# moresize: More font sizes with LaTeX

Christian Cornelssen (`cornsub1@flintstone.ukbf.fu-berlin.de`)

v1.9 [1999/07/26]

## 1 Introduction

When writing yet another LaTeX introduction with a demonstration of font sizes, I discovered that `\Huge` and `\huge` were set to the same size. No good for a demonstration! I localized the definition in the `size12.clo` class option file and found that the font size for `\Huge` had to be increased from 24.88pt to 29.86pt.

`\HUGE`      Therefore, I produced a package moresize redefining `\Huge` and defining an even "bigger" font size command `\HUGE`, because I wanted to be able to use font sizes up to 35.83pt, as provided by the EC fonts' standard edition.

`\ssmall`      Additionally, I defined a size command `\ssmall` filling the gap between the sizes selectable by `\tiny` and `\scriptsize`.

The documentation in the LaTeX $2_\varepsilon$ distribution's source file `fontdef.dtx` reminded me to also declare math sizes for the new text sizes. When studying the docs, I found that the math script sizes defined in `fontdef.dtx` grew faster than the text sizes, and that at least the script sizes for 24.88pt seemed to be absolutely inappropriate. Therefore, I designed two sets of text/math size combinations including the new 29.86pt and 35.83pt text sizes, one set just fixing the 25pt math script and scriptscript sizes, and another one with somewhat smaller scriptscript sizes in order to maintain the typical 10 : 7 : 5 ratio for text sizes from 10pt upwards. The math size changings can be activated by means of options passed to the moresize package.

For testing the new sizes and for visualizing the effects of the math size settings, a LaTeX file `msizetst.tex` containing sample text and math expressions at various sizes is provided.

## 2 User Interface

The moresize package can be loaded by the LaTeX $2_\varepsilon$ command

     `\usepackage[`⟨*options*⟩`]{moresize}`

By default, moresize neither redefines existing macros nor changes the standard math size declarations. However, the package provides the additional font size selection commands `\HUGE` and `\ssmall`, and declares math sizes for the new 29.86pt and 35.83pt text sizes.

Available Options are:

**10pt** This option provides the same settings as the default. `\HUGE` is defined to select a 29.86pt font size, and `\ssmall` is made to give 6pt. These definitions continuously extend the set of size commands given in the `size10.clo` standard class option file.

**11pt** This option makes `\HUGE` and `\ssmall` select 29.86pt and 7pt, respectively, which is suitable for use with the settings given in the `size11.clo` standard class option file.

**12pt** With this option, `\HUGE` and `\ssmall` are defined to select 35.83pt and 7pt, respectively.

Additionally, the size selection command `\Huge` is redefined to give 29.86pt. This is done because the setting given in the `size12.clo` standard class option file saturates the `\Huge` font size at the same level as given by `\huge` (24.88pt)—presumably with the CM font restrictions of old LaTeX 2.09 in mind. Nowadays, with LaTeX$2_\varepsilon$'s flexible NFSS, font size selection commands are not associated with a predetermined font family any longer, hence such restrictions are out of place. Note that NFSS automatically resorts to lower sizes if a demanded font size is not available, so standard CM font users will not be affected by the new size settings, but EC font users can benefit from them.

**ecpatch** This is a bug fix for the EC font declarations given in the LaTeX$2_\varepsilon$ releases from December 1996 through December 1997. Therein, the declarations of the EC fonts do not include font sizes above 24.88pt, although the EC font standard edition provides font metrics files for 29.86pt and 35.83pt. As a consequence, NFSS would deny usage of those font sizes and substitute 24.88pt instead.

The `ecpatch` option installs macros which override any declaration of an EC font with an own declaration including the 29.86pt and 35.83pt sizes.

Regard this patch as a temporary bug fix; the proper way requires updating the LaTeX$2_\varepsilon$ distribution source file `cmfonts.fdd` from which the `t1cm*.fd` files are generated, and remaking the LaTeX$2_\varepsilon$ format file. Indeed, a proper fix has been incorporated into the LaTeX$2_\varepsilon$ distribution shortly after December 1997; therefore, recent distributions should not require the `ecpatch` option.

As can be concluded from the above, the decision whether to use `ecpatch` or not should be determined by the availibility of EC (*not* DC!) fonts and by the version of the LaTeX$2_\varepsilon$ base distribution both of which may vary independently from the author's main LaTeX input file. Therefore, the appropriate spot for invoking `ecpatch` is a configuration file placed somewhere in the LaTeX$2_\varepsilon$ input directory tree with filename `moresize.cfg`, containing a line like

```
\ExecuteOptions{ecpatch}
```

This frees authors of LaTeX$2_\varepsilon$ input files from the need of specifying `ecpatch` themselves when using the moresize package and thus preserves the portability of LaTeX$2_\varepsilon$ documents across different LaTeX$2_\varepsilon$ releases.

Note that `ecpatch` is useful only if the following conditions are met:

1. The "official" EC fonts are available, including TFM files for the sizes 29.86pt and 35.83pt. With DC fonts instead, `ecpatch` will not work correctly.[1]

2. The LaTeX 2$_\varepsilon$ base has been configured to support EC fonts (i.e. `cmfonts.fdd` has been unpacked by `ec.ins`[2]).

3. The LaTeX 2$_\varepsilon$ base (or at least its `cmfonts.fdd`) has a release date before 1998. Later releases do not need `ecpatch`.[3]

Anyway—upgrading your system to a recent LaTeX 2$_\varepsilon$ base with "official" EC fonts would be a better solution than using `ecpatch`.

**normalscripts** This option decreases the script and scriptscript sizes associated with the 24.88pt text size by one magstep, in order to damp the script size explosion defined in the LaTeX 2$_\varepsilon$ source file `fontdef.dtx`. With this setting, a formula like $\frac{X}{Y}$ within a `\Huge` paragraph (e.g. a document title) does no longer affect line spacing, as should be.

**smallscripts** Starting from the `normalscripts` math size settings, this option further reduces the scriptscript sizes associated with text sizes of 17.28pt and above. The idea behind this is to maintain an approximate $10 : 7 : 5$ ratio between text, script, and scriptscript sizes in bigger-than-normal writings.

The math script size option names roughly describe the changes to the standard LaTeX setup. Therefore, `normalscripts` may not look "normal" in the sense of "just right", but differs from the standard setup in 24.88pt writings only. I prefer `smallscripts` which results in scriptscript sizes that are not really small—they are simply not as big as the standard ones. Note that this option only affects math expressions with nested fractions and/or sub-/superscripts in large writings (17.28pt text size and above), e.g. in titles or section headers. Therefore, it is quite likely that you have not encountered an example of that kind so far. But once you have, you should remember this package...

## 3  The test file `msizetst.tex`

A test file `msizetst.tex` is provided for evaluating the math size settings. You may rename it and then manipulate it as you wish, typically for testing the options passed to the moresize package. The test document shows sample text and math expressions at various sizes, each framed above and below by horizontal lines delimiting the amount of vertical space that is normally available for math expressions within paragraphs of text. If the expressions overprint the delimiters, they are likely to cause stretching of the baseline skip when used within a paragraph.

However, the output for sizes above 24.88pt may be erroneous or misleading under certain circumstances:

- Even when using EC text fonts, math expressions may still use CM fonts which are limited to 24.88pt in their declarations to NFSS. This does not

---

[1] `ecpatch` may work with the last ("new") DC font release if provided with TFM files for all standard EC font sizes. Typically, the 29.86pt size would be missing.

[2] Using `newdc.ins` for DC fonts with all TFM files mentioned above could also work.

[3] Applying `ecpatch` to newer LaTeX 2$_\varepsilon$ releases has not caused any harm in my testings so far.

matter for scripts and scriptscripts because even `\HUGE` writings do not request script sizes greater than 24.88pt. However, the main (i. e. non-script) symbols of math expressions in `\HUGE`, or even in `\Huge` (for `12pt` documents), will be shrinked to 24.88pt size, thereby issuing NFSS warnings about unavailable font sizes for OT1, OML, or OMS encodings. (If there are messages about unavailable T1 font sizes, you should try the `ecpatch` option or get a recent version of LaTeX and of the EC fonts.)

- As another effect of the main symbol size shrinking, the vertical position of the main math axis is lowered, thereby moving fraction rules towards the denominators.

- The calculation of the amount of vertical space available within a normal line yields wrong results if fonts with the desired size are not available. In detail, the appearant amount of free vertical space gets increased by twice the difference between the requested font size and the size chosen by NFSS.

  If you use properly-declared EC fonts for testing the samples, this problem should not occur.

By adding extreme magnification factors to the CM font declarations, I have verified that the aforementioned artifacts disappear when math fonts in the requested sizes are made available (though heavily "bolded" by the magnification). However, I have not included that hack for demonstration, in order not to provide any means for over-using the CM math fonts. According to the LaTeX $2_\varepsilon$ newsletters `ltnews09.tex` and `ltnews11.tex`, TeX users can look forward to a modern math font concept allowing easy setup and use of alternate math fonts. Therefore, making documents which explicitly misuse CM fonts for huge math expressions would be counterproductive. However, even without math fonts at more than 24.88pt, the improvements in typesetting huge math expressions can be demonstrated by varying the options passed to the moresize package in the test file `msizetst.tex`.

## 4 Package Code

This section documents the contents of the ready-to-use package file `moresize.sty`.

1 ⟨∗package⟩

### 4.1 Preliminaries

Announce the file as a LaTeX $2_\varepsilon$ package.

2 `\ProvidesPackage{moresize}[1999/07/26 v1.9 (more font sizes)]`

`\@xxxpt` As a convention, declare macros for the new point sizes.
`\@xxxvipt`
3 `\providecommand\@xxxpt{29.86}`
4 `\providecommand\@xxxvipt{35.83}`

## 4.2 Size Selection Commands

Define the author commands `\HUGE` and `\ssmall`, depending on the main document font size option. In case of `12pt`, fix the clipping of `\Huge` which occurs in `size12.clo` (`classes.dtx`).

```
 5 \DeclareOption{10pt}{
 6   \def\ssmall{\@setfontsize\ssmall\@vipt\@viipt}
 7   \def\HUGE{\@setfontsize\HUGE\@xxxpt{36}}}
 8 \DeclareOption{11pt}{
 9   \def\ssmall{\@setfontsize\ssmall\@viipt\@viiipt}
10   \def\HUGE{\@setfontsize\HUGE\@xxxpt{36}}}
11 \DeclareOption{12pt}{
12   \def\ssmall{\@setfontsize\ssmall\@viipt\@viiipt}
13   \def\Huge{\@setfontsize\Huge\@xxxpt{36}}
14   \def\HUGE{\@setfontsize\HUGE\@xxxvipt{43}}}
```

The definitions given here are similar to most size command definitions in LaTeX $2_\varepsilon$'s `classes.dtx`. The following rules have been observed and taken as guidelines:

- Size commands other than `\footnotesize`–`\normalsize` are not intended for typesetting document structures more complex than paragraphs. Therefore, their definitions do not contain specifications of dimension parameters for list environments or displayed equations.

- Baselineskips for text sizes below 8pt are chosen to be 1pt higher than the text size. Baselineskips above 10pt are set to somewhat rounded pt sizes around 1.2 times the text size.

## 4.3 Math Size Options

Consider the declarations in LaTeX $2_\varepsilon$'s `fontdef.dtx`:

```
\DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
\DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
\DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
\DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
\DeclareMathSizes{\@xviipt}{\@xviipt}{\@xiipt}{\@xpt}
\DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xiipt}
\DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xviipt}
```

The script and scriptscript sizes for `\@xxvpt` are one magstep bigger than the sizes extrapolated from the context. This seems to be an error. And in fact, fractions like $\frac{X}{Y}$ placed in 25pt text strech the baseline skip, whereas this does not happen at lower sizes.

This will be fixed when given the option `normalscripts`:

```
15 \DeclareOption{normalscripts}{
16   \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xviipt}{\@xivpt}}
```

In any case, declare math sizes for the new point sizes.

```
17 \DeclareMathSizes{\@xxxpt}{\@xxxpt}{\@xxpt}{\@xviipt}
18 \DeclareMathSizes{\@xxxvipt}{\@xxxvipt}{\@xxvpt}{\@xxpt}
```

There remains a one-magstep ratio between the script and scriptscript sizes for text sizes above `\@xivpt`, whereas for text sizes between `\@xpt` and `\@xivpt`,

approximately two magsteps are taken. An option is defined for applying the two-magsteps rule to text sizes above `\@xivpt`, including the new `\@xxxpt` and `\@xxxvipt` sizes.

```
19 \DeclareOption{smallscripts}{
20   \DeclareMathSizes{\@xviipt}{\@xviipt}{\@xiipt}{8}
21   \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xpt}
22   \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xviipt}{\@xiipt}
23   \DeclareMathSizes{\@xxxpt}{\@xxxpt}{\@xxpt}{\@xivpt}
24   \DeclareMathSizes{\@xxxvipt}{\@xxxvipt}{\@xxvpt}{\@xviipt}}
```

## 4.4   EC Font Declaration Patch

`\UndeclareFontFamily`   `\UndeclareFontFamily{⟨ENC⟩}{⟨family⟩}` makes NFSS forget the font information for the font family ⟨*family*⟩ with encoding ⟨*ENC*⟩ and thus causes the corresponding font declaration file (⟨`enc`⟩⟨`family`⟩`.fd`) to be reloaded on demand.

```
25 \newcommand\UndeclareFontFamily[2]{%
26   \global\expandafter\let\csname#1+#2\endcsname\relax}
```

The EC font declaration patch is an optional bug fix, and possibly obsolete.

```
27 \DeclareOption{ecpatch}{
28   \typeout
29     {*** Using EC font declaration patch for pre-1998 LaTeX2e releases ***}
```

`\EC@family`   Declaration of font families from the EC or the previously released DC edition is done by the macro `\EC@family`. It is defined in the relevant font definition files by means of `\providecommand`. Hence, predefining this macro will override the definition given in the `fd` files. Now take the original definition and complete the size specification with 29.86pt and 35.83pt entries. Since `\nfss@catcodes` have not been activated, white space in the `\DeclareFontShape` call must be avoided.

```
30   \def\EC@family#1#2#3#4#5{%
31     \DeclareFontShape{#1}{#2}{#3}{#4}{%
32       <5><6><7><8><9><10><10.95><12><14.4>%
33       <17.28><20.74><24.88><29.86><35.83>genb*#5}{}}
```

`\EC@ttfamily`   For the typewriter families, a similar macro is used.

```
34   \def\EC@ttfamily#1#2#3#4#5{%
35     \DeclareFontShape{#1}{#2}{#3}{#4}{%
36       <5><6><7><8>#50800%
37       <9><10><10.95><12><14.4><17.28><20.74><24.88>%
38       <29.86><35.83>genb*#5}{}}
```

Now undeclare font families that are usually mapped to EC or DC fonts. On demand, they will be redeclared under control of the predefined `\EC@family` or `\EC@ttfamily` macros.

```
39   \UndeclareFontFamily{T1}{cmr}
40   \UndeclareFontFamily{TS1}{cmr}
41   \UndeclareFontFamily{T1}{cmss}
42   \UndeclareFontFamily{TS1}{cmss}
43   \UndeclareFontFamily{T1}{cmtt}
44   \UndeclareFontFamily{TS1}{cmtt}
45   \UndeclareFontFamily{T1}{cmvtt}
46   \UndeclareFontFamily{TS1}{cmvtt}
```

```
47   \UndeclareFontFamily{T1}{cmsc}% from 'smallcap' package
48   \UndeclareFontFamily{T1}{cmdh}
49   \UndeclareFontFamily{T1}{cmfib}
50   \UndeclareFontFamily{T1}{cmfr}
```

The patch is now complete.

```
51 }
```

Of course, a proper way in NFSS for adding size specifications to existing font shape declarations would be appreciated. But once the fonts are declared properly, this should not be necessary at all.

## 4.5   Configuration

In the standard classes, 10pt writing is the default. Therefore, the same is assumed here. Global options or explicitly given package options may override this.

```
52 \ExecuteOptions{10pt}
```

Before processing the package options, a configuration file `moresize.cfg` may be input, typically containing `\ExecuteOptions{ecpatch}` for systems with installed EC fonts and a LATEX 2$_\varepsilon$ version released before 1998.

```
53 \InputIfFileExists{moresize.cfg}{}{}
54 \ProcessOptions
55 ⟨/package⟩
```

# 5   Test Code

This section documents the contents of the test document source file `msizetst.tex`.

```
56 ⟨∗testfile⟩
```

This is a LATEX 2$_\varepsilon$ document.

```
57 %% This is a test file for text and math size settings.
58 %% You might want to test several configurations.  Therefore,
59 %% it might be a good idea to save different versions of this file
60 %% under different names.
61 \documentclass[11pt,a4paper]{article}
62 \usepackage{exscale}     %% for scaled \sum, \int etc.
63
```

Here the package of interest is loaded.

```
64 %% Available options for the 'moresize' package:
65 %%    10pt, 11pt, 12pt   (in \documentclass) determines font sizes
66 %%    normalscripts      avoid script size explosion in huge writings
67 %%    smallscripts       smaller, more appropriate script sizes
68 %%    ecpatch            tell NFSS that huge EC fonts are available
69 \usepackage[smallscripts]{moresize}
70
```

For testing the 29.86pt and 35.83pt sizes, the T1-encoded EC or DC fonts may be needed.

```
71 %% Use the 't1enc' package for testing with DC/EC fonts
72 \usepackage{t1enc}
73
```

**\calcmathspace**
**\mathht**
**\mathdp**

The amount of vertical space usable for math expressions within a paragraph without probably stretching the baseline skip can be calculated by means of the macro \calcmathspace. When called, \calcmathspace uses the caller's current text size and \baselineskip settings as well as height and depth of the characters [ and ] from the roman text font for estimating the height and depth which an inline math expression may have. The results are stored in the length parameters \mathht and \mathdp, respectively.

```
74 %% Consider a paragraph with a text line of depth d (below the baseline),
75 %% followed by a line with an inline formula,
76 %% followed by a text line with height h (above the baseline).
77 %% Then the formula's height may not exceed \baselineskip - d,
78 %% and its depth may not exceed \baselineskip - h,
79 %% or the baseline skips will be stretched.
80 %% The roman-text brackets '[' and ']' are used for estimating d and h.
81 %% Note that the space for the formula seems to get larger if NFSS
82 %% substitutes a smaller font, because \baselineskip does not shrink
83 %% accordingly!
84 \newlength\mathht        %% height available for an inline formula
85 \newlength\mathdp        %% depth available for an inline formula
86 \newcommand\calcmathspace{%
87         \settodepth\mathht{\textrm{[}}%
88         \setlength\mathht{-\mathht}%
89         \addtolength\mathht{\baselineskip}%
90         \settoheight\mathdp{\textrm{]}}%
91         \setlength\mathdp{-\mathdp}%
92         \addtolength\mathdp{\baselineskip}}
93
```

**\linebox**

The command \linebox{⟨text⟩} typesets ⟨text⟩ in a box with fixed height and depth corresponding to a line that extends so high and so deep that it just touches the preceding and following lines which are assumed to contain ordinary text. The upper and lower bounds of the box are delimited by horizontal lines.

```
94 %% The following command shows its argument, typeset in text style,
95 %% between two horizontal lines showing the available space in a text line.
96 %% If the argument overprints the lines, it would stretch
97 %% the baseline skip in a paragraph.
98 \newcommand\linebox[1]{\calcmathspace\begin{tabular}{@{}l@{}}\hline
99         \raisebox{0pt}[\mathht][\mathdp]{#1}\\\hline\end{tabular}}
100
```

**\msample**

The macro \msample should contain fairly simple math expressions with nested sub- or superscripts and/or fractions which should fit into paragraphs of text without stretching the baseline skip.

```
101 %% A macro for sample math expressions.
102 %% The original definition gives expressions which should fit into
103 %% paragraphs without enlarging the baseline skip.
104 %% For variable-sized \sum or \int symbols, \usepackage{exscale} is needed.
105 %% Caution: Main (i.e. non-script) symbols are 24.88pt maximum
106 %% with CM math fonts. You may try using text fonts instead,
107 %% but some vertical positionings and spacings still go wrong then.
108 \newcommand\msample{\ensuremath{
109     \frac{X}{Y}
110 \,  g^{i_1\ldots i_n}_{j_1\ldots j_n}
```

```
111 \,  \frac{t^2}{2!}
112 \,  e^{-\frac{t}{\tau}}}
113 }}
114
```

**\Size**   Interested testers may like to directly specify point sizes instead of using author commands. The command `\Size{⟨font size⟩}{⟨baselineskip⟩}` is provided for easing such specifications. ⟨*font size*⟩ and ⟨*baselineskip*⟩ must be specified as numbers of pt, with the `pt` omitted.

```
115 %% None of the size options 10pt, 11pt, and 12pt offers the entire spectrum
116 %% of available font sizes.
117 %% For example, the 10pt option offers no command for selecting 10.95pt,
118 %% and the 12pt option does not provide a command for 9pt.
119 %% The 11pt option fills both gaps, but misses the extremes 35.83pt and 5pt.
120 %% Therefore, a shorthand for lower-level size switches is provided.
121 %% E.g. \Size{20.74}{25} selects a 20.74pt font size with 25pt baselineskip.
122 \newcommand\Size[2]{\fontsize{#1}{#2}\selectfont}
123
```

Now typesetting can begin. Only one page should be needed. Remember to echo the options used.

```
124 \begin{document}
125 \parindent0pt
126
127 %% The test output should fit onto one page
128 \pagestyle{empty}
129 \enlargethispage{1in}
130
131 %% Don't forget to list the moresize options in the test output!
132 \textsf{moresize} options:
133 %%[10pt]
134 [11pt]
135 %%[12pt]
136 %%[normalscripts]
137 [smallscripts]
138 %%[ecpatch]
139
140 %% Are you using HUGE fonts?
141 %%Using CM fonts only.
142 %%Using DC fonts for text, CM fonts for math.
143 Using EC fonts for text, CM fonts for math.
144
```

Now comes a table containing samples at sizes down to 5pt. The original font size directives are intended for use with the `11pt` option; with `12pt` (`10pt`) instead, the 35.83pt (5pt) size would appear twice.

```
145 \begin{center}
146 \newcommand\n{\\\\[-1ex]}          %% for separating the tabular lines
147 \begin{tabular}{@{}ll@{}}
148          \Size{35.83}{43}\linebox{36pt}
149                                  & \Size{35.83}{43}\linebox{\msample}
150 \n      \HUGE\linebox{HUGE}      & \HUGE\linebox{\msample}
151 \n      \Huge\linebox{Huge}      & \Huge\linebox{\msample}
152 \n      \huge\linebox{huge}      & \huge\linebox{\msample}
```

```
153 \n       \LARGE\linebox{LARGE}    & \LARGE\linebox{\msample}
154 \n       \Large\linebox{Large}    & \Large\linebox{\msample}
155 \n       \large\linebox{large}    & \large\linebox{\msample}
156 \n       \normalsize\linebox{normalsize}
157                                   & \normalsize\linebox{\msample}
158 \n       \small\linebox{small}    & \small\linebox{\msample}
159 \n       \footnotesize\linebox{footnotesize}
160                                   & \footnotesize\linebox{\msample}
161 \n       \scriptsize\linebox{scriptsize}
162                                   & \scriptsize\linebox{\msample}
163 \n       \ssmall\linebox{ssmall} & \ssmall\linebox{\msample}
164 \n       \tiny\linebox{tiny}      & \tiny\linebox{\msample}
165 \n       \Size{5}{6}\linebox{5pt}
166                                   & \Size{5}{6}\linebox{\msample}
167 \end{tabular}
168 \end{center}
```

That's all on one test page.

```
169 \end{document}
170 ⟨/testfile⟩
```