Typesetting Poetry Collections with poemscol*

John Burt
burt@brandeis.edu

July 14, 2007

Abstract

poemscol provides macros for LaTeX for setting collections of poetry. It is especially suited for setting collections of poetry in which several volumes are combined, such as in a critical edition of a poet's Collected Poems. It provides the structures required to produce a critical edition of the kind specified by the Modern Language Association's Committee on Scholarly Editions, and it automatically marks every occasion where a stanza break falls on a page break.

Contents

*This document corresponds to poemscol 2.53, dated 14 July 2007.

1 Introduction

poemscol provides the structures necessary for editing a critical edition of a volume of poems or of a collection of a poet's works. poemscol numbers the lines, and produces separate, formatted endnote sections for emendations, textual collations, and explanatory notes (or, optionally, multi-layered footnotes), tying each note the number of the line upon which it is a comment. It also automatically generates a table of contents, an index of titles and first lines, and divider pages for the sections of the volumes. It produces running headers of the form "Emendations to pp. xx-yy" for the note sections. And it marks occasions when a stanza break falls at a page break.

Producing line numbers for verse is something for which LaTeX would seem to be especially suited. A line of verse, after all, is not a carriage return but a logical unit, an element of versification more than of typesetting, perhaps running over several physical lines, or perhaps split into half-lines as speakers or subjects change. With poemscol you mark lines, stanzas, and entire poems up as logical units, and LaTeX does the formatting and counting. Once you have marked out the logical units of the poem, poemscol will automatically mark every textual note, emendation, or explanatory note with the range of line numbers to which it applies.

In poetry which does not have a regular stanzaic form, it is useful to be able to mark automatically occasions where there is a stanza break at the bottom of a page which the reader might not notice. Doing this by hand is not only tedious and easy to get wrong, but also a process you will have to start over again if anything about your volume changes, if you add a poem, say, or even decide to break a title across two lines. poemscol takes care of this process, so that the editor need never worry about it, automatically marking cases where the page break coincides with a stanza break with a symbol.

Editions of Collected Poetry might also require special structures to reflect the fact that they are made up of the contents of several volumes of poetry. In particular, such editions require special structures for setting up specially formatted divider pages between volumes. They also require tables of contents and other front matter. poemscol provides these structures.

The best features of poemscol are of course simply that it is TeX: it uses TeX's automatic kerning and setting of ligatures, its algorithm for justifying lines (in prose sections), and LaTeX's way of setting verse.

Using LaTeX to typeset critical editions offers more advantages than simply the ability to automate tedious and easy to fumble tasks. Most modern TeX compilers have the ability to produce output in Adobe pdf format. Adobe pdf output can be used as camera ready copy, saving your publisher time and expense, and perhaps making a marginally economic critical edition a bit easier to bring to press. Furthermore, since typesetting the edition yourself in LaTeX obviates the publisher's own typesetting of your text, it removes another possible source of new errors. (You should expect to work with your publisher on the final design of the book, but LaTeX is a flexible language which will enable you to reproduce most book designs.)

Editions made in the formats of proprietary software such as Word© or Quark© will become not only obsolete but unreadable if those programs pass from use. Since your LaTeX sources for your edition are in ASCII, they provide a permanent record of your local intentions at every point in your edition, whether or not those who wish to consult your files have access to a LaTeX compiler, or indeed whether or not they can read LaTeX code.

Finally, although LaTeX is a typesetting language, not a content markup language, the ability to create new commands which the language offers comes very close to enabling one to realize the ideal of completely separating content markup from formatting. One advantage of this kind of markup is that even if the appearance of the poem on the page may be ambiguous, the editor's intentions about the logical structure of the poem will be preserved in the LaTeX source. Should you wish later to produce an electronic edition of your work, either using XML or the SGML markup approved by the Text Encoding Initiative, transforming your texts from LaTeX to XML would largely (although not entirely) be a matter of performing a series of global search-and-replaces, and could conceivably be done with a perl script.

poemscol is also suited for verse drama, and the package dramatist has been modified to work with poemscol. poemscol can provide line numbers and notes for prose sections such as the author's introduction or prose poetry, and it can also handle inset prose passages in the midst of verse. poemscol is not, however, designed for typesetting large scale prose works. For critical editions of prose works, several other packages are available, including the EDMAC format, a TeX format analogous to but distinct from LaTeX, by John Lavagnino and Dominik Wujastyk, or ledmac, a port of EDMAC into LaTeX by Peter Wilson, or ednotes, a completely independent set of macros for critical editions which builds on manypar and lineno, by Uwe Lück.

2 Dependencies and compatibility with other packages

poemscol depends upon several other packages, which you should be sure you have in your preamble and search path: fancyhdr for managing the running headers, makeidx and multicol for managing the index, geometry, and ifthen and keyval to simplify page geometry. If you are using the `\JHpoemtitle` alternate form of the `\poemtitle` command, you need to add mparhack to your package list, to make sure that the marginal references `\JHpoemtitle` adds appear on the correct side of the page. You should make sure that poemscol is the very last package you load in your preamble, or at least is loaded after those packages upon which it depends.

If you are planning to generate multiple indices, substitute splitindex for makeidx in your list of packages.

2.1 Prose sections

If you plan to number lines in prose contexts, and to make textual notes, emendations, or explanatory notes in prose contexts, you will need to add the lineno package to your preamble. I use the "right" and "modulo" options with this package, but you can set the options however you wish to make how you number the lines of prose sections consistent with how you number the lines of verse sections.

2.2 Multi-layer footnotes

If you plan to use paragraph-formatted footnotes rather than endnotes, you should load manyfoot. You should load it with the "ruled," and "para" options.

(poemscol's footnote commands are just a wrapper around manyfoot.) manyfoot inherited some of the limitations of LaTeX in dealing with long inserts at the end of the page, so you may have to do some fiddling. As fixes to these problems emerge, I will incorporate them into poemscol. manyfoot does not allow multiple paragraph footnotes when typesetting in "para" mode. This is probably a feature, not a bug, since one would want such notes to be set in several paragraphs, rather than running them together into one, as manyfoot would naturally want to do in para mode. For that reason, I have defined the two traditional classes of note in which multiple paragraph notes are likely to appear, "sources," and "explanatory notes," to be typeset in "plain" mode, which opens a new paragraph with each note. If your textual notes or emendations sections also have multiple paragraph notes, you should re-define them in order to produce footnotes in plain rather than in para mode, if you wish to produce footnotes rather than endnotes.

2.3 Verse drama and dramatist

Massimiliano Dominici has made his dramatist package compatible with poemscol, for which I am very grateful. Versions 1.2a or later of dramatist are compatible with versions 2.3 or later of poemscol.

2.4 Compatibility with earlier versions of poemscol

Making poemscol compatible with lineno required me also to change the commands for turning verse line numbering on and off, which means that version 2.3 of poemscol is incompatible with prior versions. The old command `\makelinenumbers` has been replaced with `\makeverselinenumbers` to distinguish it from a command from the lineno world. (`\makelinenumbers` will still work, but I have deprecated it.) To turn on verse line numbering, issue `\makeverselinenumbers`. Because the counter "linenumber" has been changed to "verselinenumber," you must use `\global\verselinenumbersfalse` to turn off line numbering, rather than `\global\linenumbersfalse` as before.

Up to version 2.44 poemscol borrowed the code for `\sidepar` non-floating marginal paragraphs from the memoir class, and also made `\leftsidepar` and `\rightsidepar` commands on their model. Version 2.46 modified Peter Wilson's original code for `\sidepar`, so therefore I have changed the names of `\sidepar`, `\leftsidepar`, and `\rightsidepar` to `\pmclsidepar`, `\pmclleftsidepar` and `\pmclrightsidepar` respectively. If you used any of these commands in your files, you will need to change their names.

Up to version 2.5 the `\definenewnotetype` command took four arguments, using the third argument for both the running header for the notes section and for the entry in the table of contents. In order to make it easier to handle titles in languages other than English I have found it necessary to separate these two uses, so `\definenewnotetype` now takes five arguments. You will need to retrofit files that use the old version of this command. You can see the details at section 8, page 20.

2.5 Incompatibility with memoir and verse

Unfortunately, poemscol is not compatible with the memoir class and the verse package, because they share some command names. I am preparing a workaround for this problem.

2.6 Crop marks

poemscol no longer provides cropmarks, but is compatible with the style packages that do. I use crop, with the letter,pdftex,cam, and center options.

3 Marking up individual poems

First, a word about the command names. The command names may seem ugly and long. And there are separate commands for many tasks that seem closely related, such as a command to mark the title of a section of a poetic sequence, and a separate command to mark a subsection. But the names do describe pretty much what each macro does, and they do specify exactly what the object they mark is supposed to be. The markup is designed to look like content markup, marking objects as a poem title, as a stanza, as a line, and so on. poemscol gives all of these content terms typographical meaning.

There are also many commands for special purposes whose necessity may not seem clear until the editor finds him or herself in the jam the command was designed for. For instance, poemscol normally encourages a page break before the title of a poem, or before the title of a section of a poetic sequence, to discourage page breaks between the title and the poem. But for the first section of a poetic sequence, or the first section of a poem in sections, one does not want to encourage LaTeX to break the page before the title, since that would leave the title of the sequence or the title of the larger poem as a widow on the previous page. `\sequencefirstsectiontitle` and its siblings are designed for this situation. Although all commands like `\sequencefirstsectiontitle` discourage page breaks before the title (and all of the commands discourage page breaks after the title), they will still sometimes happen, particularly if you have a multi-line sequence title followed by a multi-line section title. For these cases, page breaks will have to be explicitly issued.

3.1 The title of the poem

`\poemtitle` `\poemtitle{}`, as its name implies, sets the title of the poem. poemscol gives default values for such things as the font size, the separation between the top of the title and the bottom of the previous poem, the separation between the bottom of the title and the first line, and so on. It also sets penalties in order to encourage page breaks just before a title, and to discourage page breaks between a title and a poem. You can change all these values in your header file by changing the value of the parameters listed below in the implementation section using either `\setlength` or `\renewcommand` (depending upon what you are changing).

`\poemtitlefont` globally sets the font size (and leading) for all poem titles in the main text. `\contentspoemtitlefont` globally sets the font size (and leading) for poem titles in the table of contents. `\afterpoemtitleskip` sets the vertical separation between a poem and its title. `\afterpoemskip` sets the vertical separation between the end of a poem and the title of the next poem. `\poemtitlepenalty` encourages but does not require a page break just before a poem title. A complete list of all of the special commands for setting font sizes, skips, and penalties, is in section 12.1 below.

In the actual text of the poem, you need only use the `\poemtitle{}` macro to specify the title of the poem. The macro takes one argument — the title, of course. The macro typesets the title in the body of the volume (testing first to see whether there is enough space at the bottom of the page to get the title and a couple of lines of the poem in), typesets the title in the table of contents (with the page number), typesets the title in the textual notes (adding the page number, and checking to see whether there is room enough on the page), and prepares similar entries in the lists of emendations and in the explanatory notes (if you need them). Multiple line titles, titles of subsections of poems, titles of sequences of poems, and titles of the elements of a sequence of poems are all special cases, with special commands, which will be dealt with below.

`\poemsubtitle` Subtitles of poems should be issued as the arguments to the `\poemsubtitle{}` macro.

`\titleindent` For indented parts of titles. In the macros for setting the sections and subsections of poetic sequences, poemscol uses `\titleindent` internally to set up a hierarchy of indentations. (A section of a sequence is indented one `\titleindent`. A subsection is indented two. And so on.) `\titleindent` is also used internally in the commands for multiple line titles. Instructions about changing the value of `\titleindent` can be found in section 12.1.

3.2 The body of the poem

`poem` The body of every poem should be placed in a `poem` environment. Putting the body of the poem between `\begin{poem}` and `\end{poem}` resets the line counter to 1, and puts the poem in a `verse` environment (to handle run over lines automatically). poemscol slightly modifies the `verse` environment from the standard LaTeX definition, increasing the indentation used for run over lines, in order to make the difference between the indented run over portion of a long line, on one hand, and an explicitly indented second line, more obvious in the output.

poemscol turns off automatic hyphenation in poetry environments. The idea here is that every hyphen in the printed poem is authorial, obviating the need for you to compile a hyphenated-lines list to distinguish between authorial hyphens and hyphens you added for lineation purposes. Line-ending hyphens should not be a feature of verse anyway, I think. (You may wish to change this for your own edition, in order to improve its look; but if you do so you must keep track of added hyphens yourself. This list will be easy to compile, however, because only authorial hyphens will appear in your source code. Automatically added hyphens will appear only in the output. You might even modify the output routine so that automatically added hyphens have a different look. That would be non-trivial, but Donald Knuth has an exercise about doing just that in *The TeXbook.*) poemscol turns automatic hyphenation back on in prose contexts, so if you wish to keep a hyphenation list for such things as authorial prefaces and so on, you must do so yourself manually. (Alternatively, you can turn automatic hyphenation off in those contexts as well, by setting the `\language` to 255. For an example of how to do this, look at the definition of the `\poem` environment in the implementation section below. If you do turn automatic hyphenation off, it would be wise to restrict the change to some particular environment, rather than changing the `\language` globally. You might wish, for instance, to turn automatic hyphenation off in the `\prosesection` environment.)

`stanza`

Every stanza should be placed in its own stanza environment. Every poem should have at least one stanza. Marking the beginning and end of every stanza (with `\begin{stanza}` and `\end{stanza}`) provides poemscol with a way of detecting cases in which a page boundary falls on a stanza break, since in those cases a page turn happens when one is *inside* a poem environment but *not* inside a stanza environment. Further, marking the beginning and end of every stanza makes the logical structure of the poem (and the editor's intentions about it) clear to readers of your source code.

poemscol adds a little bit more white space between stanzas than the standard LaTeX `verse` environment does. (I found that the standard stanza breaks did not leap out on the page as stanza breaks.)

`\stanzaatbottom`
`\nostanzaatbottom`

If you wish to change the symbol used to mark cases in which a break falls on a page turn, issue the command `\stanzaatbottom{}`, using as the argument whatever you wish to use to mark such cases. The default is *, which is suitable for a published volume. If you are preparing a typescript for submission, you may wish to set `\stanzaatbottom` to "stanza break."

If you wish also to mark all cases where there is NOT a stanza break at the bottom of the page, issue the command `\nostanzaatbottom{}`, using as the argument whatever you wish to mark such cases. The default is `\relax`.

`\verseline`

`\verseline` should mark the end of every line, except the last line of every stanza (which should be marked with `\end{stanza}`). `\verseline` marks the end of every line as a prosodic unit (since a line of verse is not simply a carriage return), and advances the line counter. There are commands below for changing the line counter's horizontal placement (how close it is to the left or right margin of the line) and appearance. (See section 20.10 for these.)

`\setverselinemodulo`

`\setverselinemodulo` sets how often a marginal line number appears. To print a line number every five lines, issue `\setverselinemodulo{5}`. The default value is 10.

`\makeverselinenumbers`

Marginal verse line numbering is on by default. To turn line numbering off, issue `\global\verselinenumbersfalse`. To turn it back on, issue `\makeverselinenumbers`.

`\verselinenumberstoright`
`\verselinenumberstoleft`
`\verselinenumbersswitch`
`\verselinenumbersgutter`
`\verselinenumbersouter`

To put all of the line numbers to the left, issue `\verselinenumberstoleft` in your preamble. `\verselinenumberstoright` puts the line numbers in the right margin. (These two commands use non-floating marginal note macros derived from the memoir class, `\pmclleftsidepar` and `\pmclrightsidepar`, respectively.) If you wish them to appear in the outer margins, issue `\verselinenumbersouter` in your preamble. If you want the line numbers to appear in the inner margins, issue `\verselinenumbersgutter` in your preamble. (These two commands use `\pmclsidepar`, derived from `\sidepar` from the memoir class, but modified by Dan Leucking.) You can adjust the distance between the marginal line number and the text with `\setlength{\marginparsep}{}`.The default value for `\marginparsep` is 18pt, and the default value for the width of the box in which the line numbers are set, `\marginparwidth`, is 10 pt.

The line numbers should print on the baseline. You can adjust them to move them up or down by changing the value of a length called `\pmclsideparvshift`. The default value is `\setlength{\pmclsideparvshift}{0ex}`.

3.3 Special features

`\linebend` poemscol automatically runs over long lines, indenting the run over portion on the next physical line. If you are unhappy with where poemscol has run over a particular line, you can "bend" that line by issuing `\linebend` at the point where you wish it to run over. The run over portion of the line will be indented just as if poemscol had "bent" the line at your selected point. This command only works if you have chosen to bend the line at some point earlier than poemscol would have chosen on its own. If you really do want to extend a line further into the right margin, you can probably do so by using a combination of `\nobreak` and `\hbox{}`, or by turning all of the spaces in that line into unbreakable spaces, marked with ~ in your source. But poemscol will complain if you do this, and rightly so, since the result is likely to be ugly.

You may also wish to use `\linebend` to reproduce how your author broke up long lines on the page (if you know that your author cared about such things and did not leave them up to the typesetter).

`\linebend` should only be used for managing run over lines, not for cases in which a line is to be broken into separate half-lines. For cases in which a line is to be broken into half-lines, use the `\brokenline` macro. The two macros do similar (but not identical) things. But a "linebend" is a feature of typesetting, and a "broken line" is a feature of versification, and it seems best to distinguish them logically. (`\linebend`, like `\brokenline`, issues a carriage return without incrementing the line number, but `\linebend` adds indentation to the next line.)

`\runoverindent` To change the indentation for the run over portion of verse lines globally, issue the amount of indentation as the argument to `\runoverindent{}`. `\runoverindent{}` is currently set to 6 em, rather more than the standard amount in the definition of the LaTeX `verse` environment. By default, the runover portion of long lines is indented. You may wish to flush the runover portion to the right margin instead. For instructions about how to do this, see section 3.3, page 12.

`\verseindent` The `\verseindent` command is to be used for formally indented lines. It should not be used to indent the run over portion of long lines (which poemscol handles automatically anyway). `\verseindent` indents the line a bit less than the run over portion of long "bent" lines are indented, so that the two cases can be visually distinguished. By default, the length of `\verseindent` is 2em. To change it, say to 3em, issue `\renewcommand{\verseindent}{\hspace{3em}}`

`\brokenline`
`\versephantom`

`\brokenline` should be used where you wish to break the line without incrementing the line counter. Authors often break a line into two half-lines, setting the beginning of the second half-line flush with the end of the first half-line. (Sometimes these broken lines indicate a change of speaker. Sometimes they just indicate a change of subject, usually an abrupt one, a change which calls attention to itself by interrupting a line rather than waiting for a line break or stanza break). `\brokenline` is normally used with `\versephantom{}`, which adds white space exactly as long as its argument would have been had it been set in type. `\versephantom{}` thus provides an easy way of setting the beginning of the second half-line flush with the end of the first, whatever the font size or special formatting of the first line.

The sestet of Yeats's sonnet "Leda and the Swan," has such a broken line:

> A shudder in the loins engenders there
> The broken wall, the burning roof and tower
> And Agamemnon dead.
>   Being so caught up,
> So mastered by the brute blood of the air,
> Did she put on his knowledge with his power
> Before the indifferent beak could let her drop?

To set the broken line properly, issue:

```
And Agamemnon dead.\brokenline
\versephantom{And Agamemnon dead.} Being so caught up,\verseline
```

`\tweakbrokenline`

By default, `poemscol` sets the line numbers level with the ends of broken lines. If the line numbers are set to the right of the text, this seems the obvious choice. Conceivably you might want to set the line number level with the beginning of a broken line when the line numbers are set to the left. To ensure that the line numbers of broken lines are set level with the beginning of a broken line when the line number is on the left, issue `\tweakbrokenline` in your preamble.

You should issue `\tweakbrokenline` with caution, because although it changes how `\brokenline` is handled, it does not change the handling of long lines with runover. Runover lines will still set the line number level with the end of the line. You can have runover lines set their line number level with the beginning of the line by issuing `\startverseline` at the beginning of the line.

`\brokenlineatbeginning`

Conceivably you might want `\brokenline` to set the line number level with the beginning of the line in every case. To do this, issue `\brokenlineatbeginning` in your preamble. If you do this, you should be prepared to mark the beginning of runover lines with `\startverseline`.

`\startverseline`

If you begin a verse line with `\startverseline` the line number will be set level with the beginning of the line rather than with the end. This is useful, for instance, if you want to ensure that the line number of a line with runover is set level with the beginning of the line. You can, if you don't want to keep track of where runover lines are, start every line with `\startverseline`.

\tweakstartverseline

Rather than setting the line number of every runover line level with the beginning of the line, it may make more sense for you to set the line number level with the beginning in cases where the line number is to the left, and level with the end in cases where the line number is to the right. You might want to do this, for instance, if you have decided to set all the verse lines at the outer margin (by issuing \verselinenumbersouter in your preamble), or have decided to set all the verse lines at the inner margin (by issuing \verselinenumbersgutter in your preamble). To ensure that \startverseline does this, issue \tweakstartverseline in your preamble.

\stanzalinestraddle

Some poets (such as Robert Penn Warren) occasionally introduce a stanza break in the middle of a broken line, considering the line to be a single metrical unit despite the fact that it straddles a stanza break. To record these cases, mark the end of the first half-line with \end{stanza} as usual. But instead of opening the next stanza with \begin{stanza} issue \stanzalinestraddle instead. This will make sure that the line counter counts the straddling line as only one line, despite the stanza break. \stanzalinestraddle is usually used with \versephantom{}.

rightflushverse
\rightversebegin

By default, poemscol indents the runover portion of long lines by the amount \runoverindent (by default 6 em). It is possible to set verse so that the runover portion is flushed to the right margin. To do this, you should put the \rightflushverse environment inside the \poem environment. And you must mark the beginning of each verse line with \rightversebegin. Christian Ebert has reminded me that the \rightflushverse environment must be issued *inside* the poem environment. Otherwise it simply flushes every line to the right.

It is rather clumsy to need a special environment for verse with right flushed runover. I plan to provide a command for switching between a verse environment that indents runover lines and a verse environment that flushes runover lines to the right. I have seen examples of this kind of environment from Markus Kohm and Christian Ebert, but so far I haven't been able to persuade them to play well with poemscol

\poemlinelabel

\poemlinelabel{} enables crossreferences by line number. To make a line label, issue \poemlinelabel{text of label} right after the line to which you wish to refer, where "text of label" is some distinctive label you can use for a reference elsewhere. (You should issue the command *after* the \verseline or \end{stanza} command, to make sure that the line number will be correct.) To produce the reference, just issue \ref{text of label} as usual. (\poemlinelabel is just a crudely hacked version of \label from LaTeX.) \pageref{text of label} will set the page number of the page on which the label appears.

4 Making a table of contents

\makepoemcontents

I found the normal LaTeX macros for making tables of contents unsuited to poetry, so I have provided my own. To make a table of contents for your poems, issue \makepoemcontents. It might seem as though you would want to issue this command in your preamble. In fact it is better, for reasons explained below, to issue it immediately after issuing the \putpoemcontents, the command which inputs and typesets a table of contents from a prior run. Information for your table of contents will be written to an external file with the extension .ctn. You may need to run your book through LaTeX twice in order to generate a table of contents, the first time to generate the correct page numbers, and the second time to set the table of contents. (Under some circumstances — if you have an index, for instance — you may need three passes rather than two.) `\makepoemcontents` takes an optional argument, which sets the page number for the table of contents. If your publisher wishes to add some front matter so that the contents will appear on page vii, you should issue `\makepoemcontents{7}`.

All of the commands that generate lines in the table of contents are somewhat messy and somewhat rigid, but you can modify them if you wish. Lines in the table of contents usually include an indent (how much of one depends upon what the entry is — whether it is a volume title, a poem title, or something else), then the title itself. If the entry is for a poem, for a section of a poem, or for a section of a poetic sequence, a slash follows the title, followed by the page number. If you wish to give *all* your entries a different look altogether, modify the `\c@ntentsinfo` macro below (see section 20.13.1, page 66. To change the appearance of a *particular* kind of entry (for all sequences of poems, for instance), go to the definition of the title of that kind of entry (e.g. `\sequencetitle`) below, find the line where it calls the `\c@ntentsinfo` macro, and change the arguments to that macro to suit you.

`\setcontentsleaders` By default the title and the page are separated by a slash (actually by `~/~`). If you wish to change this, say to put dot leaders in instead, issue what you wish to use as the argument to `\setcontentsleaders`, as for instance `\setcontentsleaders{\poemdotfill}`, which will flush the number to the right margin, and insert leading dots up to the number.

`\putpoemcontents` `\putpoemcontents` will input the table of contents file if it exists, and complain on your log file if it does not. Because the table of contents uses a special page style and special fonts `\putpoemcontents` restores the default fancy page style once it has input the contents, and restores `\normalfont` as well. Be sure to reset the style of page numbering (from `\pagenumbering{roman}` to `\pagenumbering{arabic}`) after issuing `\putpoemcontents`, since when you read in the external file created by `\makepoemcontents` the page numbering will be set to roman. The command `\resetpagestyle` (which is called by `\putpoemcontents`) will reset the fancy page style to that used in the main body of the volume, but it won't change the page numbering to arabic, in case there is some frontmatter between the contents and the main text, so you should be sure to do so yourself. If you neglect to issue `\pagenumbering{arabic}` before you begin to get to your poems your page numbering will remain in roman numerals. The notes sections use the page numbers from the poem to construct running headers of the form "Notes to pp. xx–yy." If those numbers are roman numerals rather than arabic ones, poemscol will not recognize them as numbers, and you will get the infamous "Missing Number" error that puzzles so many users of LaTeX.

Because `\putpoemcontents` uses the table of contents file from a prior run, you should issue `\putpoemcontents` *before* issuing `\makepoemcontents`. That sounds counterintuitive. But remember that `\putpoemcontents` will read in the contents file you made on a previous run, and `\makepoemcontents` will over-write that file and start a new one. You should issue `\makepoemcontents` just *after* issuing `\putpoemcontents`.

4.1 Entries for notes sections and the index

`\putpoemindex`

Remember that if you are making an index of titles and first lines you may need to run your file through LaTeX after running MakeIndex, to put the entry for the index into your table of contents. `\putpoemindex` inputs the formatted index, and adds an entry for it into the table of contents. Use `\putmultiplepoemindex` if you are making several indices. For instructions about `\putmultiplepoemindex` see section 5.1, p. 15.

`\puttextnotes`
`\putemendations`
`\putexplanatory`

`\puttextnotes`, and its siblings `\putemendations` and `\putexplanatory`, like `\putpoemindex`, input the formatted notes sections, and put entries for them into your table of contents. As with `\putpoemindex`, you will have to run LaTeX again to set the table of contents properly. The first run sets a label to find the page number to add to the table of contents, and the second run actually sets the table of contents. (These commands are described below, in section 10.) All of these commands also restore normal hyphenation.

`\finish`

Rather than issuing `\putpoemcontents` in the proper place, and `\puttextnotes` and its siblings in their proper places, you can issue the `\finish` command just after your last poem. `\finish` turns automatic hyphenation back on, closes and reads back in the external files for the emendations, the textual collations, and the explanatory notes, sets the index of first lines and titles, puts the information about these sections in the table of contents, closes the external file for the contents, and reads it back in. One disadvantage of this method is that you print the book with the contents at the end, so when you print it you must retrieve the contents pages and put them in the proper place. I found the convenience worth the hassle, but you might not. This command is deprecated now. But I have retained it, since the few users who gave me their opinions of the matter told me that they preferred to use `\finish` than `\puttextnotes`.

5 Making an index of titles and first lines

`\indexingontrue`

To set up the Index section of your volume, issue `\indexingontrue` in your preamble. This will create the external file for your index information and send that file the typesetting information for its title. You must process this external file (the .idx file) with MakeIndex and run your book through LaTeX again in order to set a properly organized and formatted index. MakeIndex sorts the various multi-level "index cards" generated by all of the `\index` commands in your source files. Once you have processed your .idx file with MakeIndex, and generated the sorted and formatted .ind file for your book, the `\putpoemindex` command will insert the typeset index where you issue it, and put an entry for your index in your table of contents. You will have to run your file through LaTeX again to make the page number for that entry correct.

poemscol sets up a two-column index with a simple running header. (Setting up a two-column index, however, requires that you include the multicol package in your `\usepackage{}` list.) To turn indexing off, issue `\global\indexingonfalse`.

Enter the individual index entries in the poems as you normally would for use by MakeIndex (say, after each title and first line). One of the nice features of MakeIndex is that every entry has two parts, one for specifying the sort key of the entry (or subentry), the other for specifying the font and typesetting information of the entry (or subentry). This enables you to drop all of the initial uses of "the" in titles for sorting purposes, while still including the "the" in the title. It also enables you to distinguish titles from first lines (by italicizing them), and to include typesetting information (such as italicization) in titles or first lines without messing up their sort order.

5.1 Making Multiple Indices

poemscol can work with splitindex to generate more than one index. You might, for instance, wish a separate index of titles and index of first lines, or an index of proper names. To do this, call splitindex in your list of packages. Then, in your preamble, declare the indices you are planning to use, e.g. `\newindex[Index of Titles]{idx}` `\newindex[Index of First Lines]{fir}` `\newindex[Onomastic Index]{ono}`. The second argument is the "shortcut" defined by splitindex, and will be used to mark index entries in your text (see below), and as the extension for the external index files used by MakeIndex.

Then mark up individual entries. Some entries (in the example above, the "Index of Titles" examples) can be marked up with `\index` just as if you were using only one index. Entries for other indices should be marked using the `\sindex` command from splitindex, as follows

```
\sindex[fir]{Of Man's first disobedience and the fruit}
```

where the first argument is the "shortcut" and the second is the index entry. If you are loading splitindex with the split option, the different types of index entry will be sent to separate output streams, which each must be processed separately with MakeIndex. TEX has only a limited number of such output streams, however, so it is probably wiser not to choose that option, but to use splitindex.tex, splitindex.pl, or one of the other means provided with the splitindex package to break up the index into the separate parts, each of which must be processed with MakeIndex separately.

To include the indices, and to add entries for them to the table of contents, issue `\putmultiplepoemindex` for each one. `\putmultiplepoemindex` takes four arguments, which are: 1. The three letter "shortcut" used by `\splitindex`, 2. The title of the Index on its first page, 3. The running header for that index section, and 4. The name of the section as you wish it to appear in the Table of Contents. So, for instance

```
\putmultipleindex{fir}{INDEX OF FIRST LINES]{Index of First
Lines}{First Line Index}
```

includes an index with shortcut “fir,” title “INDEX OF FIRST LINES,” running header “Index of First Lines,” and contents entry “First Line Index.”

6 Collections with multiple volumes

`volumetitlepage`
`maintitlepage`

The `\volumetitlepage` environment is an environment for divider pages in collections made up of several volumes. Volume title pages will always appear on recto pages. poemscol will automatically create a blank verso page preceding the volume title page if it necessary. The `\maintitlepage` environment is for the title page of the whole book. The main title page will also automatically always be on a recto page. These divider pages have special page styles, with no page numbers and no running headers.

`\wholebooktitle`
`\volumetitle`
`\volumetitlefirstline`
`\volumetitlemiddleline`
`\volumetitlelastline`
`\volumesubtitle`
`\volumesectiontitle`

`\wholebooktitle{}` is for the title of the entire collection. `\volumetitle{}` is for the title of an individual volume within a Collected Poems. The `\volumetitlefirstline{}`, `\volumetitlemiddleline{}`, and `\volumetitlelastline{}` macros are for multi-line titles of volumes gathered into the collection. (Use these last commands if you want to specify how a long title is broken up both in the text and in the table of contents. poemscol will break up long titles on its own anyway, but you might not be happy with how it breaks up the lines.) `\volumesubtitle{}` is for setting the subtitle to a volume. `\volumesectiontitle{}` is for setting the title of a section of a volume. In addition to setting the text in the body of your edition, these commands send the title (and the formatting information about it) to the table of contents file and to the textnotes file.

`\volumeepigraph`
`\volumeattribution`
`\volumeheader`
`\leftheader`
`\rightheader`

`\volumeepigraph{}` and `\volumeattribution{}` are for epigraphs and attributions on the divider pages. `\volumeheader{}` or `\rightheader{}` is the right running header for that volume (except for divider pages, the table of contents, the notes sections, and other special cases). Normally that is just the name of the particular volume. `\leftheader{}` is the left running header for the entire book (except for divider pages, the table of contents, the notes sections, and other special cases). Normally that is the name of the entire book.

7 Recording textual notes

7.1 Setup for sections for textual notes, emendations, and explanatory notes.

`\makeemendations`
`\maketextnotes`
`\makeexplanatorynotes`

If you issue these commands, poemscol puts all of the emendations, textual collations, and explanatory notes in the backmatter of your volume. poemscol opens external files for emendations, textual collations, and explanatory notes (with extensions .emd, .ent, and .enx respectively), and reads them back in at the place you specify with `\puttextnotes`, `\putexplanatory`, or `\putemendations`, or at the end (if you use the `\finish` command).

To collect emendations, issue `\makeemendations` in your preamble. To collect textual collations, issue `\maketextnotes` in your preamble. To collect explanatory notes, issue `\makeexplanatorynotes` in your preamble. These commands set up the external files, send to them the commands for typesetting the title of each section (e.g. Explanatory Notes) and for modifying the running headers of the notes sections to xx–yy format (e.g. "Explanatory Notes to pp. 55–7").

If you wish to use multi-level footnotes rather than endnotes, you do not need to issue `\maketextnotes` or its relatives (but you should of course issue `\textnotesatfoot` or its relatives). You can mix footnotes or endnotes as you wish. You may, for instance, wish to set explanatory notes as footnotes, and textual notes and emendations as endnotes. Instructions for setting up poemscol to make footnotes are in section 9 below.

In end note sections poemscol opens an entry (headed with the title and page of the poem in the text) in the textual notes section for every poem (since, minimally, you want to include information about the variants and copy text of the poem), but only opens entries in the emendations or explanatory notes sections if you actually have emendations or explanatory notes on that poem. The notes for each poem are set in a block paragraph, with a new paragraph for every `\poemtitle` command, and the line counter for the notes is reset to 1 with every `\begin{poem}`. (For long poems in books or cantos, it would be wise to use `\poemtitle` for the canto names or numbers, setting the poem title in some other way, perhaps with `\volumetitle` described above.)

7.2 Textual notes of various kinds

`\sources`

To record information about the copy text, editions and publication history of individual poems, or any information not tied to specific lines in the poem, you should place that information in the argument to the `\sources{}` macro. Typically, you should issue this macro after you have issued `\poemtitle{}` and before you issue `\begin{poem}`. If you wish to send information to the textual notes file (such as to force a page break), you can do so by using `\sources{}`. You can send typesetting information to other sections by using `\literalemend{}`, `\literalexplain{}`, or `\literalcontents{}`. There is also a `\literaltextnote{}` command, which is equivalent to `\sources{}`.

`\textnote`
`\sameword`
`\missingpunct`
`\emendation`
`\explanatory`

`\textnote{}` is used to capture variants and tie them to the correct line number. You don't have to count line numbers. poemscol keeps track of them for you. Issue `\textnote{}` immediately after the `\verseline` command which marks the ending of the line you wish to comment upon. Put the text of your note (which may be simply the recording of a variant in standard notation) into the argument of the macro. You should put both the lemma and the variants or comment in the argument to the `\textnote{}` macro. To put the $\sim$ glyph in your note (used for recording places where the variant and the copy text have the same word, as for instance when recording a variation of punctuation) use `\sameword`. To put the $_\wedge$ glyph into your text (used for recording places where a punctuation mark is missing in a variant), use `\missingpunct`.

For notes that refer to ranges of lines, `\textnote`, `\emendation`, and `\explanatory` have an optional argument, which records how many lines the note refers to. There are two ways of using this argument. The grubby but simple way is simply to put into the argument the number of lines it refers to. If, for instance, you issue `\textnote[4]{lemma] rest of note}`, poemscol, rather than printing merely the number of the line which closed just before you issued `\textnote`, will print a line range, beginning at that line, and ending at the fourth line after it. If that `\textnote` were issued after line 214, this is to say, the output in the textual notes would be: 214–218: lemma] rest of note.

It seems clumsy to count the lines referred to by a note, especially since you don't have to count lines to set where the note begins. A more sophisticated way, and one which involves no counting, is to make a unique poem line label for the last line of the range by issuing `\poemlinelabel{some unique label}` just after the `\verseline` or `\end{stanza}` that ends the range. (To use `\poemlinelabel` see section 3.3.) Then, back after the `\verseline` or `\end{stanza}` command that closes the first line of the range, put the text of the label into the optional argument of `\textnote` as follows: `\textnote[some unique label]{lemma]rest of note}`. This is the preferred method. The line-counting method is deprecated, but retained for backwards compatibility with earlier versions. If you use the label method, you will have to run your file through LaTeX twice, once to define the labels, and once to set the text.

One advantage of using an optional argument to specify line ranges in notes is that nested or overlapping lemmas, or lemmas that include stanza breaks, do not pose any difficulties.

`\emendation{}` and `\explanatory{}` are used exactly as `\textnote{}` is. Issue the emendation or the explanatory note as the argument to the command. Place the command immediately after the `\verseline` that concludes the line to which it is a comment. These commands also can take optional arguments to specify line ranges.

If you quote verse *within* a textual note, an explanatory note, or an emendation, you may want to use the normal LaTeX markup for verse for that quotation, rather than the special markup for poemscol. poemscol will not complain about this.

If a textual note *ends* with a quotation in verse, you should follow the `\textnote{}` entry with `\quotedversecorrectiontextnote`. I found that if I quoted verse in the note sections a dirty trick I used to send the notes to the external files would cause the following notes on that poem in the notes sections to be set in a new paragraph. (This is of course not a problem if the quoted verse is the last thing in the notes to that poem, since the next poem will open a new paragraph in the notes anyway. It is also not a problem if more of the individual textnote follows the verse. It's only a problem if the last thing in the note is quoted verse, and the note is not the last one on that poem.) The command should set it flush as it is, but I have found odd cases I can't understand in which that doesn't work, so the command has an optional argument, which should be the additional length you need to adjust the next line by to make it flush. The default is 0pt. `\quotedversecorrectionexplanatory` and `\quotedversecorrectionemendation` should follow explanatory notes or emendations that end with snippets of verse. These three commands are an embarrassing kludge on my part. They simply send commands to the relevant note sections that prohibit opening a new paragraph. You should issue this command in the main text right after the close of the textnote (not in the text of the textnote itself).

`\accidental`

`\accidental{}` behaves exactly like `\textnote{}`. If you wish to distinguish between accidentals and substantives, this command provides a way of doing so. If you wish to exclude accidentals from your textual notes, issue `\global\includeaccidentalsfalse` in your preamble. Accidentals are included by default, but if you issue `\includeaccidentalstrue` in your preamble, you will remind yourself of that fact.

poemscol does not provide for a separate backmatter section for accidentals, but it would be trivial to construct one, using `\definenewnotetype`, described in section 8 below. Be sure to delete the existing definition of `\accidental` from poemscol.sty before doing this.

If you wish to exclude accidentals from your printed output, but to mark them in your source files, so that your published collation consists only of substantives, issue `\global\includeaccidentalsfalse` in your preamble. Many publishers are reluctant to publish accidentals, believing that they are, well, less substantive than substantives. Using the `\accidental{}` command allows you to exclude accidentals from the published version should your publisher insist, while preserving the information about them should the publisher's mind change. In the very worst case, if you have marked all the accidentals in this way you can still produce a list of accidentals for later use, and other scholars can search for accidentals in your source files simply by searching for the string `\accidental`.

`\tsvariant`
`\tsentry`

Some publishers may turn up their noses at typescript or manuscript variants in just the way they turn up their noses at accidentals. If you wish to distinguish between published variants and typescript, manuscript, or galley variants, `\tsvariant` (for a whole note) and `\tsentry` (for a typescript entry in a list of variants in a textual note that includes other kinds of variant) provide ways of doing so. By default, typescript variants are included. To exclude typescript variants, issue `\global\includetypescriptsfalse` in your preamble. You can issue `\global\includetypescriptstrue` in your preamble, to remind yourself that you are including typescript variants. In that (default) case `\tsvariant{}` will behave exactly like `\textnote{}`, and entries in other notes marked with `\tsentry` will be included in your output.

Here is how to use `\tsentry`: if you wish to include typescript entries in a single note including those entries in a list with variants from other published versions (as for instance when a comma appears in a typescript but only in the second edition of the published poem), simply issue `\textnote{}` as usual, marking the relevant variant in the list of variants with the `\tsentry{}` macro. If typescript variants are excluded, the typescript entry will also be excluded.

Here is a typical use of the `\tsentry{}` command:

```
Of moonlit desert. A stallion, white and flashing, slips,\verseline
\textnote{Of moonlit] Of the moonlit {\em NY\/}\tsentry{, SP85TS
(revised in black pen to SP85)}}
```

Notice that since the `\tsentry{}` comes in the middle of the list, it begins with a comma. You might expect an `\unskip` command before the comma, to prevent LaTeX from adding white space before the comma. poemscol automatically provides the `\unskip` if the entry starts with a comma or a semicolon.

You can mark individual variants with `\tsentry{}` in the arguments to the `\explanatory{}` and `\emendation{}` commands as well, and `\tsentry` will also behave normally in any endnote or footnote section you define using `\definenewnotetype`.

poemscol does not provide for a separate backmatter section for typescript variants, but it would be trivial to construct one, again using `\definenewnotetype`. To do this, you would also have to copy all of the emendations, explanatory notes, and textual notes which have a `\tsentry` in their arguments, and construct `\tsvariant{}` commands in those places. For information about `\definenewnotetype` see section 8.

8 Creating new kinds of endnote

`\definenewnotetype` The `\definenewnotetype` macro can be used to create new kinds of endnote, in case the kinds poemscol defined are not enough. `\definenewnotetype` takes five arguments. They are

1. A base name used to construct
   - some `\if` variables (e.g. `\ifmynoteson` to see whether you are collecting "mynotes", and `\ifnomynoteyet` to see whether you have written a "mynote" for a particular poem yet). poemscol will only open an entry for a poem in the "mynotes" section if there actually is a "mynote" on that poem. Note that the `\if` that refers to the section is plural (`\ifmynoteson`) and that the `\if` that refers to a singular note is singular (`\ifnomynoteyet`).
   - The name for the output channel for writing the notes to the appropriate external file. This name will always be plural.
   - The name for the "literal" command for writing literal, unexpanded text on the file (as `\literaltextnote{}` does on the textnotes file), and for the note command that sends the actual textual note (line number and all) to the section (e.g. `\mynote{}`), also the command to write notes in line-numbered prose (e.g. `\prosemynote{}`) or prose numbered by paragraph and sentence (e.g. `\pmmynote{}`). These commands will always be singular.
   - The name of the `\put` command (e.g. `\putmynotes`) which closes the external file for the notes section, reads it back in to your document, and adds an entry in the table of contents for that notes section. This command will always be plural.

- The name of the `\margref` command for putting a marginal mark next to a line giving the page number in the apparatus where a note is to be found (e.g. `\margrefmynote`)

2. The extension for the external file which will hold the notes as you generate them. "Emendations" for instance, are written to an external file called `\jobname.emd` The file name will always be `\jobname`, and the different species of endnotes will all have distinctive extensions. (`\jobname` is a macro that always expands to the name of your top level source file. If, for instance, I am setting running a file called "myfile.tex" through LaTeX `\jobname.emd` translates to "myfile.emd.")
3. The title of the notes section as you wish it to appear in the running header (e.g. "Emendations").
4. The title of the notes section as you wish it to appear on the first page of the notes section (e.g. "EMENDATIONS").
5. The title of the notes section as you wish it to appear in the table of contents.

So, for instance,

```
\definenewnotetype{xyznote}{xyz}{Xyznotes to}{XYZNOTES}{Xyznotes}
```

creates the booleans `\ifxyznoteson` and `\ifnoxyznoteyet`, the commands `\xyznote`, `\literalxyznote`, `\prosexyznote`, `\pmxyznote`, and `\putxyznotes` (and, although you don't ever have to use it explicitly, `\firstxyznote`) the output channel xyznotes, the external file `\jobname.xyz`, an entry in the contents for "Xyznotes", and an endnotes section with a running header of the form "Xyznotes to pp. xx–yy" and a title XYZNOTES. `\xyznote` takes the same arguments that `\explanatory` does (which is to say, an optional argument for a label for entering line ranges, and an argument for the lemma and the comment). There won't be an entry for a poem in the "Xyznotes" endnotes section unless there actually is an xyznote for the poem. `\prosexyznote` takes the same arguments `\proseexplanatory` does (which is to say, an aptional argument for a label for entering line ranges, an argument for a label to capture the line number of the first line, and an argument for the lemma and the comment). `\pmxyznote` behaves exactly like `\pmexplanatory`.

9 Multi-level footnotes

poemscol will set multiple levels of footnotes in block paragraphs, separating each level with a rule. To enable this feature, call the manyfoot package as follows in your preamble: `\usepackage[ruled,para*]{manyfoot}`.

To set your textual notes as footnotes, issue `\textnotesatfoot` in your preamble. If you do this, poemscol will actually create two sets of footnotes for textnotes, one for those that are called with `\textnote`, which will produce an unindented block of notes, and one for those that are called with `\sources`, which will produce individual indented paragraphs, opening a new paragraph with every note. `\sources` notes often have several paragraphs, so they are inappropriate for block-paragraph or "para" mode, and because those paragraphs are often in fact prose paragraphs, rather than blocks of notations, they are indented, so that they will be read as prose. To set your emendations as footnotes, issue `\emendationsatfoot` in your preamble. These notes will be set by default as block paragraphs. If you have multiple-paragraph notes in your emendations, you should redefine these notes to set plain rather than para style notes. (See the next paragraph for how to do this.) To set your explanatory notes as footnotes, issue `\explanationsatfoot`. These notes will also, by default, be plain style notes, with one indented paragraph per note. You do not need to issue `\maketextnotes`, `\makeemendations` or `\makeexplanatorynotes` if you set those notes as footnotes. If you use `\literaltextnote` anywhere in your file, however, you will lose the text it would have put in your textnotes, since `\literaltext` just writes its argument onto the external file for textual notes.

To change the style of your footnotes between plain and para modes, globally change the values some of the following booleans. `\ifsourcesfootnotespara`, obviously, is true if footnotes from the `\sources` command are set in "para" mode (default is false). `\iftextfootnotespara` is true if footnotes from `\textnote` are set in "para" mode (default is true). `\ifemendationfootnotespara`, or `\ifexplanfootnotespara` are the booleans which set the style of footnotes from `\emendation` and `\explanatory` respectively. By default, footnotes from `\emendation` are set in "para" mode, and footnotes from `\explanatory` are set in "plain" mode. g explanatory footnotes from plain to para mode, for instance, issue `\global\explanfootnotesparatrue`. Counterintuitive as it sounds, you must issue this *before* issuing `\textnotesatfoot` (or the command for whichever other kind of note you are wishing to change the appearance of).

When `\sources` is defined to send endnotes to the textual notes section, they usually appear immediately underneath the title of the volume or poem they comment upon. If you are having `\sources` output footnotes, it would be a good idea to open the note with the title of the volume or poem it is a comment upon, in case there is more than one poem on the page.

When note classes are set as footnotes, the symbol which marks cases in which a page turn falls on a stanza break will be set *beneath* the footnotes. This is a consequence of the way poemscol uses the `\mark` mechanism to keep track of these cases, so I have not figured out how to change it. But it may well be more of an idiosyncrasy than a bug, particularly if the symbol is changed from * to something like "stanza break at page turn."

To create a new level of footnote, you must first declare it to make it known to manyfoot, which requires each level of footnote to be given a unique suffix (usually a capital letter of the alphabet). You can declare a new footnote level called `\footnoteE{}` for instance, by issuing `\DeclareNewFootnote[para]{E}`. You will want notes which don't put a mark in the body of the text, and begin with the line number or range of line numbers of the lemma. To define a footnote called `\mynote{}`, first declare new footnote `\footnoteE` (A, B, C and D are reserved for sources, textual notes, emendations, and explanatory notes, respectively). Then issue `\newcommand{\mynote}[2][0]{\setlemmarange{#1} \FootnotetextE{\citerange}{#2} \resetlemmacounters}` (`\setlemmarange`, `\citerange`, and `\resetlemmacounters` enable your footnote to handle line ranges in the lemma.) If a `\mynote` refers to a range of lines, you can issue the same kinds of optional argument that are used with `\textnote`.

10 Pulling it all together

`\testforauxonfirstrun`

If you don't want to listen to LaTeX complain about missing references during your first run, issue `\testforauxonfirstrun`, which won't let `\finish`, `\putexplanatory`, `\puttextnotes`, or `\putemendations` run unless there is an .aux file for a previous run. *Be sure to issue this command before issuing* `\begin{document}`, *since that reads in the existing .aux file, and overwrites it with a new one.*

`\putemendations`

This macro closes the external file for emendations and reads it in. It also puts an entry for the emendations section into your table of contents. After `\putemendations` has read in the external file for emendations, you will need to run your file through LaTeX again, to put the correct page number in the table of contents file.

`\puttextnotes`
`\putexplanatory`

`\puttextnotes` and `\putexplanatory` do the same thing for the textual notes and for the explanatory notes.

All of these last three commands restore normal hyphenation. Since all three are printed with the `\small` font size, the hyphen character is defined only for that size. This should not create other issues, however, because the commands that prohibit hyphenation in the `poem` environment (and in the `contentsentry` environment) are local to those environments, so the restoration of the hyphen character may be redundant.

`\putpoemindex`

`\putpoemindex` reads in the index at this point, and adds an entry for the index to external file for the table of contents.

`\putmultiplepoemindex`

If you are generating several indices using splitindex `\putmultiplepoemindex` reads in one of them at this point. For instructions about this command, see section 5.1, p. 15.

`\finish`

The `\finish` macro is deprecated but still available. It is designed to be placed just before `\end{document}` (which, in fact, it replaces — but you should still put an `\end{document}` at the end of your document for form's sake). `\finish` closes the external files for the three notes sections and reads them back into the document (turning the automatic hyphenation back on for these sections). It adds entries to the table of contents file for the notes sections. It reads in the formatted index file (which you have made from your raw .idx file using MakeIndex between LaTeX runs) and adds an entry for the index to the table of contents. Then it closes the external file for the table of contents and reads that in (putting it at the end of the document, as I explained before). Then it adds a page explaining that the symbol (by default *) at the bottom of certain pages marks places where a stanza break and a page break coincide. If you have an editor's introduction preceding your notes section, or a list of abbreviations, you may wish to consult the `\finish` macro for suggestions about how to add the lines to read those files in and add entries for them in the table of contents.

11 Special cases

11.1 Epigraphs, attributions etc.

`\epigraph`
`\headnote`
`\attribution`
`\poemdedication`

`\epigraph`, `\headnote`, `\attribution`, and `\poemdedication` set their arguments as epigraphs, headnotes, attributions, or dedications. If you have dedications, epigraphs, or attributions on the divider pages of editions that combine several volumes of poetry, use `\volumededication{}`, `\volumeepigraph{}`, and `\volumeattribution{}`.

11.2 Inset prose quotations

`\saveverselinenumber`
`\restoreverselinenumber`

Sometimes you will have inset prose passages, but you will not wish to start the line numbering over after the passage. The way to set this is to issue `\end{stanza}` at the end of the stanza before the prose insert. Then issue `\saveverselinenumber`. Then issue `\end{poem}`. Insert the prose quotation (perhaps followed by `\attribution{}`). Then issue `\begin{poem}` followed by `\restoreverselinenumber` and `\begin{stanza}`. Then begin entering the rest of the poem as usual. Longer, independent prose sections (such as the author's preface or prose poems) require different handling. To provide line numbers and notes for such prose sections, see section 13.

11.3 Multiline poem titles

`\poemtitlefirstline`
`\poemtitlemiddleline`
`\poemtitlelastline`
`\poemtitlenocontents`
`\poemtitlebaretitle`

Very long titles present special problems. In the first place, they require special provisions to make sure that page breaks don't happen in the middle of a title, or immediately after a title. In the second place, they raise questions about how the author wished to break them up over several lines. If you think it might matter, these macros make it possible for you to ensure that long titles are broken in exactly the same way in your text and in your table of contents. (I didn't pursue this question for titles in notes sections, and in notes sections these commands simply concatenate a long title.)

If you wish to set the titles differently in the text and in the contents, you can set the title in the text and notes with `\poemtitlenocontents{}` and send a differently formatted entry to the table of contents file using `\literalcontents{}`. (Remember that you will have to send not only the title of the poem to the contents, but also the other formatting information, such as the indentation, the slash, and the label for the page number.) `\poemtitlebaretitle{}` simply sets the title in the text, enabling you to use `\literalcontents{}` and `literaltextnotes{}` to produce whatever effect you wish.

Alternatively, you can just let poemscol handle the long titles by itself. If you are pretty sure that the author did not care how long titles were broken up, this would probably be the best thing to do. Remember also that if poemscol breaks a title across two lines, you can always force it not break the line by using enclosing the argument to `\poemtitle{}` in `\hbox{}`. You may get a complaint on your log file about an overfull `\hbox` if you do this, however.

11.4 Italicized poem titles and empty poem titles

Titles with italics or other special formatting posed special problems in earlier versions (before version 1.7) since the typesetting commands gummed up how I passed parameters around that I was unable to solve using `\protect`. All of these commands are obsolete now, but are retained for backward compatibility.

`\poemtitleitalic`

`\poemtitleitalic{}{}` takes two arguments. The first is the title of the poem with the italics (or small capitals or international characters or what have you), for typesetting, and the second is the title without the commands, for cross reference purposes. (Actually, now that poemscol no longer needs to treat italics in titles in a different way, it can be anything you want, since the second argument is discarded.)

`\poemtitlenotitle`

`\poemtitlenotitle{}`, with its paradoxical name, is for poems without formal titles. Usually such poems are referred to by their first lines. This macro does not produce a title in the text, but sets its argument in the table of contents and in the notes. Usually you use the first line as the moral equivalent of a title.

`\poemtitlenotitleitalic`

`\poemtitlenotitleitalic{}{}`, with its even more paradoxical name, is for poems without formal titles but with italicized words in their first lines. `\poemtitlenotitleitalic{}{}` takes two arguments. The first is the first line with the italics (for typesetting purposes), and the second is the first line without the italics (for crossreference purposes). This command is obsolete now that italics in titles no longer require special handing, but is retained for backwards compatibility.

11.5 Poems in sections

`\poemsectiontitle`

`\poemsectiontitle{}` and (`\poemsubsectiontitle{}`, as their names imply, are for the titles of sections or subsections of poems that have them. `\poemfirstsectiontitle` is for the first section of a poem with many sections. (It differs from `\poemsectiontitle` in that it tries to make it harder for a page break to separate the title of a poem from the title of its first section.) `\poemsectiontitlefirstline{}`, `\poemsectiontitlemiddleline`, `\poemsectiontitlelastline{}`, `\poemsectiontitlenocontents{}`, and `\poemsectiontitlebaretitle{}`), as their titles imply, are used for special cases such as long section titles, if you wish to specify explicitly how the title is to be broken up in the text and in the table of contents. You can construct variants of these macros for dealing with special cases I didn't anticipate by using `\poemsectiontitlebaretitle{}` and `\literalcontents{}` and its siblings.

11.6 Poetic sequences

`\sequencetitle`
`\sequencesectiontitle`
`\sequencefirstsectiontitle`

Sequences of related poems, perhaps with an over-title, demand special handling in the text, in the notes sections, and in the table of contents. (A moment's thought will show that they are different in some ways from poems in sections. For one thing, a sequence manifests a different relationship between part and whole than a poem in sections does, and often provides stronger experiences of momentary closure between sections.) `\sequencetitle{}` sets the title of the sequence, and `\sequencesectiontitle{}` sets the titles of the sections of the sequence. `\sequencefirstsectiontitle{}` gives a little more vertical space between the main title and the first section title than `\sequencesectiontitle{}` gives between adjacent sections later on in the sequence (since the over-title for the sequence is in larger type) and adjusts the page breaking penalties to reflect the fact that such titles should not occur near the bottom of a page, since there should be no page break between the sequence title and the title of the first section of the sequence (although page breaks are permissible, indeed even slightly favored, between later sections). There should also never be a page break between a section title and the first or second line of the section. The penalties poemscol sets should take care of these automatically, but you may still have cases which require you to specify a page break with `\pagebreak` or `\newpage`. There are a host of other commands, all with self-explanatory titles, that deal with some special situations that arise in setting the elements of poetic sequences, such as cases where you wish to specify how long titles will be broken up into lines, or where you wish no entry to be made for the section in the table of contents or in the notes sections. These macros don't of course exhaust the dizzying possibilities, but from them you can construct whatever other macros you may need.

Here is a list of all of the defined title commands.

- Macros for Poem titles
  - `\poemtitle`
  - `\poemtitlefirstline`
  - `\poemtitlemiddleline`
  - `\poemtitlelastline`
  - `\poemtitlenotitle`
  - `\poemtitlenocontents`
  - `\poemtitlebaretitle`
  - `\poemsubtitle`
- Poems in Sections
  - `\poemsectiontitle`
  - `\poemsectiontitlefirstline`
  - `\poemsectiontitlemiddleline`

  - `\poemsectiontitlelastline`
  - `\poemsectiontitlenocontents`
  - `\poemsectiontitlebaretitle`
  - `\poemfirstsectiontitle`
  - `\poemfirstsectiontitlebaretitle`
  - `\poemsubsectiontitle`
- Sequence Titles
  - `\sequencetitle`
  - `\sequencetitlefirstline`
  - `\sequencetitlemiddleline`
  - `\sequencetitlelastline`
  - `\sequencetitlenonotes`
- Sequence Sections
  - `\sequencesectiontitle`
  - `\sequencesectiontitlenocontents`
  - `\sequencesectiontitlenonotes`
  - `\sequencesectiontitlebaretitle`
  - `\sequencesectiontitlefirstline`
  - `\sequencesectiontitlemiddleline`
  - `\sequencesectiontitlelastline`
  - `\sequencefirstsectiontitle`
  - `\sequencefirstsectiontitlenocontents`
  - `\sequencefirstsectiontitlenonotes`
  - `\sequencefirstsectiontitlefirstline`
  - `\sequencefirstsectiontitlemiddleline`
  - `\sequencefirstsectiontitlelastline`
  - `\sequencesectionsubtitle`
- Sequence Subsections
  - `\sequencesubsectiontitle`
  - `\sequencesubsectiontitlenocontents`
  - `\sequencefirstsubsectiontitle`
  - `\sequencefirstsubsectiontitlenocontents`
  - `\sequencesubsectiontitlefirstline`

- \sequencesubsectiontitlemiddleline
- \sequencesubsectiontitlelastline
- \sequencefirstsubsectiontitlefirstline
- \sequencefirstsubsectiontitlemiddleline
- \sequencefirstsubsectiontitlelastline
- \sequencesubsubsectiontitle
- \sequencesubsubsectiontitlenocontents
- \sequencefirstsubsubsectiontitlenocontents

11.7 Customizing titles

poemscol provides a large number of ways of setting titles, but conceivably you may wish to define your own. Here is a template for designing a new kind of title. Suppose you wish to create a kind of title called `\mytitle{}`. It would take one argument, which would be the title itself. The macro should in turn call three macros, `\set@p@emtitle`, to set the title in the main text and make the label for use by the contents and notes sections, `\c@ntentsinfo` to set the title in the table of contents, and `\t@xtnotesinfo`, which sets the title in the notes sections. (You should enclose the whole thing between `\makeatletter` and `\makeatother`.) `\set@p@emtitle` takes six arguments, which are, in order, (1) the title (which, if you are calling this command inside of the definition of `\mytitle` should just be `#1`), (2) commands to be issued before setting the title (minimally, a penalty, such as `\poemtitlepenalty`), (3) the font size in which the title is to be set in the text (such as `\poemtitlefont` or `\large`), (4) any indentation of the title (`\relax` for `\poemtitle`, or `\titleindent` for `\sequencesectiontitle`), (5) commands to be issued after the title is set (such as `\afterpoemtitleskip`) and (6) the amount overrun lines in the title should be indented. `\c@ntentsinfo` takes four arguments, which are, in order, (1) the title, (2) the indentation of the title in the table of contents (such as `\contentsindentone`), (3), the font size in which the title should be set (such as `\contentspoemtitlefont`) and (4) the amount overrun lines in the table of contents should be indented (such as `\contentsindenttwoamount`). `\t@xtnotesinfo` takes one argument, the title. By default, poemscol does not indent the overrun portion of titles relative to the first line of the title, but if that first line is indented will set the overrun portions to be indented the same amount. Overrun portions of titles in the table of contents will be indented by one `\contentsindentamount` more than the first line is indented. Both of these behaviors can be changed by changing the arguments to `\set@p@emtitle` or `\c@ntentsinfo`.

If you wish to design a title which might have multiple lines, you will need to write three macros: one for the first line, one for middle lines, and one for the last lines. Before issuing `\set@p@emtitle` issue `\firstlinesettings`, `\middlelinesettings`, or `\lastlinesettings` (as the case may require). And issue `\restoresinglelinesettings` last of all in each of your three macros. The lines will be broken in the table of contents and in the body of the text in the same way, but will be run in in the notes sections.

`\JHpoemtitle`

In his edition of the poetry of José Maria Heredia, Tilmann Altenberg placed a marker in the margin next to poem titles with a reference to where in the apparatus notes on that poem could be found. To do this, use the `\JHpoemtitle` command. This command takes two arguments: first, the title of the poem, second, the label used by the `\label` command on the page of the notes section to which the title refers. So, you would issue the poem title command this way: `\poemtitle{Los desvelados}{EP_LosDesvelados_AC}`. Up to version 2.49 of poemscol you had to be sure to define the label in the notes sections by issuing `\sources{\label{EP_LosDesvelados_AC}}`. This is no longer necessary, and `\JHpoemtitle` defines both the `\label` and the `\pageref`. Instructions are given in section 20.13.4, p. 77 for how to construct similar commands for sequences, sequence sections, and so on.

The following related commands are defined already `\JHsequencetitle`, `\JHsequencefirstsectiontitle`, `\JHsequencesectiontitle`, `\JHsequencesubsectiontitle`, `\JHpoemsectiontitle`, `\JHpoemfirstsectiontitle`, `\JHpoemsubtitle`, `\JHepigraph`, `\JHprosesectiontitle`, and `\JHdedication`.

If you are using `\JHpoemtitle` and its siblings, you need to be sure to add mparhack to your package list. You can prevent the marginal reference from switching sides on alternate pages by issuing `\@mparswitchfalse` (be sure to put `\makeatletter` and `\makeatother` around this command). You can control the side on which the marginal reference appears with `\normalmarginpar` (to place them on the right) or `\reversemarginpar` (to place them on the left). `\JHpoemtitle` and its siblings place the marginal reference using `\marginpar`.

`\setmpargpoemtitle`

`\setmargpoemtitle{}{}` is a more general version of `\JHpoemtitle{}{}`, with the second argument being any commands you want to be issued while the title is being processed.

`\margreftextnote`

Another feature of Altenberg's Heredia edition is a marginal mark attached to lines to which text notes refer showing where in the apparatus a comment upon that line is to be found. `\margreftextnote` sets a label in the textual notes section, and set a marginal note with a page reference to that label. Issue `\margreftextnote` *before* issuing the `\verseline` or `\end{stanza}` that closes the line to which it refers. (The textual note itself can be an ordinary `\textnote`, and should be place, as usual, *after* the `\verseline` or `\end{stanza}`.

`\margreftextnote` generates its own label, so there is no need to make up one yourself.

`\margreftextnote` uses `\marginpar` rather that `\pmclsidepar` to set the note, because I had to use `\marginpar` for `\JHpoemtitle` and its siblings, and I wanted the whole family of markers to be controlled by the same settings. You use the settings that control the behavior of `\marginpar` (e.g. `\mparswitch` and `\reversemarginpar`) to control the behavior of these marks.

If you define a new endnote type using `\definenewnotetype`, `\definenewnotetype` will also define a new `\margref` command to go with it.

`\setmargrefmarker` Altenberg's edition used a ding from the pifonts collection in the marginal marks made by `\margreftextnote` and its siblings. I have set the default to `\dag`, so as not to depend upon yet another style package. You can choose another mark using `\setmargrefmarker` (e.g. `\setmargrefmarker{\ding{228}}`).

`\margrefemendation` `\margrefemendation` is a version of `\margreftextnote` for emendations.

`\margrefexplanatory` `\margrefexplanatory` is a version of `\margreftextnote` for explanatory notes.

11.8 Visual formatting

`\longpage` `\shortpage` The ideal in LaTeX is to separate content markup and formatting as completely as possible. In the text itself you simply mark the logical units (a poem, a title, a stanza, a textual note), and a header program like this interprets the meaning of those terms typographically. However, one still winds up having to do some visual formatting. Especially in prose sections, such as in the textual notes, you will sometimes have to enlarge or shrink a page by one line in order to avoid an intractable widow or club line, and when one does so one must be careful to do the same to the other member of the two-page spread of which that page is a part. These commands enable one to wiggle pagelength enough to handle these problems. Since verse sections should be set with `\raggedbottom` in effect, the commands are not necessary in verse sections.

12 Customizing page geometry

12.1 Font sizes, skips, sinks indents, and penalties

poemscol sets the font sizes, but not the font family, of fonts for many common tasks, in order to create a hierarchy of sizes. They can be changed with a `\renewcommand` of the following form, where the first number is the font size in points and the second is the leading:

```
\renewcommand{\volumetitlefont}{\fontsize{18}{23}\selectfont}
```

The following table gives the default values for the fonts. The titles of the fonts give their uses. Backmatterheaderfont is for setting the titles of back matter sections. Sequence section titles use poemtitlefont. Backmattertext font is `\small`, backmatterintrofont is `\normalsize`.

Font	**Size**	**Leading**
volumetitlefont	18	23
volumesubtitlefont	14	18
poemtitlefont	14	18
sequencetitlefont	16	20
subsectiontitlefont	12	14.5
backmatterheaderfont	16	20
backmattervolumefont	12	14.5
backmattervolumesubtitlefont	11	13
contentsvolumefont	13	20
contentsvolumesubtitlefont	12	14.5
contentssequencetitlefont	10	14
contentspoemtitlefont	10	14

The following table gives various skips, sinks, penalties, and indentations. The titles of these should be self-explanatory. These should be changed with `\renewcommand` as follows:
`\newcommand{\volumetitlesink}{\leavevmode\vspace{23.5pt}}`

Name	**Argument**	**Comment**
backmattersink	21.5pt	
backmatterafterheadersink	11pt	
afterpoemtitleskip	`\smallskip`	
afterpoemskip	`\bigskip`	
aftersequencetitleskip	`\medskip`	
stanzaskip	`\medskip`	Extra skip between stanzas
titleindent	`\titleindentamount` (1pc)	use `\setlength` to change
parindent	titleindentamount	
volumetitleshift	volumetitleshiftamount	use `\setlength` to change
contentsindentone	24pt	
contentsindenttwo	42pt	
contentsindentthree	60pt	
contentsindentfour	72pt	
runoverindentvalue	6em	for runover lines in verse
verseindent	2em	for formally indented lines
notesclubpoempenalty	-300	
notessequenceclubpenalty	-1500	
notesvolumetitlepenalty	-1500	

12.2 Page styles

poemscol also sets several default page styles, defined below. The main page style is “fancy” (defined below), used for prose frontmatter (such as introductions), the main body of the poetry, and the endnotes sections. “Volumefirststyle” is for the first page of a volume of poems. “Volumetitlestyle” is for the title page of a volume of poems. The external file for the table of contents redefines “fancy,” so you should issue `\resetpagestyle` after issuing `\putpoemcontents`.

12.3 Page geometry and type leading

You have to give geometry the values that suit you. I include the values I used in my own book, just to give you a starting place. Using these numbers out of the box might make your page look strange, since I also used the now obsolete cropmarks package, which defined a few other lengths and moved the page box around on the page. Also, the default values for marginparsep and marginparwidth are different now (18pt and 10pt respectively). With that warning, here is a list of the values I used:

tmargin	33pt
textwidth	318.21pt
textheight	550pt
marginparsep	7pt
marginparwidth	10pt
headheight	15pt
headsep	19pt
footskip	15pt
lmargin	132pt

You can reproduce this geometry by issuing the command `\tightgeometry`.

In my own book I also, strictly to save money, and at the price of some ugliness, changed the definition of `\normalsize` and `\small`, so that the former was 10 pt type on 11 pt leading (rather than 12 pt), and that the latter was 9 pt type on 10 pt leading. You can reproduce this setting by issuing `\tightleading`, although I would advise not doing so unless you are under financial constraints.

13 Prose Sections

poemscol uses the lineno package to make marginal line numbers for prose sections such as the Author's Introduction, Afterword, Author's Notes, or such prose interludes or prose poems the volume may include. To control the appearance, placement, and frequency of marginal line numbers in prose sections, you may wish to consult the manual to lineno. I have added a minimal user interface to make lineno cohere with poemscol. You may have to fiddle with the parameters of both poemscol and lineno to persuade them to set marginal line numbers that look the same way in prose and verse contexts. The default values should work as is.

Line numbers in prose sections are placed using a different `\length` from that used in verse sections. Essentially, verse sections, and prose sections which use paragraph and sentence numbering — see below, in section 14 — place the marginal line numbers using `\marginparsep`. Prose sections which are numbered by line use `\linenumbersep` instead, which is a length defined in the lineno package. I have given values for both lengths which seem to place line numbers on the left and on the right side in prose sections roughly where they are placed in verse sections. You can change these values with `\setcounter{countername}{length}`.

`\verselinenumberstoleft` and `\verselinenumberstoright` also place the line numbers to the left or to the right, respectively, in prose sections. If you wish the numbering to appear on a different side in prose contexts you can set the side by issuing `\leftlinenumbers` or `\rightlinenumbers`. These last are commands from the lineno package. The two forms in which line numbers switch sides in prose contexts can't at the moment be used, because of a clash between how lineno handles line numbers that switch sides and the way I pass the line numbers to notes sections. If you have set the verse line numbers to switch sides, prose line numbers will appear on the right side. I hope soon to find a workaround for this problem.

By default, the line modulo for prose sections is the same as that for verse sections; if you have a line number every 5 lines in verse, you will have the same in prose. You can set the prose line modulo independently with `\setprosemodulo`.

poemscol may deal with rather long stretches of prose, but it is not designed to be used for critical editions of extended prose texts, as EDMAC, ledmac, or ednotes are. The method of entering line ranges into notes in prose contexts is something of a hack, and rather a worse one than the one employed in verse contexts, since it involves using two labels for each note, one to capture the line number of the beginning, the other to capture the line number of the end. Other limitations of this method may show up with longer prose extracts.

`\prosesectiontitle`

Prose sections may be titled with `\prosesectiontitle{}` which is essentially just `\poemtitle`, or `\prosesectiontitlenotitle`, which is essentially just `\poemtitlenotitle`. (Remember that if you use the latter command, you must specify a "dummy" title to use in the note and contents sections.) They will be formatted in the notes and contents sections as poems are. Alternatively, you can use `\literalcontents` and `\literaltextnotes` to set such titles as you please.

`prosesection`

Prose sections should be enclosed in a `prosesection` environment (between `\begin{prosesection}` and `\end{prosesection}`). Entering this environment sets the modulo repeat for line numbering to the same value that prevails in verse contexts, and resets the prose line number to zero. If for some reason you do not wish to reset the line number but continue on from where it left off at the end of the last prose section, use the `prosesectionnoreset` environment instead.

`\setprosemodulo`

If you wish to set the modulo repeat for line numbering in prose to some value different for that from verse contexts, issue the new modulo as the argument to `\setprosemodulo{}`.

It is possible, if a prose section is interpolated in the midst of verse, to number the prose and verse continuously. To move from a verse passage to a prose passage, just before the `\end{poem}` at the end of the first verse passage issue `\saveverselinenumber` to save the number of the verse passage. After you issue `\begin{prosesection}` issue `\resetlinenumber[\theverselineumberscratch]` to apply that number to the prose section. To number continuously while moving from prose to verse is more tricky. One way might be to mark a label at the end of the prose passage with `\proselinelabel{prose end label}` (or some other suitably unique label). This label will record the line number of the last prose line. Pass that value to `\restoreverselinenumber` by issuing `\setcounterfromref{verselinenumberscratch}{prose end label}`. Then issue `\restoreverselinenumber` just after issuing the `\begin{poem}` that restarts the verse again. I would recommend against numbering verse and prose continuously, however, since a verse line and a prose line are different categories of thing, the one being an element of prosody, the other being only an element of typography.

`\proselinelabel`

`\proselinelabel` is simply `\linelabel` from lineno. I have provided the new name to make it easy to distinguish from `\poemlinelabel`.

`\prosetextnote`
`\prosemendation`
`\proseexplanatory`

Special versions of `\textnote`, `\emendation` and `\explanatory` are available for use in prose sections. `\prosetextnote`, `\proseemendation`, and `\proseexplanatory` differ from the verse versions in that they each require an extra argument, a unique label to be used in locating the line number for use by the notes commands. If `\maketextnotes`, `\makeemendations`, or `\makeexplanatorynotes` have been issued, the endnotes will be sent to the usual sections, where all the notes from that prose section will be grouped into a paragraph, with the entry headed by the title given by `\prosesectiontitle{}`, just as if that were the title of a poem. `\textnotesatfoot`, `\emendationsatfoot`, and `\explanationsatfoot` will send the notes from prose contexts to the usual footnote levels. Notes commands in prose contexts take three arguments. As in verse sections, in prose sections textnotes, explanatory notes, and emendations have an optional first argument for recording line ranges. As in verse sections, you can record these line ranges either by simply putting in the argument how many lines the lemma will refer to (a deprecated method retained for backward compatibility reasons), or the argument of a `\proselinelabel` command on the last line of the lemma. (The only difference from the verse version here is that you should make the label using the `\proselinelabel` command, rather than with the `\poemlinelabel` command.) The second argument is a label to identify each note uniquely, which is used to find the line number at which the note begins. You will need a different label for every note. The third argument is the text of the note, which should include both the lemma and the rest of the note, as in the notes for verse sections. `\proseemendation[]{}{}` and `\proseexplanatory[]{}{}` are coded the same way. So, if you wished to make a textual note on a passage that begins with “Call me Ishmael.” and ends “a way of driving away the spleen,” you would issue at the point where the note is to occur (at the beginning of the passage):

```
\prosetextnote[a way of driving away the spleen] {Call me Ishmael}
{Call me Ishmael \ldots\ a way of driving away the spleen] text of
textual note}
```

, and on last line of the passage to which the note refers you would issue:
`\proselinelabel{a way of driving away the spleen}`.

`\prosetsvariant`
`\proseaccidental`
`\prosetsaccidental`

You can issue `\sources` and `\tsentry` in prose contexts as usual. For typescript variants and accidental variants you need to issue `\prosetsvariant[]{}{}`, `\proseaccidental{}{}`, and `\prosetsaccidental{}{}`. `\prosetsvariant` is entered just as `\prosetextnote[]{}{}` is, with an optional argument (for use with line ranges, as usual), and two mandatory arguments (the first a label to capture the line number, the second the lemma and comment).

`\proseaccidental` and `\prosetsaccidental` take only the two mandatory arguments, since by definition an accidental variant is confined to one line.

One caveat: because the line numbers for notes in prose contexts are generated by a label/ref mechanism, each note will result in a complaint about an undefined reference the first time the file is run through LaTeX. Either run the first pass in quiet mode, or be prepared to ignore a lot of griping until LaTeX has generated the .aux file at the end of the first run.

14 Paragraph and Sentence Annotation

Annotating a prose section by sentence number, or by paragraph number and sentence number, has some advantages over annotating it by line number. The principal advantage is that the annotation and reference system can remain stable over multiple editions, even if they are set in different formats. The most familiar of such systems is the chapter and verse system in the Bible, introduced by Archbishop Stephen Langton in the thirteenth century (who divided the Vulgate edition into numbered chapters) and by Robert Estienne in the sixteenth century (who added the verse numbers). Such a form of annotation might be especially useful for medieval prosimetra, a form combining verse and prose. (Hence the commands that follow often begin with `pm`.)

`pmsection` You should put prose marked up for sentence and paragraph annotation in a `pmsection` environment. You do not need to put prose marked up for sentence and paragraph annotation in a `prosesection` environment. That environment is only needed if you are line-numbering prose. You also do not need to load the lineno package if you are using the sentence and paragraph system for annotating prose. You might, however, still wish to set the title with `\prosesectiontitle`.

`\setprosebysentence` `\setprosebysentence` sets poemscol to number prose by sentence rather than by line number. This command is mostly a placeholder now, used only for setting the value of `\marginparsep` to something better for prose. (`\marginparsep` sets how far the line numbers are from the text block.) `\verselinenumberstoleft`, `\verselinenumberstoright`, and `\verselinenumbersswitch` each pick convenient values for `\marginparsep`. By default these values all line up on the page, but if you change any of them, you may have to use reset `\marginparsep` to some value you will have to find by trial and error, since they each use `\marginparsep` in slightly different ways. You can change `\marginparsep` either by using `\setlength{\marginparsep}{length}` or, better, by using `\renewcommand` to change the setting of `\marginparsep` in `\verselinenumberstoleft` or whatever else you intend to use.

`\pmsentence` Each sentence must begin with `\pmsentence`. `\pmsentence` increments the sentence counter, optionally puts the sentence number in the running text, and optionally puts the sentence number in the margin.

`\runningsentencenumberstrue` sets sentence numbers in the running text. To set sentence numbers in the margin, issue `\marginsentencenumberstrue`. You can issue both, either, or neither of these commands.

`\setpmmodulo`

If you are putting sentence numbers in the margin, you need not number every sentence. Indeed, if there is more than one sentence on a line, the numbers will overwrite each other (unless you use the marn.sty package, and substitute `\marn` for `\pmclrightsidepar`, `\pmclleftsidepar` or `\pmclsidepar` in the code for `\pmsentence`). You can also avoid overlapping sentence numbers by using `\pmsentencetwo` and its siblings, defined below. You can set the modulo for sentence marginal numbers by issuing `\setpmmodulo{}`.

`\pmsentencetwo`
`\pmsentencethree`
`\pmsentencefour`

If you know, from a prior run, that there will be a second or third sentence on some particular line, marking them with these commmands will ensure that the marginal sentence numbers do not overwrite each other. These macros are something of a kludge, but they will do until I figure out how to do this better.

`\pmnumberstoright`
`\pmnumberstoleft`
`\pmnumbersgutter`
`\pmnumbersouter`

If you wish to control the margin in which sentence and paragraph numbers appear, issue `\pmnumbersright`, `\pmnumbersleft`, `\pmnumbersgutter`, or `\pmnumbersouter`. By default, these numbers will line up with the line numbers in verse sections. If you change the value of `\marginparsep` for prose sections from those used by their verse equivalents, it is a good idea to issue the verse equivalent command (`\versenumberstoright` or whatever) again when you move from a prose to a verse environment.

`\runningsentencenumberformat` sets the format of running sentence numbers. Currently it is `\textsuperscript` and `\tiny`. You can change this with `\renewcommand`. The format of marginal sentence numers is set by `\marginsentencenumberformat`. Currently it is `\tiny`. You can change this with `\renewcommmand`.

`\pmpara`
`\pmsentencebypara`
`\pmsparainmartrue`
`\pmpararunningtrue`

You can also number paragraphs. Each paragraph must begin with `\pmpara`. `\pmpara` increments the paragraph counter, optionally puts the paragraph number in the running text, and optionally puts the paragraph number in the margin. To reset the sentence counter at the beginning of each paragraph, issue `\pmsentencebyparatrue` in your preamble. To set paragraph numbers in the margin, issue `\pmparainmartrue`. To set paragraph numbers in the running text, issue `\pmpararunningtrue`. You can issue both, either, or neither of these last two commands. Which margin these numbers are set in is controlled by the same commands that control placement of marginal sentence numbers.

If you begin a paragraph with `\pmpara` poemscol will assume that you intend to include the paragraph number in your notes (in the form *paragraph: sentence:* for simple references, or *firstparagraph: firstsentence: – ;astparagraph: lastentence:* for ranges of sentences) even if you do not reset the sentence number at the beginning of each paragraph. Normally you would not issue `\pmpara` if you are not resetting the sentence number at the beginning of each paragraph, since under those circumstances you don't really need to keep track of the paragraph numbers, in which case each note would simply include the sentence number (just as in verse it would include the line number). In the unusual event in which you wish to put the paragraph number in the margin but refer only to sentence numbers in the notes, simply follow every use of `\pmpara` with `\pmparasfalse`.

`\pmpararunningformat`
`\pmparmarformat`

The format of running paragraph numbers is set by `\pmparrunningformat`. Currently it is `\textbf` and `\small`. The format of marginal paragraph numbers is set by \pmparmarformat. Currently it is \textbf and \small. Either can be changed with \renewcommand.

14.1 Cross Reference by Sentence and Paragraph

\sentencelabel

To make labels that refer to sentences or paragraphs, issue \sentencelabel{}. The label will be either both the paragraph number and the sentence number (if you are resetting sentence numbers by paragraph), or only the sentence number (if you are not resetting sentence numbers by paragraph).

14.2 End notes by Sentence and Paragraph

\pmtextnote
\pmexplanatory
\pmemendation
\pmaccidental
\pmtsvariant
\pmtsaccidental

\pmtextnote, \pmexplanatory, \pmemendation, \pmaccidental, \pmtsvariant and \pmtsaccidental are special versions of the usual endnote commands for use with annotations by sentence and paragraph. If you have not issued \pmsentencebypara these commands will write the sentence number at the head of each note, where the line number would usually go. If you have issued \pmsentencebypara, these commands will issue both the paragraph and sentence number, in the format <paragraph number>:<sentence number>. You can refer to ranges of sentences (or paragraphs and sentences) in a way very similar to how you do so in verse sections. At the end of the range, issue a \sentencelabel command, with a unique label. At the beginning of the range, issue your \pmtextnote command with the optional argument in square brackets before the main argument, giving your unique label as the value of that argument, e.g. \pmtextnote[some label]{text of lemma] comment}. At the end of your range, issue \sentencelabel{some label}, using the same label you put in the square brackets for the \pmtextnote. If you have not issued \pmsentencebypara, the range will simply be the range of sentence numbers. If you have issued \pmsentencebypara, the range will include paragraph and sentence numbers, like this: 4:12–6:13.

\sources needs no special handling if you are annotating by sentence and paragraph.

14.3 Footnotes by Sentence and Paragraph

\textnoteatfoot shifts \pmtextnotes to the foot, as with \prosetextnote. \emendationsatfoot and \explanationsatfoot work the same way.

14.4 Bible Chapter and Verse

Since \pmsentence and \pmpar are the moral equivalents of bible verse and chapter, \biblechapter and \bibleverse are provided for convenience.

15 Parallel Text Editions

poemscol does not make parallel texts in which the texts are set in parallel on the same page, but it can help one make parallel text editions in which the contrasting versions are set on facing verso and recto pages. poemscol does not provide the page breaks for parallel editions, and even if it did, one would still probably have to specify page breaks every two or three pages to keep the texts in register. If you are willing to set the page breaks yourself, however, it does provide macros for keeping the line count correct on successive recto or verso pages, and for keeping track of whether or not one was in the middle of a stanza when one closed the previous page on a particular side of the book. And it provides, in the `\synchrolabel` and `\synchroref` commands, a way of marking which lines on the recto side correspond to particular lines on the verso side. `\synchroref` will even tag the marginal line number it makes with a marker to specify the text to which it refers.

`\startparalleltexts`
`\startrectopage`
`\finishrectopage`
`\startversopage`
`\finishversopage`

You must begin a region of parallel text on a *verso* page. Setting parallel texts requires one to redefine the poem and stanza environments. A parallel text region must begin with `\startparalleltexts` to redefine those environments. Verso pages should begin with `\startversopage` and end with `\finishversopage`. Recto pages should begin with `\startrectopage` and end with `\finishrectopage`. You have to declare the page breaks manually, so be sure you are putting the two `\finish` commands in places which will work.

Although usually with poemscol it is up to you where to put the marginal line numbers, when you issue `\startparalleltexts` the positions are fixed as follows: marginal line numbers are in the left margin, line cross references (marking which lines in the recto text correspond to which lines in the verso text) are in the right margin.

`\synchrolabel`
`\synchroref`
`\setsynchroflag`

To mark these cross references, you need to make a label on the verso side, and a corresponding reference on the recto side. You may also wish to add a mark to that reference to specify what edition it refers to. So, for instance, if your verso text is the 1805 version of Wordsworth's *The Prelude*, and your recto text is the 1850 version, you may wish to specify `\setsynchroflag{1850: }`, which will preface the references put out by the `\synchroref` command with '1850: '. Suppose you wished to tie the line "Oh, what can ail thee, man at arms?" on the verso side to the line "Oh, what can ail thee, wretched wight?" on the recto. To make the label, issue the `\synchrolabel` command on the verso side just after the `\verseline` or `\end{stanza}` which closes the line you wish to label:

```
Oh, what can ail thee, man at arms?\verseline
\synchrolabel{Oh, what can ail thee, man at arms?}
```

You don't have to quote the whole line in the label, but it should be long enough to uniquely pick out that line in the poem. (If the line is repeated in the poem, you will have to exercise your ingenuity in making up a unique label.)

Over on the recto side, you will want to use `\synchroref` in order to print the marginal cross reference. The `\synchroref` command should come *before* the `\verseline` that closes the line it is tied to, as follows:

```
Oh, what can ail thee, wretched wight?
\synchroref{Oh what can ail thee, knight at arms?}\verseline
```

16 Using poemscol in a Multilingual World

\changecontentsname

By default, poemscol uses English language titles for such things as the Table of Contents, the Explanatory Notes, and so on, and uses English language running headers. As of version 2.5 these are easily changed. To change the name of the Table of Contents (the default is CONTENTS) to Inhaltsverzeichnis, issue `\changecontentsname{Inhaltsverzeichnis}`. To change the running header on contents pages, issue `\changecontentsheader{your header}`.

\changenotesname

To change the name for the endnote section for textual notes (the default is TEXTUAL NOTES), issue `\changenotesname{your version}`. To change the running header (by default "Textual Notes to") issue `\changenotesheader{your version}`. To change the name for the textual notes in the table of contents issue `\changetextnotescontentsname{your version}`.

\changesinglepageabbrev
\changemultiplepageabbrev

Running headers refer to single pages with the English abbreviation "p." and to multiple pages with the English abbreviation "pp." To change these issue `\changesinglepageabbrev{new abbrev}` or `\changemultiplepageabbrev{new abbrev}` respectively. The ultimate outcome is to translate a header of the form "Textual Notes to pp. xx-yy" to the form appropriate for your language.

\changeemendationsname

To change the name for the endnote section for emendations (the default is EMENDATIONS), issue `\changeemendationsname{your version}`. To change the running header (by default "Emendations to") issue `\changeemendationsheader{your version}`. To change the name for the emendations in the table of contents issue `\changeemendationscontentsname{your version}`.

\changeexplanationsname

To change the name for the endnote section for explanatory notes (the default is EXPLANATORY NOTES), issue `\changeexplanationsname{your version}`. To change the running header (by default "Explanatory Notes to") issue `\changeexplanationsheader{your version}`. To change the name for the explanatory notes in the table of contents issue `\changeexplanationscontentsname{your version}`.

\changepoemindexname

To change the name for the index (the default is "Index of Titles and First Lines", issue `\changepoemindexname{your version}`. To change the running header (the default is "Index of Titles and First Lines") issue `\changepoemindexheader{your version}`. To change the contents entry for the index (the default is "Index of Titles and First Lines") issue `\changepoemindexcontentsname{your version}`.

17 A sample driver file

If you are compiling a collected poems, you are probably preparing your volume from many files. You may (or may not) wish to subdivide your edition by the original volumes. Or you may subdivide by year, or by the significant phases of the poet's career. One way to keep in control of the process is to create a driver file, which calls all the appropriate .sty files and sets the variables globally, and then read in the sections of the edition using `\input`. Here is a version of the driver file I used for *The Collected Poems of Robert Penn Warren*:

```
\documentclass[10pt,twoside]{article}
\usepackage{fancyhdr,kdgreek,makeidx,times,multicol,geometry,}
\usepackage{keyval,ifthen,mparhack,manyfoot,poemscol}
\begin{document}
\leftheader{The Collected Poems of Robert Penn Warren}
\makeexplanatorynotes
\makeemendations
\maketextnotes
\makepoemcontents
\makeverselinenumbers
\global\indexingontrue
\global\includeaccidentalstrue
\global\includetypescriptstrue
\input maintitle.tex
\input poems1922_1943.tex
\input thirtysix.tex
\input eleven.tex
\input sp43.tex
\input promises1.tex
\input promises2.tex
\input promises3.tex
\input yeo1.tex.tex
\input yeo2.tex.tex
\input yeo3.tex.tex
\input tale1.tex
\input tale2.tex
\input incarnations.tex
\input audubon.tex
\input orelse1.tex
\input orelse2.tex
\input arcturus.tex
\input nowthen.tex
\input beinghere.tex
\input rumor1.tex
\input rumor2.tex
\input cj1.tex.tex
\input cj2.tex.tex
\input altitudes.tex.tex
\input poems1943_1989.tex
\input introduction.tex
```

```
\finish
```

18 A sample poem markup

Here is a poem Robert Penn Warren wrote in his first volume, *Thirty-Six Poems* (1935). The poem is the first poem of his sequence “Kentucky Mountain Farm.” I include the over-title, to demonstrate how to mark up such cases.

```
 \sequencetitle{Kentucky Mountain Farm}
\index{Kentucky Mountain Farm@{\em Kentucky Mountain Farm\/}}
\sources{Text: TSP. Variants: SP43, SP66 (Deletes ``The Cardinal,'' ``The
Jay,'' and ``Watershed''), SP75 (Same sections as SP66), SP85 (Restores
``Watershed''), {\em Helsinki\/} (includes only ``Rebuke of
the Rocks'' and
``At the Hour of the Breaking of the Rocks'').
``The Owl'' (above) was marked as a section of ``Kentucky
Mountain Farm'' when it first appeared in {\em Poetry,\/} but it was never
included in any book version of the entire sequence. The sequence in {\em
Poetry\/} included, in this order, ``The Owl,'' ``The Cardinal,'' and
``Watershed.'' TSP uses lower case Roman numerals in the section
titles. The typescript drafts in the Beinecke Library do not seem to
be setting copies.}
\sequencefirstsectiontitle{I. Rebuke of the Rocks}
\index{Rebuke of the Rocks@{\em Rebuke of the Rocks\/}}
\sources{Text: TSP. Variants: {\em Nation\/}, 11 Jan.\ 1928,
p.~47, {\em Literary
Digest,\/} 28 Jan.\ 1928, p.~32, {\em Vanderbilt Masquerader,\/} 10
(Dec.\ 1933), p.~16, SP43, SP66, SP75, SP85, {\em Helsinki\/},
Broadside: The Press at Colorado College, printed on paper handmade by
Thomas Leech for the American Poetry Society, April~26, 1985. This
poem was not included in SP85 until the second set of galleys, in
which a photocopy of the SP75 text is a stapled insert.}
\begin{poem}
\begin{stanza}
Now on you is the hungry equinox,\verseline
\index{Now on you is the hungry equinox}
O little stubborn people of the hill,\verseline
\accidental{hill,] \sameword--- {\em Nation,\/} {\em Literary Digest\/}
\sameword, {\em Vanderbilt\/} (I include the reading
from {\em Vanderbilt\/}
even though it is the same as in TSP, because
{\em Vanderbilt\/} was published
after the other magazine versions but before TSP.)}
The season of the obscene moon whose pull\verseline
Disturbs the sod, the rabbit, the lank fox,\verseline
Moving the waters, the boar's dull blood,\verseline
And the acrid sap of the ironwood.\end{stanza}
```

```
\begin{stanza}
But breed no tender thing among the rocks.\verseline
Rocks are too old under the mad moon,\verseline
\accidental{old\missingpunct] \sameword, {\em Vanderbilt\/}}
Renouncing passion by the strength that locks\verseline
The eternal agony of fire in stone.\end{stanza}

\begin{stanza}
Then quit yourselves as stone and cease\verseline
\accidental{stone\missingpunct] \sameword, {\em Vanderbilt\/}}
To break the weary stubble-field for seed;\verseline
Let not the naked cattle bear increase,\verseline
Let barley wither and the bright milkweed.\verseline
\accidental{milkweed.] milk-weed. {\em Vanderbilt\/}}
Instruct the heart, lean men, of a rocky place\verseline
That even the little flesh and fevered bone\verseline
May keep the sweet sterility of stone.\end{stanza}
\end{poem}
```

19 A sample divider page

```
\begin{volumetitlepage}

\volumetitle{Being Here}
\volumeheader{Being Here}
\volumesubtitle{Poetry 1977--1980}
\volumededication{To Gabriel Thomas Penn\\
(1836--1920)}
\bigskip

\volumeepigraph{OLD MAN: You get old and you can't do anybody
any good any more.\\
BOY: You do me some good, Grandpa. You
tell me things.}


\bigskip

\volumeepigraph{There is in short no absolute time standard.}
\volumeattribution{{\em Van Nostrand's Scientific Encyclopedia,\/}\\
Fifth Edition, p. 2203}

\bigskip


\volumeepigraph{I thirst to know the power and nature of Time\ldots.}
\volumeattribution{St.\ Augustine: {\em Confessions,\/}\\
Book {\em XI,\/} Chapter {\em XXIII\/}\\
Translated by Albert C. Outler}


\bigskip

\volumeepigraph{Time is the dimension in which God
strives to define His own Being.}

\sources{The typescript material in the Beinecke Library
indicates that this
book went through many drafts, and some rethinking about what to
include and how to order the poems. Until a late stage of
production---very late, since the typescript includes a designed
mock-up of the proposed title page---the title of this volume was {\em
Life is a Fable.\/} Indeed, as late as the time the book was set in
galleys it was called {\em Life is a Fable\/}, although the title was
corrected on the galleys. Like {\em Now and Then\/}, the typescript
was assembled from photocopies of re-typed copies of poems made after
their magazine appearance, with annotations giving the name of the
magazine in which they occurred. Unlike {\em Now and Then\/}, the
typescript seems to have been typed in one piece, and the annotations
added for the convenience of the permissions department at
Random House. The page numbers have been added---presumably
later---in blue pen. The copy is marked up in Warren's hand in black
pencil, and in another hand in red pencil. (Unless marked otherwise,
all of the mark-up described here is in black pencil.) \tsentry{Like most
Warren typescripts BHTS is single-spaced. The dedication to Gabriel
Thomas Penn and the dialogue between the old man and the boy appeared
on a separate page, and the three epigraphs appeared on another
separate page. BHTS gives 1838 as the year of Gabriel
Thomas Penn's
birth (the error is corrected on the galleys). In the quotation from
Augustine, the second word is corrected to ``thirst'' from ``yearn,''
and the translator's name is added in pencil, not in Warren's
handwriting.} In one of Warren's own
copies at Western Kentucky University, he has corrected the date of
Gabriel Thomas Penn's birth to 1837 (copy 310). Some revisions in this book
may have come very late
indeed. A letter to Erskine of January 28, 1980, refers to changes
made in a bound manuscript that Warren kept after returning the
copy-edited typescripts. There are revisions not only on the galleys,
but also on the page proofs. There are even some corrections on the
``blues'' for this volume, including some new copy on a card referred
to in a note on the cover of the ``blues'' but now lost. The Secker
and Warburg London edition is, except for the title page, printed from
the American plates. There is a copy of this book, marked up by
Warren with
proposed revisions for Stuart Wright's use on May 25, 1987, in the
Special Collections
```

```
Department of the Emory University Library. I quote Warren's
notes by permission.
Warren has also marked up, slightly, a copy of this book now in the
Robert Penn Warren Room of the Library of the Kentucky Museum at
Western Kentucky University. I quote that volume too by permission.
The setting typescript for the section of SP85 drawn from this volume
consisted of taped-in photocopies of passages from BH. Because the
revisions for SP85 are sometimes a vexed question, I have indicated
where the SP85 revisions are made on SP85TS.}


\sources{Title Page] SP85 omits the dialogue between the old man and
the young boy. SP85 capitalizes ``time standard'' in the quotation
from Van Nostrand's Scientific Encyclopedia, and does not capitalize
``Time'' in the quotation from Augustine.}



\end{volumetitlepage}
```

20 Implementation

20.1 Page geometry and crop marks

`\tightgeometry`

There's no reason why you should use the page geometry I specify here. If you issue `\tightgeometry` in your preamble, you will have the geometry I have used. If you issue `\tightleading` you will redefine `\normalsize` and `\small` to use tighter leading. Normal 10 point LaTeX uses 12 point leading. Strictly to save money, I redefined `\normalsize` to use 11 point leading. I use the geometry package, since it's simpler to figure out. I give my page geometry here chiefly to remind the reader of some important variables that should be kept track of, such as marginparsep, which controls the separation of the marginal line numbers from the verse line, marginparwidth, which controls how wide the block for the marginal line numbers will be, headsep, which controls how far the text will be from the running headers, and footskip, which controls how far the last verse line will be from the marker which tells the reader that there is a stanza break at the end of the page (if there is one). I've also set the odd side margin and the even side margin for two-sided printing. Finally, I have set the default handling of page endings to `\raggedbottom`. Verse should always be set with a ragged bottom margin, otherwise LaTeX will stretch the spaces between stanzas to make the bottom perfectly flush. When you enter prose sections, such as the editor's introduction, you should enter the command `\flushbottom` to make sure that the prose sections have flush page bottoms.

```
\newcommand{\tightgeometry}{\geometry{verbose, tmargin= 33pt,
textwidth=318.21pt,textheight=550pt,
marginparsep=7pt,marginparwidth=10pt,
 headheight= 15pt,
 headsep=19pt,
 footskip=15pt,
 lmargin=132pt}
\setlength{\oddsidemargin}{60pt}
\setlength{\evensidemargin}{65.41pt}}
\newcommand{\tightleading}{%
\makeatletter
\renewcommand\normalsize{%
   \@setfontsize\normalsize\@xpt\@xipt
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
   \belowdisplayskip \abovedisplayskip
   \let\@listi\@listI}
\makeatother
\normalsize
\makeatletter
\renewcommand\small{%
   \@setfontsize\small\@ixpt{10}%
   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus2\p@
   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 4\p@ \@plus2\p@ \@minus2\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
   \belowdisplayskip \abovedisplayskip
}
\makeatother}


```

20.1.1 Internal font size commands

You will probably never need to use these font commands explicitly. I defined these for internal use within poemscol, so that a command will always call the same fonts.

As a way of making global changes to repeated features I have defined various special fonts for special purposes. For instance, in a Collected Poems in which the poems are arranged by volume, with divider pages for the beginning of each volume, the first command sets the font for the volume title on the divider page, using the default font family, setting it in 18 point type on 23 point leading. The second command, as the name of the command indicates, sets the font for the subtitle of the volume to be 14 point type on 18 point leading. You can change these either by changing the values in your (renamed) style file, or by using `\renewcommand`. For instance, you can change the font parameters for the font for setting poem titles from 14 on 18 to 12 on 14 by issuing `\renewcommand{\poemtitlefont}{\fontsize{12}{14}\selectfont}` in your preamble. Notice that there are special fonts for titles in the back matter and in the table of contents as well.

```
\newcommand{\volumetitlefont}{\fontsize{18}{23}\selectfont}
\newcommand{\volumesubtitlefont}{\fontsize{14}{18}\selectfont}
\newcommand{\poemtitlefont}{\fontsize{14}{18}\selectfont}
\newcommand{\sequencetitlefont}{\fontsize{16}{20}\selectfont}
\newcommand{\subsectiontitlefont}{\fontsize{12}{14.5}\selectfont}
\newcommand{\backmatterheaderfont}{\fontsize{16}{20}\selectfont}
\newcommand{\volumetitlesink}{\leavevmode\vspace{23.5pt}}
\newcommand{\backmattersink}{\leavevmode\vspace{21.5pt}}
\newcommand{\backmatterafterheadersink}{\leavevmode\vspace{11pt}}
\newcommand{\backmattertextfont}{\small}
\newcommand{\backmatterintrofont}{\normalsize}
\newcommand{\backmattervolumefont}{\fontsize{12}{14.5}\selectfont}
\newcommand{\backmattervolumesubtitlefont}{\fontsize{11}{13}\selectfont}
\newcommand{\contentsvolumefont}{\fontsize{14}{20}\selectfont}
\newcommand{\contentsvolumesubtitlefont}{\fontsize{12}{14.5}\selectfont}
\newcommand{\contentssequencetitlefont}{\fontsize{10}{14}\selectfont}
\newcommand{\contentspoemtitlefont}{\fontsize{10}{14}\selectfont}
```

20.2 Miscellaneous dimensions and constants

poemscol sets relatively large penalties for widows and clubs, both in the main text, and in the notes sections. There are also special penalties after volume titles in the notes sections, since one does not want to leave a volume title in the notes sections stranded at the bottom of a page. (Setting penalties may not automatically solve all the problems, however. You can fix an intractable widow problem in the contents or in the notes sections by issuing `\literalcontents{\pagebreak}` or `\literalemend{\pagebreak}` just before you issue the `\poemtitle{}` or `\volumetitle{}` that resulted in the problematic title line.) poemscol also manipulates penalties for places like the space between a poem's title and its first line. It defines special values for the white space before and after poems and for the amount of a stanza break. `\contentsindentone`, `\contentsindenttwo`, and so on are amounts to indent different hierarchical levels, say of poetic sequences and subsequences, in the table of contents.

```
\widowpenalty=300
\clubpenalty=300
\newcommand{\notespoemclubpenalty}{\penalty-300}
\newcommand{\notessequenceclubpenalty}{\penalty-1500}
\newcommand{\notesvolumetitlepenalty}{\penalty-1500}
```

Since the right margin of verse is ragged anyway, I allowed some flexibility for slightly longer lines.

```
\hfuzz=2pt
```

For explicitly declared indented lines:

```
\newcommand{\verseindent}{\hspace{2em}}
```

A hierarchy of indentations for sequence section titles and subsection titles.

```
\newdimen\titleindentamount \titleindentamount=1pc
\newdimen\titleindenttwoamount \titleindenttwoamount=2pc
\newdimen\titleindentthreeamount \titleindentthreeamount=3pc
\newcommand{\titleindent}{\hspace{\titleindentamount}}
\setlength{\parindent}{\titleindentamount}
```

White space after poem titles, after poems, and after sequence titles.

```
\newcommand{\afterpoemtitleskip}{\smallskip}
\newcommand{\afterpoemskip}{\bigskip}
\newcommand{\aftersequencetitleskip}{\medskip}
```

A little extra white space between stanzas.

```
\newcommand{\stanzaskip}{\medskip}
```

A hierarchy of indentations for the table of contents:

```
\newdimen\volumetitleshiftamount \volumetitleshiftamount=1pc
\newcommand{\volumetitleshift}{\hbox{\hspace{\volumetitleshiftamount}}}
\newdimen\voladditionalamount \voladditionalamount=2pc
\newdimen\contentsindentoneamount \contentsindentoneamount=24pt
\newcommand{\contentsindentone}{\hspace{\contentsindentoneamount}}
\newdimen\contentsindenttwoamount \contentsindenttwoamount=42pt
\newcommand{\contentsindenttwo}{\hspace{\contentsindenttwoamount}}
\newdimen\contentsindentthreeamount \contentsindentthreeamount=60pt
\newcommand{\contentsindentthree}{\hspace{\contentsindentthreeamount}}
\newdimen\contentsindentfouramount \contentsindentfouramount=72pt
\newcommand{\contentsindentfour}{\hspace{\contentsindentfouramount}}
\newdimen\contentsindentfiveamount \contentsindentfiveamount=80pt
\newcommand{\contentsindentfive}{\hspace{\contentsindentfiveamount}}
```

Penalties to encourage page breaks before poem titles, before multi-line poem titles, and before sequence titles.

```
\newcommand{\poemtitlepenalty}{\penalty-1000}
\newcommand{\multilinetitlepenalty}{\penalty-3000}
\newcommand{\sequencetitlepenalty}{\penalty-3000}
\newcommand{\multilinesequencepenalty}{\penalty-3500}
```

20.3 Verse

The verse environment here is taken from LaTeX, slightly modified to indent run over lines more. The idea is to distinguish the automatic indentation of run over lines from the explicitly declared indentation of indented lines. You can change the amount of indentation of runover lines with `\runoverindent{}`.

```
\def\makeatletter{\catcode`\@=11}
\def\makeatother{\catcode`\@=12 }
\newcommand{\runoverindent}[1]{\global\edef\runoverindentvalue{#1}
}
\runoverindent{6em}
% pmclverse (runover indented  more than it is in the normal verse
% environment)
```

```
\newenvironment{pmclverse}{%
\let\\=\@centercr%
\list{}{\itemsep\z@ \itemindent -\runoverindentvalue\listparindent \itemindent
          \rightmargin 1sp\leftmargin 1sp\advance\leftmargin
          \runoverindentvalue}\item[]}
{\endlist}
% rfoverse (flushes runover to right) (Just a placeholder now.)
\newif\ifrfv
\rfvfalse
\newcommand{\rightflushrunoververse}{\global\rfvtrue}
\newcommand{\indentrunoververse}{\global\rfvfalse}
\newenvironment{rfoverse}{\begin{pmclverse}}{\end{pmclverse}}
% verse environment with a switch
\renewenvironment{verse}{\ifrfv\begin{rfoverse}\else\begin{pmclverse}\fi}
{\ifrfv\end{rfoverse}\else\end{pmclverse}\fi}
```

20.4 Miscellaneous internal counters

`\verselinenumber` is the line number, obviously. `\printlineindex` records how many lines have passed since the last marginal line number was output. `\lineindexrepeat` governs how often the marginal line numbers appear. `\lemmalines` and `\lemmaend` are for setting number ranges for multiline lemmas.

```
\newcounter{verselinenumber}
\newcounter{printlineindex}
\newcounter{verselinenumberscratch}
\newcounter{printlineindexscratch}
\newcounter{lineindexrepeat} \setcounter{lineindexrepeat}{9}
\newcounter{lemmalines}
\setcounter{lemmalines}{0}
\newcounter{lemmaend}
\setcounter{lemmaend}{0}
```

This next counter is used to make labels for each poem for the contents and notes sections.

```
\newcounter{poemnumber}
\setcounter{poemnumber}{1}
```

These next two are used to set the page numbers in the running headers of the various notes sections, which are of the form "Textual notes to pp. xx–yy."

```
\newcounter{notepageholdertitle} \setcounter{notepageholdertitle}{1}
\newcounter{notepageholdernote}  \setcounter{notepageholdernote}{1}
```

20.5 Miscellaneous token lists

Since the explanatory notes section and the emendations section should only open an entry for poems for which there actually are explanatory notes or emendations, the current are held in token lists in case they are needed.`\titlesofar` and `\titleincrement` are used to concatenate multi-line titles.

```
\newtoks{\fulltitleholder}
```

```
\newtoks{\titlesofar}
\newtoks{\titleincrement}
```

20.6 Miscellaneous booleans

Most of these have self-explanatory names.

```
\newif\ifinstanza
\newif\ifinpoem
\newif\ifpoemcontentson
\newif\iftextnoteson
\newif\ifverselinenumbers
\newif\ifexplanon
\newif\ifemendationson
 \global\verselinenumberstrue
\newif\ifindexingon
\global\indexingontrue
\ifindexingon
  \makeindex
\fi
\newif\ifincludeaccidentals
\global\includeaccidentalstrue
\newif\ifincludetypescripts
\global\includetypescriptstrue
\newif\ifredundantemendations
\global\redundantemendationsfalse
\newif\ifnoemendyet
\newif\ifnoexplainyet
\global\noemendyettrue
\global\noexplainyettrue
\newif{\ifmiddlecontentsline}
\middlecontentslinefalse
\newif{\iflastcontentsline}
\lastcontentslinefalse
\newif{\ifputpagenumberincontents}
\putpagenumberincontentstrue
\newif{\ifputpagenumberinnotes}
\putpagenumberinnotestrue
\newif{\ifsinglelinetitle}
\singlelinetitletrue
\newif{\iftitlefirstline}
\titlefirstlinefalse
\newif{\iftitlemiddleline}
\titlemiddlelinefalse
\newif{\iftitlelastline}
\titlelastlinefalse
\newif{\ifverserightflush}
\verserightflushfalse
\newif\ifrangelemma
\rangelemmafalse
```

```
\newcommand{\makeverselinenumbers}{\global\verselinenumberstrue}
\let\makelinenumbers=\makeverselinenumbers% for compatibility
\newif\iftextnotesatend
\textnotesatendtrue
\newif\ifemendationsatend
\emendationsatendtrue
\newif\ifexplanatend
\explanatendtrue
\newif\iftextfootnotespara
\textfootnotesparatrue
\newif\ifsourcesfootnotespara
\sourcesfootnotesparafalse
\newif\ifemendationfootnotespara
\emendationfootnotesparatrue
\newif\ifexplanfootnotespara
\explanfootnotesparafalse
```

20.7 Page styles

These define several special page styles. To use them you must have the fancyhdr package installed. First, the default running headers: the right side (`\volumeheader`) is the title of the volume in a multivolume edition, and the left (`\leftheader`) is the title of the general collection.

\volumeheader
\leftheader

```
\newcommand{\volumeheader}[1]{\global\edef\volumeheadervalue{#1}
}
\newcommand{\leftheader}[1]{\global\edef\leftheadervalue{#1}
}
\leftheader{\relax}
\volumeheader{\relax}
\newcommand{\rightheader}[1]{\volumeheader{#1}}
```

The default page style. Note the use of the mark mechanism to keep track of cases where the stanza breaks and page breaks overlap.

\pagestylefancy

```
\newlength{\headoffsetlength}
\setlength{\headoffsetlength}{-1sp}
%\addtolength{\headoffsetlength}{\parindent}
\raggedbottom
\pagestyle{fancy}
\fancyhead{}
\fancyfoot{}
\fancyhfoffset[RO,LE]{\headoffsetlength}
\fancyhead[RO]{{\small\textrm{\thepage}}}
\fancyhead[LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{\hfil{\small{\em \volumeheadervalue}}}
\fancyhead[CE]{{\small{\em \leftheadervalue}}\hfil}
\fancyfoot[CE]{{\small \botmark}\hfil}
```

```
\fancyfoot[CO]{\hfil{\small \botmark}}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
\fancypagestyle{main}{%
\fancyhead{}
\fancyfoot{}
\fancyhfoffset[RO,LE]{\headoffsetlength}
\fancyhead[RO]{{\small\textrm{\thepage}}}
\fancyhead[LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{\hfil{\small{\em \volumeheadervalue}}}
\fancyhead[CE]{{\small{\em \leftheadervalue}}\hfil}
\fancyfoot[CE]{{\small \botmark}\hfil}
\fancyfoot[CO]{\hfil{\small \botmark}}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
}
```

Style for the first page of every volume in a Collected Poems. You don't have to call this style explicitly. It is called by the `\volumetitlepage` environment. But of course if you want to change the style of those pages, you should do it here.

```
\fancypagestyle{volumefirststyle}{%
\fancyhf{}
\fancyfoot[C]{\hfil{\small \botmark}}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
}
```

Style for plain pages.

```
\fancypagestyle{plain}{%
\fancyhf{}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
}
```

Style for divider pages between volumes

```
\fancypagestyle{volumetitlestyle}{%
\fancyhf{}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
}
```

`\resetpagestyle`

`\resetpagestyle` restores the fancy page style of the main sections of the book. `\putpoemcontents` changes the fancy page style, and `\resetpagestyle`, which is called by `\putpoemcontents` as it finishes, restores the original style

```
 \newcommand{\resetpagestyle}{\pagestyle{main}
% \fancyhead{}
% \fancyfoot{}
% \fancyhead[RO,LE]{{\small\textrm{\thepage}}}
% \fancyhead[CO]{\hfil{\small{\em \volumeheadervalue}}}
% \fancyhead[CE]{{\small{\em \leftheadervalue}\hfil}}
% \fancyfoot[CE]{{\small \botmark}\hfil}
% \fancyfoot[CO]{\hfil{\small \botmark}}
% \renewcommand{\headrulewidth}{0pt}
% \renewcommand{\footrulewidth}{0pt}
}
```

`\clearemptydoublepage` This macro is useful for making sure that the verso of a title page is blank.

```
 \newcommand{\clearemptydoublepage}{\newpage{\pagestyle{empty}\cleardoublepage}}
%
```

`\mymarks` Here the marks mechanism keeps track of pages from which notes come. Marks are used different in the notes sections and in the body of the volume. In the body, they are used to keep track of cases where a stanza break falls on a page break. In the notes sections, they set the running headers.

```
\newtoks{\singlepageabbrev}
\singlepageabbrev={p.}
\newcommand{\changesinglepageabbrev}[1]{\singlepageabbrev={#1}}
\newtoks{\multiplepageabbrev}
\newcommand{\changemultiplepageabbrev}[1]{\multiplepageabbrev={#1}}
\multiplepageabbrev{pp.}
\newcommand{\mymarks}{%
  \ifthenelse{\equal{\firstmark}{\botmark}}%
       {\unskip ~\the\singlepageabbrev~\firstmark}% if equal
       {\unskip ~\the\multiplepageabbrev~\firstmark--\botmark}}%if not equal
\newcommand{\poemdotfill}{\dotfill}
```

20.8 Environments: poem, and stanza

Here are definitions of the poem and stanza environments, and of the `\verseline` and `\stanzalinestraddle` macros.

`stanza` Page breaks are slightly encouraged in stanza breaks. Notice that entering a stanza sets the mark to `\relax` and leaving it sets the mark to * (the default value of `\stanzaatbottom`). If you want to use some other symbol for this purpose, use `\stanzaatbottom{}` or `\nostanzaatbottom{}` (if you also want to mark cases where there is *no* stanza break at the bottom of the page) to change it.

```
\newenvironment{stanza}{\penalty-100\instanzatrue\mark{\nostanzaatbottomvalue\ }}
 {\nobreak\mark{\stanzaatbottomvalue\ }\nobreak\ifspeciallinelock%
\relax\else%
\nolinebreak\incrementverselinenumber%
\fi%
\speciallinelockfalse%
\nobreak
  \instanzafalse

  \stanzaskip
}

```

```
\newenvironment{marginenvironment}{}{}
\newcommand{\stanzalinestraddle}
    {\begin{stanza}\addtocounter{verselinenumber}{-1}%
    \addtocounter{printlineindex}{-1}%
    }
```

The next two environments are for contents entries and titles. The idea is to handle overrun titles the way overrun lines are handled inverse. Contents entries indent the overrun slightly. Titles do not indent the overrun relative to the first line, but preserve the indentation of that line.

```
\newlength{\contentsentryoverrun}
\setlength{\contentsentryoverrun}{\contentsindentoneamount}
\newenvironment{contentsentryenvironment}{%
\language=255%no hyphenation in contents
\let\\=\@centercr%
\list{}{\topsep\z@%
\partopsep\z@%
\itemsep\z@ \itemindent -\contentsentryoverrun\listparindent \itemindent
          \rightmargin 1sp\leftmargin 1sp\advance\leftmargin
          \contentsentryoverrun}\item[]}{\endlist}
\newlength{\titleentryoverrun}
\setlength{\titleentryoverrun}{\z@}
\newenvironment{titleentryenvironment}{%
\language=255%no hyphenation in poem titles
\let\\=\@centercr%
\list{}{\topsep\z@%
\partopsep\z@%
\itemsep\z@ \itemindent -\titleentryoverrun\listparindent \itemindent
          \rightmargin 1sp\leftmargin 1sp\advance\leftmargin
          \titleentryoverrun}\item[]}{\endlist}
```

The poem environment. The little trick about setting the language number is designed to suppress automatic hyphenation in the poem environment. The booleans tell the `\emendation{}` and `\explanatory{}` macros that there have not yet been emendations or explanatory notes for this poem. The little trick with `\mark` sets the value of `\mark` to the current value of `\stanzaatbottom` (by default *) if the current location is in a poem but not in a stanza (which is to say, in a stanza break). The `\pagestyle` for poetry sets the value of `\mark` in the footer, which is * if the page break happens at a stanza break, and `\relax` if the page break happens anywhere else.

`poem`

```
\newenvironment{poem}{\raggedright%
     \language=255%no hyphenation in verse
     \noemendyettrue%
     \noexplainyettrue%
     \setcounter{verselinenumber}{0}\setcounter{printlineindex}{0}%
     \nobreak\begin{verse}%
     \inpoemtrue\nobreak\mark{\relax}%
     }{\end{verse}%
```

```
    \inpoemfalse\mark{\relax}%
    \goodbreak\afterpoemskip%\bigskip
    }
```

rightflushverse

Sets verse with the runover portions of long lines flushed to the right. Each line must begin with `\rightversebegin`.This code was suggested to me by Mark Wooding on comp.text.tex, who describes the code as "very unpleasant."

```
\newenvironment{rightflushverse}{\let\oldleftskip=\leftskip%
\leftskip=1 \leftskip\advance\leftskip by 0pt plus 1fill}%
{\let\leftskip=\oldleftskip}
```

20.9 Environments: main title page, divider pages

Environments for the main title page, and for the divider pages for individual volumes in a Collected Poems.

volumetitlepage
maintitlepage
booksectionpage

```
\newenvironment{volumetitlepage}
{\par\clearpage{\pagestyle{volumetitlestyle}\cleardoublepage}
\thispagestyle{volumetitlestyle}}
    {\newpage\thispagestyle{volumetitlestyle}
\cleardoublepage\thispagestyle{volumefirststyle}}
\newenvironment{maintitlepage}
{\par\cleardoublepage\thispagestyle{volumetitlestyle}
}{\newpage\thispagestyle{volumetitlestyle}\cleardoublepage}
\newenvironment{booksectionpage}
{\par\newpage\thispagestyle{volumetitlestyle}}{\newpage}
```

20.10 Marginal line numbers, verse lines, line cross references, etc.

The default is that the line numbers are on the right side, and that the `\marginparsep` is 18pt. If you use `\pmclleftsidepar` the placement of the number is controlled by `\marginparsep` and `\marginparwidth`. You should experiment with different values until you find a set that suits you. A rule of thumb for me is that the `\marginparsep` if the numbers is on the right should be the negative of the value if it is on the left. First, some lengths and ifs:

```
 \setlength{\marginparsep}{18pt}
 \setlength{\marginparwidth}{10pt}
\newif\ifnumbersswitch \numbersswitchfalse
\newif\ifnumbersright \numbersrighttrue
```

Next, code for `\pmclsidepar`, taken from the memoir class, by Peter Wilson. Modified by Dan Luecking.

\pmclsidepar

```
\newcommand{\pmclsidepar}{\@dblarg{\@pmclsidepar}}
%\long\def\@pmclsidepar[#1]#2{\@bsphack\leavevmode\vadjust{%
\long\def\@pmclsidepar[#1]#2{\@bsphack\strut\vadjust{% added
\checkoddpage
  \ifpmclsideparswitch
    \ifpmclreversesidepar
      \ifoddpage
        \oddpagefalse
      \else
        \oddpagetrue
      \fi
    \fi
  \else
    \oddpagetrue
    \ifpmclreversesidepar
      \oddpagefalse
    \fi
  \fi
  \rlap{\kern-\parindent
    \if@twocolumn
      \if@firstcolumn     % put at left
        \kern -\marginparsep \kern -\marginparwidth % at left
      \else               % put at right
        \kern \columnwidth \kern \marginparsep % at right
      \fi
    \else
      \ifoddpage            % put at right
        \kern \textwidth \kern \marginparsep        % at right
      \else                 % put at left
        \kern -\marginparsep \kern -\marginparwidth % at left
      \fi
    \fi
%     \vbox to 0pt{%
%      \kern \pmclsideparvshift%           % vertical shift to align top text lines
%      \begin{minipage}{\marginparwidth}%
    \setbox0=\vtop to 0pt{% added
     \begin{minipage}[t]{\marginparwidth}% added
     \ifoddpage #2\else #1\fi%
      \end{minipage}%
%  \vss}}}\@esphack}
\vss}% added
 \vtop to 0pt{\kern\pmclsideparvshift % default should be 0pt
  \kern-\dp\strutbox \kern-\ht0 \box0 \vss}}}\@esphack}% added


```

`\thepmemc@@page` The value of the page counter.

```
\gdef\thepmemc@@page{\the\c@page}

```

`\pmemlabel` A version of `\label` that uses `\pmemprotected@write`.

```
\providecommand{\pmemlabel}[1]{\@bsphack
  \pmemprotected@write\@auxout{}%
  {\string\newpmemlabel{#1}{\thepmemc@@page}}%
 \@esphack}
```

\pmemprotected@write

\pmemprotected@write is a modified version of the kernel's \protected@write.

```
\long\def\pmemprotected@write#1#2#3{%
 \begingroup
 \let\thepmemc@@page\relax
 #2%
 \let\protect\@unexpandable@protect
 \edef\reserved@a{\write#1{#3}}%
 \reserved@a
 \endgroup
 \if@nobreak\ifvmode\nobreak\fi\fi}

```

\newpmemlabel
\pmemlabelref

We have to be able to cope with a particular label not (yet) being in the the aux file when we come to check the page number.

```
\providecommand{\newpmemlabel}[2]{{\global\@namedef{m@#1}{#2}}}
\providecommand{\pmemlabelref}[1]{%
 \expandafter\ifx\csname m@#1\endcsname\relax
  0%
 \else
  \csname m@#1\endcsname
 \fi}

```

\@memcnta

We need a scratch count register.

```
\newcount\@memcnta
```

\checkoddpage

This is the user level command to check for odd/even page. It does a robust check for \strictpagecheck otherwise the simple minded check. It sets \ifoddpage to TRUE if the page is odd, otherwise it sets it to FALSE.

This is now fixed so that it should work for non-arabic page numbering. It uses a new label/pageref variant based on the page counter value, not its printed representation. This also gets rid of worrying about hyperref!! The problem was discovered by Bastiaan Niels Veelo

```
\DeclareRobustCommand{\checkoddpage}{%
 \oddpagefalse%
 \ifstrictpagecheck%
  \stepcounter{cp@cntr}\pmemlabel{\cplabel\thecp@cntr}%
  \@memcnta=\pmemlabelref{\cplabel\thecp@cntr}\relax
  \ifodd\@memcnta\oddpagetrue\fi
 \else
  \ifodd\c@page\oddpagetrue\fi
 \fi}

```

\ifpmclreversesidepar
\ifpmclsideparswitch

Analogues of \marginpar controls.

```
\newif\ifpmclreversesidepar
%  \pmclreversesideparfalse
  \pmclreversesidepartrue
\newif\ifpmclsideparswitch
  \pmclsideparswitchfalse
\if@twoside \pmclsideparswitchtrue \fi

```

\ifoddpage
\ifstrictpagecheck
\c@cp@cntr
\cplabel

The boolean \ifoddpage is TRUE if the checked page is odd. The boolean \ifstrictpagecheck is for turning on (TRUE) and off (FALSE) the strictest method of page checking. The counter cp@cntr is used to make unique labels, which start with \cplabel.

```
\newif\ifoddpage
\newif\ifstrictpagecheck
  \strictpagecheckfalse
\newcounter{cp@cntr}
\newcommand{\cplabel}{^_}
```

Two-column and one-column printing is again realized via a switch which is defined in the kernel. The default is single column printing.

\pmclsideparvshift

Vertical shift for sidepar to align with text line

```
\newlength{\pmclsideparvshift}
%  \setlength{\pmclsideparvshift}{-2.08ex}% seems to work for all font sizes
%\setlength{\pmclsideparvshift}{-2.58ex}
\setlength{\pmclsideparvshift}{0ex}
```

My code resumes here

\verselinenumberstoleft
\verselinenumberstoright

```
\newcommand{\verselinenumberstoleft}{%
    \setlength{\marginparsep}{18pt}%
    \setlength{\marginparwidth}{10pt}%
    \numbersswitchfalse\numbersrightfalse}
\newcommand{\verselinenumberstoright}{%
    \setlength{\marginparsep}{18pt}%
    \setlength{\marginparwidth}{10pt}%
    \numbersswitchfalse\numbersrighttrue}
```

\verselinenumbersswitch
\verselinenumbersgutter
\verselinenumbersouter

```
\newcommand{\verselinenumbersswitch}{%
    \strictpagechecktrue%
    \setlength{\marginparsep}{18pt}%
    \setlength{\marginparwidth}{10pt}%
    \numbersswitchtrue\numbersrightfalse\pmclsideparswitchtrue}
\newif\ifnumbersgutter \numbersgutterfalse%
\newcommand{\verselinenumbersgutter}{\verselinenumbersswitch%
\pmclreversesideparfalse%
\numbersguttertrue%
}
\newcommand{\verselinenumbersouter}{\verselinenumbersswitch%
\pmclreversesidepartrue%
\numbersgutterfalse}
```

\putverselinenumber

```
\newcommand{\putverselinenumber}{\nolinebreak\begin{marginenvironment}%
  \nolinebreak%
% \ifnumbersswitch\marginpar{\hfil\textrm{\scriptsize\theverselinenumber}}%
%     \else%
%     \ifnumbersright\marginpar{\hfil\textrm{\scriptsize\theverselinenumber}}%
%       \else%
%       \marginpar{\textrm{\scriptsize\theverselinenumber}\hfil}%
%       \fi%
%     \fi
  \ifnumbersswitch\pmclsidepar{\hfil\textrm{\scriptsize\theverselinenumber}}%
    \else%
    \ifnumbersright\pmclrightsidepar{\hfil\textrm{\scriptsize\theverselinenumber}}%
      \else%
      \pmclleftsidepar
{\textrm{\scriptsize\theverselinenumber}\hfil}%
      \fi%
    \fi%
%  \marn{\hfil\textrm{\theverselinenumber}}
%  \marginpar{\hfil\textrm{\scriptsize\theverselinenumber}}
%   \linenumberside{\textrm{\scriptsize\theverselinenumber}\hfil}
% \pmclrightsidepar{\textrm{\scriptsize\theverselinenumber}}
  \nolinebreak%
  \end{marginenvironment}%
  \nolinebreak%
}
```

\verseline

```
\newif\ifspeciallinelock
\speciallinelockfalse
\newcommand{\verseline}{\ifspeciallinelock%
\relax\else%
\nolinebreak\incrementverselinenumber%
\fi%
\speciallinelockfalse%
\nolinebreak\\
}
\newcommand{\incrementverselinenumber}{%
\nolinebreak\addtocounter{verselinenumber}{1}%
  \addtocounter{printlineindex}{1}%
  \ifnum\theprintlineindex>\thelineindexrepeat%
    \ifverselinenumbers\nobreak%
      \putverselinenumber\nobreak%
    \fi\nobreak%
    \setcounter{printlineindex}{0}%
  \fi%
}
\newcommand{\setverselinemodulo}[1]{%
\setcounter{lineindexrepeat}{#1}%
\addtocounter{lineindexrepeat}{-1}%
}
```

\linebend

A linebend is a a normal linebreak from LaTeX, with no incrementing of the line number counter, but the indentation makes the line look like a run over line. If you change the value of the run over in the definition of the \verse environment, be sure to change it here too. \linebend should be used for tweaking how poemscol runs over long lines.

```
\newcommand{\linebend}{\\
\hspace{\runoverindentvalue}%
 }
```

\saveverselinenumber and \restoreverselinenumber are useful for setting prose interjections in the midst of poems.

\saveverselinenumber
\restoreverselinenumber

```
\newcommand{\saveverselinenumber}{\setcounter{verselinenumberscratch}%
    {\value{verselinenumber}}%
   \setcounter{printlineindexscratch}{\value{printlineindex}}}
\newcommand{\restoreverselinenumber}{\setcounter{verselinenumber}%
     {\value{verselinenumberscratch}}%
    \setcounter{printlineindex}{\value{printlineindexscratch}}}
```

\rightversebegin

By default, poemscol indents the runover portion of long lines by the amount \runoverindent (by default 6 em). It is possible to set verse so that the runover portion is flushed to the right margin. To do this, you should enclose the \poem environment in the \rightflushverse environment. And you must mark the beginning of each verse line with \rightversebegin

```
\newcommand{\rightversebegin}
{\leavevmode\nobreak\hskip0pt plus -1fill\ignorespaces}
```

\hour

\hour{} is useful for typesetting the AM and the PM in the time.

```
\newcommand{\hour}[1]{{{\textsc{#1}}\ $\!\!\!$}
 }
```

\brokenline
\versephantom

\brokenline and \versephantom{} are a pair, useful for setting cases where there is a line break in the midst of a single metrical line. \tweakbrokenline ensures that the line number will mark the first partial line if the line numbers are on the left, the last partial line if the line numbers are on the right. If you have issued \tweakbrokenline in your preamble, the setting of runover lines may not be consistent with the setting of broken lines, because the line number will be set on the level of the end of the line. This is not a problem if the line numbers are set on the right, but it may look odd if the numbers are set on the left. \startverseline will ensure that the line number marks the beginning of any line it begins. If you put `\startverseline` at the beginning of the runover lines, it will ensure that the line number marks the beginning, not the end, of the line. If you wish `\startverseline` to align the number with the beginning of runover lines when the line number is on the left, and with the end of runover lines when the line number is on the right, issue `\tweakstartverseline` in your preamble.

```
\newcommand{\brokenline}{\\}
\newcommand{\versephantom}[1]{\leavevmode\phantom{#1}}
```

`\tweakbrokenline` Causes `\brokenline` to set the line number level with the beginning of the line if the number is on the left, level with the end of the line if the number is on the right. Issue it in the preamble.

```
\newcommand{\tweakbrokenline}{%
  \renewcommand{\brokenline}{\ifnumbersswitch\relax%
  \ifnumbersgutter% shift on odd pages for gutter numbers
  \ifodd\c@page \relax \else \speciallinelocktrue\incrementverselinenumber\fi
  \else % shift on even pages for outer numbers
  \ifodd\c@page \speciallinelocktrue\incrementverselinenumber \fi
  \fi
     \else \ifnumbersright \relax%
       \else \speciallinelocktrue\incrementverselinenumber%
       \fi%
     \fi%
\\}
}
```

`\brokenlineatbeginning` Issue `\brokenlineatbeginning` if you want broken line to set the line number level with the beginning of the line, regardless of whether the line number is at the right or at the left

```
\newcommand{\brokenlineatbeginning}{%
  \renewcommand{\brokenline}{%
  \speciallinelocktrue\incrementverselinenumber%
  \\
  }
}
```

`\startverseline` Put `\startverseline` at the beginning of every line where you wish to ensure that the line number is level with the beginning of the line. This is particularly useful for overrun lines when `\tweakbrokenline` or `\brokenlineatbeginning` is in effect.

```
\newcommand{\startverseline}{%
\speciallinelocktrue%
\leavevmode%
%\kern-3pt
\incrementverselinenumber%
}
```

`\tweakstartverseline` You may want `\startverseline` to align the line number with the beginning of the line if the line number is on the left, and with the end of the line if the line number is on the right. To do this, issue `\tweakstartline` in your preamble.

```
\newcommand{\tweakstartverseline}{%
\renewcommand{\startverseline}{%
\leavevmode%
\ifnumbersswitch%
% gutter case: shift on odd pages to top of runover line
  \ifnumbersgutter%
      \ifodd\c@page \relax \else \speciallinelocktrue\incrementverselinenumber\fi
% outer case: shift on even pages to top of runover line
  \else
      \ifodd\c@page \speciallinelocktrue\incrementverselinenumber \fi
  \fi% end of ifnumbersgutter
\else \ifnumbersright \relax% numbers don't switch. Are they to the right?
       \else \speciallinelocktrue\incrementverselinenumber% or to the left?
       \fi% end of ifnumbersright
\fi% end of ifnumbersswitch
}% end of renewcommand
}% end of tweakverseline
```

`\poemlinelabel`

`\poemlinelabel` is a hacked version of `\label` for crossreferences by line.

```
\def\poemlinelabel#1{\@bsphack
 \protected@write\@auxout{}%
 {\string\newlabel{#1}{{\theverselinenumber}{\thepage}}}%
\@esphack
}
```

20.11 Setup for contents, textual notes, emendations, and explanatory notes

First, the table of contents:

`\makepoemcontents`

An external file is created with the extension .ctn, and the filename of your driver file. The channel to that file is called `\poemcontents`. The rest of the command is the page style for the contents section. You will notice the running head Contents, and the title of the Contents section. I am assuming that the table of contents starts on page 5, but it may be different in different books. If you want to change that page number (say, to page 3) without messing with this file, issue `\literalcontents{\setcounter{page}{3}` just after you issue `\makepoemcontents` Alternatively, you can issue `\makepoemcontents{}` with its optional argument, putting the page number in the argument.

```
\newtoks{\pmclcontentsname}
\pmclcontentsname={CONTENTS}
\newtoks{\pmclcontentsheader}
\pmclcontentsheader={Contents}
\newcommand{\changecontentsname}[1]{%
\pmclcontentsname={#1}}
\newcommand{\changecontentsheader}[1]{%
\pmclcontentsheader={#1}}
\newcommand{\makepoemcontents}[1][5]{\global\poemcontentsontrue
```

```
  \newwrite\poemcontents
  \immediate\openout\poemcontents=\jobname.ctn
  \literalcontents{\flushbottom\normalfont
{\par\clearpage{\pagestyle{volumetitlestyle}\cleardoublepage}
\pagestyle{fancy}\thispagestyle{volumetitlestyle}}
\pagenumbering{roman}
\setcounter{page}{#1}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{{\small{\em \the\pmclcontentsheader}}}
\fancyhead[CE]{{\small{\em \the\pmclcontentsheader}}}
\fancyfoot{}
\backmattersink
\begin{center}{\normalfont \backmatterheaderfont
\the\pmclcontentsname}\end{center}
\lefthyphenmin=2\backmatterafterheadersink\tolerance=500\language=0}
\literalcontents{  \relax}}
```

Next, the textual collations:

`\maketextnotes` Like the `\makecontents` macro, `\maketextnotes` opens an external file with the extension .ent. It makes sure that line numbering is on (since it makes no sense to make notes without numbering the lines, particularly since the `\textnote{}` macro uses the line number). And it makes sure that the notes section is set with a flush bottom line, rather than with the ragged bottom line used for the verse sections. Also, it sets a label for sending the page number of the textual notes to the table of contents.

```
\newtoks{\pmclnotesname}
\pmclnotesname={TEXTUAL NOTES}
\newcommand{\changenotesname}[1]{\pmclnotesname={#1}}
\newtoks{\notesheadername}
\notesheadername={Textual Notes to}
\newcommand{\changenotesheader}[1]{\notesheadername={#1}}
\newtoks{\textnotescontentsname}
\textnotescontentsname={Textual Notes}
\newcommand{\changetextnotescontentsname}[1]{\textnotescontentsname={#1}}
\newcommand{\maketextnotes}{\global\textnotesontrue
  \global\verselinenumberstrue
  \newwrite\textnotes
  \immediate\openout\textnotes=\jobname.ent
  \literaltextnote{\flushbottom{\par\clearpage
  {\pagestyle{volumetitlestyle}\cleardoublepage}
\thispagestyle{volumetitlestyle}}
\fancyhead{}
\fancyfoot{}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{{\small{\em \the\notesheadername~\mymarks}}}
\fancyhead[CE]{{\small{\em \the\notesheadername~\mymarks}}}
\fancyfoot[C]{}
\mark{3}
\backmattersink
\begin{center}{\normalfont \backmatterheaderfont \the\pmclnotesname}\end{center}
\label{textualnotes}
\lefthyphenmin=2\backmatterafterheadersink\tolerance=500\language=0
\normalfont \backmattertextfont}
}
```

\makeemendations Next, the emendations section:

```
\newtoks{\emendationsname}
\emendationsname={EMENDATIONS}
\newcommand{\changeemendationsname}[1]{\emendationsname={#1}}
\newtoks{\emendationsheadername}
\emendationsheadername={Emendations to}
\newcommand{\changeemendationsheader}[1]{\emendationsheadername={#1}}
\newtoks{\emendationscontentsname}
\emendationscontentsname={Emendations}
\newcommand{\changeemendationscontentsname}[1]{%
\emendationscontentsname={#1}}
\newcommand{\makeemendations}{\global\emendationsontrue
  \newwrite\emendations
  \global\verselinenumberstrue
   \immediate\openout\emendations=\jobname.emd
   \literalemend{\flushbottom{\par\clearpage
   {\pagestyle{volumetitlestyle}\cleardoublepage}
\thispagestyle{volumetitlestyle}}
\fancyhead{}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{{\small{\em \the\emendationsheadername~\mymarks}}}
\fancyhead[CE]{{\small{\em \the\emendationsheadername~\mymarks}}}
\fancyfoot{}
\mark{3}
\backmattersink
\begin{center}{\normalfont \backmatterheaderfont
\the\emendationsname}\end{center}
\label{emendationnotes}
\lefthyphenmin=2\backmatterafterheadersink\tolerance=500\language=0
\normalfont \backmattertextfont}
}
```

\makeexplanatorynotes And the explanatory notes:

```
\newtoks{\explanationsname}
\explanationsname={EXPLANATORY NOTES}
\newcommand{\changeexplanationsname}[1]{\explanationsname={#1}}
\newtoks{\explanationsheadername}
\explanationsheadername={Explanatory Notes to}
\newcommand{\changeexplanationsheader}[1]{\explanationsheadername={#1}}
\newtoks{\explanationscontentsname}
\explanationscontentsname={Explanatory Notes}
\newcommand{\changeexplanationscontentsname}[1]{%
\explanationscontentsname={#1}}
\newcommand{\makeexplanatorynotes}{\global\explanontrue
```

```
    \global\verselinenumberstrue
    \newwrite\explanations
    \immediate\openout\explanations=\jobname.enx
    \literalexplain{\flushbottom{\par\clearpage
 {\pagestyle{volumetitlestyle}\cleardoublepage}
\thispagestyle{volumetitlestyle}}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{{\small{\em \the\explanationsheadername~\mymarks}}}
\fancyhead[CE]{{\small{\em \the\explanationsheadername~\mymarks}}}
\fancyfoot{}
\mark{3}
\backmattersink
\begin{center}{\normalfont \backmatterheaderfont
\the\explanationsname}\end{center}
\label{explanatorynotes}
\lefthyphenmin=2\backmatterafterheadersink\tolerance=500\language=0
\normalfont \backmattertextfont }
}
```

Auxiliary commands for note and title sections

```
%
\newcommand{\@pagemarktotextnotes}[1]{%
\immediate\write\textnotes{\string\setcounterfrompageref\string{notepageholdertitle\string}%
\string{#1\string}\string\unskip}%
\immediate\write\textnotes{\string\mark%
\string{\string\thenotepageholdertitle\string}}%
}
\newcommand{\@poempagetotextnotes}[1]{%
\immediate\write\textnotes{\string\par\string\argpageref\string{#1\string}\string\unskip}%
\@pagemarktotextnotes{#1}}
\newcommand{\@poemtitletotextnotes}[1]{\literaltextnote{\textbf{#1}}}
```

20.12 Book, volume, and volume section titles

\wholebooktitle
\booksection

```
\newcommand{\wholebooktitle}[1]{{\Huge\begin{center}
\hbox{#1}\end{center}}
}
\newcommand{\booksection}[1]{\volumesection{#1}}
```

\volumetitle

```
\newcommand{\volumetitle}[1]{%
  \set@p@emtitle{#1}{\volumetitlesink\par}{\volumetitlefont}
  {\relax}{\nobreak\par\nobreak}{\z@}
  \putpagenumberincontentsfalse
  \putpagenumberinnotesfalse
  \literalcontents{\goodbreak}
  \c@ntentsinfo{#1}{\relax}{\contentsvolumefont}
  {\contentsindentoneamount}
  \literalcontents{\vspace{5pt}\par\nobreak}
  \literaltextnote{\notesvolumetitlepenalty\bigskip\goodbreak\par}
  \t@xtnotesinfo{\backmattervolumefont #1 \backmattertextfont\nobreak}
  \putpagenumberincontentstrue
  \putpagenumberinnotestrue}
```

\volumetitlefirstline
\volumetitlelastline
\volumesubtitle
\volumesectiontitle

```
\newcommand{\volumetitlefirstline}[1]{\firstlinesettings%
  \set@p@emtitle{#1}{\volumetitlesink\par}
  {\volumetitlefont}
  {\relax}{\nobreak\par\nobreak}{\z@}
  \putpagenumberincontentsfalse
  \putpagenumberinnotesfalse
  \c@ntentsinfo{#1}{\relax}{\contentsvolumefont}
  {\contentsindentoneamount}
%  \literalcontents{\vspace{5pt}\par\nobreak}
%  \literaltextnote{\notesvolumetitlepenalty\bigskip\par}
  \t@xtnotesinfo{\backmattervolumefont #1 \backmattertextfont\nobreak}
  \putpagenumberincontentstrue
  \putpagenumberinnotestrue}
\newcommand{\volumetitlemiddleline}[1]{\middlelinesettings%
  \set@p@emtitle{#1}{\relax}
  {\volumetitlefont}
  {\relax}{\nobreak\par\nobreak}{\z@}
  \putpagenumberincontentsfalse
  \putpagenumberinnotesfalse
  \c@ntentsinfo{#1}{\contentsindentone}{\contentsvolumefont}
  {\contentsindenttwoamount}
  \t@xtnotesinfo{\backmattervolumefont #1 \backmattertextfont\nobreak}
  \putpagenumberincontentstrue
  \putpagenumberinnotestrue}
\newcommand{\volumetitlelastline}[1]{\lastlinesettings%
  \set@p@emtitle{#1}{\relax}
  {\volumetitlefont}
  {\relax}{\nobreak\par\nobreak}{\z@}
  \putpagenumberincontentsfalse
  \putpagenumberinnotesfalse
  \c@ntentsinfo{#1}{\contentsindentone}{\contentsvolumefont}
  {\contentsindenttwoamount}
  \literalcontents{\vspace{5pt}\par\nobreak}
  \t@xtnotesinfo{\backmattervolumefont #1 \backmattertextfont\nobreak}
  \putpagenumberincontentstrue
  \putpagenumberinnotestrue}
\newcommand{\volumesubtitle}[1]{{\volumesubtitlefont
\volumetitleshift #1 \par}
  \ifpoemcontentson
    \literalcontents{{\contentsvolumesubtitlefont
#1\vspace{3pt}\par\nobreak}}
  \fi
  \iftextnoteson\iftextnotesatend
```

```
    \immediate\write\textnotes{\string\backmattervolumesubtitlefont\ #1}
    \immediate\write\textnotes{\string\par \string\backmattertextfont}
    \immediate\write\textnotes{\string\nobreak}
  \fi\fi
}
\newcommand{\volumesectiontitle}[1]
{\clearpage\thispagestyle{volumefirststyle}
\begin{center} {\volumesubtitlefont\textbf{#1}}
\end{center} \bigskip\bigskip
  \ifpoemcontentson
  \literalcontents{\goodbreak{\contentsvolumesubtitlefont #1\par\nobreak}}
  \fi
  \iftextnoteson\iftextnotesatend
    \immediate\write\textnotes{\string\notesvolumetitlepenalty}
    \immediate\write\textnotes{\string\par \string\begin{bf}}
    \immediate\write\textnotes{#1 \string\end{bf}}
    \immediate\write\textnotes{\string\par}
    \immediate\write\textnotes{\string\nobreak}
  \fi\fi
}
```

20.13 Commands for setting titles of poems and sequences

First, commands for sending labels to the table of contents and to the notes sections. Since these are internal commands, not intended to be issued in the body of the text, they might have been marked out with "at" signs to prevent accidental redefinition. But since they are repeatedly written out to the external files, it seemed more economical to run the risk of redefinition than to write out `\makeatletter` and `\makeatother` with every note.

```
\newcommand{\makepoemlabel}[1]{\label{#1}}
\newcommand{\argpageref}[1]{\pageref{#1}}
```

20.13.1 Poem Titles

`\poemtitle` All of the other commands in this section are variations on this one, changing the penalties, the skips, the indentations, and the fonts. The command breaks into three sections. First, the title is set in the text. `\poemtitlepenalty` tests whether there is room for the title and the first couple of lines. `\poemtitlefont` globally sets the fonts for all poem titles. You can put the argument in an `\hbox{}` to make sure that the poem title is not broken across two lines, and the `\label` records the page for the contents and notes sections. `\afterpoemtitleskip` is the amount of white space after a poem title. `\fulltitleholder` and `\@compoundlabelscratch` record the title for concatenation later. All of these commands can manage titles with control sequences in them, such as for italics or international characters. In an earlier version there were special commands for such titles, using two arguments (one with the control sequence, one without it). Those commands are no longer necessary, but have been retained for backward compatibility with earlier versions.

Second, the poem is entered in the external file for the table of contents. `\contentsindentone` sets it in one level in the hierarchy of indentations, and `\contentspoemtitlefont` is the type size used for poems in the table of contents. The `\pageref{}` holds the page number of the poem.

Third, a new paragraph is opened in the textual notes for textual collations. `\@poempagetotextnotes{}` sends the page number to the textual notes. `\@poemtitletotextnotes{}` sends the title.

```


\newcommand{\m@kep@emlabel}{\ifsinglelinetitle%
 \addtocounter{poemnumber}{1}%
 \makepoemlabel{poem\thepoemnumber}%
 \else\iftitlefirstline%
     \addtocounter{poemnumber}{1}%
     \makepoemlabel{poem\thepoemnumber}%
     \else \relax
\fi
\fi
}

\newcommand{\m@ken@teholder}[1]{%
\ifsinglelinetitle
\fulltitleholder={#1}%
\global\edef\@compoundlabelscratch{poem\thepoemnumber}%
\else\iftitlefirstline
       \fulltitleholder={#1}%
       \global\edef\@compoundlabelscratch{poem\thepoemnumber}%
     \else\iftitlemiddleline
     \titlesofar=\fulltitleholder
     \titleincrement={~#1}%
     \global\edef\titleconcat{\the\titlesofar \the\titleincrement}%
     \fulltitleholder=\expandafter{\titleconcat}%
     \global\edef\@compoundlabelscratch{poem\thepoemnumber}%
  \else
      \titlesofar=\fulltitleholder
      \titleincrement={~#1}%
      \global\edef\titleconcat{\the\titlesofar \the\titleincrement}%
      \fulltitleholder=\expandafter{\titleconcat}%
      \global\edef\@compoundlabelscratch{poem\thepoemnumber}%
   \fi
       \fi
  \fi
}

\newcommand{\set@p@emtitle}[6]{\setlength{\titleentryoverrun}{#6}%
{#2}%
{#3 \begin{titleentryenvironment}{#4 #1}\end{titleentryenvironment}}%
{#5}%
\m@kep@emlabel
```

```
\m@ken@teholder{#1}%
\setlength{\titleentryoverrun}{\z@}}

\newcommand{\setcontentsleaders}[1]{%
   \def\contentsleaders{#1\nobreak%
}}
\setcontentsleaders{~/~}
\newcommand{\c@ntentsinfo}[4]{\ifpoemcontentson
\immediate\write\poemcontents{\string\setlength\string\contentsentryoverrun\string{#4\string
\immediate\write\poemcontents{\string\begin\string{contentsentryenvironment\string}}%
\literalcontents{#2{#3#1}}%
  \ifputpagenumberincontents
\immediate\write\poemcontents{\string\contentsleaders}%
\immediate\write\poemcontents{\string\pageref
\string{poem\thepoemnumber\string}}%
  \else
  \literalcontents{\relax}%
  \fi
\immediate\write\poemcontents{\string\end\string{contentsentryenvironment\string}}%
\immediate\write\poemcontents{\string\par}%
\immediate\write\poemcontents{\string\setlength\string\contentsentryoverrun
\string{\string\contentsindentoneamount\string}}%
\fi
}

\newcommand{\t@xtnotesinfo}[1]{\iftextnoteson
\iftextnotesatend
 \immediate\write\textnotes{\string\notespoemclubpenalty}%
  \ifputpagenumberinnotes
   \@poempagetotextnotes{poem\thepoemnumber}%
  \fi
   \@poemtitletotextnotes{#1}%
%    \else
%   \relax
   \fi\fi
}
\newcommand{\poemtitle}[1]{%
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\z@}%
\c@ntentsinfo{#1}{\contentsindentone}{\contentspoemtitlefont}%
{\contentsindenttwoamount}%
\t@xtnotesinfo{#1}%
}
\newcommand{\poemtitlenocontents}[1]{%
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\z@}%
\t@xtnotesinfo{#1}%
}
\newcommand{\poemtitlebaretitle}[1]{%
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\z@}%
}
```

\poemtitleitalic

Poems with italic titles or other formatting (such as \hour{}) take two arguments. The second argument is discarded. This command is to maintain backward compatibility with earlier versions.

```
\newcommand{\poemtitleitalic}[2]{%
\poemtitle{#1}}
```

For poem subtitles, see the entry on \sequencesectionsubtitle{}.

\poemtitlefirstline
\poemtitlemiddleline
\poemtitlelastline

Next, macros for setting multiline poem titles. Notice the special larger penalty for page breaks in the middle of multiple line title.

```
\newcommand{\firstlinesettings}{\singlelinetitlefalse
\titlefirstlinetrue
\putpagenumberincontentsfalse
\putpagenumberinnotestrue}

\newcommand{\middlelinesettings}{\singlelinetitlefalse
\titlemiddlelinetrue
\putpagenumberincontentsfalse
\putpagenumberinnotesfalse}

\newcommand{\lastlinesettings}{\singlelinetitlefalse
\titlelastlinetrue
\putpagenumberincontentstrue
\putpagenumberinnotesfalse}

\newcommand{\restoresinglelinesettings}{\putpagenumberinnotestrue
\putpagenumberincontentstrue
\titlefirstlinefalse
\titlemiddlelinefalse
\titlelastlinefalse
\singlelinetitletrue}


\newcommand{\poemtitlefirstline}[1]{
\firstlinesettings
\set@p@emtitle{#1}{\multilinetitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak}{\z@}%
\c@ntentsinfo{#1}{\contentsindentone}{\contentspoemtitlefont}%
{\contentsindenttwoamount}%
\t@xtnotesinfo{#1}%
\restoresinglelinesettings
}
```

Notice the use of token registers and \edef here to concatenate the first and second lines.

```
\newcommand{\poemtitlemiddleline}[1]{%
\middlelinesettings
\set@p@emtitle{#1}{\par\nobreak}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak}{\z@}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\restoresinglelinesettings
}
\newcommand{\poemtitlelastline}[1]{%
\lastlinesettings
\set@p@emtitle{#1}{\par\nobreak}{\poemtitlefont}%
{\relax}{\nobreak\par\afterpoemtitleskip\nobreak}{\z@}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\restoresinglelinesettings
}
```

Titles for sections of poems, and multi-line titles of sections of poems

`\poemsectiontitle`
`\poemsectiontitlefirstline`
`\poemsectiontitlemiddleline`
`\poemsectiontitlelastline`

```
\newcommand{\poemsectiontitle}[1]{%
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
}
\newcommand{\poemsectiontitlefirstline}[1]{%
\firstlinesettings
\set@p@emtitle{#1}{\multilinetitlepenalty}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak}{\titleindentamount}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\restoresinglelinesettings
}
\newcommand{\poemsectiontitlemiddleline}[1]{%
\middlelinesettings
\set@p@emtitle{#1}{\relax}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak}{\titleindentamount}%
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}%
{\contentsindentfouramount}%
\t@xtnotesinfo{#1}%
\restoresinglelinesettings
}
\newcommand{\poemsectiontitlelastline}[1]{%
\lastlinesettings
\set@p@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\afterpoemtitleskip\nobreak}%
{\titleindentamount}%
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}%
{\contentsindentfouramount}%
\t@xtnotesinfo{#1}%
\restoresinglelinesettings
}
```

\poemsectiontitlenocontents
\poemsectionbaretitle
\poemsubsectiontitle
\poemtitlenotitle
\poemtitlenotitleitalic

```
\newcommand{\poemsectiontitlenocontents}[1]{%
  \sequencesubsectiontitlenocontents{#1}}%
\newcommand{\poemfirstsectiontitle}[1]{%
\set@p@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
}
\newcommand{\poemfirstsectiontitlebaretitle}[1]{%
\set@p@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}}
\newcommand{\poemsectiontitlebaretitle}[1]{%
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}%
}
\newcommand{\poemsubsectiontitle}[1]{%
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}%
{\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindenttwoamount}%
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}%
{\contentsindentfouramount}%
\t@xtnotesinfo{#1}%
}
\newcommand{\poemtitlenotitle}[1]{%
\m@kep@emlabel
\m@ken@teholder{#1}%
\c@ntentsinfo{#1}{\contentsindentone}{\contentspoemtitlefont}%
{\contentsindenttwoamount}%
\t@xtnotesinfo{#1}%
}
\newcommand{\poemtitlenotitleitalic}[2]{%
\poemtitlenotitle{1}}
```

20.13.2 Poetic Sequences: Setting the Main Title

\sequencetitle

Variants here for multiple line titles and titles with font commands. Also some all-purpose kludges to work around other problems.

```
\newcommand{\sequencetitle}[1]{
\set@p@emtitle{#1}{\sequencetitlepenalty}{\sequencetitlefont}
{\relax}{\nobreak\par\nobreak\aftersequencetitleskip\nobreak}{\z@}
\putpagenumberincontentsfalse
\c@ntentsinfo{#1}{\contentsindentone}{\contentssequencetitlefont}
{\contentsindenttwoamount}
\putpagenumberincontentstrue
\t@xtnotesinfo{#1}
}
```

\sequencetitlefirstline
\sequencetitlemiddleline
\sequencetitlelastline

To set the title of a poetic sequence if it requires several lines to do so.

```
\newcommand{\sequencetitlefirstline}[1]{
\firstlinesettings
\set@p@emtitle{#1}{\multilinesequencepenalty}{\sequencetitlefont}
{\relax}{\nobreak\par\nobreak}{\z@}
\c@ntentsinfo{#1}{\contentsindentone}{\contentssequencetitlefont}
{\contentsindenttwoamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencetitlemiddleline}[1]{
\middlelinesettings
\set@p@emtitle{#1}{\par\nobreak}{\sequencetitlefont}
{\relax}{\nobreak\par\nobreak}{\z@}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentssequencetitlefont}
{\contentsindentthreeamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencetitlelastline}[1]{
\lastlinesettings
\set@p@emtitle{#1}{\par\nobreak}{\sequencetitlefont}
{\relax}{\nobreak\par\aftersequencetitleskip\nobreak}{\z@}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentssequencetitlefont}
{\contentsindentthreeamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
```

20.13.3 Sections of poetic sequences

\sequencesectiontitle

First, the easy case: a simple section of a poetic sequence.

```
\newcommand{\sequencesectiontitle}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
{\contentsindentthreeamount}
\t@xtnotesinfo{#1}
}
```

\sequencefirstsectiontitle The first poem in a sequence is a special case.

```
\newcommand{\sequencefirstsectiontitle}[1]{
\set@p@emtitle{#1}{\par\nobreak\afterpoemtitleskip}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
{\contentsindentthreeamount}
\t@xtnotesinfo{#1}
}
```

\sequencesectiontitlenonotes If you don't have any notes on a sequence section, it seems a shame to open an entry in the textual notes for that section:

```
\newcommand{\sequencetitlenonotes}[1]{
\set@p@emtitle{#1}{\par\nobreak}{\sequencetitlefont}
{\relax}{\nobreak\par\aftersequencetitleskip\nobreak}{\z@}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentssequencetitlefont}
{\contentsindentthreeamount}
%\t@xtnotesinfo{#1}
}
\newcommand{\sequencesectiontitlenonotes}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
{\contentsindentthreeamount}
%\t@xtnotesinfo{#1}
}
```

For multiple line titles of sequence sections

```
\newcommand{\sequencesectiontitlefirstline}[1]{
\firstlinesettings
\set@p@emtitle{#1}{\multilinetitlepenalty}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak}{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
{\contentsindentthreeamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencesectiontitlemiddleline}[1]{
\middlelinesettings
\set@p@emtitle{#1}{\par\nobreak}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak}{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
{\contentsindentfouramount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencesectiontitlelastline}[1]{
\lastlinesettings
\set@p@emtitle{#1}{\par\nobreak}{\poemtitlefont}
{\titleindent}{\nobreak\par\afterpoemtitleskip\nobreak}{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
{\contentsindentfouramount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
```

The following macros are to kludge around situations where the section title is just a number. This macro has two arguments. Originally, the first was the title of the section. The second is the title of the sequence as a whole. I've retained the second argument for backward compatibility with earlier versions, but you can set it to anything, since it's discarded now.

```
\newcommand{\sequencefirstsectiontitlenocontents}[2]{
\set@p@emtitle{#1}{\par\nobreak\afterpoemtitleskip}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
%\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencesectiontitlenocontents}[2]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
%\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
\t@xtnotesinfo{#1}
}
```

If the first section of a sequence is a multiple line title, the following macros can handle the special problems that situation poses.

```
\newcommand{\sequencefirstsectiontitlefirstline}[1]{
\firstlinesettings
\set@p@emtitle{#1}{\par\nobreak\afterpoemtitleskip}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
{\contentsindentthreeamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencefirstsectiontitlemiddleline}[1]{
\sequencesectiontitlemiddleline{#1}}

\newcommand{\sequencefirstsectiontitlelastline}[1]
{\sequencesectiontitlelastline{#1}}

\newcommand{\sequencefirstsectiontitlenonotes}[1]{
\set@p@emtitle{#1}{\par\nobreak\afterpoemtitleskip}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentamount}
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}
{\contentsindentthreeamount}
%\t@xtnotesinfo{#1}
}
```

`\sequencesectionsubtitle`
`\poemsubtitle`

```
\newcommand{\sequencesectionsubtitle}[1]{{\subsectiontitlefont \hbox{\titleindent #1}}
  \nobreak\par\nobreak\afterpoemtitleskip\nobreak
}
\newcommand{\poemsubtitle}[1]{\sequencesectionsubtitle{#1}
}
```

Another (former) kludge for special situations:

```
\newcommand{\sequencesectiontitleitalicnonotes}[2]
{\sequencesectiontitlenonotes{#1}
}
```

Italic titles used to require two arguments. These commands are retained only for backward compatibility.

`\sequencesectiontitleitalic`

```
\newcommand{\sequencesectiontitleitalic}[2]{
\sequencesectiontitle{#1}}
\newcommand{\sequencesectiontitlefirstlineitalic}[2]
{\sequencesectiontitlefirstline{#1}}
\newcommand{\sequencesectiontitlebaretitle}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\titleindentamount}
}
```

20.13.4 Subsections of sequences

`\sequencesubsectiontitle`

Continuing several levels down: macros for setting subsections and subsubsections of sequences.

```
\newcommand{\sequencesubsectiontitle}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindenttwoamount}
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
{\contentsindentfouramount}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencesubsectiontitlenocontents}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindenttwoamount}
%\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}{\contentsindentfouramount}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencefirstsubsectiontitle}[1]
```

```
{\set@p@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindenttwoamount}
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
{\contentsindentfouramount}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencefirstsubsectiontitlenocontents}[1]
{\set@p@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindenttwoamount}
%\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
%{\contentsindentfouramount}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencesubsectiontitlefirstline}[1]{
\firstlinesettings
\set@p@emtitle{#1}{\multilinetitlepenalty}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak}{\titleindenttwoamount}
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
{\contentsindentfouramount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencesubsectiontitlemiddleline}[1]{
\middlelinesettings
\set@p@emtitle{#1}{\nobreak}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak}{\titleindenttwoamount}
\c@ntentsinfo{#1}{\contentsindentfour}{\contentspoemtitlefont}
{\contentsindentfiveamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencesubsectiontitlelastline}[1]{
\lastlinesettings
\set@p@emtitle{#1}{\nobreak}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\afterpoemtitleskip\nobreak}
{\titleindenttwoamount}
\c@ntentsinfo{#1}{\contentsindentfour}{\contentspoemtitlefont}
{\contentsindentfiveamount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencesubsubsectiontitle}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}
{\titleindent\titleindent\titleindent}
{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentthreeamount}
\c@ntentsinfo{#1}{\contentsindentfour}{\contentspoemtitlefont}
{\contentsindentfiveamount}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencesubsubsectiontitlenocontents}[1]{
\set@p@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}
{\titleindent\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentthreeamount}
%\c@ntentsinfo{#1}{\contentsindentfour}{\contentspoemtitlefont}
%{\contentsindentfiveamount}
\t@xtnotesinfo{#1}
}
\newcommand{\sequencefirstsubsubsectiontitlenocontents}[1]{
\set@p@emtitle{#1}{\nobreak\par\nobreak}{\subsectiontitlefont}
{\titleindent\titleindent\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}
{\titleindentthreeamount}
%\c@ntentsinfo{#1}{\contentsindentfour}{\contentspoemtitlefont}{\contentsindentfiveamount}
\t@xtnotesinfo{#1}
}


\newcommand{\sequencefirstsubsectiontitlefirstline}[1]{
\firstlinesettings
\set@p@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}
{\titleindent\titleindent}{\nobreak\par\nobreak}{\titleindenttwoamount}
\c@ntentsinfo{#1}{\contentsindentthree}{\contentspoemtitlefont}
{\contentsindentfouramount}
\t@xtnotesinfo{#1}
\restoresinglelinesettings
}
\newcommand{\sequencefirstsubsectiontitlemiddleline}[1]{
\sequencesubsectiontitlemiddleline{#1}}
\newcommand{\sequencefirstsubsectiontitlelastline}[1]{
\sequencesubsectiontitlelastline{#1}}
```

Interjections in sequences, as in Robert Penn Warren's *Or Else*.

```
\newcommand{\interjectiontitlefirstline}[1]{
\sequencefirstsubsectiontitlefirstline{#1}}
\newcommand{\interjectiontitlelastline}[1]{
\sequencefirstsubsectiontitlelastline{#1}
}
\newcommand{\interjectiontitlemiddleline}[1]{
\sequencefirstsubsectionmiddleline{#1}
}
```

\setmargpoemtitle

Titles with a marginal reference to the notes giving the page in the notes where a comment on the poem appears, as in Altenberg's Heredia edition. \setmargpoemtitle takes as its second argument anything you want to add to the line while the title is being processed. \JHpoemtitle is just \setmargpoemtitle with the second argument passed to \JHlabel, which actually makes the marginal mark.

```
\newcommand{\setmargpoemtitle}[2]{%
\set@margp@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\z@}{#2}%
\c@ntentsinfo{#1}{\contentsindentone}{\contentspoemtitlefont}%
{\contentsindenttwoamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\set@margp@emtitle}[7]{\setlength{\titleentryoverrun}{#6}%
{#2}%
{#3 \begin{titleentryenvironment}%
{#4 #1 #7}\end{titleentryenvironment}}%
{#5}%
\m@kep@emlabel%
\m@ken@teholder{#1}%
\setlength{\titleentryoverrun}{\z@}}
```

`\JHpoemtitle` Macros to set marginal line references.

```
% \newcommand{\JHpoemtitle}[2]{%
% \setmargpoemtitle{#1}{\JHlabel{#2}}%
% }
\newcommand{\JHpoemtitle}[2]{%
\set@margp@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\z@}%
{\JHlabel{#2}}%
\c@ntentsinfo{#1}{\contentsindentone}{\contentspoemtitlefont}%
{\contentsindenttwoamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHprosesectiontitle}[2]{%
\set@margp@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\relax}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}{\z@}%
{\JHlabel{#2}}%
\c@ntentsinfo{#1}{\contentsindentone}{\contentspoemtitlefont}%
{\contentsindenttwoamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHlabel}[1]{%
\marginpar{{\scriptsize{\the\margrefmarker~p.~\pageref{#1}}}}}
```

`\JHsequencetitle` Macros to set marginal page references for sequences, sequence sections, poem sections, poem subtitles, dedications, and epigraphs.

```
\newcommand{\JHsequencetitle}[2]{%
\set@margp@emtitle{#1}{\sequencetitlepenalty}{\sequencetitlefont}%
{\relax}{\nobreak\par\nobreak\aftersequencetitleskip\nobreak}{\z@}%
{\JHlabel{#2}}%
\putpagenumberincontentsfalse%
\c@ntentsinfo{#1}{\contentsindentone}{\contentssequencetitlefont}%
```

```
{\contentsindenttwoamount}%
\putpagenumberincontentstrue%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHsequencesectiontitle}[2]{%
\set@margp@emtitle{#1}{\poemtitlepenalty}{\poemtitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}{\JHlabel{#2}}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHsequencefirstsectiontitle}[2]{%
\set@margp@emtitle{#1}{\par\nobreak\afterpoemtitleskip}{\poemtitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}{\JHlabel{#2}}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHpoemsectiontitle}[2]{%
\set@margp@emtitle{#1}{\poemtitlepenalty}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}{\JHlabel{#2}}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHpoemfirstsectiontitle}[2]{%
\set@margp@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}{\JHlabel{#2}}%
\c@ntentsinfo{#1}{\contentsindenttwo}{\contentspoemtitlefont}%
{\contentsindentthreeamount}%
\t@xtnotesinfo{#1}%
\sources{\label{#2}}%
}
\newcommand{\JHsequencesectionsubtitle}[2]{%
\set@margp@emtitle{#1}{\par\nobreak}{\subsectiontitlefont}%
{\titleindent}{\nobreak\par\nobreak\afterpoemtitleskip\nobreak}%
{\titleindentamount}{\JHlabel{#2}}%
\sources{\label{#2}}%
}
\newcommand{\JHpoemsubtitle}[2]{\JHsequencesectionsubtitle{#1}{#2}
}
\newcommand{\JHdedication}[2]{\nopagebreak\afterpoemtitleskip\begin{epigraphquote}%
```

```
\emph{#1}\JHlabel{#2}\end{epigraphquote}%
\sources{\label{#2}}\afterpoemtitleskip\nopagebreak}
\newcommand{\JHepigraph}[2]{\nopagebreak\afterpoemtitleskip\begin{epigraphquote}%
\emph{#1}\JHlabel{#2}\end{epigraphquote}%
\sources{\label{#2}}\afterpoemtitleskip\nopagebreak}
```

`\margreftextnote` Macros to set marginal page references showing where in the apparatus a comment upon the marked line is to be found. `\margrefspecial{}` is for occasions when you might need to set the label by hand. For instance, suppose you have several textnotes on a line, and the note you wish to draw attention to is on a different page from the other notes. `\margreftextnote` under those circumstances will point only to the first note. Use `\margrefspecial` to point to the other note, giving `\margrefspecial` a unique argument, and opening that note with a command using the same argument.

```
\newcounter{margrefnumber}
\setcounter{margrefnumber}{1}
\newtoks{\margrefmarker}
\margrefmarker={\dag}
\newcommand{\setmargrefmarker}[1]{\margrefmarker={#1}}
\newcommand{\makemargreflabel}{\string\label\string{margref\themargrefnumber\string}}
\newcommand{\setmargref}
{\marginpar{\scriptsize {\the\margrefmarker~p.~\pageref{margref\themargrefnumber}}}
}
\newcommand{\margreftextnote}{%
\iftextnoteson\addtocounter{margrefnumber}{1}%
\immediate\write\textnotes{\makemargreflabel}%
\setmargref\else\relax\fi%
}
\newcommand{\margrefexplanatory}{%
\ifexplanon\addtocounter{margrefnumber}{1}%
\immediate\write\explanations{\makemargreflabel}%
\setmargref\else\relax\fi%
}
\newcommand{\margrefemendation}{%
\ifemendationson\addtocounter{margrefnumber}{1}%
\immediate\write\emendations{\makemargreflabel}%
\setmargref\else\relax\fi%
}
\newcommand{\margrefspecial}[1]{%
\marginpar{\scriptsize {\the\margrefmarker~p.~\pageref{#1}}}
}
```

To change other title making commands to produce titles which add something while processing the title, you should: 1. add one to the number of arguments it uses 2. replace the use of `\set@p@emtitle` with `\set@margp@emtitle` 3. add `{#2}` or `\JHlabel{#2}` or whatever as the last (seventh) argument to `\set@margp@emtitle`. So, to change `\sequencetitle{}` to `\JHsequencetitle{}{}` start with:

```
\newcommand{\sequencetitle}[1]{
\set@p@emtitle{#1}{\sequencetitlepenalty}{\sequencetitlefont}
{\relax}{\nobreak\par\nobreak\aftersequencetitleskip\nobreak}{\z@}
\putpagenumberincontentsfalse
\c@ntentsinfo{#1}{\contentsindentone}{\contentssequencetitlefont}
{\contentsindenttwoamount}
\putpagenumberincontentstrue
\t@xtnotesinfo{#1}
}
```

and change it to

```
\catcode‘\@=11
\newcommand{\JHsequencetitle}[2]{
\set@margp@emtitle{#1}{\sequencetitlepenalty}{\sequencetitlefont}
{\relax}{\nobreak\par\nobreak\aftersequencetitleskip\nobreak}{\z@}
{\JHlabel{#2}}\putpagenumberincontentsfalse
\c@ntentsinfo{#1}{\contentsindentone}{\contentssequencetitlefont}
{\contentsindenttwoamount}
\putpagenumberincontentstrue
\t@xtnotesinfo{#1}
}
\catcode‘\@=12
```

20.14 Epigraphs, headnotes, attributions, dedications

Notice that epigraphs to volumes are handled differently from epigraphs to poems.

\epigraph
\headnote
\attribution
\poemdedication
\volumededication
\volumeepigraph
\volumeattribution

```
\newcommand{\epigraph}[1]{\nopagebreak\afterpoemtitleskip\begin{epigraphquote}
  \emph{#1}\end{epigraphquote}\afterpoemtitleskip\nopagebreak}
\newcommand{\headnote}[1]{\epigraph{#1}}
\newcommand{\attribution}[1]{\nopagebreak\afterpoemtitleskip\begin{epigraphquote}
 {\small\emph{#1}}\end{epigraphquote}\afterpoemtitleskip\nopagebreak}
\newcommand{\poemdedication}[1]{\nopagebreak\afterpoemtitleskip\begin{epigraphquote}
 \emph{#1}\end{epigraphquote}\afterpoemtitleskip\nopagebreak}
\providecommand{\dedication}[1]{\poemdedication{#1}}
\newcommand{\volumededication}[1]{\par\bigskip\begin{volumetitlepagequote}
  \emph{#1}\end{volumetitlepagequote}}
\newcommand{\volumeepigraph}[1]{\par\bigskip\begin{volumetitlepagequote}
  \emph{#1}\end{volumetitlepagequote}}
\newcommand{\volumeattribution}[1]{\par\smallskip\begin{volumetitlepagequote}
 {\small \emph{#1}}\end{volumetitlepagequote}}
```

20.15 Tools used for making note sections

The main tool is a dirty trick borrowed from John Lavagnino's package for endnotes, endnotes, which allows for writing out literal characters to an external file.

The trick involves redefining the space character as the newline character and treating the text of the note as the body of a macro that \meaning specifies. The result is that the text is written to the external file in a long ribbon one word wide.

```
\def\strip#1>{}
\newcommand{\literaltextnote}[1]{\iftextnotesatend
     \begingroup
        \def\next{#1}%
        \newlinechar='40
        \immediate\write\textnotes{\expandafter\strip\meaning\next}%
     \endgroup\fi
 }
\newcommand{\literalemend}[1]{\ifemendationsatend
     \begingroup
        \def\next{#1}%
        \newlinechar='40
        \immediate\write\emendations{\expandafter\strip\meaning\next}%
     \endgroup\fi
}
\newcommand{\literalexplain}[1]{\ifexplanatend
     \begingroup
        \def\next{#1}%
        \newlinechar='40
        \immediate\write\explanations{\expandafter\strip\meaning\next}%
     \endgroup\fi
}
\newcommand{\literalcontents}[1]{\ifpoemcontentson
     \begingroup
        \def\next{#1}%
        \newlinechar='40
        \immediate\write\poemcontents{\expandafter\strip\meaning\next}%
     \endgroup\fi
}
\newcommand{\literaltextnoteshort}[1]{
\iftextnotesatend
        \begingroup
        \def\next{#1}%
        \immediate\write\textnotes{\expandafter\strip\meaning\next}%
     \endgroup\fi}
\newcommand{\literalemendshort}[1]{\ifemendationsatend
        \begingroup
        \def\next{#1}%
        \immediate\write\emendations{\expandafter\strip\meaning\next}%
     \endgroup\fi}
\newcommand{\literalexplainshort}[1]{\ifexplanatend
        \begingroup
        \def\next{#1}%
        \immediate\write\explanations{\expandafter\strip\meaning\next}%
     \endgroup\fi}
```

```
\newcommand{\literalcontentsshort}[1]{\ifpoemcontentson
        \begingroup
        \immediate\write\poemcontents{\expandafter\strip\meaning\next}%
    \endgroup\fi}
\newcommand{\sameword}{$\sim$}
\newcommand{\missingpunct}{${}_{\wedge}$}
```

`\pmccheckifinteger` `\pmccheckifinteger{`⟨*num*⟩`}` checks if ⟨*num*⟩ is an integer. If it is, then `\ifinteger` is set TRUE, otherwise it is set FALSE. (Code taken from memoir class, and based on Donald Arseneau's `\Lpack{cite}` package).

```
\newcommand{\pmccheckifinteger}[1]{%
  \protected@edef\@vsa{#1}%
  \ifcat _\ifnum9<1\pmcgobm{#1} _\else A\fi
    \integertrue%
  \else
    \integerfalse%
  \fi%
}
\newif\ifinteger
% \begin{macro}{\pmcgobm}
% |\pmcgobm{|\meta{num}|}| is defined as \meta{num}. It could be defined as: \\
% |\newcommand{\pmcgobm}[1]{\ifx-#1\expandafter\gobm\else#1\fi}| \\
% which would remove a leading minus sign (hyphen) from its argument
% (|gobm| = gobble minus sign).
% (Code from memoir class and a posting to comp.text.tex by Donald Arseneau on 1997/07/21).
%    \begin{macrocode}
\newcommand{\pmcgobm}[1]{#1}
```

20.16 Commands to make notes and send info to contents

`\setlemmarange` `\setlemmarange` calculates the range of line numbers for multi-line lemmas.

```
% \newcommand{\setlemmarange}[1]{
% \setcounter{lemmalines}{#1}
% \ifthenelse{\equal{\value{lemmalines}}{0}}{\rangelemmafalse}{\rangelemmatrue}
% \setcounter{lemmaend}{\theverselinenumber}
% \addtocounter{lemmaend}{\thelemmalines}
% }
\newcommand{\setlemmarange}[1]{%
\pmccheckifinteger{#1}%
\ifinteger
\setcounter{lemmalines}{#1}%
\ifthenelse{\equal{\value{lemmalines}}{0}}{\rangelemmafalse}{\rangelemmatrue}%
\setcounter{lemmaend}{\theverselinenumber}%
\addtocounter{lemmaend}{\thelemmalines}%
\else
\setcounterfromref{lemmaend}{#1}\rangelemmatrue
\fi
\makeatother}
```

\citerange

\citerange sets the range of line numbers for multi-line lemmas

```
\newcommand{\citerange}{%
\ifrangelemma
\theverselinenumber --\thelemmaend
\else
\theverselinenumber
\fi}
```

\resetlemmacounters

\resetlemmacounters resets the counts for multiline lemmas.

```
\newcommand{\resetlemmacounters}{\rangelemmafalse
\setcounter{lemmalines}{0}%
\setcounter{lemmaend}{0}%
}
```

\checknoteheaders

\checknoteheaders Checks and corrects the values in the running headers of notes sections. The running headers are of the form “Notes to pages xx–yy.” Every title and every note checks whether the values in the running header should be changed. This command essentially gives a list of strings to be written on the external file for the note section. When these strings are read back in when the file is set, they make a little program that recalculates the values in the headers during the processing of every note. Also typesets the line number or line number range in notes sections.

```
\newcommand{\checknoteheaders}{\string\setcounter\string{notepageholdernote\string}%
\string{\thepage\string}\string\ifhmode\string\unskip\string\fi%
\string\ifthenelse\string{\string\value\string{notepageholdernote\string}>%
\string\value\string{notepageholdertitle\string}\string}%
\string{\string\mark\string{\string\thenotepageholdernote\string}\string}%
\string{\string\mark\string{\string\thenotepageholdertitle\string}\string}\string\unskip
\string\unskip\string\relax\ \citerange :\string~\string\nolinebreak}%
```

\textnote

The basic \textnote{} command, on which everything else is based, has several parts. First, it writes out a little program on the external endnotes file which, when it is read back in when the endnotes are typeset, sets a variable with the value of the page of the line the note is commenting upon and checks to see whether that value is equal to or greater than that of previous notes on that page of notes. Depending upon the outcome of that test, it sets the value of \mark with the correct numbers to print the proper running header of the form “Textual Notes to pp.xx–yy.” Second, it sends the line number of the line it is a comment upon and the text of the note to the notes section. The optional first argument is the number of lines covered by a multiline lemma. If there is no optional first argument specified, the default is 0. The second argument is the text of the note, which includes the rest of the lemma, other than the line number. It is up to you to suitably abbreviate long lemmas.

```
\newcommand{\textnote}[2][0]{%
\iftextnoteson
    \setlemmarange{#1}%
\immediate\write\textnotes{\checknoteheaders}%
    \literaltextnote{#2}%
    \fi
\resetlemmacounters
}
```

\accidental

Accidentals: As it is, the command just tests to see whether accidentals are being included or not, and sets the accidental as a textnote if accidentals are being included. It would not be hard to divert accidentals to another external file in order to compile a separate list of accidentals. Accidentals, by their nature, cannot have multiline lemmas.

```
\newcommand{\accidental}[1]{%
\iftextnoteson%
 \ifincludeaccidentals%
   \iftextnotesatend\immediate\write\textnotes{\string\nobreak}\fi%
\textnote{#1}%
\iftextnotesatend\immediate\write\textnotes{\string\nobreak}\fi%
 \fi\fi
}
```

\tsvariant
\tsaccidental

Typescript variants. Treated like accidentals. If they are being collected, they are sent to the textual notes. Again, it would not be hard to collect them separately. Typescript variants can have multiline lemmas.

```
\newcommand{\tsvariant}[2][0]{%
\iftextnoteson
 \ifincludetypescripts%
 \iftextnotesatend\immediate\write\textnotes{\string\nobreak}\fi%
\textnote[#1]{#2}%
\iftextnotesatend\immediate\write\textnotes{\string\nobreak}\fi%
  \fi\fi
}
\newcommand{\tsaccidental}[1]{\iftextnoteson\ifincludetypescripts\accidental{#1}\fi\fi}
```

\tsentry

Typescript entries. Allows one to to selectively include or exclude typescript entries from lists of variants. If an entry begins with a comma (as it commonly will, since it will typically be a member of a list of entries, you need not put \unskip before the comma, since poemscol will do it for you. Thanks to Donald Arseneau for suggesting how to do this.

```
\newcommand{\tsentry}[1]{\iftextnoteson\ifincludetypescripts{\@ifnextchar,{\unskip}{%
\@ifnextchar;{\unskip}{}%
}#1}\fi\fi\unskip}
```

\sources

`\sources{}` is essentially a `\textnote{}` with no line number and no page checking. Useful for typesetting the sources of the text (hence the name) and for sending literal characters to the endnotes file for other purposes.

```
\newcommand{\sources}[1]{%
\iftextnoteson
    \begingroup
```

```
        \newlinechar='40
        \def\next{#1}%
        \immediate\write\textnotes{\expandafter\strip\meaning\next}%
     \endgroup
 \fi
}
```

20.17 Emendations and explanatory notes

The difference between these kinds of notes and ordinary `\textnotes{}` is that they must first test to see whether there have been any emendations or explanatory notes for the poem in question. If there have not been prior notes, then a new paragraph must be opened and the page number and title of the poem set in the note section.

`\titletoothernotes`

`\titletoothernotes` sends the page and title information to the Emendations or Explanatory Notes sections. It is called only for the first emendation or explanatory note for a poem. It is designed so that if the user wishes to define another category of notes in which might not apply to all of the poems to be set — not all poems, for instance, have emendations — this command can be used for those notes as well.

```
\newcommand{\titletoothernotes}{\string\par
\string\argpageref\string{\@compoundlabelscratch\string}%
\string\setcounterfrompageref\string{notepageholdertitle\string}%
\string{\@compoundlabelscratch\string}%
\string\textbf\string{\ \the\fulltitleholder\string}}%
```

`\emendation`

```
\newcommand{\emendation}[2][0]{
\ifemendationson
    \setlemmarange{#1}%
    \ifredundantemendations
        \iftextnoteson
 \ifrangelemma
     \textnote[#1]{#2}%
 \else
     \textnote{#2}%
 \fi % ifrangelemma}
       \fi %iftextnoteson
   \fi % ifredundantemendations
    \ifnoemendyet % firstemendation
    \firstemendation
      \fi %noemendyet
      \global\noemendyetfalse
\immediate\write\emendations{\checknoteheaders}%
    \literalemend{#2}%
\fi % ifemendationson
\resetlemmacounters
}
```

This next is an internal command, called by \emendation. There's no need to use it explicitly.

```
\newcommand{\firstemendation}{\ifemendationsatend
\immediate\write\emendations{\titletoothernotes}\fi
}
```

\explanatory Again, \firstexplanatory is internal, called by \explanatory.

```
\newcommand{\explanatory}[2][0]{%
\ifexplanon
    \setlemmarange{#1}%
    \ifnoexplainyet % first explanation
    \firstexplanatory
      \fi %noexplainyet
      \global\noexplainyetfalse
\immediate\write\explanations{\checknoteheaders}%
    \literalexplain{#2}%
\fi % ifexplanon
\resetlemmacounters
}

\newcommand{\firstexplanatory}{\ifexplanatend
\immediate\write\explanations{\titletoothernotes}\fi
}

```

20.18 Making new notes sections

\definenewnotetype This monster macro has nine sections, and does all of the things required to set up a new end notes section. If you decide to mark a new kind of note in your text with, say \mynote, but decide at the last moment against including those notes in your edition, simply don't issue \putmynotes at the point it might have appeared, and neither the notes section nor the contents entry for it will be included.

```
\makeatletter
\long\def\appendtomacro#1#2{%
\begingroup
    \toks@\expandafter{#1#2}%
    \xdef#1{\the\toks@}%
  \endgroup}
\makeatother
\newcommand{\definenewnotetype}[5]{%
% 1. make an if and set it to true by default. make an if for the first
% instance of the command per poem and  set it to false
% Thanks to Igor Pechtchanski and Ulrich Schwarz.
\expandafter\newif\csname if#1son\endcsname
\global\csname #1sontrue\endcsname
\expandafter\newif\csname ifno#1yet\endcsname
\csname no#1yettrue\endcsname
% 2. open a new output stream
```

```
\expandafter\newwrite\csname #1s\endcsname
% 3. open a file to associate with the stream
\immediate\expandafter\openout\csname #1s\endcsname=\expandafter\jobname .#2
% 4. set up a literalwrite command
 \expandafter\newcommand\csname literal#1\endcsname[1]{
\begingroup
\def\next{##1}%
\newlinechar='40
\immediate\write\expandafter\csname #1s\endcsname{\expandafter\strip\meaning\next}%
\endgroup}
% 5. send heading information to output file
% first, use literaltext to send commands without parameters
\csname literal#1\endcsname{\flushbottom{\par\clearpage
  {\pagestyle{volumetitlestyle}\cleardoublepage}
\thispagestyle{volumetitlestyle}}
\fancyhead{}
\fancyfoot{}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}}
% then immediate write to send the parameter for the marks in the
% headers
\immediate\write\csname #1s\endcsname{\string\fancyhead[CO]\string{\string{\string\small
\string{\string\em\ #3\string~\string\mymarks\string}\string}\string}}
\immediate\write\csname #1s\endcsname{\string\fancyhead[CE]\string{\string{\string\small
\string{\string\em\ #3\string~\string\mymarks\string}\string}\string}}
% then literal to finish the headers
\csname literal#1\endcsname{\fancyfoot[C]{}
\mark{3}
\backmattersink}
% then immediate write to send the title of the section to print at
% the top of the page
\immediate\write\csname #1s\endcsname{
\string\begin\string{center\string}
\string{\string\normalfont \string\backmatterheaderfont\ #4\string}
\string\end\string{center\string}}
% immediate write to set a label for the table of contents
\immediate\write\csname #1s\endcsname{
\string\label\string{#1notes\string}}
% literal write to turn hyphenation on and set the font for the note
% section
\csname literal#1\endcsname{\lefthyphenmin=2\backmatterafterheadersink
\tolerance=500\language=0
\normalfont \backmattertextfont}
% 6. set up a command to write the page and title of the poem for the first
% instance of this kind of note in a poem
\expandafter\newcommand\csname first#1\endcsname{
\immediate\write\expandafter\csname#1s\endcsname{\titletoothernotes}}
% 7. set up a command to write a note, complete with line numbers
\expandafter\newcommand\csname #1\endcsname[2][0]{%
    \setlemmarange{##1}%
    \csname ifno#1yet\endcsname%
    \csname first#1\endcsname%
    \fi%
\global\csname no#1yetfalse\endcsname%
\immediate\write\expandafter\csname #1s\endcsname{\checknoteheaders}%
\csname literal#1\endcsname{##2}%
\resetlemmacounters%
}
% prose  version
\expandafter\newcommand\csname prose#1\endcsname[3][0]{%
\unskip\proselinelabel{##2}%
\setproselemmarange{##2}{##1}%
\csname ifno#1yet\endcsname%
\csname first#1\endcsname%
\fi%
\global\csname no#1yetfalse\endcsname%
\immediate\write\expandafter\csname #1s\endcsname{\checkprosenoteheaders}%
\csname literal#1\endcsname{##3}%
\resetlemmacounters\unskip%
}
% pm version
\expandafter\newcommand\csname pm#1\endcsname[2][0]{%
    \setpmlemmarange{##1}%
    \csname ifno#1yet\endcsname%
    \csname first#1\endcsname%
    \fi%
\global\csname no#1yetfalse\endcsname%
\immediate\write\expandafter\csname #1s\endcsname{\pmchecknoteheaders\string~}%
\immediate\write\expandafter\csname #1s\endcsname{\pmciterange}%
\csname literal#1\endcsname{##2}%
\pmresetlemmacounters\unskip%
}
% 8.  modify the end of the poem environment to reset the first
% command per poem boolean (so that the page and title will be set in
% the notes section when the first note is called for a poem)
%  thanks to Heiko Oberdiek
\makeatletter
\appendtomacro{\poem}{\global\expandafter\csname no#1yettrue\endcsname}
\makeatother
\makeatletter
\appendtomacro{\prosesection}{\global\expandafter\csname no#1yettrue\endcsname}
\makeatother
\makeatletter
\appendtomacro{\pmsection}{\global\expandafter\csname no#1yettrue\endcsname}
\makeatother

% 9. command for closing the output file
%  command for reading in and printing the output file
\expandafter\newcommand\csname put#1s\endcsname{
\newpage\hyphenationforsmall
\immediate\expandafter\closeout\csname #1s\endcsname
```

```
\expandafter\input \jobname.#2
\ifpoemcontentson
  \immediate\write\poemcontents{\string\contentspoemtitlefont\ #5}
  \immediate\write\poemcontents{\string~ \string\contentsleaders
  \string~\ \string\pageref{#1notes} \string\par}
  \immediate\write\poemcontents{\string\par \string\smallskip}
\fi
  }
% 10. command for making a marginal mark at a line, giving the page
% number in the apparatus where a note is to be found
\expandafter\newcommand\csname margref#1\endcsname{%
\addtocounter{margrefnumber}{1}
\immediate\write\expandafter\csname #1s\endcsname{\makemargreflabel}%
\setmargref
}
}
```

20.19 Prose sections

poemscol calls on lineno to do all the heavy lifting for making line numbers in prose sections and for making endnotes of various kinds for prose sections. First, set up a counter for the line number and for the modulo line number for prose sections, and set it to equal that used in verse sections. Then set the font for marginal line numbers to the size used in marginal line numbers in verse sections.

```
\newcounter{proselinenumber}
\setcounter{proselinenumber}{1}
\newcounter{prosemodulo}
\setcounter{prosemodulo}{\value{lineindexrepeat}}
\addtocounter{prosemodulo}{1}
\newcommand{\setprosemodulo}[1]{\modulolinenumbers[#1]}
```

Then commands for setting titles of prose sections. These are just versions of `\poemtitle`. Fancier versions will follow if there is need.

```
\newcommand{\prosesectiontitle}[1]{\poemtitle{#1}}
\newcommand{\prosesectiontitlenotitle}[1]{\poemtitlenotitle{#1}}
```

Next, environments for prose sections. The environment resets (or doesn't, in the second case) the marginal line number. The default modulo line number is that prevailing in verse sections, but you can change it at will.

```
\setcounter{prosemodulo}{\value{lineindexrepeat}}%
\addtocounter{prosemodulo}{1}%
\ifx\modulolinenumbers\undefined%
 \relax\else
\modulolinenumbers[\value{prosemodulo}]% if lineno is called
\fi
\newenvironment{prosesection}
{\resetlinenumber\begin{linenumbers}%
\renewcommand{\linenumberfont}{\scriptsize}%
\ifnumbersright\rightlinenumbers*\setlength{\linenumbersep}{-\marginparsep}%
%\else\ifnumbersswitch\switchlinenumbers\setlength{\linenumbersep}{\marginparsep}%
\else\ifnumbersswitch\rightlinenumbers*\setlength{\linenumbersep}{-\marginparsep}%
\else\leftlinenumbers*\setlength{\linenumbersep}{\marginparsep}%
\addtolength{\linenumbersep}{5pt}% seems about right
\fi\fi
\noemendyettrue%
\noexplainyettrue%
}
{\end{linenumbers}}
\newenvironment{prosesectionnoreset}
{\begin{linenumbers}%
\renewcommand{\linenumberfont}{\scriptsize}
\ifnumbersright\rightlinenumbers*\setlength{\linenumbersep}{-\marginparsep}%
%\else\ifnumbersswitch\switchlinenumbers\setlength{\linenumbersep}{\marginparsep}%
\else\ifnumbersswitch\rightlinenumbers*\setlength{\linenumbersep}{-\marginparsep}%
\else\leftlinenumbers*\setlength{\linenumbersep}{\marginparsep}%
\addtolength{\linenumbersep}{5pt}% seems about right
\fi\fi
% \noemendyettrue%
% \noexplainyettrue%
}{\end{linenumbers}}
```

Set up commands for notes sections.

`\setproselemmastart` Finds the line number at the beginning of a lemma. Thanks to Robin Fairbairns and Heiko Oberdiek for `\ifrefundefined`. Fairbairns version

```
% \newcommand*\ifrefundefined[1] {%
%  \expandafter\ifx\csname r@#1\endcsname\relax
% }
% Oberdiek version
\makeatletter
\newcommand*{\ifrefundefined}[1]{%
  \expandafter\ifx\csname r@#1\endcsname\relax
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
\newcommand*{\@extract@ref}[2]{%
  \expandafter\expandafter\expandafter#1\csname
r@#2\endcsname{}{}\@nil
}

\newcommand*{\@extractref}{%
  \@extract@ref\@car
}

\newcommand*{\@extractpageref}{%
  \@extract@ref\@secondcar
}
```

```

\long\def\@secondcar#1#2#3\@nil{#2}

\newcommand*{\setcounterfromref}[2]{%
  \ifrefundefined{#2}{%
    \protect\G@refundefinedtrue
    \@latex@warning{Reference ‘#2’ on page \thepage \space
        undefined}%
    \setcounter{#1}{0}%
  }{%
    \setcounter{#1}{\@extractref{#2}}%
  }%
}
\newcommand*{\setcounterfrompageref}[2]{%
  \ifrefundefined{#2}{%
    \protect\G@refundefinedtrue
    \@latex@warning{Reference ‘#2’ on page \thepage \space
        undefined}%
    \setcounter{#1}{0}%
  }{%
    \setcounter{#1}{\@extractpageref{#2}}%
  }%
}
%
\makeatother
\newcommand{\setproselemmastart}[1]{%
\unskip
\setcounterfromref{proselinenumber}{#1}}
```

`\setproselemmarange` Finds the end of a multi-line lemma.

```
% \newcommand{\setproselemmarange}[2]{%
% \setproselemmastart{#1}%
% \setcounter{lemmalines}{#2}%
% \ifthenelse{\equal{\value{lemmalines}}{0}}{\rangelemmafalse}{\rangelemmatrue}%
% \setcounter{lemmaend}{\theproselinenumber}%
% \addtocounter{lemmaend}{\thelemmalines}\unskip%
% }
\newcommand{\setproselemmarange}[2]{%
\setproselemmastart{#1}%
\pmccheckifinteger{#2}%
\ifinteger
\setcounter{lemmalines}{#2}%
\ifthenelse{\equal{\value{lemmalines}}{0}}{\rangelemmafalse}{\rangelemmatrue}%
\setcounter{lemmaend}{\theproselinenumber}%
\addtocounter{lemmaend}{\thelemmalines}\unskip%
\else
% \ref{#2}
\setcounterfromref{lemmaend}{#2}\rangelemmatrue
\fi\unskip
}
```

`\proseciterange` Sets the range note for in a multi-line lemma.

```
\newcommand{\proseciterange}{%
\ifrangelemma%
\theproselinenumber --\thelemmaend%
\else%
\theproselinenumber%
\fi}
```

`\checkprosenoteheaders` Updates running header for note sections. Typesets line number in notes.

```
\newcommand{\checkprosenoteheaders}{\string\setcounter\string{notepageholdernote\string}%
\string{\thepage\string}\string\ifhmode\string\unskip\string\fi
\string\ifthenelse\string{\string\value\string{notepageholdernote\string}>
\string\value\string{notepageholdertitle\string}\string}%
\string{\string\mark\string{\string\thenotepageholdernote\string}\string}%
\string{\string\mark\string{\string\thenotepageholdertitle\string}\string}\string\unskip
\string\unskip\string\relax\ \proseciterange :\string~\string\nolinebreak}%
```

`\prosetextnote`

```
\newcommand{\proselinelabel}[1]{\unskip\linelabel{#1}}
\newcommand{\prosetextnote}[3][0]{%
\iftextnoteson%
\unskip\proselinelabel{#2}%
\setproselemmarange{#2}{#1}%
\immediate\write\textnotes{\checkprosenoteheaders}%
\literaltextnote{#3}%
\resetlemmacounters%
\fi\unskip%
}
```

`\proseemendation` Notice that it calls `\firstemendation` as usual.

```
\newcommand{\proseemendation}[3][0]{%
\ifemendationson%
\unskip\proselinelabel{#2}%
\setproselemmarange{#2}{#1}%
     \ifredundantemendations
       \iftextnoteson
 \ifrangelemma
     \prosetextnote[#1]{#2}{#3}%
 \else
     \prosetextnote{#2}{#3}%
 \fi % ifrangelemma}
       \fi %iftextnoteson
   \fi % ifredundantemendations
    \ifnoemendyet % firstemendation
     \firstemendation
      \fi%noemendyet
      \global\noemendyetfalse
\immediate\write\emendations{\checkprosenoteheaders}%
    \literalemend{#3}%
```

```
\resetlemmacounters%
\fi\unskip%
}
```

\proseexplanatory

```
\newcommand{\proseexplanatory}[3][0]{%
\ifexplanon%
\unskip\proselinelabel{#2}%
\setproselemmarange{#2}{#1}%
    \ifnoexplainyet% firstexplanation
     \firstexplanatory%
      \fi%noexplainyet
      \global\noexplainyetfalse%
\immediate\write\explanations{\checkprosenoteheaders}%
    \literalexplain{#3}%
\resetlemmacounters%
\fi\unskip%
}
```

\proseaccidental

```
\newcommand{\proseaccidental}[2]{
\iftextnoteson
 \ifincludeaccidentals
    \immediate\write\textnotes{\string\nobreak}%
\prosetextnote{#1}{#2}%
\immediate\write\textnotes{\string\nobreak}%
 \fi\fi
}
\newcommand{\prosetsvariant}[3][0]{\iftextnoteson
 \ifincludetypescripts
   \immediate\write
\textnotes{\string\nobreak}%
\prosetextnote[#1]{#2}{#3}%
\immediate\write\textnotes{\string\nobreak}%
  \fi\fi
}
\newcommand{\prosetsaccidental}[2]{\iftextnoteson
\ifincludetypescripts\proseaccidental{#1}{#2}\fi\fi}
```

20.20 Annotation by Sentence and Paragraph Number

Marking sentences. First, some `\if`s, token lists, and counters.

```
\newif{\ifprosebysentence}
\newif{\ifrunningsentencenumbers}\runningsentencenumberstrue
\newif{\ifmarginsentencenumbers}\marginsentencenumberstrue

\newtoks{\pmnoteheader}

\newcounter{pmsentencenumber}\setcounter{pmsentencenumber}{0}
```

```
\newcounter{pmmodulo}\setcounter{pmmodulo}{3}
\newcounter{pmindexcount}\setcounter{pmindexcount}{0}
```

A starter. Mostly a placeholder. For changes to a whole section.

```
\newcommand{\setprosebysentence}{\prosebysentencetrue%
}
\newenvironment{pmsection}{
\noemendyettrue%
\noexplainyettrue%
}{}
```

Formatting for sentence numbers in the running text and in the margin.

```
\newcommand{\runningsentencenumberformat}[1]{\nobreak%
\textsuperscript{\scriptsize{#1}}\unskip\nobreak}
\newcommand{\marginsentencenumberformat}[1]{\scriptsize{#1}}
\newcommand{\setpmmodulo}[1]{\setcounter{pmmodulo}{#1}%
\addtocounter{pmmodulo}{-1}%
}
\newcommand{\pmnumberstoleft}{%\catcode`\@=11%
\numbersswitchfalse\numbersrightfalse%
\setlength{\marginparsep}{18pt}%
}
\newcommand{\pmnumberstoright}{%\catcode`\@=11%
\numbersswitchfalse\numbersrighttrue%
\setlength{\marginparsep}{-18pt}
% \setlength{\marginparsep}{18pt}
}
\newcommand{\pmnumbersswitch}{\strictpagechecktrue\numbersswitchtrue\numbersrightfalse%
\setlength{\marginparsep}{18pt}%
}
\newcommand{\pmnumbersgutter}{%\catcode`\@=11%
\pmnumbersswitch%
\pmclreversesideparfalse\numbersguttertrue%
\setlength{\marginparsep}{18pt}%
}
\newcommand{\pmnumbersouter}{%\catcode`\@=11%
\pmnumbersswitch%
\pmclreversesidepartrue\numbersgutterfalse%
\setlength{\marginparsep}{18pt}%
}
```

Macros for actually writing the numbers in the margin.

```
\newcommand{\putpmsentencenumber}{%
\putpmmarginnumber{\thepmsentencenumber}%
{\marginsentencenumberformat}}
\newcommand{\putpmmarginnumber}[2]{%
\nolinebreak\begin{marginenvironment}%
  \nolinebreak%
  \ifnumbersswitch%\marginpar{\hfil #2{#1}}
  \pmclsidepar{\hfil #2{#1}}%
    \else%
    \ifnumbersright%\marginpar{\hfil #2{#1}}
    \pmclrightsidepar{\hfil #2{#1}}%
       \else% \marginpar{#2{#1}\hfil}
       \pmclleftsidepar
{#2{#1}\hfil}%
       \fi%
    \fi%
  \nolinebreak%
  \end{marginenvironment}\nolinebreak%
}
```

\pmsentence Put this before every sentence. (This is the opposite of the convention in verse sections, where `\verseline` is at the end of each line.

```
\newcommand{\pmsentence}{%
% increment sentence number at beginning of sentence
\addtocounter{pmsentencenumber}{1}%
\ifrunningsentencenumbers% add running number
\runningsentencenumberformat{\thepmsentencenumber}%
\fi%
\ifmarginsentencenumbers% add marginal number
  \addtocounter{pmindexcount}{1}%
  \leavevmode%
\ifthenelse{\value{pmindexcount}>\value{pmmodulo}}%
{     \putpmsentencenumber\nobreak%
    \setcounter{pmindexcount}{0}%
}{\relax}%
\fi%
% create header for notes if any are to occur
\ifpmparas%
\global\edef\noteheaderconcat{\thepmparagraph:\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\else%
%\global\edef\noteheaderconcat{\thepmsentencenumber:}%
\global\edef\noteheaderconcat{\thepmsentencenumber}
\pmnoteheader=\expandafter{\noteheaderconcat}%
\fi%
}
```

\pmsentencetwo Kludge for the second sentence in a line.

```
\newcommand{\pmsentencetwo}{%
% increment sentence number at beginning of sentence
\addtocounter{pmsentencenumber}{1}%
\ifrunningsentencenumbers% add running number
\runningsentencenumberformat{\thepmsentencenumber}%
\fi%
\ifmarginsentencenumbers% add marginal number
  \addtocounter{pmindexcount}{1}%
  \leavevmode%
\ifthenelse{\value{pmindexcount}>\value{pmmodulo}}%
{     \putpmsentencenumbertwo\nobreak%
```

```
    \setcounter{pmindexcount}{0}%
}{\relax}%
%   \ifnum\thepmindexcount>\thepmmodulo%
%       \putpmsentencenumbertwo\nobreak%
%      \setcounter{pmindexcount}{0}%
%    \fi%
\fi%
% create header for notes if any are to occur
\ifpmparas%
\global\edef\noteheaderconcat{\thepmparagraph:\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\else%
\global\edef\noteheaderconcat{\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\fi%
}

\newcommand{\putpmsentencenumbertwo}{%
\putpmmarginnumber{\rlap{\phantom{1},~\thepmsentencenumber}}%
{\marginsentencenumberformat}}
```

\pmsentencethree Kludge for the third sentence in a line.

```
\newcommand{\pmsentencethree}{%
% increment sentence number at beginning of sentence
\addtocounter{pmsentencenumber}{1}%
\ifrunningsentencenumbers% add running number
\runningsentencenumberformat{\thepmsentencenumber}%
\fi%
\ifmarginsentencenumbers% add marginal number
  \addtocounter{pmindexcount}{1}%
  \leavevmode%
\ifthenelse{\value{pmindexcount}>\value{pmmodulo}}%
{     \putpmsentencenumberthree\nobreak%
    \setcounter{pmindexcount}{0}%
}{\relax}%
%   \ifnum\thepmindexcount>\thepmmodulo%
%       \putpmsentencenumberthree\nobreak%
%     \setcounter{pmindexcount}{0}%
%    \fi%
\fi%
% create header for notes if any are to occur
\ifpmparas%
\global\edef\noteheaderconcat{\thepmparagraph:\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\else%
\global\edef\noteheaderconcat{\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\fi%
}

```

```
\newcommand{\putpmsentencenumberthree}{%
\putpmmarginnumber{\rlap{\phantom{1,~2},~\thepmsentencenumber}}%
{\marginsentencenumberformat}}
```

\pmsentencefour

```
\newcommand{\pmsentencefour}{%
% increment sentence number at beginning of sentence
\addtocounter{pmsentencenumber}{1}%
\ifrunningsentencenumbers% add running number
\runningsentencenumberformat{\thepmsentencenumber}%
\fi%
\ifmarginsentencenumbers% add marginal number
  \addtocounter{pmindexcount}{1}%
  \leavevmode%
\ifthenelse{\value{pmindexcount}>\value{pmmodulo}}%
{     \putpmsentencenumberfour\nobreak%
    \setcounter{pmindexcount}{0}%
}{\relax}%
\fi%
% create header for notes if any are to occur
\ifpmparas%
\global\edef\noteheaderconcat{\thepmparagraph:\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\else%
\global\edef\noteheaderconcat{\thepmsentencenumber}%
\pmnoteheader=\expandafter{\noteheaderconcat}%
\fi%
}

\newcommand{\putpmsentencenumberfour}{%
\putpmmarginnumber{\rlap{\phantom{1,~2,~3},~\thepmsentencenumber}}%
{\marginsentencenumberformat}}

```

Macros for numbering paragraphs

```
\newcounter{pmparagraph}
\setcounter{pmparagraph}{0}
\newif{\ifpmparas}\pmparasfalse
\newif{\ifpmparainmar}\pmparainmartrue
\newif{\ifpmpararunning}\pmpararunningfalse
\newif{\ifpmsentencebypara}% reset sentence counter every paragraph
\pmsentencebyparatrue
```

Formatting marginal and running paragraph numbers, printing them.

```
\newcommand{\pmparmarformat}[1]{\textbf{{\small #1}}}
\newcommand{\pmpararunningformat}[1]{\textbf{\small #1}}
\newcommand{\putpmmarparanumber}{%
\putpmmarginnumber{\llap{\thepmparagraph~~}}{\pmparmarformat}}
```

\pmpara Mark the beginning of each paragraph with \pmpara

```
\newcommand{\pmpara}{\leavevmode%
\pmparastrue\addtocounter{pmparagraph}{1}%
\ifpmparainmar% put in marginal paragraph number
   \putpmmarparanumber
\fi
\ifpmpararunning% put in running number
  \pmpararunningformat{\thepmparagraph}%
\fi
\ifpmsentencebypara% reset sentence counter
   \setcounter{pmsentencenumber}{0}%
   \setcounter{pmindexcount}{0}%
\fi}
```

`\sentencelabel` Makes a label either by sentence number or by paragraph and sentence number, for use by `\ref` and by the various notes commands.

```
\makeatletter
\def\sentencelabel#1{\@bsphack%
\ifpmparas%
\protected@write\@auxout{}%
 {\string\newlabel{#1}{{\thepmparagraph :\thepmsentencenumber}{\thepage}}}%
\else%
\protected@write\@auxout{}%
 {\string\newlabel{#1}{{\thepmsentencenumber}{\thepage}}}%
\fi%
\@esphack%
}
\makeatother
```

Macros to set up endnote commands for paragraph and sentence annotation.

```
\newtoks{\pmrangeend} % for notes that refer to ranges
% setting the range for the lemma:
\newcommand{\setpmlemmarange}[1]{%
\pmccheckifinteger{#1}%
\ifinteger%
\setcounter{lemmalines}{#1}%
\ifthenelse{\equal{\value{lemmalines}}{0}}{\rangelemmafalse}{\rangelemmatrue}%
\setcounter{lemmaend}{\thepmsentencenumber}%
\addtocounter{lemmaend}{\thelemmalines}%
\pmrangeend={\thelemmaend}%
\else%
\pmrangeend={\ref{#1}}\rangelemmatrue%
\fi%
}
% Setting the string to include ranges in notes:
\newcommand{\pmciterange}{%
\ifrangelemma{\the\pmnoteheader --\the\pmrangeend:}%
\else%
\the\pmnoteheader:%
\fi}
% For running headers
```

```
\newcommand{\pmchecknoteheaders}{\string\setcounter\string{notepageholdernote\string}%
\string{\thepage\string}\string\ifhmode\string\unskip\string\fi
\string\ifthenelse\string{\string\value\string{notepageholdernote\string}>
\string\value\string{notepageholdertitle\string}\string}%
\string{\string\mark\string{\string\thenotepageholdernote\string}\string}%
\string{\string\mark\string{\string\thenotepageholdertitle\string}\string}\string\unskip}
% to reset lemma counters at end of note
\newcommand{\pmresetlemmacounters}{\rangelemmafalse%
\setcounter{lemmalines}{0}%
\setcounter{lemmaend}{0}%
\pmrangeend={\relax}%
}
```

\pmtextnote

```
\newcommand{\pmtextnote}[2][0]{%
\iftextnoteson%
\setpmlemmarange{#1}%
\immediate\write\textnotes{\pmchecknoteheaders\string~}%
    \immediate\write\textnotes{\pmciterange}%
    \literaltextnote{#2}%
    \fi%
\pmresetlemmacounters%
\unskip}
```

\pmexplanatory

```
\newcommand{\pmexplanatory}[2][0]{%
\ifexplanon%
\setpmlemmarange{#1}%
    \ifnoexplainyet % first explanation
    \firstexplanatory%
      \fi %noexplainyet
      \global\noexplainyetfalse%
\immediate\write\explanations{\pmchecknoteheaders\string~}%
\immediate\write\explanations{\pmciterange}%
    \literalexplain{#2}%
\fi % ifexplanon
\pmresetlemmacounters%
\unskip%
}
```

\pmemendation

```
\newcommand{\pmemendation}[2][0]{%
\ifemendationson%
    \setpmlemmarange{#1}%
    \ifredundantemendations%
       \iftextnoteson%
 \ifrangelemma%
     \pmtextnote[#1]{#2}%
 \else%
```

```
     \pmtextnote{#2}%
 \fi % ifrangelemma}
       \fi %iftextnoteson
   \fi % ifredundantemendations
    \ifnoemendyet % firstemendation
     \firstemendation%
      \fi %noemendyet
      \global\noemendyetfalse%
\immediate\write\emendations{\pmchecknoteheaders\string~}%
\immediate\write\emendations{\pmciterange}%
    \literalemend{#2}%
\fi % ifemendationson
\pmresetlemmacounters%
}
```

\pmaccidental

```
\newcommand{\pmaccidental}[1]{%
\iftextnoteson%
 \ifincludeaccidentals%
    \immediate\write\textnotes{\string\nobreak}%
\pmtextnote{#1}%
\immediate\write\textnotes{\string\nobreak}%
 \fi\fi%
}
```

\pmtsvariant

```
\newcommand{\pmtsvariant}[2][0]{\iftextnoteson%
 \ifincludetypescripts%
   \immediate\write\textnotes{\string\nobreak}%
\pmtextnote[#1]{#2}%
\immediate\write\textnotes{\string\nobreak}%
  \fi\fi%
}
\newcommand{\pmtsaccidental}[1]{\iftextnoteson\ifincludetypescripts\pmaccidental{#1}\fi\fi}
```

20.21 Using Footnotes

\texnotesatfoot has to redefine \sources and a few others as well.

\textnotesatfoot

```
\newcommand{\textnotesatfoot}{% first redefine textnotes
\global\textnotesontrue%
\global\textnotesatendfalse%
\ifsourcesfootnotespara%
\DeclareNewFootnote[para]{A}%
\else%
\DeclareNewFootnote[plain]{A}%
\fi%
\iftextfootnotespara%
\DeclareNewFootnote[para]{B}%
\else%
\DeclareNewFootnote[plain]{B}%
\fi%
\renewcommand{\textnote}[2][0]{\setlemmarange{##1}%
\FootnotetextB{\relax}{\citerange:~##2}%
\resetlemmacounters}%
% then redefine sources
\renewcommand{\sources}[1]{\FootnotetextA{\relax}{##1}}%
% then prose textnote
\renewcommand{\prosetextnote}[3][0]{%
\unskip\proselinelabel{##2}%
\setproselemmarange{##2}{##1}%
\FootnotetextB{\relax}{\proseciterange:~##3}\resetlemmacounters}%
% then pmtextnote
\renewcommand{\pmtextnote}[2][0]{\setpmlemmarange{##1}%
\FootnotetextB{\relax}{\pmciterange~##2}\pmresetlemmacounters}
}
```

\emendationsatfoot

```
\newcommand{\emendationsatfoot}{%
\global\emendationsontrue%
\global\emendationsatendfalse%
\ifemendationfootnotespara%
\DeclareNewFootnote[para]{C}%
\else%
\DeclareNewFootnote[plain]{C}%
\fi%
\renewcommand{\emendation}[2][0]{\setlemmarange{##1}%
\FootnotetextC{\relax}{\citerange:~##2}%
\resetlemmacounters}%
\renewcommand{\proseemendation}[3][0]{\unskip\proselinelabel{##2}%
\setproselemmarange{##2}{##1}%
\FootnotetextC{\relax}{\proseciterange:~##3}\resetlemmacounters}%
\renewcommand{\pmemendation}[2][0]{\setpmlemmarange{##1}%
\FootnotetextC{\relax}{\pmciterange~##2}\pmresetlemmacounters}
}
```

\explanationsatfoot

```
\newcommand{\explanationsatfoot}{%
\global\explanontrue%
\global\explanatendfalse%
\ifexplanfootnotespara%
\DeclareNewFootnote[para]{D}%
\else%
\DeclareNewFootnote[plain]{D}%
\fi%
\renewcommand{\explanatory}[2][0]{\setlemmarange{##1}%
\FootnotetextD{\relax}{\citerange:~##2}%
\resetlemmacounters}%
\renewcommand{\proseexplanatory}[3][0]{\unskip\proselinelabel{##2}%
\setproselemmarange{##2}{##1}%
\FootnotetextD{\relax}{\proseciterange:~##3}}
\renewcommand{\pmexplanatory}[2][0]{\setpmlemmarange{##1}%
\FootnotetextD{\relax}{\pmciterange~##2}\pmresetlemmacounters}%
}
```

\bibleverse

```
\newcommand{\bibleverse}{\pmsentence}
\newcommand{\biblechapter}{\pmpara}
```

20.22 Embarrassing kludges

Because the dirty trick mentioned above writes out textnotes as a ribbon one word wide, quoted poetry in the notes sections will leave a blank line at the end, which will be interpreted as a paragraph break when the file is read back in by LaTeX. These kludges correct that problem.

```
\newcommand{\quotedversecorrectiontextnote}[1][0pt]
{\iftextnoteson%
\immediate\write\textnotes{\string\noindent\string\kern-#1}%
\fi}
\newcommand{\quotedversecorrectionemendation}[1][0pt]
{\ifemendationson%
\immediate\write\emendations{\string\noindent\string\kern-#1}%
\fi}
\newcommand{\quotedversecorrectionexplanatory}[1][0pt]
{\ifexplanon%
\immediate\write\explanations{\string\noindent\string\kern-#1}%
\fi}
```

20.23 Marking stanza breaks on page turns

Use the macro below to change the symbol used to mark cases in which there is or is not a stanza break at a page turn.

```
\newcommand{\stanzaatbottom}[1]{\global\edef\stanzaatbottomvalue{#1}}
\stanzaatbottom{*}
\newcommand{\nostanzaatbottom}[1]{\global\edef\nostanzaatbottomvalue{#1}}
\nostanzaatbottom{\relax}
```

20.24 Parallel Texts

First, two utility macros from the memoir class:

\cleartorecto A repeat of \cleardoublepage; clears to a recto (odd-numbered) page.

```
\def\cleartorecto{\clearpage\if@twoside \ifodd\c@page\else
  \hbox{}\thispagestyle{empty}%
  \newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}

```

`\cleartoverso` Clears to a verso (even-numbered) page.

```
\def\cleartoverso{\clearpage\if@twoside
  \ifodd\c@page\hbox{}\thispagestyle{empty}%
  \newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}

```

Macros for forcing the marginal line numbers to the left side or the right side. These marginal line numbers, unlike those made with `\marginpar`, do not float. This code is modelled after code from the memoir class, by Peter Wilson.

`\pmclleftsidepar` As in the memoir class, these sidepars do not float.

```
\newdimen\pmclsavsk
\newcount\pmclsavsf
\def\pmclbsphack{%
  \relax
  \ifhmode
    \pmclsavsk\lastskip
    \pmclsavsf\spacefactor
  \fi}
\def\pmclesphack{%
  \relax
  \ifhmode
    \spacefactor\pmclsavsf
    \ifdim\pmclsavsk>0pt
      \ignorespaces
    \fi
  \fi}
% \newcommand{\pmclleftsidepar}[1]{%
% \pmclbsphack\leavevmode\vadjust{%
%     \rlap{\kern-\parindent%
%    \kern -\marginparsep \kern -\marginparwidth % at left
%       \vbox to 0pt{%
%        \kern \pmclsideparvshift%           % vertical shift to align top text lines
%        \begin{minipage}{\marginparwidth}%
%  #1%
%  \end{minipage}%
%     \vss}}}\pmclesphack%
% } % version 2.44
\newcommand{\pmclleftsidepar}[1]{%
\pmclbsphack\strut\vadjust{% changed
\rlap{\kern-\parindent%
\kern -\marginparsep \kern -\marginparwidth% at left
%       \vbox to 0pt{%
%        \kern \pmclsideparvshift%           % vertical shift to align top text lines
%        \begin{minipage}{\marginparwidth}%
\setbox0=\vtop to 0pt{% added
\begin{minipage}[t]{\marginparwidth}% added
#1%
\end{minipage}%
%     \vss}}}\pmclesphack%
\vss}% changed
\vtop to 0pt{\kern\pmclsideparvshift % default should be 0pt
\kern-\dp\strutbox \kern-\ht0 \box0 \vss}}}\pmclesphack}% changed

\newcommand{\pmclrightsidepar}[1]{%
% \pmclbsphack\leavevmode\vadjust{%
\pmclbsphack\strut\vadjust{%
\rlap{\kern-\parindent%
%    \kern \textwidth \kern -\marginparsep        % at right
\kern\textwidth\kern\marginparsep% at right
%       \vbox to 0pt{%
%        \kern \pmclsideparvshift%           % vertical shift to align top text lines
%        \begin{minipage}{\marginparwidth}%
\setbox0=\vtop to 0pt{% added
\begin{minipage}[t]{\marginparwidth}% added
#1%
\end{minipage}%
%     \vss}}}\pmclesphack%
\vss}% changed
\vtop to 0pt{\kern\pmclsideparvshift% default should be 0pt
\kern-\dp\strutbox \kern-\ht0 \box0 \vss}}}\pmclesphack}% changed

```

Booleans for parallel texts

```
\newif\ifparalleltexts
\paralleltextsfalse
\newif\ifrecto
\rectofalse
\newif\ifverso
\versofalse
```

Booleans are needed to describe situations in which poems and stanzas are left open at the page break. \finishrectopage and \finishversopage will close poems and stanzas (so as not to generate errors because of environments left open), although as far as the poem is concerned the stanza or poem may still be open. (I will say that the stanza or poem in such a case is "artificially closed.") There are two of them for each case of artificial closure. The first boolean (for each stanza or poem artificially closed) must reflect whether the stanza or poem has been (artificially) closed, to reopen on the next page of the same kind (recto or verso). If a poem or stanza has been (artificially) closed, the \mark will not be changed (so that it will still reflect whether the reader should understand there to be a stanza break at the bottom of the page or not). The second boolean for each case of artificial closure reflects whether the \mark has been changed or not.

```
\newif\ifrectopoempending
\rectopoempendingfalse
\newif\ifversopoempending
\versopoempendingfalse
\newif\ifrectostanzapending
```

```
\rectostanzapendingfalse
\newif\ifversostanzapending
\versostanzapendingfalse
\newif\ifrectostanzastillopen
\rectostanzastillopenfalse
\newif\ifversostanzastillopen
\versostanzastillopenfalse
\newif\ifrectopoemstillopen
\rectopoemstillopenfalse
\newif\ifversopoemstillopen
\versopoemstillopenfalse
```

A further set of booleans is required for emendations and explanatory notes. Again the idea is that if a poem is artificially closed, but has already had its first emendation or explanatory note written to the external notes files, poemscol will remember that the next emendation or explanatory note it sees is not the first one for that poem and thus won't set the heading information for that poem in the notes section all over again.

```
\newif\ifrectoemendationspending
\rectoemendationspendingfalse
\newif\ifversoemendationspending
\versoemendationspendingfalse
\newif\ifrectoexplanationspending
\rectoexplanationspendingfalse
\newif\ifversoexplanationspending
\versoexplanationspendingfalse
```

Counters for parallel texts: New counters are needed, not only for the recto and verso lines, but also for the scratch value which determines whether or not it is time to output a marginal line number. These are saved at the end of every page, and restored at the beginning of every page of the same kind.

```
\newcounter{rectolinecounter}
\setcounter{rectolinecounter}{1}
\newcounter{rectoindexscratch}
\setcounter{rectoindexscratch}{1}
\newcounter{versolinecounter}
\setcounter{versolinecounter}{1}
\newcounter{versoindexscratch}
\setcounter{versoindexscratch}{1}
```

\startparalleltexts Clears to a verso (even-numbered) page, fixes where to put line numbers, modifies poem and stanza environments, changes where marginal line numbers are set, sets the booleans.

```
\newcommand{\startparalleltexts}{%
\leavevmode
\cleartoverso
\paralleltextstrue
\renewenvironment{poem}{\raggedright
    \language=255%no hyphenation in verse
    \noemendyettrue
```

```
     \noexplainyettrue
     \setcounter{verselinenumber}{0}\setcounter{printlineindex}{0}
     \nobreak\begin{verse}
     \inpoemtrue\nobreak\mark{\relax}}
     % must end poem if parallel text. Don't worry,
     % begin recto page and begin verso page will
     % reopen the poem and reset the line counter to the right value
     % if it was finish recto page or finish verso page
     % that closed the poem rather than your own declaration.
{\ifversopoemstillopen
       \end{verse}
\inpoemfalse\mark{\relax}
   \else
      \ifrectopoemstillopen
  \end{verse}
  \inpoemfalse\mark{\relax}
      \else
\end{verse}
     \inpoemfalse\mark{\relax}
     \goodbreak\afterpoemskip%\bigskip
\fi\fi     }
\renewenvironment{stanza}{\penalty-100\instanzatrue\mark{\nostanzaatbottomvalue\ }}
% must end stanza if parallel text. Don't worry,
% begin recto page and begin verso page will
% reopen the stanza and reset the line counter to the right value
% if it was finish recto page or finish verso page
% that closed the stanza rather than your own declaration.
{\ifversostanzastillopen
       \relax
   \else
      \ifrectostanzastillopen
  \relax
      \else
 \nobreak\mark{\stanzaatbottomvalue\ }\nobreak
\ifspeciallinelock%
\relax\else%
\nolinebreak\incrementverselinenumber%
\fi%
\speciallinelockfalse%
 \nobreak
 \instanzafalse

 \stanzaskip
\fi\fi
}
\renewcommand{\putverselinenumber}{\nolinebreak\begin{marginenvironment}%
  \nolinebreak
% \marn{\hfil\textrm{\theverselinenumber}}
% \marginpar{\hfil\textrm{\scriptsize\theverselinenumber}}
  \pmclleftsidepar{\textrm{\scriptsize\theverselinenumber}}%
  \nolinebreak
  \end{marginenvironment}%
  \nolinebreak
}
}
```

\saveversoline

Macros for saving and restoring line numbers:

```
\newcommand{\saveversoline}{\setcounter{versolinecounter}%
 {\value{verselinenumber}}%
\setcounter{versoindexscratch}{\value{printlineindex}}}
```

\restoreversoline

```
\newcommand{\restoreversoline}{\setcounter{verselinenumber}%
  {\value{versolinecounter}}%
\setcounter{printlineindex}{\value{versoindexscratch}}}
```

\saverectoline

```
\newcommand{\saverectoline}{\setcounter{rectolinecounter}%
 {\value{verselinenumber}}%
\setcounter{rectoindexscratch}{\value{printlineindex}}}
```

\restorerectoline

```
\newcommand{\restorerectoline}{\setcounter{verselinenumber}%
  {\value{rectolinecounter}}%
\setcounter{printlineindex}{\value{rectoindexscratch}}}
```

Macros for line cross references:

You will probably want to use line cross references to show which lines on the recto page correspond to which lines on the verso page. When you do this, you will set, in the margin of a line on the recto page, the line number of the corresponding line on the verso side, in square brackets. You may mark that line number with a distinctive identifier. `\synchroflag` is a token register to hold that identifier, and `\setsynchroflag` is a command to put the identifier in the register. For instance, if you are setting the 1805 version of Wordsworth's *The Prelude* on the verso, and the 1850 version on the recto, you might wish to set the "synchroflag" to "1805" so that the line cross reference will look something like "[1805: 147]".

\setsynchroflag

```
\newtoks{\synchroflag}
\newcommand{\setsynchroflag}[1]{\synchroflag={#1}}
\setsynchroflag{\relax}
```

Here are commands for defining and calling "synchrolabels", line cross references that tie verso and recto pages:

\synchrolabel

```
\newcommand{\synchrolabel}[1]{\poemlinelabel{#1}}
\newcommand{\hfilll}{\hskip 0pt plus 1 filll}
\newcommand{\synchroref}[1]{%
\pmclrightsidepar{{\textrm{\scriptsize[\the\synchroflag\ref{#1}]}}}%
%  \hfill\rlap{{\textrm{\scriptsize[\the\synchroflag\ref{#1}]}}}
%  \ifnum\theprintlineindex=\thelineindexrepeat
%  \hspace{-7pt}
% \fi
 }
```

Finally, the \startversopage, \finishversopage, \startrectopage and \finishrectopage macros:

\startversopage

```
\newcommand{\startversopage}{
\ifrectostanzastillopen
  \end{stanza}
  \rectostanzastillopenfalse
  \end{poem}
   \rectopoemstillopenfalse
\fi
\ifrectopoemstillopen % from previous page
  \end{poem}
  \rectopoemstillopenfalse
\fi
\ifversopoempending % from previous verso
   \begin{poem}
  \global\versopoempendingfalse
\fi
\ifversostanzapending % from previous verso
  \begin{stanza}
  \global\versostanzapendingfalse
\fi
\ifversoemendationspending
  \noemendyetfalse
  \global\versoemendationspendingfalse
\fi
\ifversoexplanationspending
  \noexplainyetfalse
  \global\versoexplanationspendingfalse
\fi
\versotrue\restoreversoline}
```

\finishversopage

```
\newcommand{\finishversopage}{
\saveversoline\versofalse\
\ifinpoem
  \global\versopoempendingtrue
  \versopoemstillopentrue
\fi
\ifinstanza
   \global\versostanzapendingtrue
   \versostanzastillopentrue
   \global\versopoempendingtrue
     \versopoemstillopentrue
\fi
\ifnoemendyet
 \relax
 \else
   \global\versoemendationspendingtrue
    \noemendyetfalse
\fi
\ifnoexplainyet
   \relax
    \else
       \global\versoexplanationspendingtrue
       \noexplainyetfalse
\fi
% \clearpage % keep in comment until footnotes problems are worked out
}
```

\startrectopage

```
\newcommand{\startrectopage}{
\ifversostanzastillopen
  \end{stanza}
  \versostanzastillopenfalse
  \end{poem}
  \versopoemstillopenfalse
\fi
\ifversopoemstillopen
  \end{poem}
  \versopoemstillopenfalse
\fi
\ifrectopoempending
  \begin{poem}
  \global\rectopoempendingfalse
\fi
\ifrectostanzapending
  \begin{stanza}
  \global\rectostanzapendingfalse
\fi
\ifrectoemendationspending
  \noemendyetfalse
  \global\rectoemendationspendingfalse
\fi
\ifrectoexplanationspending
  \noexplainyetfalse
  \global\rectoexplanationspendingfalse
\fi
\rectotrue\restorerectoline}
```

\finishrectopage

```
\newcommand{\finishrectopage}{\saverectoline\rectofalse
\ifinpoem
  \global\rectopoempendingtrue
  \rectopoemstillopentrue
\fi
\ifinstanza
   \global\rectostanzapendingtrue
   \rectostanzastillopentrue
   \global\rectopoempendingtrue
     \rectopoemstillopentrue
\fi
\ifnoemendyet
  \relax
  \else
    \global\rectoemendationspendingtrue
    \noemendyetfalse
\fi
\ifnoexplainyet
  \relax
  \else
    \global\versoexplanationspendingtrue
    \noexplainyetfalse
\fi
% \clearpage % until footnote problem is solved
}
```

20.25 Finishing up

\finish The \finish macro reads in the notes, the index (previously created from your .aux file by MakeIndex), and the contents sections and sets them. The notes sections send page information to the contents file as they are set. Then the page information for the index is sent to the contents file. Finally the contents file is closed and read in. If you have an editor's introduction or a list of abbreviations, copy this macro, uncomment out the commented out lines, and issue it all as a \renewcommand for \finish. Be sure to have the command \label{editorsintroduction} on the first page of your introduction.

It usually takes three passes to get the table of contents and the index files correct.

The consequence of doing using \finish is that the table of contents comes last. There are ways around this that are described earlier. The notes sections are all typeset in the \small size.

```
\newcommand{\finish}{
\small\bf \hyphenchar\font=45
\small\it \hyphenchar\font=45
\small\rm \hyphenchar\font=45

\ifemendationson
   \newpage
   \immediate\closeout\emendations
   \input \jobname.emd
\fi
\iftextnoteson
   \newpage
   \immediate\closeout\textnotes
   \input \jobname.ent
   \newpage
 \fi
 \ifexplanon
      \immediate\closeout\explanations
      \input \jobname.enx
      \newpage
 \fi
 \ifindexingon
 \printindex
 \fi
 \ifpoemcontentson
%      \immediate\write\poemcontents{\string\vspace{10pt}\string\par}
%      \immediate\write
% \poemcontents{\string\contentspoemtitlefont\ Introduction to the Notes}
%      \immediate\write
% \poemcontents{\string~ /  \string~\ \string\pageref{editorsintroduction} \string\par}
%      \immediate\write\poemcontents{\string\par \string\smallskip}
%      \immediate\write\poemcontents{\string\contentspoemtitlefont\ Abbreviations}
%      \immediate\write
% \poemcontents{\string~ /  \string~\ \string\pageref{abbreviations} \string\par}
%      \immediate\write\poemcontents{\string\par \string\smallskip}
     \ifemendationson
      \immediate\write\poemcontents{\string\contentspoemtitlefont\
      \the\emendationscontentsname}
      \immediate\write
\poemcontents{\string~ /  \string~\ \string\pageref{emendationnotes} \string\par}
      \immediate\write\poemcontents{\string\par \string\smallskip}
     \fi
     \iftextnoteson
      \immediate\write\poemcontents{\string\contentspoemtitlefont\
      \the\textnotescontentsname}
      \immediate\write
\poemcontents{\string~ /  \string~\ \string\pageref{textualnotes} \string\par}
      \immediate\write\poemcontents{\string\par \string\smallskip}
     \fi
     \ifexplanon
       \immediate\write
\poemcontents{\string\contentspoemtitlefont\ \the\explanationscontentsname}
       \immediate\write
\poemcontents{\string~ /  \string~\ \string\pageref{explanatorynotes}}
     \fi
     \ifindexingon
        \immediate\write\poemcontents{\string\par \string\smallskip}
        \immediate\write
\poemcontents{\string\contentspoemtitlefont\ \the\poemindexcontentsname}
        \immediate\write
\poemcontents{\string~ /  \string~\ \string\pageref{indexpage}\string\par}
    \fi
    \immediate\closeout\poemcontents
    \input \jobname.ctn
 \fi
 \clearpage
\thispagestyle{volumetitlestyle}
\vspace{106pt}
\begin{center}
{\poemtitlefont Note}
\end{center}

\noindent The symbol \stanzaatbottomvalue\ is used to indicate
a space between sections of a poem
wherever such spaces are lost in pagination.
\cleardoublepage
}
```

`\testforauxonfirstrun` Put this in the preamble. It suppresses `\finish`, `\puttextnotes`, `\putemendations`, and `\putexplanatory` unless there is an .aux file from a previous run, this avoiding some error messages.

```
\newcommand{\testforauxonfirstrun}{\IfFileExists{\jobname.aux}{\relax}
{\let\finish=\relax
\let\puttextnotes=\relax
\let\putexplanatory=\relax
\let\putemendations=\relax
\typeout{Run again to read in endnotes files}}}
```

`\putpoemcontents` You may wish to do the tasks done by `\finish` separately. This will give you more flexibility. `\putpoemcontents` inserts the table of contents file from a prior run. Be sure to issue it *before* you issue `\makepoemcontents`.

```
\newcommand{\putpoemcontents}{
\IfFileExists{\jobname.ctn}{\input \jobname.ctn
\clearpage \resetpagestyle\normalsize}
{\typeout{Run again to input contents file here}}
}
```

`\hyphenationforsmall` This command restores hyphenation for the backmatter sections of the book. Hyphenation is turned off in the main body of the book, because verse should not be hyphenated. This macro turns hyphenation on, but only for small roman, italic, and bold fonts. Backmatter sections are all set in `\small`. `\putemendations`, `\puttextnotes`, and `\putexplanatory` all call `\hyphenationforsmall`.

```
\newcommand{\hyphenationforsmall}{\small\bf \hyphenchar\font=45
\small\it \hyphenchar\font=45
\small\rm \hyphenchar\font=45
}
```

`\putemendations` This macro closes external file for emendations, reads it in, and puts an entry for the emendations into the table of contents file. The macro tests to see whether there is an .aux file from a previous run before doing this, and complains in the .log file if the .aux file does not exist.

```
\newcommand{\putemendations}{
\ifemendationson
   \newpage
   \hyphenationforsmall
   \immediate\closeout\emendations
   \input \jobname.emd
   \ifpoemcontentson
       \ifemendationson
\immediate\write\poemcontents{\string\contentspoemtitlefont\ \the\emendationscontentsname}
\immediate\write
  \poemcontents{\string~ \string\contentsleaders  \string~\
  \string\pageref{emendationnotes} \string\par}
\immediate\write\poemcontents{\string\par \string\smallskip}
        \fi
\fi
\else
 \relax
\fi}
```

`\putexplanatory` Does the same for explanatory notes.

```
\newcommand{\putexplanatory}{
\ifexplanon
   \newpage
   \hyphenationforsmall
   \immediate\closeout\explanations
   \input \jobname.enx
   \ifpoemcontentson
       \ifexplanon
\immediate\write\poemcontents{\string\contentspoemtitlefont\ \the\explanationscontentsname}
\immediate\write
  \poemcontents{\string~ \string\contentsleaders  \string~\
  \string\pageref{explanatorynotes} \string\par}
\immediate\write\poemcontents{\string\par \string\smallskip}
        \fi
\fi
\else
 \relax
\fi}
```

`\puttextnotes` Does the same for textual notes.

```
\newcommand{\puttextnotes}{
\iftextnoteson
   \newpage
   \hyphenationforsmall
   \immediate\closeout\textnotes
   \input \jobname.ent
   \ifpoemcontentson
       \iftextnoteson
\immediate\write\poemcontents{\string\contentspoemtitlefont\ \the\textnotescontentsname}
\immediate\write
  \poemcontents{\string~ \string\contentsleaders  \string~\
  \string\pageref{textualnotes} \string\par}
\immediate\write\poemcontents{\string\par \string\smallskip}
       \fi
\fi
\else
 \relax
\fi}
```

\putpoemindex Reads in the index of titles and first lines and adds an entry for the index into the table of contents.

```
\newtoks{\poemindexname}
\poemindexname={Index of Titles and First Lines}
\newcommand{\changepoemindexname}[1]{\poemindexname={#1}}
\newtoks{\poemindexheadername}
\poemindexheadername={Index of Titles and First Lines}
\newcommand{\changepoemindexheader}[1]{\poemindexheadername={#1}}
\newtoks{\poemindexcontentsname}
\poemindexcontentsname={Index of Titles and First Lines}
\newcommand{\changepoemindexcontentsname}[1]{\poemindexcontentsname={#1}}
\newcommand{\putpoemindex}{\ifindexingon
 \printindex
 \fi
 \ifpoemcontentson\ifindexingon
     \immediate\write\poemcontents{\string\par \string\smallskip}
     \immediate\write
\poemcontents{\string\contentspoemtitlefont\ \the\poemindexcontentsname}
     \immediate\write
\poemcontents{\string~\string\contentsleaders \string~\ \string\pageref{indexpage}\string\pa
  \fi\fi}
\newcommand{\putmultiplepoemindex}[4]{%
\ifindexingon
\poemindexname={#2}
\poemindexheadername={#3}
\poemindexcontentsname={#4}
\poemindexlabelname={#4}
\printindex[#1][#2]
\fi
\ifpoemcontentson\ifindexingon
     \immediate\write\poemcontents{\string\par \string\smallskip}
     \immediate\write
\poemcontents{\string\contentspoemtitlefont\ \the\poemindexcontentsname}
     \immediate\write
\poemcontents{\string~ \string\contentsleaders \string~\
\string\pageref{\the\poemindexlabelname}\string\par}
```

```
  \fi\fi
}
\newcommand{\noteaboutstanzamarkpage}{\begin{volumetitlepage}
%\thispagestyle{volumetitlestyle}
\vspace{106pt}
\begin{center}
{\poemtitlefont Note}
\end{center}

\noindent The symbol \stanzaatbottomvalue\ is used to indicate
a space between sections of a poem
wherever such spaces are lost in pagination.
\end{volumetitlepage}}
```

20.26 Visual formatting

The ideal in LaTeX is to separate content markup and formatting as completely as possible. In the text itself you simply mark the logical units (a poem, a title, a stanza, a textual note), and a header program like this interprets the meaning of those terms typographically. However, one still winds up having to do some visual formatting. Sometimes you have to enlarge or shrink a page by one line in order to avoid an intractable widow or club line, and when one does so one must be careful to do the same to the other member of the two-page spread of which that page is a part. These commands enable one to wiggle pagelength enough to handle these problems.

```
\newcommand{\longpage}{\enlargethispage{\baselineskip}}
\newcommand{\shortpage}{\enlargethispage{-\baselineskip}}
```

20.27 Special environments: cjquotation and theindex

An environment for special prose quotations within verse as in “Chief Joseph of the Nez Perce”

cjquotation

```
\newenvironment{cjquotation}
                {\tolerance=2000%
                \vspace{-\bigskipamount}
               \restoreverselinenumber
               \language=255
               \noindent\relax
               \list{}{\listparindent 0em%
                        \itemindent 0em%
                        \rightmargin 2em%
                        \leftmargin 2em%
                        \topsep 0em%
                        \parskip 0em%
                        \itemindent    \listparindent
                        \rightmargin   \leftmargin
                        \parsep        \z@ \@plus\p@}%
                \item[]
                }
               {\endlist}
```

theindex And the environment for the index:

```
\newtoks{\poemindexlabelname}
\poemindexlabelname={indexpage}
\newcommand{\changepoemindexlabelname}[1]{\poemindexlabelname={#1}}
\renewenvironment{theindex}
     {{\par\clearpage{\pagestyle{volumetitlestyle}\cleardoublepage}
     \thispagestyle{volumetitlestyle}}
     \makeatletter
     \def\pmclidxitem{\par\hangindent 1em}
\fancyhead[RO,LE]{{\small\textrm{\thepage}}}
\fancyhead[CO]{{\small{\em \the\poemindexheadername}}}
\fancyhead[CE]{{\small{\em \the\poemindexheadername}}}
\fancyfoot{}
     \label{\the\poemindexlabelname}
     \lefthyphenmin=2\tolerance=500\language=0
     \normalfont \backmattertextfont
               \raggedright
                \begin{multicols}
{2}[\backmattersink\begin{center}{\normalfont
\backmatterheaderfont \the\poemindexname}\end{center}
\backmatterafterheadersink]%
                \let\item\pmclidxitem}
               {\end{multicols}\clearpage}
\makeatother
```

Finally, some other miscellaneous environments.

```
\newenvironment{epigraphquote}{\list{}{\rightmargin 1.5em%
                                       \leftmargin  1.5em%
                                       \topsep 0pt%
                                       \partopsep 0pt%
                                       \listparindent 0pt%
                                       }%
                \item\relax}
               {\endlist}
\newenvironment{volumetitlepagequote}{\list{}{\rightmargin \voladditionalamount%
                                              \leftmargin \voladditionalamount%
                                       \topsep 0pt%
                                       \partopsep 0pt%
                                       \listparindent 0pt
                                       }%
                \item\relax}
               {\endlist}
```