

The `lips` LaTeX package

Chicago Manual text ellipses (Frankenstein's lips)

Matt Swift <swift@alum.mit.edu>

Version: 2 Date: 1999/02/28

Documentation revision: 1999/02/28

Abstract

The `\lips` command generates text ellipses that are closer to what *The Chicago Manual of Style* suggests than what `\dots` produces. It does the right thing in most circumstances, and so is easy to use.

Contents

I	Discussion	2
1	Text Ellipses	2
2	Line breaking considerations	2
3	Limitations	3
II	Implementation	4
4	Version control	4
5	The macro	4
III	Testing	8
6	Ante- <code>\lips</code>	8
7	Post- <code>\lips</code>	8

source	result
Hello\lips.\And	Hello. . . And
Hello\lips.\And	Hello. . . And
Hello.\lips\And	Hello. . . And
Hello.\lips\And	Hello. . . And
one,\lips,\three	one, . . . , three
one,\lips,\three	one, . . . , three
Hello\lips\and	Hello . . . and
Hello\lips\and	Hello . . . and
Hello!\lips\And	Hello! . . . And
Hello!\lips\And	Hello! . . . And

Figure 1: Examples of correct usage of `\lips`.

Part I

Discussion

1 Text Ellipses

`\lips` The macro `\lips` should be used for all text ellipses, that is, ellipses not part of a mathematical expression.

The Chicago Manual of Style (University of Chicago Press 1993) does not acknowledge different spacings after different marks of punctuation, distinguish interword from intersentence space, or give rules about where to break a line near an ellipsis. Given a context where such spaces do vary and rules for linebreaks can be given, I arrived at the following rules for handling them around ellipses.

1. Pre-, post-, and intra-ellipsis space is normal intra-word space, which should be 3-to-em. In \LaTeX there is also some glue (it's a "rubber space").
2. When non-period punctuation p follows `\lips`, p is followed by the space that normally follows p .
3. A period following `\lips` is treated as if the period preceded `\lips`. Conceptually, an ellipsis never precedes a period, so conceptually there is only one case; but it is convenient to be able to type either `\lips.` or `.\lips`.
4. Lines are a tiny bit breakable before ellipses and a tiny bit breakable after ellipses that are not followed by punctuation.

In figure 1 are examples of correct usage and in figure 2 is a comparison of `\lips` to `\dots`. You can substitute any of the characters `!"()*+,-/:;=?@[]' '<>~{|}` for the exclamation point in the examples. There can be additional contiguous spaces after `\lips`, just like after any command name. There can also be contiguous spaces before `\lips`.

2 Line breaking considerations

`\LPNobreakList` When `\lips` is followed by certain punctuation (`, : ; ? ! '] }`/ by default), we never

source	result
Hello\dotsc And	Hello... And
Hello\lipsc And	Hello. . . And
Hello\dotsc And	Hello . . . And
Hello\lipsc And	Hello. . . And
Hello.\dotsc And	Hello... And
Hello.\lipsc And	Hello. . . And
Hello.\dotsc And	Hello. . . And
Hello.\lipsc And	Hello. . . And
one,\dotsc three	one,..., three
one,\lipsc three	one, . . . , three
one,\dotsc three	one, . . . , three
one,\lipsc three	one, . . . , three
Hello\dotsc and	Hello...and
Hello\lipsc and	Hello . . . and
Hello\dotsc and	Hello . . and
Hello\lipsc and	Hello . . . and
Hello!\dotsc And	Hello!... And
Hello!\lipsc And	Hello! . . . And
Hello!\dotsc And	Hello! . . . And
Hello!\lipsc And	Hello! . . . And

Figure 2: Comparison of \lips with \dots.

want to break a line after the ellipsis and before that punctuation. For the other punctuation, such as open parenthesis and open quote, we want normally breakable space following the ellipsis. `\LPNobreakList` is a list of those characters before which `\lips` should *never* break a line. To force unbreakable space following `\lips`, follow `\lips` with a tie (`~`).

3 Limitations

`\lips` does the right thing after `\par`, but after `\indent` or `\noindent` there is an extra preceding space (see the final section on testing for an example). Suggestions on how to handle this problem are welcome.

Part II

Implementation

4 Version control

```
\fileinfo These definitions must be the first ones in the file.
\DoXUsepackage 1 \def\fileinfo{Chicago Manual text ellipses (Frankenstein's lips)}
\HaveECitationS 2 \def\DoXPackageS {}
\fileversion 3 \def\initelyHaveECitationS {}
\filedate 4 \def\fileversion{v2}
\docdate 5 \def\filedate{1999/02/28}
\PPOptArg 6 \def\docdate{1999/02/28}
7 \edef\PPOptArg {%
8 \filedate\space \fileversion\space \fileinfo
9 }

If we're loading this file from a \ProcessDTXFile command (see the compsci
package), then \JustLoadInformation will be defined; otherwise we assume it is
not (that's why the FunkY Name).

If we're loading from \ProcessDTXFile, we want to load the packages listed in
\DoXPackageS (needed to typeset the documentation for this file) and then bail
out. Otherwise, we're using this file in a normal way as a package, so do nothing.
\DoXPackageS, if there are any, are declared in the dtx file, and, if you're reading
the typeset documentation of this package, would appear just above. (It's OK to
call \usepackage with an empty argument or \relax, by the way.)

10 \makeatletter% A special comment to help create bst files. Don't change!
11 \@ifundefined{JustLoadInformation} {%
12 }{% ELSE (we know the compsci package is already loaded, too)
13 \UndefinedCS\JustLoadInformation
14 \SaveDoXVarS
15 \eExpand\csname DoXPackageS\endcsname\In {%use \csname in case it's undefined
16 \usepackage{#1}%
17 }%
18 \RestoreDoXVarS
19 \makeatother
20 \endinput
21 }% A special comment to help create bst files. Don't change!

Now we check for LATEX2ε and declare the LaTeX package.
22 \NeedsTeXFormat{LaTeX2ε}
23 \ProvidesPackage{lips}[\PPOptArg]
```

5 The macro

`\lips` If we're in math mode, raise an error. If we're in vertical mode, leave it and do an ellipsis not breakable at the end. If we're in horizontal mode, call `\lp@lips@hmode`.

Normally I would use `\NewRobustCommand`, but this seems not enough reason to require the *moredefs* package. We reserve the name `\lips` by hand. In the future, this package may require *moredefs*.

```
24 \newcommand*\lips {}
```

```

25 \let\lips\relax
26 \DeclareRobustCommand\lips {%
27   \ifmmode
28     \def\sc@t@a {%
29       \PackageError{lips}
30         {Don't use \protect\lips \space in math mode; use \protect\dots.}\@ehc
31     }%
32   \else
33     \ifvmode
34       \def\sc@t@a {%
35         \leavevmode
36         .\nobreak\ .\nobreak\ .\nobreak\ %
37       }%
38     \else
39       \let\sc@t@a\lp@lips@hmode
40     \fi
41   \fi
42   \sc@t@a
43 }

```

`\lp@lips@hmode`
`\GobbleIgnoreSpaces`

The horizontal mode case. Then we remove the last skip on the horizontal list. If the next character is a period, we want to act as if the period preceded `\lips`, so we plonk down a period, a minimally-breakable space, then an ellipsis ending with normally-breakable space. This keeps the ellipsis on the same line as the sentence it follows if at all possible, but allows it to spill onto the next line if there's no other way to avoid an overfull line. Then we want to gobble up the period that's about to come, and ignore any spaces after it, since we've already inserted proper space.

If the next character is not a period, we output a minimally-breakable space and an ellipsis. To decide whether to follow it with unbreakable or normally-breakable space, we need to know more about the next character than just that it is not a period, so we `\let` it to scratch `a` and call `\lp@lips@check@punct`.

```

44 \newcommand\lp@lips@hmode {%
45   \unskip
46   \@ifnextchar . {%
47     .\penalty9999\ .\nobreak\ .\nobreak\ .\ %
48     \GobbleIgnoreSpaces
49   }{% ELSE
50     \penalty9999\ .\nobreak\ .\nobreak\ .%
51     \futurelet\sc@t@a\lp@lips@check@punct
52   }%
53 }
54 \newcommand\GobbleIgnoreSpaces [1] {%
55   \ignorespaces
56 }

```

`\LPNobreakList`
`\lp@<punctuation>`

We'd like to use `\ifcat.\sc@t@a` for this, but that won't work when we get a command sequence in scratch `a`. Bummer. Do I really need to define all this punctuation now? It protects against something becoming an active char or otherwise changing catcodes after the *lips* package is loaded.

```

57 \newcommand*\lp@bang {}
58 \newcommand*\lp@rparen {}
59 \newcommand*\lp@comma {}

```

```

60 \newcommand*\lp@slash {}
61 \newcommand*\lp@colon {}
62 \newcommand*\lp@semic {}
63 \newcommand*\lp@q {}
64 \newcommand*\lp@rbrack {}
65 \newcommand*\lp@rquote {}
66 \newcommand*\lp@tilde {}
67
68 \let\lp@bang=!
69 \let\lp@rparen=)
70 \let\lp@comma=,
71 \let\lp@slash=/
72 \let\lp@colon=:
73 \let\lp@semic=;
74 \let\lp@q=?
75 \let\lp@rbrack=]
76 \let\lp@rquote='
77 \let\lp@tilde=~
78
79 % \egroup
80

```

I'm making an attempt to put these in order of expected frequency. A doublequote could be at the beginning or end of a quotation, so half the time you'll have to correct by adding {}, so let's leave it out so there's one less to check.

```

81 \newcommand \LPNobreakList {%
82   \lp@comma \lp@colon \lp@semic
83   \lp@q \lp@bang
84   \lp@rparen \lp@rquote \lp@rbrack \egroup
85   \lp@slash
86 }

```

`\lp@lips@check@punct` Add nonbreakable space if scratch a is in `\LPNobreakList`; add normally-breakable space otherwise. Add no space at all if scratch a is a tie—the tie will add non-breakable space. We use just the guts of `\@tfor` for efficiency.

```

87 \newcommand\lp@lips@check@punct {%
88   \ifx\sc@t@a\lp@tilde
89   \else
90     \@tempwatrue
91     \let\@fortmp\PunctList
92     \expandafter\@tforloop\LPNobreakList\@nil\@nil\@@\reserved@a {%
93       \ifx\sc@t@a\reserved@a
94       \@tempwafalse
95       \@break@tfor
96         \fi
97       }%
98       \if@tempwa
99         \%
100      \else
101        \nobreak\%
102      \fi
103    \fi
104 }

```

You might want to compare the L^AT_EX version (as of 95/12/01). `\fontdimen3` is the interword stretchability, not interword space.

```
\DeclareTextCommandDefault{\textellipsis}{%
  .\kern\fontdimen3\font
  .\kern\fontdimen3\font
  .\kern\fontdimen3\font
}
\def\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}
\DeclareRobustCommand{\dots}{%
  \ifmmode\mathellipsis\else\textellipsis\fi
}
\let\ldots\dots
```

Part III

Testing

6 Ante-\lips

This is some testing. We need some text here as the first paragraph after the \section.

```
:...
:... text.
:... )
;...
;... text.
;... )
!...
!... text.
!... )
t...
t... text.
t... )
T...
T... text.
T... )
,...
,... text.
,... )
... and (\par \lips and)
and (\par and)
... (\par\indent \lips and)
and (\par\indent and)
... and (\par\noindent \lips and)
and (\par\noindent and)
And...
(And.\lips \par)
And.
(And. \par)
```

7 Post-\lips

\lips before a tie: One two three. . . tiedtolips (I doubt you'd want to do this.)
\lips before), e.g., punctuation transparent to spacefactors:
(One two three. . .) This follows the) with interword space, to compare.
(One two three. . .) This should follow the) with intersentence space.
(One two three. . .) This should follow the) with intersentence space.
\lips before shortverb: . . . shortverb verbatim stuff
\lips before \textsf: . . . sansserif

\lips before \\: . . .
second line of a center environment

`\lips` before `\#`: ... `#`
`\lips` before `<`: ... `;`
`\lips` before an abbrev: ... `Frankenstein`
`\lips` before `\cite`: ... (University of Chicago Press 1993)
`\lips` before `$`: ... $f = ma$
`\lips` before `&`: ... second column of a `tabular`

References

University of Chicago Press. 1993. *The Chicago Manual of Style*. 14th ed.
Chicago: University of Chicago Press.