

Typesetting ‘lettrines’ in L^AT_EX 2_ε documents

Daniel FLIPO

daniel.flipo@free.fr

1 Introduction

The file `lettrine.dtx`¹, provides a command `\lettrine` which requires two mandatory arguments, and an optional one.

Adding `\usepackage{lettrine}` in the preamble of a document defines the command `\lettrine`, the simplest use of which is `\lettrine{<letter>}{<text>}`. It produces a dropped capital `<letter>` (2 lines high), followed by `<text>` typeset in small caps, and the rest of the paragraph is wrapped around the dropped capital.

Various parameters are provided to control the size and layout of the dropped capital and match the requirements described in the books

- “Lexique des règles typographiques en usage à l’Imprimerie nationale” troisième édition (1994), ISBN-2-11-081075-0,
- “Mise en page et impression” Yves PERROUSSEAUX, ISBN-2-911220-01-3.

The parameters can be set using David Carlisle’s `keyval.sty` syntax:

- `lines=<integer>` sets how many lines the dropped capital will occupy (default=2);
- `depth=<integer>` sets the number of lines to be reserved under the baseline, this is meant for dropped capital with positive depth, like Q (default=0);
- `lhang=<decimal>` ($0 \leq \text{lhang} \leq 1$) sets how much of the dropped capital’s width should hang into the margin (default=0);
- `loversize=<decimal>` ($-1 < \text{loversize} \leq 1$) enlarges the dropped capital’s height: with `loversize=0.1` its height is enlarged by 10% so that it raises above the top paragraph’s line (default=0);
- `lraise=<decimal>` does not affect the dropped capital’s height, but moves it up (if positive), down (if negative); useful with capitals like J or Q which have a positive depth, (default=0);
- `findent=<dimen>` (positive or negative) controls the horizontal gap between the dropped capital and the indented block of text (default=0pt);
- `nindent=<dimen>` shifts all indented lines, starting from the second one, horizontally by `<dimen>` (this shift is relative to the first line, default=0.5em);
- `slope=<dimen>` can be used with dropped capitals like A or V to add `<dimen>` (positive or negative) to the indentation of each line starting from the third one (no effect if `lines=2`, default=0pt);

¹The file described in this section has version number v1.9 and was last revised on 2015/08/31.

- **ante**=<text> can be used to typeset <text> before the dropped capital (typical use is for French guillemets starting the paragraph).
- **image**=<true> (new to version 1.6) will force `\lettrine` to replace the letter normally used as dropped capital by an image in eps format (latex) or in pdf, jpg, etc. format (pdflatex); this needs the `graphicx` package to be loaded in the preamble of course. `\lettrine[image=true]{A}{n exemple}` or just `\lettrine[image]{A}{n exemple}` will load `A.eps` or `A.pdf` instead of letter A. This was suggested by Bill Jetzer. Redefining `\LettrineFont` as `\LettrineFontEPS` still works for compatibility but is deprecated.
- **grid**=<true> (new to version 1.8) will force the vertical skip added above the paragraph starting with `\lettrine` to be rounded up to an integer number of `\baselineskip`. This option is meant for grid typesetting.
- **novskip**=<dimen> (new to version 1.8) overrides `\DiscardVskip` (default=0.2pt). In some cases (see options `lraise`, `loversize` or accented dropped capitals,...) the top of the dropped capital will raise above the top of following text (usually in small caps), this will trigger a corresponding vertical skip above the paragraph starting with `\lettrine`, *only if* this skip exceeds `\DiscardVskip`. Consider enlarging `novskip` (or `\DiscardVskip`) to prevent small vertical skips from being rounded up to `\baselineskip` when using the ‘grid’ option.
- **realheight**=<true> (new to version 1.9) will compute the default height of the letrine so that the top of it is exactly aligned with the top of the text entered as second mandatory argument of `\lettrine` taking possible accents into account. Otherwise, the default height is computed using a customisable string `\LettrineSecondString` instead of the real argument. For backward compatibility, option `realheight` defaults to false and `\LettrineSecondString` to ‘x’.

You probably don’t need this option if you choose to typeset the second mandatory argument of `\lettrine` in small caps (the default). If you change `\LettrineTextFont` to `\relax` or `\upshape`, consider these two examples:

`\lettrine{H}{ello}` you probably would like the top of the ‘L’ to be aligned with the top of the ‘ll’ rather than with the top of the ‘e’, adding option `realheight` does the trick: `\lettrine[realheight]{H}{ello}`.
Global variants : `\LettrineRealHeighttrue` or (without the `realheight` option) `\renewcommand{\LettrineSecondString}{l}`.

`\lettrine{L}{a misère}` option `realheight=true` would align with the top of the ‘L’ with the top of the grave accent, the default is probably better (top of the ‘L’ aligned with the top of the non accented letters).

Example: `\lettrine[lines=4, lraise=0.1, nindent=0em, slope=-.5em]%
{V}{oici} un exemple ...`

Coloured letrines are possible in conjunction with package `color`, examples:
`\lettrine{\textcolor{red}{A}}{n exemple}` or
`\lettrine{\textcolor[gray]{0.5}{A}}{nother}` one
see package `color` for the syntax of colour commands. Another possibility to colour letrines globally is described below, see `\LettrineFontHook`.

Three dimensions, `\LettrineWidth`, `\LettrineHeight` and `\LettrineDepth`, store the final size of the lettrine.

Once `lettrine.sty` will be installed (run `latex` on `lettrine.ins` to extract it), compile and print `demo.tex` to see the possible usage of these parameters.

The default settings can be customized either in a config file `lettrine.cfg` (for a global usage), or on a per document basis, in the preamble of each document. The following list shows the syntax to set them and their default values:

- `\setcounter{DefaultLines}{2}`,
- `\setcounter{DefaultDepth}{0}`,
- `\renewcommand*{\DefaultLoversize}{0}`,
- `\renewcommand*{\DefaultLraise}{0}`,
- `\renewcommand*{\DefaultLhang}{0}`,
- `\LettrineImagefalse`,
- `\LettrineOnGridfalse`,
- `\LettrineRealHeightfalse`,
- `\setlength{\DefaultFindent}{0pt}`,
- `\setlength{\DefaultNindent}{0.5em}`,
- `\setlength{\DefaultSlope}{0pt}`.
- `\setlength{\DiscardVskip}{0.2pt}`.

Instead of giving optional parameters to the `\lettrine` command, it is possible, from version 1.5, to set them on a per character basis in a second config file (suggested by Pascal Kockaert): `\renewcommand{\DefaultOptionsFile}{filename}` in the preamble (or anywhere in the document) will force each call to `\lettrine` to read the file `filename`. See examples of such config files in the subdirectory `contrib`.

The dimensional parameters `findent`, `nindent` and `slope`, can be set in `filename` relative to `\LettrineWidth` if needed. The settings read from this file will be overridden by the optional arguments eventually given to `\lettrine`.

`\LettrineTextFont` sets the font used for the second argument of `\lettrine`, its default definition is `\newcommand{\LettrineTextFont}{\scshape}` (second argument in small caps, this can be changed using `\renewcommand`).

`\LettrineFont` sets the font used for the dropped capital, usually the current font in a (large) size, computed automatically from the number of lines it will fill: the font size is computed so that, a *standard* dropped capital (say X, not Å) when sitting on its baseline, gets its top aligned with the top of the following text (provided `loversize = 0` and `lines ≥ 2`). When `lines = 1`, size is computed as if `lines` was 2.

A hook `\LettrineFontHook` is provided to change the font used for the dropped capital, syntax follows L^AT_EX's low-level font interface (see L^AT_EX Companion,

p.187–192), the `\selectfont` command is issued by `\LettrineFont:`
`\renewcommand{\LettrineFontHook}{\fontfamily{ppl}\fontseries{bx}}%`
`\fontshape{sl}},`

selects Palatino bold expanded slanted for the dropped capital.

`\LettrineFontHook` can also be used to change the colour of all letrines in a (part of) document: `\renewcommand{\LettrineFontHook}{\color[gray]{0.5}}` will colour the letrines following this command in grey.

Important notice: the sizing works fine with *fully scalable* fonts (like the standard PostScript fonts), but might not work well with CM/EC fonts which have two limitations: only a limited number of sizes is available by default (precise adjustments are impossible), and the largest size (25pt or 35pt) is often too small. The CM fonts are now available in PostScript type1 format for free (courtesy of BlueSky/Y&Y), to make them fully scalable, it is mandatory to add `\usepackage{type1cm}` in the preamble of your document. The EC fonts are also available in type1 format for free (thanks to Vladimir Volovich, they are called cm-super), and adding `\usepackage{type1ec}`² in the preamble will make them fully scalable too. So, if you want `lettrine.sty` to work properly with CM or EC fonts, you will need *PostScript versions* of these fonts *and* one of the packages `type1cm.sty` or `type1ec.sty`.

The LM fonts are a good replacement for both CM and EC fonts they are fully scalable, so you should use them instead of CM or EC fonts. `\usepackage{lmodern}` is the command to switch them on (add `\usepackage[T1]{fontenc}` when composing in one of the western languages other than English in order to get proper hyphenation).

You can also consider using one of the standard PostScript fonts (Palatino, Times, Utopia...), or any OpenType font, they are fully scalable too!

Known problems:

- nothing is done to prevent page-breaking in a paragraph starting with a dropped capital; when it happens to hang into the footer, page-breaking has to be done manually;
- `\lettrine` works within ‘quote’ ‘quotation’, ‘abstract’ environments but does not work within ‘center’ environments (except with option `[lines=1]`);
- `\lettrine` does not work within lists;
- if a *list* has to be included in a paragraph starting with a ‘lettrine’, it is necessary to add the command `\parshape=0` just after the end of the list (starting a new paragraph just before or just after the list works too). Remember that ‘quote’, ‘quotation’, ‘abstract’ environments are implemented as *lists* in L^AT_EX.

²This package, available on CTAN, was first released on 2002/07/30.

2 T_EXnical details

This package only runs with L^AT_EX 2_ε and requires keyval.sty

```
1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \RequirePackage{keyval}
```

Default initializations: define the necessary counters, lengths, and commands to hold the default settings and set these default settings. They can be overwritten in file lettrine.cfg.

```
3 \newcounter{DefaultLines}
4 \setcounter{DefaultLines}{2}
5 \newcounter{DefaultDepth}
6 \newcommand*\DefaultOptionsFile{\relax}
7 \newcommand*\DefaultLoversize{0}
8 \newcommand*\DefaultLraise{0}
9 \newcommand*\DefaultLhang{0}
10 \newdimen\DefaultFindent
11 \setlength{\DefaultFindent}{\z@}
12 \newdimen\DefaultNindent
13 \setlength{\DefaultNindent}{0.5em}
14 \newdimen\DefaultSlope
15 \setlength{\DefaultSlope}{\z@}
16 \newdimen\DiscardVskip
17 \setlength{\DiscardVskip}{0.2\p@}
18 \newif\ifLettrineImage
19 \newif\ifLettrineOnGrid
20 \newif\ifLettrineRealHeight
```

Then let's define the necessary internal counters, lengths, and commands.

```
21 \newsavebox{\L@lbox}
22 \newsavebox{\L@tbox}
23 \newcounter{L@lines}
24 \newcounter{L@depth}
25 \newdimen\L@Pindent
26 \newdimen\L@Findent
27 \newdimen\L@Nindent
28 \newdimen\L@lraise
29 \newdimen\L@first
30 \newdimen\L@next
31 \newdimen\L@slope
32 \newdimen\L@height
33 \newdimen\L@novskip
34 \newcommand*\L@file{}
35 \newcommand*\L@hang{}
36 \newcommand*\L@oversize{}
37 \newcommand*\L@raise{}
38 \newcommand*\L@ante{}
39 \newif\ifL@image
40 \newif\ifL@grid
41 \newif\ifL@realh
```

Provide commands for the fonts used to typeset the two mandatory arguments of

`\lettrine.`

`\LettrineTextFont` In French, small caps usually follow the dropped capital.

```
42 \newcommand*{\LettrineTextFont}{\scshape}
```

`\LettrineFontHook` The default size for the dropped capital is computed so that the top of it is exactly aligned with the top of the following text; an extra height (positive or negative) may be added with `Defaultloversize` or with an optional argument `loversize=`. If `lines=1`, the default size for the dropped capital is computed as if `lines=2`.

`\LettrineFont`

`\Lettrine@height` computes the wished height for the dropped capital and stores it into `\L@height`. `\L@height` depends only on `L@lines`, `\L@oversize` and, in case option `realheight=true` on the height of `\L@tbox`. So options *must* be read and `\L@tbox` must be properly initialised *before* executing `\Lettrine@height` (see below in `\@lettrine` code). A default initialisation of `\L@tbox` is provided just in case `\LettrineFont` would be used outside `\lettrine`.

As `\baselineskip` might be a rubber length, we convert it into a ‘dimen’ using `\@tempdima`. `\LettrineFontHook` enables to select another font for the dropped capital. Its default definition is empty (the current text font is used).

```
43 \def\Lettrine@height{%
44   \@tempdima=\baselineskip
45   \setlength{\L@height}{\value{L@lines}\@tempdima}%
46   \ifnum\value{L@lines}>1
47     \addtolength{\L@height}{-\@tempdima}%
48   \fi
49   \ifvoid\L@tbox
50     \sbox{\L@tbox}{\LettrineTextFont{\LettrineSecondString}}%
51   \fi
52   \addtolength{\L@height}{\ht\L@tbox}%
53   \addtolength{\L@height}{\L@oversize\L@height}%
54 }
55 \newcommand*{\LettrineFontHook}{}
56 \newcommand*{\LettrineTestString}{ABCDEFGHIJKLMNOQPRSTUVWXYZ}
57 \newcommand*{\LettrineSecondString}{x}
58 \newcommand*{\LettrineFont}{%
59   \Lettrine@height
```

`\L@height` now holds the exact height required for the dropped capital, setting `\fontsize` to that height would not give the expected result (capital too small), some computing has to be done: we measure the maximal capitals’ height and compute a scaling factor (always ≥ 1). All capitals are expected to have the same height, in case this assumption would be wrong for some special font, `\LettrineTestString` can be customised to any non empty subset of capitals.

```
60   \sbox{\@tempboxa}{\LettrineFontHook\fontsize{\L@height}{\L@height}%
61     \selectfont \LettrineTestString}%

```

Arithmetic calculations convert the dimensions into integers (in sp) and compute a (4 decimal accurate) scaling factor.

```
62   \@tempcntb=\ht\@tempboxa
63   \@tempcnta=\L@height
64   \multiply\@tempcnta by 100
```

```

65 \divide\@tempcntb by 100
66 \divide\@tempcnta by \@tempcntb
67 \advance\@tempcnta by -9999
68 \ifnum\@tempcnta>0
69 \def\@tempa{1.\the\@tempcnta}%
70 \else
71 \def\@tempa{1}%
72 \fi
73 \LettrineFontHook
74 \fontsize{\@tempa\L@height}{\@tempa\L@height}%
75 \selectfont
76 }

```

`\LettrineFontEPS` The following definition is for use with dropped capitals defined as images: EPS, PDF, JPG, PNG files (see examples in `demo.tex`). Its use requires the `graphicx` package to be loaded in the preamble with `\usepackage{graphicx}`. The required size is computed just as in the standard case, `\includegraphics` prints the EPS file at this size.

```

77 \newcommand*\LettrineFontEPS}{%
78 \Lettrine@height\LettrineFontHook
79 \includegraphics[height=\L@height]%
80 }

```

Set up keyval initializations.

```

81 \define@key{L}{lines}{\setcounter{L@lines}{#1}}
82 \define@key{L}{depth}{\setcounter{L@depth}{#1}}
83 \define@key{L}{lhang}{\renewcommand*\L@hang}{#1}}
84 \define@key{L}{loversize}{\renewcommand*\L@oversize}{#1}}
85 \define@key{L}{lraise}{\renewcommand*\L@raise}{#1}}
86 \define@key{L}{ante}{\renewcommand*\L@ante}{#1}}
87 \define@key{L}{findent}{\setlength{\L@Findent}{#1}}
88 \define@key{L}{nindent}{\setlength{\L@Nindent}{#1}}
89 \define@key{L}{slope}{\setlength{\L@slope}{#1}}
90 \define@key{L}{image}[true]{\csname L@image#1\endcsname}
91 \define@key{L}{grid}[true]{\csname L@grid#1\endcsname}
92 \define@key{L}{realheight}[true]{\csname L@realh#1\endcsname}
93 \define@key{L}{novskip}{\setlength{\L@novskip}{#1}}

```

`\LettrineOptionsFor` This command is to be used in an optional config file (the name of which is found in `\DefaultOptionsFile`) to set the values of parameters on a per character basis, for instance:

`\LettrineOptionsFor{A}{slope=0.6em, findent=-1em, nindent=0.6em}` creates an internal command (`\l@A-keys` in this example), which expands to the options given as second argument of `\LettrineOptionsFor` for letter ‘A’.

```

94 \newcommand*\LettrineOptionsFor}[2]{\@namedef{l@#1-keys}{#2}}
95 \newdimen\LettrineWidth
96 \newdimen\LettrineHeight
97 \newdimen\LettrineDepth

```

`\lettrine` Now let's define `\lettrine`.

```
98 \def\lettrine{\@ifnextchar[\@lettrine{\@lettrine[]}}
99 \def\@lettrine[#1]#2#3{%
```

First reset the parameters to their default values:

```
100 \setcounter{L@lines}{\value{DefaultLines}}%
101 \setcounter{L@depth}{\value{DefaultDepth}}%
102 \renewcommand*{\L@hang}{\DefaultLhang}%
103 \renewcommand*{\L@oversize}{\DefaultLoversize}%
104 \renewcommand*{\L@raise}{\DefaultLraise}%
105 \renewcommand*{\L@ante}{}%
106 \setlength{\L@Findent}{\DefaultFindent}%
107 \setlength{\L@Nindent}{\DefaultNindent}%
108 \setlength{\L@slope}{\DefaultSlope}%
109 \setlength{\L@novskip}{\DiscardVskip}%
110 \ifLettrineImage\L@imagetrue\else\L@imagefalse\fi
111 \ifLettrineOnGrid\L@gridtrue\else\L@gridfalse\fi
112 \ifLettrineRealHeight\L@realhtrue\else\L@realhfalse\fi
```

`\LettrineFont` and `\LettrineFontEPS` both call `\Lettrine@height` to set `\L@height` which depends on `\L@tbox`. The content of `\L@tbox` depends on option `realheight`, so we have to read the lettrine's options and initialise the `\L@tbox` content now ³.

```
113 \setkeys{L}{#1}%
114 \sbox{\L@tbox}{\LettrineTextFont{\LettrineSecondString}}%
115 \ifL@realh
116   \def\@tempa{#3}
117   \ifx\@tempa\@empty
118     \PackageWarning{lettrine.sty}%
119       {Empty second argument,\MessageBreak
120         ignoring option 'realheight';}%
121   \else
122     \sbox{\L@tbox}{\LettrineTextFont{#3}}%
123   \fi
124 \fi
```

Then try to read an optional file (its name is given by `\DefaultOptionsFile`), do this inside a group, and define a global command `\l@LOkeys` which will expand to the list of options given by `\LettrineOptionsFor` for the current lettrine (defined by #2)...

```
125 \if\DefaultOptionsFile\relax
126 \else
127 \begingroup
128 \InputIfFileExists{\DefaultOptionsFile}%
129 {}%
130 {\PackageWarning{lettrine.sty}%
131   {File \DefaultOptionsFile\space not found}}%
132 }%
```

Gobble the colour commands, just keep the letter argument.

³Now means before eventually reading the config file.


```

133 \def\color##1##{\l@color{##1}}%
134 \let\l@color\@gobbletwo
135 \def\textcolor##1##{\l@textcolor{##1}}%
136 \def\l@textcolor##1##2##3{##3}%

```

Save the list of options relevant to the letter in #2 in \l@LOKeys (list is empty eventually).

```

137 \expandafter\ifx\csname l@#2-keys\endcsname\relax
138     \gdef\l@LOKeys{}%
139     \else
140     \xdef\l@LOKeys{\csname l@#2-keys\endcsname}%
141     \fi
142 \endgroup

```

Now apply these options (the following code executes \setkeys{L}{\l@LOKeys} where \l@LOKeys is expanded, see keyval.sty).

```

143 \def\KV@prefix{KV@L@}%
144 \let\@tempc\relax
145 \expandafter\KV@do\l@LOKeys,\relax,

```

As some parameters' values `findent`, `nindent` and `slope` — which do not influence the lettrine size — may be given relative to `\LettrineWidth`, this has to be done again after measuring the lettrine for `\LettrineWidth` to be set properly.

```

146 \sbox{\L@lbox}{\LettrineFont #2}%
147 \setlength{\LettrineWidth}{\wd\L@lbox}%
148 \def\KV@prefix{KV@L@}%
149 \let\@tempc\relax
150 \expandafter\KV@do\l@LOKeys,\relax,

```

As local options prevail on those held in the config file, we have to read local options again:

```

151 \setkeys{L}{#1}%
152 \fi

```

Options and optional config file have been taken into account, we can now save the first mandatory argument of `\lettrine` properly scaled into `\L@lbox`. Depending on the boolean `image`, `\LettrineFont` or `\LettrineFontEPS` is used.

```

153 \ifL@image
154     \sbox{\L@lbox}{\LettrineFontEPS{#2}}%
155 \else
156     \sbox{\L@lbox}{\LettrineFont #2}%
157 \fi

```

Height calculations done, let's reset `\L@tbox`'s content (mandatory in case `realheight=false`):

```

158 \sbox{\L@tbox}{\LettrineTextFont{#3}}%

```

Start a new paragraph, skipping the necessary amount of space if the dropped capital sticks out the top of paragraph. We use `\L@first` to compute the amount of space to be skipped. Again, as `\baselineskip` might be a rubber length, we convert it into a 'dimen' using `\@tempdima`.

```

159 \@tempdima=\baselineskip
160 \ifnum\value{L@lines}=1
161   \setlength{\L@first}{\ht\L@lbox}%
162   \addtolength{\L@first}{-\ht\L@tbox}%
163   \setlength{\L@lraise}{\z@}%
164 \else
165   \setlength{\L@first}{-\value{L@lines}\@tempdima}%
166   \addtolength{\L@first}{\@tempdima}%
167   \sbox{\@tempboxa}{\LettrineTextFont x}%
168   \addtolength{\L@first}{-\ht\@tempboxa}%

```

Now, `\L@first` holds (the opposite of) the raw height of a standard dropped capital (like 'X'), excluding the effect of `\L@oversize`. This is the basis for `\L@raise` (and `\L@oversize`, see `\LettrineFont`).

```

169   \setlength{\L@lraise}{-\L@raise\L@first}%
170   \addtolength{\L@first}{\L@lraise}%
171   \addtolength{\L@first}{\ht\L@lbox}%
172   \addtolength{\L@lraise}{-\value{L@lines}\@tempdima}%
173   \addtolength{\L@lraise}{\@tempdima}%
174 \fi
175 \par

```

`\L@first` now holds the height of the needed `\vskip`; if too small it will be discarded.

```

176 \ifdim\L@first>\L@novskip

```

When the 'grid' option is true, let's round up `\L@first` to the next integer number of `\baselineskip`.

```

177 \ifL@grid
178   \@tempdima=\baselineskip
179   \@tempdimb=\@tempdima
180   \advance\@tempdimb by \L@novskip
181   \@tempcnta=1
182   \loop\ifdim\L@first>\@tempdimb
183     \advance\@tempcnta by 1
184     \advance\L@first by -\@tempdima
185   \repeat
186   \vskip\@tempcnta\baselineskip
187 \else
188   \vskip\L@first
189 \fi
190 \fi

```

Again, we (mis)use the length `\L@first` to compute the width of the text eventually coming before the dropped capital. It is reset later on to hold the first line's length.

```

191 \setlength{\L@Pindent}{\wd\L@lbox}%
192 \addtolength{\L@Pindent}{-\L@hang\wd\L@lbox}%
193 \settowidth{\L@first}{\L@ante}%
194 \addtolength{\L@Pindent}{\L@first}%
195 \addtolength{\L@Pindent}{\L@Findent}%
196 \setlength{\L@first}{\linewidth}%
197 \addtolength{\L@first}{-\L@Pindent}%

```

Now let's compute `\L@Nindent` and `\L@next` for the next lines.

```
198 \addtolength{\L@Nindent}{\L@Pindent}%
199 \setlength{\L@next}{\linewidth}%
200 \addtolength{\L@next}{-\L@Nindent}%
```

This is for quotation, quote, abstract... environments: `\linewidth` is set by these environments, all we have to do is to shift our text left by `\rightmargin` (amount of space locally added to `\leftmargin` in these environments).

```
201 \addtolength{\L@Pindent}{\rightmargin}%
202 \addtolength{\L@Nindent}{\rightmargin}%
```

Store the letrine's final dimensions:

```
203 \setlength{\LettrineWidth}{\wd\L@lbox}%
204 \setlength{\LettrineHeight}{\ht\L@lbox}%
205 \setlength{\LettrineDepth}{\dp\L@lbox}%
```

Now, set up the shape of the new paragraph (designed by `\parshape`).

```
206 \addtocounter{L@lines}{1}%
207 \addtocounter{L@lines}{\value{L@depth}}%
208 \def\L@parshape{\c@L@lines \the\L@Pindent \the\L@first}%
209 \@tempcnta=\tw@
210 \@whilenum \@tempcnta<\c@L@lines\do{%
211   \edef\L@parshape{\L@parshape \the\L@Nindent \the\L@next}%
212   \addtolength{\L@Nindent}{\L@slope}%
213   \addtolength{\L@next}{-\L@slope}%
214   \advance\@tempcnta\@ne}%
215 \edef\L@parshape{\L@parshape \rightmargin \the\linewidth}%
216 \noindent
217 \parshape=\L@parshape\relax
```

Write the dropped capital into the left margin, and wrap the rest of paragraph around it.

```
218 \smash{\llap{\mbox{\L@ante}\raisebox{\L@lraise}{\usebox{\L@lbox}}}%
219   \hskip \the\L@Findent}}%
220 \usebox{\L@tbox}%
221 }
```

This ends the definition of `\lettrine`.

Load a local config file if present in L^AT_EX's search path.

```
222 \InputIfFileExists{lettrine.cfg}
223   {\typeout{Loading lettrine.cfg}}
224   {\typeout{lettrine.cfg not found, using default values}}
```

3 Configuration file

```
225 %% lettrine.cfg: configuration file for lettrine.sty
226 %%
227 %% If you want to customize lettrine, please *do not* hack into the
228 %% code, copy this file to the directory where lettrine.sty lies
```

```

229 %% and customize it as you like.
230 %%
231 %% Uncomment these lines and change the parameters' values to fit
232 %% your needs (see letrine.dtx).
233 %%
234 %%\setcounter{DefaultLines}{2}
235 %%\setcounter{DefaultDepth}{0}
236 %%
237 %% These are *decimal* numbers:
238 %%\renewcommand{\DefaultLoversize}{0}
239 %%\renewcommand{\DefaultLraise}{0}
240 %%\renewcommand{\DefaultLhang}{0}
241 %%
242 %% These are *lengths* (don't forget the unit):
243 %%\setlength{\DefaultFindent}{Opt}
244 %%\setlength{\DefaultNindent}{0.5em}
245 %%\setlength{\DefaultSlope}{0mm}
246 %%\setlength{\DiscardVskip}{0.2pt}
247 %%
248 %% These are *flags* (value=true/false):
249 %%\LettrineImagefalse
250 %%\LettrineOnGridfalse
251 %%\LettrineRealHeightfalse
252 %%
253 %% These are *commands* (value=string, only its height matters):
254 %%\renewcommand{\LettrineTestString}{ABCDEFGHJKLMNOQRSTUVWXYZ}
255 %%\renewcommand{\LettrineSecondString}{x}
256 %%
257 %% In case you want to set parameters for some letters
258 %% in file 'optfile.cfl'
259 %%\renewcommand{\DefaultOptionsFile}{optfile.cfl}

```

Change History

letrine-0.81		work properly in quote, quotation, abstract environments...
\letrine: \DefaultLoversize		but do not change \linewidth
added.	8	which is set by these environments.
letrine-0.9		
General: \LettrineFontEPS	7	letrine-1.2
\letrine: Calculations of length		General: \newlength changed to
\L@first changed. Do not		\newdimen, to correct a bug
'vskip' small lengths (<0.2pt),		with seminar.cls (pointed out
they are just rounding errors. .	9	by Peter Münster).
\LettrineFont: \Lettrine@height		\letrine: \baselineskip may be
added.	6	a rubber length, we convert it
\LettrineFontHook added.	6	to a dimen.
Size of the dropped capital		\LettrineFont: \baselineskip
changed when 'lines' value is 1		may be a rubber length, we
(was \Huge).	6	convert it to a dimen.
letrine-1.1		letrine-1.3
\letrine: Add \rightmargin to		General: Correct the documenta-
\L@Pindent for \Lettrine to		

tion to mention the cm-super fonts and the typelec package by Vladimir Volovich.	4	pointed out by David Monniaux. Correction by Enrico Gregorio.	11
lettrine-1.4		lettrine-1.65	
<code>\lettrine</code> : <code>\lettrine</code> still didn't work properly in quote, quotation, abstract environments, pointed out by Matthias C. Schmidt. <code>\rightmargin</code> was added too early to <code>\L@nindent</code> , thus making <code>\L@next</code> too short by <code>\rightmargin</code>	11	<code>\lettrine</code> : Measure and store the lettrine's final dimensions. . . .	11
lettrine-1.5		lettrine-1.7	
General: <code>\LettrineOptionsFor</code> and <code>\LettrineWidth</code> added. . . .	7	General: New counter to add lines for dropped capitals with positive depth, like Q.	1
<code>\lettrine</code> : Added reading of an optional config file <code>\DefaultOptionsFile</code>	8	lettrine-1.8	
lettrine-1.6		General: Added newif <code>\ifLettrineOnGrid</code> and new dimen <code>\DiscardVskip</code> , default (0.2pt) set for compatibility with previous releases. . . .	5
General: Add a flag to switch to images in eps or pdf format. Suggested by Bill Jetzer.	2	Added two keyval options: 'grid' (true/false) and 'novskip' to override <code>\DiscardVskip</code>	2
Added newif <code>\ifL@grid</code>	5	<code>\lettrine</code> : The 0.2pt limit for discarded vskips is now customisable through <code>\DiscardVskip</code> and option 'novskip'.	9
Added newif <code>\ifL@image</code>	5	lettrine-1.9	
Added newif <code>\ifLettrineImage</code>	5	General: New customisable string <code>\LettrineSecondString</code> to tune the lettrine's height.	3
<code>\lettrine</code> : Add braces around #3 to allow commands taking an argument (such as <code>\MakeLowercase</code>) in <code>\LettrineTextFont</code> . Suggested by Philipp Lehman.	9	New keyval option: 'realheight' (true/false) and new global flag <code>\ifLettrineRealHeight</code>	2
<code>\LettrineFontEPS</code> : Added <code>\LettrineFontHook</code> to <code>\LettrineFontEPS</code> . This is needed for color options.	7	<code>\lettrine</code> : <code>\theDefaultLines</code> changed to <code>\value{DefaultLines}</code> , same with <code>\theDefaultDepth</code> . Needed for babel-hebrew which redefines <code>\@arabic</code> . Thanks to Ulrike Fischer for providing the fix.	8
lettrine-1.63		<code>\theL@depth</code> changed to <code>\value{L@depth}</code>	11
<code>\LettrineFont</code> : Added command <code>\LettrineTestString</code> which defaults to 'ABCDEFGHJKLMNOPQRSTUVWXYZ'. In previous versions height computations were based on letter 'X' which might not exist in some (rare) fonts. Pointed out by Raphaël Pinson.	6	Use the second mandatory argument of <code>\lettrine</code> or <code>\LettrineSecondString</code> (which defaults to 'x') to compute <code>\L@height</code> . This is controlled by the 'realheight' flag. . . .	8
lettrine-1.64		<code>\LettrineFont</code> : <code>\theL@lines</code> changed to <code>\value{L@lines}</code> . Needed for babel-hebrew which redefines <code>\@arabic</code>	6
<code>\lettrine</code> : Remove \$ around <code>\smash</code> and add <code>\relax</code> . Bug			