
Changing the font size in L^AT_EX

Thomas Thurnherr

Abstract

Changing the font size in L^AT_EX can be done on two levels, affecting either the whole document or elements within it. Using a different font size on a global level will affect all normal-sized text as well as the sizes of headings, footnotes, etc. By changing the font size locally, however, a single word, a few lines of text, a large table, or a heading throughout the document may be modified. Fortunately, there is no need for the writer to juggle with numbers when doing so. L^AT_EX provides a set of macros for changing the font size locally, taking into consideration the document's global font size.

1 Changing the font size on the document-wide level

The standard classes `article`, `report` and `book` support three different font sizes: `10pt`, `11pt`, `12pt`. By default, the font size is set to `10pt` and can be modified by passing any of the previously-mentioned value as a class option. As an example, suppose you want to change the font size for normal text to `12pt` throughout the document. For the class `report`, this is how you would do that:

```
\documentclass[12pt]{report}
```

In most cases, the available font sizes for the standard classes are sufficient and you do not have to bother about loading special packages that provide more options.

1.1 Extended font sizes for basic classes

Should you ever require a different font size, however, the `extsizes` package comes in handy. Along with the standard font sizes mentioned above, it provides the following additional options: `8pt`, `9pt`, `14pt`, `17pt`, and `20pt`. As these font sizes require a reimplementations of the document classes, names are slightly different from the standard classes `article` and `report`:

```
\documentclass[9pt]{extarticle}
\documentclass[14pt]{extreport}
```

1.2 KOMA-script and memoir classes

The KOMA-script document classes work very much the same in terms of font size as the standard classes. The only difference is the default font size which is `11pt` for all classes except `scrlettr`. The latter has a default size of `12pt`.

The memoir class, however, is more flexible when it comes to font sizes. It provides additional sizes

ranging from `9pt` all the way to `60pt`. These options are available: `9pt`, `10pt`, `11pt`, `12pt`, `14pt`, `17pt`, `20pt`, `25pt`, `30pt`, `36pt`, `48pt`, and `60pt`. The following example illustrates their usage:

```
\documentclass[60pt,extrafontsizes]{memoir}
```

The example illustrates a common problem with fonts larger than `25pt` and the standard L^AT_EX font Computer Modern (in `OT1` encoding). They cannot exceed `25pt` since larger sizes are not defined and therefore not available. The memoir class solves this problem with the `extrafontsizes` option. It changes the standard font to the scalable Latin Modern in `T1` encoding. This is equivalent to the following two lines of code in the document preamble:

```
\usepackage{lmodern}
\usepackage[T1]{fontenc}
```

1.3 Other classes

The AMS document classes have a few more font sizes than the basic classes, though not as many as `extsizes`. It's always good to check the class documentation to see what's supported — not all classes are the same.

2 Changing the font size locally

A common scenario is that the author of a document needs to change the font size for a word or paragraph, decrease the font size of a large table to make it fit on a page or increase the size of a heading throughout the document. L^AT_EX implements a set of macros which allow changing font size from `Huge` to `tiny`, literally. That way, the author does not have to worry about numbers. The macros, including the exact font size in points, are summarized in table 1.

A good rule of thumb is not to use too many different sizes and not to make things too small or too big.

L^AT_EX provides two different ways to use these font size modifier macros: inline or as an environment using `\begin... \end`:

```
{\Large This is some large text.\par}
\begin{footnotesize}
This is some footnote-sized text.
\end{footnotesize}
```

The `\par` command at the end of the inline example adjusts `baselineskip`, the minimum space between the bottom of two successive lines.

2.1 More sizes: `\HUGE` and `\ssmall`

The `moresize` package adds two additional options to the list of macros above, `\HUGE` and `\ssmall`. The first provides a font size bigger than the largest

Class option	10pt	11pt	12pt
<code>\Huge</code>	25pt	25pt	25pt
<code>\huge</code>	20pt	20pt	25pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\Large</code>	14pt	14pt	17pt
<code>\large</code>	12pt	12pt	14pt
<code>\normalsize</code> (default)	10pt	11pt	12pt
<code>\small</code>	9pt	10pt	11pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\tiny</code>	5pt	6pt	6pt

<code>fontsize</code>	<code>\Huge</code>
<code>fontsize</code>	<code>\huge</code>
<code>fontsize</code>	<code>\LARGE</code>
<code>fontsize</code>	<code>\Large</code>
<code>fontsize</code>	<code>\large</code>
<code>fontsize</code>	<code>\normalsize</code>
<code>fontsize</code>	<code>\small</code>
<code>fontsize</code>	<code>\footnotesize</code>
<code>fontsize</code>	<code>\scriptsize</code>
<code>fontsize</code>	<code>\tiny</code>

Table 1: Font sizes available in standard L^AT_EX.

available by default, whereas the latter fills the gap between `\scriptsize` and `\tiny`.

Since `\HUGE` changes the font size to a number bigger than 25pt and, as mentioned above, the standard font is not scalable, L^AT_EX displays a warning saying the font size is not available and that it was replaced by the next smaller (`\Huge`). Again, one needs to use another font type, such as the Times Roman equivalent available in the PSNFSS package (see example below). This way, you can benefit from that “HUGE” font size provided by the `moresize` package. Here is an example:

```
\documentclass[11pt]{report}
\usepackage{mathptmx}
\usepackage[11pt]{moresize}
\begin{document}
{\HUGE HUGE text}
{\small Can you see a ‘‘ssmall’’ text?}
\end{document}
```

HUGE text Can you see a “ssmall” text?

2.2 Not enough?

There is an alternative, completely flexible approach. The `anyfontsize` package scales the closest bigger or smaller font size available to any size.

The usage is very similar to the inline example shown before. The package implements the `\fontsize` command which takes two arguments, the new font size and the size of the `baselineskip`.

```
\fontsize{<size>}{<baselineskip>}
```

It is recommended to use a `baselineskip` of roughly $1.2 \times$ font size in order to get a reasonable space

between two successive lines. Of course the best value depends on the document and font design.

The following example shows font sizes 50pt and 5pt and compares them with `\Huge` and `\tiny`. The difference between 5pt and `\tiny` (6pt for the 11pt class option) is barely visible.

```
\documentclass[11pt]{report}
\usepackage{mathptmx}
\usepackage{anyfontsize}
\usepackage{t1enc}
\begin{document}
{\fontsize{50}{60}\selectfont Foo}
{\fontsize{5}{6}\selectfont bar!}
{\Huge Foo}
{\tiny bar!}
\end{document}
```

Foo bar! **Foo** bar!

Again, this only works with a scalable, non-standard font.

2.3 Memoir classes

As with font size class options, the memoir class also provides additional font modifier macros at the extreme ends of the scale, `\minuscule` and `\HUGE`. These macros use font sizes of 4pt, 20pt respectively, for the 9pt class option and 20pt, 132pt respectively, for the 60pt class option. Usage is exactly the same as for the standard L^AT_EX classes.

- ◇ Thomas Thurnherr
 texblog (at) gmail dot com
<http://texblog.org>