}.}}
\let\nwanchortname\@gobble
```

Defines:

```
\nw@anchorto, never used.
\nw@anchortofin, never used.
\nwanchorto, never used.
```

This lets us hide stuff intended for use only when converting to HTML:

52b *<knoweb.sty 6a>+≡* <52a 52c>

```
\newif\ifhtml
\htmlfalse
```

Defines:

```
\ifhtml, never used.
```

2.15 Support for Prettyprinting

The following macro can be redefined to allow custom typesetting of identifiers in the index and mini-indices.

52c *<knoweb.sty 6a>+≡* <52b 52d>

```
\newcommand\nwixident{}
\let\nwixident=\relax
```

Defines:

```
\nwixident, never used.
```

The following macros can be redefined to typeset ‘\’, ‘{’ and ‘}’ correctly in non-typewriter fonts. The problem is that the built-in \LaTeX `\{` tries to produce a math symbol, which doesn’t exist in the typewriter font, so we get a brace in the wrong font and a warning. Most unpleasant. Noweave therefore attempts to emit `\nwlbrace` and `\nwrbrace` wherever it believes braces should appear. The standard noweb style is to set code in typewriter font, and so the standard definitions just select the proper characters from that font. People setting code in fonts other than typewriter are responsible for redefining those macros to work in their environment.

52d *<knoweb.sty 6a>+≡* <52c 62h>

```
\newcommand{\nwbackslash}{\ifmmode\backslash\else\textbackslash\fi}
\newcommand{\nwlbrace}{\ifmmode\lbrace\else\textbraceleft\fi}
\newcommand{\nwrbrace}{\ifmmode\rbrace\else\textbraceright\fi}
```

2.16 Language-Dependent Macros

Miguel Filgueiras (DCC-FCUP & LIACC, Universidade do Porto) provided some changes to add multilingual support for the words Noweb uses in indexing and cross-reference. He inserted macros that are defined by, e.g., `\noweboptions{english}`. The Noweb package uses the (apparently standard) \LaTeX macro `\language` to select a language at load time. If the `babel` package is loaded (with the appropriate language name) before Noweb is loaded, the Noweb package selects the language appropriately, provided it is one that Noweb supports. Mr. Filgueiras provided support for English, Portuguese, German, and French. He notes that the French is faulty; the translations may be poor, and there are bugs in the implementation that he could not solve.

53 *<execution of options 9e>+≡* (6a) <44d
`\ExecuteOptions{english}`
 Uses english 54a.

2.16.1 Support for English

This describes the original English text.

54a *<declaration of options 9a>+≡* (6a) <51a 56a>

```

\DeclareOption{english}{%
  \def\nw@langdepdef{<english: This definition is continued 54b>}%
  \def\nw@langdep cud{<english: This code is used 55a>}%
  \def\nw@langdep rtc{<english: Root chunk (not used in this document) 55b>}%
  \def\nw@langdep cwf{<english: This code is written to file 55c>}%
  \def\nw@langdep chk{<english: chunk 55d>}%
  \def\nw@langdep chks{<english: chunks 55e>}%
  \def\nw@langdep in{<english: in 55f>}%
  \def\nw@langdep and{<english: and 55g>}%
  \def\nw@langdep uss{<english: Uses 55h>}%
  \def\nw@langdep usd{<english: used 55i>}%
  \def\nw@langdep nvu{<english: never used 55j>}%
  \def\nw@langdep dfs{<english: Defines 55k>}%
  \def\nw@langdep nvd{<english: never defined 55l>}%
  \def\nw@langdep pre{<english: Underlined indices... 55m>}%
}
\DeclareOption{american}{\ExecuteOptions{english}}
\DeclareOption{USenglish}{\ExecuteOptions{english}}
\DeclareOption{canadian}{\ExecuteOptions{english}}
\DeclareOption{UKenglish}{\ExecuteOptions{english}}
\DeclareOption{british}{\ExecuteOptions{english}}

```

Defines:

```

\nw@langdep and, used in chunks 41b, 56a, 57i, 59a, and 61a.
\nw@langdep chk, used in chunks 22c, 56a, 57i, 59a, and 61a.
\nw@langdep chks, used in chunks 22c, 56a, 57i, 59a, and 61a.
\nw@langdep cud, used in chunks 19a, 56a, 57i, 59a, and 61a.
\nw@langdep cwf, used in chunks 19a, 56a, 57i, 59a, and 61a.
\nw@langdep def, used in chunks 19a, 56a, 57i, 59a, and 61a.
\nw@langdep dfs, used in chunks 44a, 56a, 57i, 59a, and 61a.
\nw@langdep in, used in chunks 19a, 44b, 56a, 57i, 59a, and 61a.
\nw@langdep nvd, used in chunks 46c, 56a, 57i, 59a, and 61a.
\nw@langdep nvu, used in chunks 44b, 56a, 57i, 59a, and 61a.
\nw@langdep pre, used in chunks 47a, 49a, 56a, 57i, 59a, and 61a.
\nw@langdep rtc, used in chunks 19a, 56a, 57i, 59a, and 61a.
\nw@langdep usd, used in chunks 44b, 56a, 57i, 59a, and 61a.
\nw@langdep uss, used in chunks 43b, 56a, 57i, 59a, and 61a.
american, never used.
british, never used.
canadian, never used.
english, used in chunk 53.
UKenglish, never used.
USenglish, never used.

```

54b *<english: This definition is continued 54b>≡* (54a)
 This definition is continued

- 55a $\langle \text{english: This code is used } 55a \rangle \equiv$ (54a)
 This code is used
- 55b $\langle \text{english: Root chunk (not used in this document) } 55b \rangle \equiv$ (54a)
 Root chunk (not used in this document)
- 55c $\langle \text{english: This code is written to file } 55c \rangle \equiv$ (54a)
 This code is written to file
- 55d $\langle \text{english: chunk } 55d \rangle \equiv$ (54a)
 chunk
- 55e $\langle \text{english: chunks } 55e \rangle \equiv$ (54a)
 chunks
- 55f $\langle \text{english: in } 55f \rangle \equiv$ (54a)
 in
- 55g $\langle \text{english: and } 55g \rangle \equiv$ (54a)
 and
- 55h $\langle \text{english: Uses } 55h \rangle \equiv$ (54a)
 Uses
- 55i $\langle \text{english: used } 55i \rangle \equiv$ (54a)
 used
- 55j $\langle \text{english: never used } 55j \rangle \equiv$ (54a)
 never used
- 55k $\langle \text{english: Defines } 55k \rangle \equiv$ (54a)
 Defines
- 55l $\langle \text{english: never defined } 55l \rangle \equiv$ (54a)
 never defined
- 55m $\langle \text{english: Underlined indices... } 55m \rangle \equiv$ (54a)
 \underline{Underlined} indices denote definitions;
 regular indices denote uses.

2.16.2 Support for Portuguese

This contains the text in Portuguese.

- 56a *<declaration of options 9a>+≡* (6a) <54a 57i>
- ```

\DeclareOption{portuges}{%
 \def\nw@langdepdef{<portuguese: This definition is continued 56b>}%
 \def\nw@langdepdpcud{<portuguese: This code is used 56c>}%
 \def\nw@langdepdprtc{<portuguese: Root chunk (not used in this document) 56d>}%
 \def\nw@langdepdpcwf{<portuguese: This code is written to file 56e>}%
 \def\nw@langdepdchk{<portuguese: chunk 56f>}%
 \def\nw@langdepdchks{<portuguese: chunks 56g>}%
 \def\nw@langdepdin{<portuguese: in 57a>}%
 \def\nw@langdepdand{<portuguese: and 57b>}%
 \def\nw@langdepduss{<portuguese: Uses 57c>}%
 \def\nw@langdepdusd{<portuguese: used 57d>}%
 \def\nw@langdepdnvu{<portuguese: never used 57e>}%
 \def\nw@langdepddfs{<portuguese: Defines 57f>}%
 \def\nw@langdepdnvd{<portuguese: never defined 57g>}%
 \def\nw@langdepdpre{<portuguese: Underlined indices... 57h>}%
}
\DeclareOption{portuguese}{\ExecuteOptions{portuges}}
\DeclareOption{brazilian}{\ExecuteOptions{portuges}}
\DeclareOption{brazil}{\ExecuteOptions{portuges}}

```
- Defines:
- brazil, never used.
  - brazilian, never used.
  - portuges, never used.
  - portuguese, never used.
- Uses \nw@langdepdand 54a, \nw@langdepdchk 54a, \nw@langdepdchks 54a, \nw@langdepdpcud 54a, \nw@langdepdpcwf 54a, \nw@langdepddef 54a, \nw@langdepddfs 54a, \nw@langdepdin 54a, \nw@langdepdnvd 54a, \nw@langdepdnvu 54a, \nw@langdepdpre 54a, \nw@langdepdprtc 54a, \nw@langdepdusd 54a, and \nw@langdepduss 54a.
- 56b *<portuguese: This definition is continued 56b>≡* (56a)
- Defini\c{c}\~ao continuada em
- 56c *<portuguese: This code is used 56c>≡* (56a)
- C\'odigo usado em
- 56d *<portuguese: Root chunk (not used in this document) 56d>≡* (56a)
- Fragmento de topo (sem uso no documento)
- 56e *<portuguese: This code is written to file 56e>≡* (56a)
- Este c\'odigo foi escrito no ficheiro
- 56f *<portuguese: chunk 56f>≡* (56a)
- fragmento
- 56g *<portuguese: chunks 56g>≡* (56a)
- fragmentos



- 57a  $\langle\text{portuguese: in 57a}\rangle\equiv$  (56a)  
no(s)
- 57b  $\langle\text{portuguese: and 57b}\rangle\equiv$  (56a)  
e
- 57c  $\langle\text{portuguese: Uses 57c}\rangle\equiv$  (56a)  
Usa
- 57d  $\langle\text{portuguese: used 57d}\rangle\equiv$  (56a)  
usado
- 57e  $\langle\text{portuguese: never used 57e}\rangle\equiv$  (56a)  
nunca usado
- 57f  $\langle\text{portuguese: Defines 57f}\rangle\equiv$  (56a)  
Define
- 57g  $\langle\text{portuguese: never defined 57g}\rangle\equiv$  (56a)  
nunca definido
- 57h  $\langle\text{portuguese: Underlined indices... 57h}\rangle\equiv$  (56a)  
Os \'indices \underline{sublinhados} indicam defini\c{c}\~oes;  
outros \'indices indicam usos.

### 2.16.3 Support for Spanish

Tentative translation to Spanish by José Riel. Improvements welcome.

- 57i  $\langle\text{declaration of options 9a}\rangle+\equiv$  (6a)  $\triangleleft 56a \ 59a \triangleright$
- ```

\DeclareOption{spanish}{%
  \def\nw@langdepdef{\langle\text{spanish: This definition is continued 58a}\rangle}%
  \def\nw@langdepdpcud{\langle\text{spanish: This code is used 58b}\rangle}%
  \def\nw@langdepdprtc{\langle\text{spanish: Root chunk (not used in this document) 58c}\rangle}%
  \def\nw@langdepdpcwf{\langle\text{spanish: This code is written to file 58d}\rangle}%
  \def\nw@langdepdchk{\langle\text{spanish: chunk 58e}\rangle}%
  \def\nw@langdepdchks{\langle\text{spanish: chunks 58f}\rangle}%
  \def\nw@langdepdin{\langle\text{spanish: in 58g}\rangle}%
  \def\nw@langdepdand{\langle\text{spanish: and 58h}\rangle}%
  \def\nw@langdepduss{\langle\text{spanish: Uses 58i}\rangle}%
  \def\nw@langdepdusd{\langle\text{spanish: used 58j}\rangle}%
  \def\nw@langdepdnvu{\langle\text{spanish: never used 58k}\rangle}%
  \def\nw@langdepdpdfs{\langle\text{spanish: Defines 58l}\rangle}%
  \def\nw@langdepdnvd{\langle\text{spanish: never defined 58m}\rangle}%
  \def\nw@langdepdppe{\langle\text{spanish: Underlined indices... 58n}\rangle}%
}

```
- Defines:
- spanish, never used.
- Uses \nw@langdepdand 54a, \nw@langdepdchk 54a, \nw@langdepdchks 54a, \nw@langdepdpcud 54a,
 \nw@langdepdpcwf 54a, \nw@langdepddef 54a, \nw@langdepdpdfs 54a, \nw@langdepdin 54a, \nw@langdepdnvd
 54a, \nw@langdepdnvu 54a, \nw@langdepdppe 54a, \nw@langdepdprtc 54a, \nw@langdepdusd 54a,
 and \nw@langdepduss 54a.

- 58a $\langle \text{spanish: This definition is continued 58a} \rangle \equiv$ (57i)
Esta definici\`on se contin\`ua
- 58b $\langle \text{spanish: This code is used 58b} \rangle \equiv$ (57i)
Esta c\`odigo se utiliza
- 58c $\langle \text{spanish: Root chunk (not used in this document) 58c} \rangle \equiv$ (57i)
Fragmento de la tapa (no usado en este documento)
- 58d $\langle \text{spanish: This code is written to file 58d} \rangle \equiv$ (57i)
Este c\`odigo se escribe al ficheiro
- 58e $\langle \text{spanish: chunk 58e} \rangle \equiv$ (57i)
fragmento
- 58f $\langle \text{spanish: chunks 58f} \rangle \equiv$ (57i)
fragmentos
- 58g $\langle \text{spanish: in 58g} \rangle \equiv$ (57i)
em
- 58h $\langle \text{spanish: and 58h} \rangle \equiv$ (57i)
e
- 58i $\langle \text{spanish: Uses 58i} \rangle \equiv$ (57i)
Usa
- 58j $\langle \text{spanish: used 58j} \rangle \equiv$ (57i)
usos
- 58k $\langle \text{spanish: never used 58k} \rangle \equiv$ (57i)
nunca usado
- 58l $\langle \text{spanish: Defines 58l} \rangle \equiv$ (57i)
Define
- 58m $\langle \text{spanish: never defined 58m} \rangle \equiv$ (57i)
nunca definido
- 58n $\langle \text{spanish: Underlined indices... 58n} \rangle \equiv$ (57i)
Los \`indices \underline{subrayados} indican definiciones;
otros \`indices indican aplicaciones.

2.16.4 Support for French

This is a tentative translation to French. Although NR has made some corrections, it should probably be reviewed by a native speaker.

There are problems with using accents: on the `\nw@langdepnvd` macro (which apparently is not used in the context of `\nwcodecomment`), and in some other macros L^AT_EX complains about missing `\endcsname`. This should be fixed by someone with experience in using T_EX. JR: The problem occurs with the OT1 font encoding, it does not occur with T1.

59a *<declaration of options 9a>+≡* (6a) <57i 61a>

```
\DeclareOption{frenchb}{%
  \def\nw@langdepdef{<french: This definition is continued 59b>}%
  \def\nw@langdep cud{<french: This code is used 59c>}%
  \def\nw@langdeprtc{<french: Root chunk (not used in this document) 59d>}%
  \def\nw@langdep cwf{<french: This code is written to file 60a>}%
  \def\nw@langdepchk{<french: chunk 60b>}%
  \def\nw@langdepchks{<french: chunks 60c>}%
  \def\nw@langdepin{<french: in 60d>}%
  \def\nw@langdepand{<french: and 60e>}%
  \def\nw@langdepuss{<french: Uses 60f>}%
  \def\nw@langdepusd{<french: used 60g>}%
  \def\nw@langdepnvu{<french: never used 60h>}%
  \def\nw@langdepdfs{<french: Defines 60i>}%
  \def\nw@langdepnvd{<french: never defined 60j>}%
  \def\nw@langdep pre{<french: Underlined indices... 60k>}
}
\DeclareOption{french}{\ExecuteOptions{frenchb}}
\DeclareOption{français}{\ExecuteOptions{frenchb}}
\DeclareOption{acadian}{\ExecuteOptions{frenchb}}
\DeclareOption{canadien}{\ExecuteOptions{frenchb}}
```

Defines:

acadian, never used.
canadien, never used.
français, never used.
french, never used.
frenchb, never used.

Uses `\nw@langdepand 54a`, `\nw@langdepchk 54a`, `\nw@langdepchks 54a`, `\nw@langdep cud 54a`,
`\nw@langdep cwf 54a`, `\nw@langdepdef 54a`, `\nw@langdepdfs 54a`, `\nw@langdepin 54a`, `\nw@langdepnvd`
`54a`, `\nw@langdepnvu 54a`, `\nw@langdep pre 54a`, `\nw@langdeprtc 54a`, `\nw@langdepusd 54a`,
and `\nw@langdepuss 54a`.

59b *<french: This definition is continued 59b>≡* (59a)
Cette d'efinition suit

59c *<french: This code is used 59c>≡* (59a)
Ce code est employ'e

59d *<french: Root chunk (not used in this document) 59d>≡* (59a)
Morceau racine (pas employ'e dans ce document)

- 60a $\langle \text{french: This code is written to file 60a} \rangle \equiv$ (59a)
Ce code est \’ecrit aux fichier
- 60b $\langle \text{french: chunk 60b} \rangle \equiv$ (59a)
le morceau
- 60c $\langle \text{french: chunks 60c} \rangle \equiv$ (59a)
les morceaux
- 60d $\langle \text{french: in 60d} \rangle \equiv$ (59a)
dans
- 60e $\langle \text{french: and 60e} \rangle \equiv$ (59a)
et
- 60f $\langle \text{french: Uses 60f} \rangle \equiv$ (59a)
Il emploie
- 60g $\langle \text{french: used 60g} \rangle \equiv$ (59a)
employ\’{e}
- 60h $\langle \text{french: never used 60h} \rangle \equiv$ (59a)
jamais employ\’{e}
- 60i $\langle \text{french: Defines 60i} \rangle \equiv$ (59a)
Il d\’{e}fine
- 60j $\langle \text{french: never defined 60j} \rangle \equiv$ (59a)
jamais defini
- 60k $\langle \text{french: Underlined indices... 60k} \rangle \equiv$ (59a)
Les index soulignes indiquent des definitions ;
d’autres index indiquent des utilisations.

2.16.5 Support for German

This is a translation to German by Sabine Broda (DCC-FCUP & LIACC, Universidade do Porto).

- 61a *<declaration of options 9a>+≡* (6a) <59a
- ```

\DeclareOption{german}{%
 \def\nw@langdepdef{<german: This definition is continued 61b>}%
 \def\nw@langdepdcud{<german: This code is used 61c>}%
 \def\nw@langdepdrtc{<german: Root chunk (not used in this document) 61d>}%
 \def\nw@langdepdcwf{<german: This code is written to file 61e>}%
 \def\nw@langdepchk{<german: chunk 61f>}%
 \def\nw@langdepchks{<german: chunks 61g>}%
 \def\nw@langdepin{<german: in 61h>}%
 \def\nw@langdepand{<german: and 62a>}%
 \def\nw@langdepuss{<german: Uses 62b>}%
 \def\nw@langdepusd{<german: used 62c>}%
 \def\nw@langdepnvu{<german: never used 62d>}%
 \def\nw@langdepdfs{<german: Defines 62e>}%
 \def\nw@langdepnvd{<german: never defined 62f>}%
 \def\nw@langdeppre{<german: Underlined indices... 62g>}
}
\DeclareOption{ngerman}{\ExecuteOptions{german}}
\DeclareOption{germanb}{\ExecuteOptions{german}}

```
- Defines:
- german, never used.
  - germanb, never used.
  - ngerman, never used.
- Uses \nw@langdepand 54a, \nw@langdepchk 54a, \nw@langdepchks 54a, \nw@langdepdcud 54a, \nw@langdepdcwf 54a, \nw@langdepdef 54a, \nw@langdepdfs 54a, \nw@langdepin 54a, \nw@langdepnvd 54a, \nw@langdepnvu 54a, \nw@langdeppre 54a, \nw@langdepdrtc 54a, \nw@langdepusd 54a, and \nw@langdepuss 54a.
- 61b *<german: This definition is continued 61b>≡* (61a)  
Diese Definition wird fortgesetzt
- 61c *<german: This code is used 61c>≡* (61a)  
Dieser Code wird benutzt
- 61d *<german: Root chunk (not used in this document) 61d>≡* (61a)  
Hauptteil (nicht in diesem Dokument benutzt)
- 61e *<german: This code is written to file 61e>≡* (61a)  
Dieser Code schreibt man zum File
- 61f *<german: chunk 61f>≡* (61a)  
Teil
- 61g *<german: chunks 61g>≡* (61a)  
Teils
- 61h *<german: in 61h>≡* (61a)  
im

- 62a  $\langle \text{german: and 62a} \rangle \equiv$  (61a)  
und
- 62b  $\langle \text{german: Uses 62b} \rangle \equiv$  (61a)  
Benutzt
- 62c  $\langle \text{german: used 62c} \rangle \equiv$  (61a)  
benutzt
- 62d  $\langle \text{german: never used 62d} \rangle \equiv$  (61a)  
nicht benutzt
- 62e  $\langle \text{german: Defines 62e} \rangle \equiv$  (61a)  
Definiert
- 62f  $\langle \text{german: never defined 62f} \rangle \equiv$  (61a)  
nicht definiert
- 62g  $\langle \text{german: Underlined indices... 62g} \rangle \equiv$  (61a)  
Unterstrichene Indizes zeigen Definitionen an;  
andere Indizes zeigen Gebrauch an.
- 62h  $\langle \text{knoweb.sty 6a} \rangle + \equiv$  <52d  
\endinput

## Chunks

Underlined indices denote definitions; regular indices denote uses.

$\langle \backslash \text{count@} \leftarrow \backslash \text{nw@hipage} - 1 \text{ 24d} \rangle$  24d  
 $\langle \backslash \text{nw@hipage} \leftarrow \backslash \text{nw@lopage} + 1 \text{ 24c} \rangle$  23c, 24c  
 $\langle B^k \leftarrow \lfloor B^k/26 \rfloor; d \leftarrow \lfloor n/B^k \rfloor \text{ 35g} \rangle$  35a, 35g  
 $\langle n \leftarrow \langle n \rangle - 1; B^k \leftarrow 26 \text{ 35c} \rangle$  35a, 35c  
 $\langle n \leftarrow n - B^k; B^k \leftarrow 26 B^k \text{ 35e} \rangle$  35a, 35e  
 $\langle n \leftarrow n - d B^k \text{ 36b} \rangle$  35a, 36b  
 $\langle \backslash \text{count@} \leftarrow \backslash \text{nw@hipage} - \backslash \text{nw@lopage} \text{ 25b} \rangle$  25a, 25b  
 $\langle \backslash \text{nwbegincode separation and penalties 13b} \rangle$  11b, 13b  
 $\langle \backslash \text{obeylines setup 13d} \rangle$  12a, 13d  
 $\langle \backslash \text{trivlist cliché (à la verbatim) 13c} \rangle$  12a, 13c  
 $\langle \text{add } \backslash \text{nwcodecommentsep if this is the first } \backslash \text{nwcodecomment 18b} \rangle$  18a, 18b  
 $\langle \text{add } \langle \text{key} \rangle \text{ to } \backslash \text{pending@sublabels list 32a} \rangle$  31a, 32a  
 $\langle \text{append range to range list 24e} \rangle$  22a, 23a, 24b, 24e, 27d  
 $\langle \text{assign } \backslash \text{nwixaddsx 50a} \rangle$  46c, 48c, 50a  
 $\langle \text{assign temporary count registers 35b} \rangle$  35a, 35b  
 $\langle \text{call } \backslash \text{newlabel for each } \langle \text{key} \rangle \text{ in } \backslash \text{pending@sublabels 32b} \rangle$  31a, 32b  
 $\langle \text{chicago: } \ell \leftarrow \backslash \text{nw@lopage}, h \leftarrow \backslash \text{nw@hipage}, 10^k \leftarrow 1 \text{ 26a} \rangle$  25c, 26a  
 $\langle \text{chicago: } \backslash \text{nw@hipage} \leftarrow h - \lfloor h/10^k \rfloor 10^k \text{ 26c} \rangle$  25c, 26c  
 $\langle \text{chicago: find } 10^k \text{ such that } \lfloor \ell/10^k \rfloor = \lfloor h/10^k \rfloor \text{ 26b} \rangle$  25c, 26b  
 $\langle \text{clear } \backslash \text{pending@sublabels 32c} \rangle$  31a, 32c  
 $\langle \text{declaration of options 9a} \rangle$  6a, 9a, 9c, 9d, 11a, 12b, 15b, 20b, 20c, 21a, 34a, 36c, 37b, 40a, 40b,  
44c, 45c, 45d, 47d, 51a, 54a, 56a, 57i, 59a, 61a  
 $\langle \text{define } \backslash \text{@tempa to be } \{ \backslash \text{nw@lopage} - \backslash \text{nw@hipage} \} \{ \} \text{ 26e} \rangle$  25c, 26e  
 $\langle \text{english: and 55g} \rangle$  54a, 55g  
 $\langle \text{english: chunk 55d} \rangle$  54a, 55d  
 $\langle \text{english: chunks 55e} \rangle$  54a, 55e  
 $\langle \text{english: Defines 55k} \rangle$  54a, 55k  
 $\langle \text{english: in 55f} \rangle$  54a, 55f  
 $\langle \text{english: never defined 55i} \rangle$  54a, 55i  
 $\langle \text{english: never used 55j} \rangle$  54a, 55j  
 $\langle \text{english: Root chunk (not used in this document) 55b} \rangle$  54a, 55b  
 $\langle \text{english: This code is used 55a} \rangle$  54a, 55a  
 $\langle \text{english: This code is written to file 55c} \rangle$  54a, 55c  
 $\langle \text{english: This definition is continued 54b} \rangle$  54a, 54b  
 $\langle \text{english: Underlined indices... 55m} \rangle$  54a, 55m  
 $\langle \text{english: used 55i} \rangle$  54a, 55i  
 $\langle \text{english: Uses 55h} \rangle$  54a, 55h  
 $\langle \text{execution of options 9e} \rangle$  6a, 9e, 12c, 21b, 34b, 36d, 41a, 44d, 53  
 $\langle \text{french: and 60e} \rangle$  59a, 60e  
 $\langle \text{french: chunk 60b} \rangle$  59a, 60b  
 $\langle \text{french: chunks 60c} \rangle$  59a, 60c

<french: Defines 60i> 59a, 60i  
 <french: in 60d> 59a, 60d  
 <french: never defined 60j> 59a, 60j  
 <french: never used 60h> 59a, 60h  
 <french: Root chunk (not used in this document) 59d> 59a, 59d  
 <french: This code is used 59c> 59a, 59c  
 <french: This code is written to file 60a> 59a, 60a  
 <french: This definition is continued 59b> 59a, 59b  
 <french: Underlined indices... 60k> 59a, 60k  
 <french: used 60g> 59a, 60g  
 <french: Uses 60f> 59a, 60f  
 <german: and 62a> 61a, 62a  
 <german: chunk 61f> 61a, 61f  
 <german: chunks 61g> 61a, 61g  
 <german: Defines 62e> 61a, 62e  
 <german: in 61h> 61a, 61h  
 <german: never defined 62f> 61a, 62f  
 <german: never used 62d> 61a, 62d  
 <german: Root chunk (not used in this document) 61d> 61a, 61d  
 <german: This code is used 61c> 61a, 61c  
 <german: This code is written to file 61e> 61a, 61e  
 <german: This definition is continued 61b> 61a, 61b  
 <german: Underlined indices... 62g> 61a, 62g  
 <german: used 62c> 61a, 62c  
 <german: Uses 62b> 61a, 62b  
 <identification 6b> 6a, 6b  
 <if two sublabels on page, assign \2on\nw@thispage 31d> 31a, 31d  
 <initial code 6c> 6a, 6c, 9b, 11b, 14d, 19a, 19b, 21c, 28a, 30b, 34c, 39c, 45b, 46b, 46c, 47c, 48b,  
48c, 50b, 50d  
 <initialize \nw@pages and \nw@pagecount 22b> 22a, 22b, 23a, 27d  
 <insert “chunk(s)” 22c> 22a, 22c, 23a  
 <knoweb.sty 6a> 6a, 6d, 6e, 7a, 7b, 7c, 7d, 8a, 8b, 8c, 10a, 10b, 10c, 10d, 12a, 13a, 13e, 14c, 15a,  
15c, 15d, 16a, 16b, 16c, 17, 18a, 20a, 21d, 22a, 23a, 23b, 23c, 24a, 26d, 26f, 27a, 27b, 27c, 27d,  
28g, 29a, 29b, 29c, 29d, 30a, 30c, 31a, 32d, 33a, 33b, 33c, 33d, 35a, 37a, 38a, 39a, 39b, 41b,  
42a, 42b, 42c, 42d, 43a, 43b, 43c, 44a, 44b, 46a, 47a, 47b, 48a, 49a, 49b, 49d, 50c, 51b, 52a,  
52b, 52c, 52d, 62h  
 <make all those damn active characters “other” 14b> 14a, 14b  
 <new range starting with #2 24b> 24a, 24b  
 <portuguese: and 57b> 56a, 57b  
 <portuguese: chunk 56f> 56a, 56f  
 <portuguese: chunks 56g> 56a, 56g  
 <portuguese: Defines 57f> 56a, 57f  
 <portuguese: in 57a> 56a, 57a  
 <portuguese: never defined 57g> 56a, 57g  
 <portuguese: never used 57e> 56a, 57e



<portuguese: Root chunk (not used in this document) 56d> 56a, 56d  
 <portuguese: This code is used 56c> 56a, 56c  
 <portuguese: This code is written to file 56e> 56a, 56e  
 <portuguese: This definition is continued 56b> 56a, 56b  
 <portuguese: Underlined indices... 57h> 56a, 57h  
 <portuguese: used 57d> 56a, 57d  
 <portuguese: Uses 57c> 56a, 57c  
 <remove ligatures, make actives \other 38b> 38a, 38b, 43b  
 <save <page no.> into \nw@thispage 31b> 31a, 31b  
 <set \@tempa to page range(s), marked with \ 25a> 24e, 25a  
 <set index environment paragraph spacing 49c> 47a, 49a, 49c  
 <spanish: and 58h> 57i, 58h  
 <spanish: chunk 58e> 57i, 58e  
 <spanish: chunks 58f> 57i, 58f  
 <spanish: Defines 58l> 57i, 58l  
 <spanish: in 58g> 57i, 58g  
 <spanish: never defined 58m> 57i, 58m  
 <spanish: never used 58k> 57i, 58k  
 <spanish: Root chunk (not used in this document) 58c> 57i, 58c  
 <spanish: This code is used 58b> 57i, 58b  
 <spanish: This code is written to file 58d> 57i, 58d  
 <spanish: This definition is continued 58a> 57i, 58a  
 <spanish: Underlined indices... 58n> 57i, 58n  
 <spanish: used 58j> 57i, 58j  
 <spanish: Uses 58i> 57i, 58i  
 <undocumented – man page: \noweboptions 45a> 45a  
 <until  $B^k > n$  35d> 35a, 35d  
 <update \nw@subpage and \nw@lastpage 31c> 31a, 31c  
 <use simple rules from Chicago style manual 25c> 25a, 25c  
 <warn of undefined reference to #1 28c> 28b, 28c  
 <warn of undefined reference to #1 and add page ?? 28b> 27a, 27b, 28b  
 <while  $B^k > 1$  35f> 35a, 35f  
 <write character corresponding to d 36a> 35a, 36a  
 <zap ligatures, fix spaces 14a> 12a, 14a

## Index

Underlined indices denote definitions; regular indices denote uses.

\@beginparpenalty: [13b](#)  
 \@endparpenalty: [13b](#)  
 \@firstnwcodecommentfalse: [18b](#), [21d](#)  
 \@firstnwcodecommenttrue: [21d](#)  
 \@nwidentuses: [43b](#)  
 \@nwindexdefn: [38a](#)  
 \@nwindexuse: [38a](#)  
 \Bf: [10c](#)  
 \code: [15d](#)  
 \eatline: [7a](#)  
 \edoc: [15d](#)  
 \if@firstnwcodecomment: [18b](#), [21d](#)  
 \ifhtml: [52b](#)  
 \ifnw@chunks preamble: [46a](#), [47a](#)  
 \ifnw@hideunuseddefs: [44a](#), [44b](#), [45b](#)  
 \ifnw@index preamble: [48a](#), [49a](#)  
 \ifnw@longchunks: [47a](#), [47b](#), [47c](#)  
 \It: [8a](#), [10c](#)  
 \LA: [8a](#), [8c](#), [47b](#)  
 \newslabel: [30b](#), [31a](#), [37a](#)  
 \nextchunklabel: [33a](#)  
 \nosublabel: [30c](#)  
 \nowebchunks: [46a](#), [46b](#)  
 \nowebchunks@external: [46c](#), [51a](#)  
 \nowebindex: [48a](#), [48b](#)  
 \nowebindex@external: [48c](#), [51a](#)  
 \nw@alsodefined: [19a](#), [20c](#)  
 \nw@anchorto: [52a](#)  
 \nw@anchortofin: [52a](#)  
 \nw@auxix: [38a](#), [39a](#)  
 \nw@chunkcommachunketc: [19a](#), [22a](#), [44b](#)  
 \nw@chunknamebox: [8a](#), [9a](#), [9b](#), [12a](#)  
 \nw@chunks: [46a](#), [46b](#), [51a](#)  
 \nw@chunkspacechunketc: [19b](#), [27d](#)  
 \nw@commafy: [22a](#), [23a](#), [41b](#), [43b](#)  
 \nw@commentpar: [16a](#), [16c](#)  
 \nw@enddeflinemarkup: [19b](#), [20b](#)  
 \nw@equivbox: [8b](#), [8c](#)  
 \nw@firstpage: [23c](#), [24b](#), [27a](#)  
 \nw@firstpagel: [22a](#), [27a](#), [27d](#)  
 \nw@g@nericref: [29d](#), [30a](#)  
 \nw@genericref: [28g](#), [29b](#), [29d](#)  
 \nw@hi: [26a](#), [26b](#), [26c](#), [26d](#)  
 \nw@hideunuseddefsfalse: [45b](#)  
 \nw@hideunuseddefstrue: [45b](#), [45c](#)  
 \nw@hipage: [23b](#), [24a](#), [24c](#), [24d](#), [25a](#), [25b](#), [26a](#), [26c](#), [26e](#)  
 \nw@hisub: [23b](#), [24a](#), [25a](#)  
 \nw@hixreftag: [24a](#), [25a](#)  
 \nw@hyperident: [39c](#), [40a](#), [40b](#)  
 \nw@hyperpagenum: [27c](#)  
 \nw@index: [48a](#), [48b](#), [51a](#)  
 \nw@indexline: [49a](#), [49b](#)  
 \nw@int@to@str: [34a](#), [36c](#)  
 \nw@ixappend: [24e](#), [28b](#), [39b](#), [42a](#), [50d](#)  
 \nw@ixcounter: [41b](#), [42c](#), [43c](#), [44a](#), [44b](#)  
 \nw@ixdefcount: [42d](#)  
 \nw@ixdefs: [42b](#), [42d](#), [43a](#)  
 \nw@ixdefs@space: [43a](#), [44c](#)  
 \nw@ixid@defs: [43a](#), [43b](#)  
 \nw@ixid@uses: [44b](#)  
 \nw@ixlistcount: [41b](#), [42c](#), [42d](#), [44a](#)  
 \nw@ixtotaluses: [43c](#), [44a](#)  
 \nw@ixusecount: [42d](#), [43c](#), [44b](#)  
 \nw@ixuses: [42b](#), [42d](#), [44b](#)  
 \nw@labeltrailers: [32b](#), [32d](#)  
 \nw@langdepand: [41b](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdepchk: [22c](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdepchks: [22c](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep cud: [19a](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep cwf: [19a](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep def: [19a](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep dfs: [44a](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep in: [19a](#), [44b](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep nvd: [46c](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep nvu: [44b](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep pre: [47a](#), [49a](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep rtc: [19a](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
 \nw@langdep usd: [44b](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)

\ nw@langdepu ss: [43b](#), [54a](#), [56a](#), [57i](#), [59a](#), [61a](#)  
\ nw@lastpage: [31c](#), [33d](#)  
\ nw@laterfilename: [17](#)  
\ nw@lo: [26a](#), [26b](#), [26d](#)  
\ nw@longalph: [35a](#), [36c](#)  
\ nw@lop age: [23b](#), [23c](#), [24a](#), [24c](#), [25a](#), [25b](#), [26a](#),  
[26e](#)  
\ nw@lo sub: [23b](#), [23c](#), [24a](#), [25a](#)  
\ nw@loxreftag: [23c](#), [25a](#)  
\ nw@makeother: [14b](#), [14c](#), [38b](#)  
\ nw@nextpage: [24a](#), [27b](#)  
\ nw@nextpagel: [22a](#), [27b](#), [27d](#)  
\ nw@nosubscriptident: [39c](#), [40a](#), [40b](#)  
\ nw@notused: [19a](#), [20c](#)  
\ nw@onechunk: [47a](#), [47b](#)  
\ nw@pagecount: [22b](#), [22c](#), [23c](#), [24a](#), [27a](#)  
\ nw@pagenumslrest: [27d](#)  
\ nw@pages: [22a](#), [22b](#), [23a](#), [24e](#), [27d](#), [28a](#), [28b](#)  
\ nw@pagetemp: [26f](#)  
\ nw@plusequivbox: [8b](#), [8c](#)  
\ nw@prevnextdefs: [19b](#), [20b](#)  
\ nw@pwrten: [26a](#), [26b](#), [26c](#), [26d](#)  
\ nw@semifilbreak: [15a](#), [15b](#)  
\ nw@startdeflinemarkup: [19b](#), [20b](#)  
\ nw@stripstar: [19a](#)  
\ nw@subpage: [31c](#), [31d](#), [32b](#), [33d](#)  
\ nw@subpagepair: [29b](#), [29c](#)  
\ nw@subpageref: [28g](#), [29a](#)  
\ nw@subscriptident: [39c](#), [40a](#), [40b](#)  
\ nw@thispage: [31b](#), [31c](#), [31d](#), [32b](#)  
\ nw@underlinedefs: [47b](#), [49b](#), [49d](#)  
\ nw@used: [19a](#), [20c](#)  
\ nw@usesondefline: [19b](#), [20b](#)  
\ nw@weblabel: [37a](#), [37b](#)  
\ nwalsodefined: [20b](#), [20c](#), [21a](#), [21c](#)  
\ nwanchorto: [52a](#)  
\ nwbegincode: [11b](#)  
\ nwbegindocs: [14d](#)  
\ nwblindhyperanchor: [30b](#), [33a](#), [37a](#), [51b](#)  
\ nwbreakcodespace: [15a](#)  
\ nwcodecomment: [18a](#), [19a](#), [43b](#), [44a](#), [44b](#)  
\ nwcodecommentsep: [18b](#), [21d](#)  
\ nwcodeindent: [8c](#), [9c](#), [10d](#), [13c](#), [18a](#), [44b](#)  
\ nwcodepenalty: [13d](#), [13e](#), [18a](#), [18b](#)  
\ nwcodetopsep: [13a](#), [13b](#)  
\ nwdocspar: [14d](#), [15b](#)  
\ nwendcode: [11b](#), [15b](#)  
\ nwenddeflinemarkup: [20b](#), [20c](#), [21a](#), [21c](#)  
\ nwenddocs: [14d](#)  
\ nwendquote: [6e](#)  
\ nwfilename: [17](#)  
\ nwhyperreference: [27c](#), [28g](#), [39c](#), [51b](#)  
\ nwidentdefs: [44a](#), [45d](#)  
\ nwidentuses: [43b](#), [45d](#)  
\ nwindexdefn: [38a](#)  
\ nwindexuse: [38a](#)  
\ nwixadd: [39a](#), [39b](#)  
\ nwixadds: [50c](#), [50d](#), [51a](#)  
\ nwixaddsx: [50a](#), [50b](#), [51a](#)  
\ nwixd: [38a](#), [39a](#), [42b](#), [49d](#)  
\ nwixident: [52c](#)  
\ nwixlogsorted: [50c](#)  
\ nwixu: [38a](#), [39a](#), [42b](#), [49d](#)  
\ nwlinkedidentc: [40a](#)  
\ nwlinkedidentq: [40a](#), [40b](#)  
\ nwmargintag: [9c](#), [9d](#)  
\ nwnewline: [7b](#)  
\ nwnextdefptr: [19b](#), [20a](#)  
\ nwnotused: [20b](#), [20c](#), [21a](#), [21c](#)  
\ nwoutput: [19a](#)  
\ nwprevdefptr: [19b](#), [20a](#)  
\ nwprevnextdefs: [20b](#), [20c](#), [21a](#), [21c](#)  
\ nwstartdeflinemarkup: [20b](#), [20c](#), [21a](#), [21c](#)  
\ nwtagstyle: [10b](#), [19b](#), [47b](#)  
\ nwthepagenum: [25a](#), [29a](#), [34a](#), [34c](#)  
\ nwtypesetcommentfont: [16a](#), [16b](#)  
\ nwused: [20b](#), [20c](#), [21a](#), [21c](#)  
\ nwusesondefline: [20b](#), [20c](#), [21a](#), [21c](#)  
\ pending@sublabels: [32b](#), [32c](#), [33b](#), [33c](#)  
\ pendingsublabel: [32a](#), [33a](#), [33b](#)  
\ plusendmoddef: [8c](#)  
\ ps@noweb: [17](#)  
\ r@nw@notdef: [46c](#)  
\ RA: [8a](#), [8c](#), [47b](#)  
\ Rm: [7c](#), [10b](#), [10c](#), [17](#), [18a](#)  
\ setupcode: [6d](#), [12a](#), [14a](#), [15d](#)  
\ sublabel: [30b](#), [37b](#)  
\ subpagepair: [27a](#), [27b](#), [29b](#)  
\ subpageref: [20a](#), [28g](#), [39c](#), [43a](#), [47b](#), [49d](#)  
\ subpages: [23a](#)

\Tt: [6d](#), [10c](#), [17](#), [19a](#), [43a](#), [44b](#), [49b](#)  
 acadian: [59a](#)  
 alphasubpage: [34a](#), [34b](#)  
 american: [54a](#)  
 brazil: [56a](#)  
 brazilian: [56a](#)  
 breakcode: [15b](#)  
 breakdefs: [44c](#), [44d](#), [45a](#)  
 british: [54a](#)  
 canadian: [54a](#)  
 canadien: [59a](#)  
 donotbreakchunknames: [9a](#)  
 english: [53](#), [54a](#)  
 externalindex: [51a](#)  
 footnotesizecode: [12b](#)  
 francais: [59a](#)  
 french: [59a](#)  
 frenchb: [59a](#)  
 german: [61a](#)  
 germanb: [61a](#)  
 hideunuseddefs: [45c](#)  
 Hugecode: [12b](#)  
 hugecode: [12b](#)  
 hyperidents: [40a](#), [41a](#)  
 hyperquotedidents: [40b](#)  
 LARGEcode: [12b](#)  
 Largecode: [12b](#)  
 largecode: [12b](#)  
 longchunks: [47d](#)  
 longstrings: [36c](#)  
 longxref: [20c](#)  
 margintag: [9c](#), [9e](#)  
 moddef: [8c](#)  
 ngerman: [61a](#)  
 nobreakdefs: [44c](#), [45a](#)  
 nohyperidents: [40a](#)  
 nohyperquotedidents: [40b](#)  
 noidentxref: [45d](#)  
 nomargintag: [9c](#), [34a](#)  
 normalsizecode: [12b](#), [12c](#)  
 noshift: [11a](#)  
 nosubpage: [34a](#)  
 nosubscriptidents: [40a](#)  
 nosubscriptquotedidents: [40b](#)  
 noxref: [21a](#)

numsubpage: [34a](#)  
 nwtypesetcomment: [16a](#)  
 portuges: [56a](#)  
 portuguese: [56a](#)  
 scriptsizecode: [12b](#)  
 shift: [11a](#)  
 shortstrings: [36c](#), [36d](#)  
 shortxref: [20b](#), [21b](#)  
 smallcode: [12b](#)  
 spanish: [57i](#)  
 subscriptidents: [40a](#)  
 subscriptquotedidents: [40b](#)  
 thenowebchunks: [46b](#), [46c](#), [47a](#)  
 thenowebindex: [48b](#), [48c](#), [49a](#)  
 tinycode: [12b](#)  
 twocolumn: [9d](#)  
 UKenglish: [54a](#)  
 USenglish: [54a](#)  
 webcode: [11b](#)  
 webnumbering: [37b](#)

## References

- [1] Anonymous. *The Chicago Manual of Style*. University of Chicago Press, Chicago, IL, USA, 14th edition, 1993. [2.9](#)
- [2] Donald E. Knuth. *The T<sub>E</sub>Xbook*. Addison-Wesley, Reading, MA, USA, 1984. [2](#)
- [3] Norman Ramsey. T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X support for noweb. Distributed with noweb package. [1.1](#)