

The fontscale package

A flexible interface for setting font sizes

Oliver Beery

Version 3.1.0 3 February 2025

1 Introduction

1.1 About

The fontscale package provides the following functionalities:

- Set font sizes using a classic or musical typographic scale (§2.1).
- Set arbitrary font sizes and font baselineskips for the standard L^AT_EX font size commands `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, and `\Huge` (§2.3).
- Set the font sizes and font baselineskips of the standard L^AT_EX font size commands mid-document or within a local group (§3.1).
- Provides expandable commands that expand to the current font step, font scale, font size, or font baselineskip (§3.2).
- Provides additional font size commands, including setting the font size by setting the font step or font scale and setting the font size relative to the current font size using more robust tools than the `scalefont` and `relsize` packages (§3.3).
- For L^AT_EX3 programmers, this package defines public `expl3` variables that store the font step, font scale, font size, and font baselineskip of each font size command from `\tiny` to `\Huge` and the current font step, font scale, font size, and font baselineskip (§4.2).

1.2 Loading the package

Requirements:

- L^AT_EX 2_ε version 2023-11-01 or newer
- l3kernel version 2023-10-10 or newer

You may need to ensure that your L^AT_EX installation is up-to-date before using this package.

This package lets the user set arbitrary font sizes. This works with most fonts produced today, but, for historical reasons, the default Computer Modern font is available only in a number of discrete font sizes. If you get a warning that Computer Modern is not available in the requested font size, you may need to add the code `\RequirePackage{fix-cm}` before `\documentclass` to make Computer Modern available at arbitrary font sizes. Alternatively, you can use the Latin Modern font by loading the `lmodern` package.

The fontscale package has no package options. Instead, this package provides the command `\fontscalesetup{key = value list}` (§3.1) which sets the package keys (§2).

When loaded, this package uses `\normalsize` after defining and initializing the font size commands from `\tiny` to `\Huge`.

Many L^AT_EX document classes have a font size option (e.g. `10pt`, `11pt`, `12pt`) which not only changes the set of document font sizes, but also modifies additional settings such as the page layout and vertical spacing which were specifically designed to work with those font sizes. For this reason, you may want to set the document class font size option close to the font size of `\normalsize` set by this package.

This package defines each font size command from `\tiny` to `\Huge` using `\fontsize` and `\selectfont`. They have no additional functionality beyond that of `\fontsize` and `\selectfont`, except that they:

- Set the kernel command `\@currsize` equal to the font size command. This is needed only for compatibility.
- Do nothing, except issue a warning, if used in math mode.

Unlike the standard L^AT_EX document classes, the font size commands from `\tiny` to `\Huge` defined by this package do not change the vertical spacing for displayed math and list structures. If the user would like to add this functionality or perhaps other features to the font size commands, they can use hooks, which are documented in `lthooks` and `ltxcmdhooks`.

1.3 The font size parameters

When using the `fontscale` package, each font size, including the standard L^AT_EX font size commands from `\tiny` to `\Huge`, has a font step, font scale, font size, and font baselineskip. The latter two parameters are the familiar first and second arguments to `\fontsize{<dimen>}{<skip>}`. (The font baselineskip should not be confused with the paragraph baselineskip `\baselineskip`.) The font scale and font step are different ways of describing the font size. The font scale is the relative font size; the ratio of the font size to `\normalsize`, which has a font scale of 1. The font step is the number of font size gradations from `\normalsize`, which has a font step of 0. Table 1 displays the font step of each font size command.

1.4 Syntax

This package defines some keys and commands that take as a value or argument a *floating point expression*, *integer expression*, *dimen expression*, or *skip expression*. This syntax has the same representation as the arguments to `\fpeval`, `\inteval`, `\dimeval`, and `\skipeval`, documented in `usrguide`.

2 Keys

This section documents the keys provided by the `fontscale` package. This package has no package options. Set the package keys using `\fontscalesetup{<key = value list>}` (§3.1).

Table 1: The font step of each font size command from `\tiny` to `\Huge`. These are constants and cannot be changed.

font size command	font step
<code>\tiny</code>	-4
<code>\scriptsize</code>	-3
<code>\footnotesize</code>	-2
<code>\small</code>	-1
<code>\normalsize</code>	0
<code>\large</code>	1
<code>\Large</code>	2
<code>\LARGE</code>	3
<code>\huge</code>	4
<code>\Huge</code>	5

2.1 The key `typographic-scale`

The font sizes of the font size commands from `\tiny` to `\Huge` are initially set by the key `typographic-scale`.

```
typographic-scale = classic-10pt | classic-11pt | classic-12pt |
musical initial = classic-10pt
classic-10pt
classic-11pt
classic-12pt
musical
```

The value of the key `typographic-scale` must be `classic-10pt`, `classic-11pt`, `classic-12pt`, or `musical`. The initial value is `classic-10pt`. Sets the font size of each font size command from `\tiny` to `\Huge` using a classic or musical typographic scale. These are common methods of choosing a set of document font sizes. The keys `classic-10pt`, `classic-11pt`, `classic-12pt`, and `musical` set the key `typographic-scale` to the corresponding value.

The classic typographic scale consists of the traditional font sizes 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 21, 24, 36, 48, 60, and 72.¹ They have been used since the sixteenth century and are the default font sizes on most computer software. Setting the value of the key `typographic-scale` to `classic-10pt`, `classic-11pt`, or `classic-12pt` sets the font size of each font size command from `\tiny` to `\Huge` to values based on their position in the classic typographic scale. The named point size — 10 pt, 11 pt, or 12 pt — is the font

1. See §3.1 of *The Elements of Typographic Style* by Robert Bringhurst.

Table 2: The font size of each font size command from `\tiny` to `\Huge` in units of pt when using a classic typographic scale.

font size command	classic-10pt	classic-11pt	classic-12pt
<code>\tiny</code>	6	7	8
<code>\scriptsize</code>	7	8	9
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	11
<code>\normalsize</code>	10	11	12
<code>\large</code>	11	12	14
<code>\Large</code>	12	14	16
<code>\LARGE</code>	14	16	18
<code>\huge</code>	16	18	21
<code>\Huge</code>	18	21	24

size of `\normalsize`. The font sizes of the other font size commands are the adjacent font sizes in the classic typographic scale. Table 2 displays the font size of each font size command when using a classic typographic scale.

<code>musical/base = <dimen expression></code>	<code>initial = 10pt</code>
<code>musical/ratio = <floating point expression></code>	<code>initial = 2</code>
<code>musical/notes = <integer expression></code>	<code>initial = 5</code>

The key `typographic-scale=musical` sets the font size of each font size command from `\tiny` to `\Huge` using a musical typographic scale. The font sizes are calculated using the formula:²

$$f_i = f_0 \times r^{i/n} \quad (1)$$

f_i is the font size with font step i . f_0 is the base font size. n is the number of musical notes — the number of font size gradations or steps above f_0 . r is the musical ratio, the ratio of the highest to the lowest note f_n/f_0 .

The key `musical/base` sets the base font size to the value of `<dimen expression>`, appending a default unit of pt. The base font size is the font size of `\normalsize`. The initial value is 10 pt. The key `musical/ratio` sets the musical ratio to the result of computing the `<floating point expression>`. The initial value is 2. The key `musical/notes` sets the number of musical notes to the value of `<integer expression>`. The initial value is 5. Table 3 displays the font size of each font size command when using a musical typographic scale with the initial values.

2. I have referenced this article by Spencer Mortensen:
<https://spencermortensen.com/articles/typographic-scale/>

Table 3: The font size of each font size command from `\tiny` to `\Huge` when using a musical typographic scale with the initial values. The font sizes are in units of pt and rounded to 2 decimal places.

font size command	musical
<code>\tiny</code>	5.74
<code>\scriptsize</code>	6.60
<code>\footnotesize</code>	7.58
<code>\small</code>	8.71
<code>\normalsize</code>	10
<code>\large</code>	11.49
<code>\Large</code>	13.20
<code>\LARGE</code>	15.16
<code>\huge</code>	17.41
<code>\Huge</code>	20

2.2 The key `baselineskip-size-ratio`

The font baselineskips of the font size commands from `\tiny` to `\Huge` are initially set by the key `baselineskip-size-ratio`.

```
baselineskip-size-ratio = <floating point expression>      initial = 1.2
```

Sets the font baselineskip of each font size command from `\tiny` to `\Huge` equal to its font size \times the result of computing the *<floating point expression>*. Sets the font baselineskip set by `\SetFontStep`, `\SetFontScale`, and `\SetFontSize` equal to the new font size \times the result of computing the *<floating point expression>* (§3.3). The initial value is 1.2.

2.3 Overwriting the previous keys

This subsection documents keys for directly setting the font sizes and font baselineskips of the font size commands from `\tiny` to `\Huge`. The font sizes can also be set by setting the font scale.

The user should take care to ensure that the lengths of the font sizes remain correctly ordered from `\tiny` to `\Huge`. This is important for typographic and syntactic consistency. If the font sizes are in the wrong order, `\fontscasesetup` will issue a warning and some package features may not work correctly.

The syntax ** represents the name of a font size command from `\tiny` to `\Huge`, omitting the backslash character.

```
<font size command>/scale = <floating point expression>    initial = <not set>
```

Sets the font size of ** by setting its font scale to the result of computing the *<floating point expression>*. Overwrites the font size set by the key `typographic-scale`. These keys are initially not set. The key `normalsize/scale` is not defined.

```
<font size command>/size = <dimen expression>            initial = <not set>
```

Sets the font size of ** to the value of *<dimen expression>*, appending a default unit of pt. Overwrites the font size set by the keys `typographic-scale` and */scale*. These keys are initially not set.

```
<font size command>/baselineskip = <skip expression>    initial = <not set>
```

Sets the font baselineskip of ** to the value of *<skip expression>*, appending a default unit of pt. Overwrites the font baselineskip set by the key `baselineskip-size-ratio`. These keys are initially not set.

```
<font size command> = <dimen expression>  
<font size command> = <dimen expression>/<skip expression>
```

Sets the keys */size = <dimen expression>* or */baselineskip = <skip expression>* in a condensed format.³ A “/” contained within a brace group will not be treated as the delimiter.

3 Commands

This section documents the commands provided by the `fontscale` package.

3.1 Setting the keys

```
\fontscalesetup <*> {<key = value list>}
```

Sets and processes the `fontscale` package keys (§2) in *<key = value list>* and then uses `\normalsize`. Adding the optional star *<*>* first resets all the `fontscale` package keys to their initial values. The order of the keys in *<key = value list>* does not affect how the keys are processed (unless a key is set more than once). Can be used mid-document. The scope of the effect is local to the current group. Does nothing, except issue a warning, if used in math mode. `\fontscalesetup` should typically be used only once in the preamble or omitted if the user is satisfied with the initial font sizes and font baselineskips.

3. The latter format resembles traditional typographic notation where the size and leading of a text can be expressed as, for example, 10/12 or 11/13. The number before/after the slash is the size/leading in units of pt.

3.2 Expandable font size parameters

```
\CurrentFontStep  
\CurrentFontScale  
\CurrentFontSize  
\CurrentFontBaselineskip
```

This package provides expandable commands that expand to the current font step, font scale, font size, or font baselineskip. They can be used for printing or within calculations.

For clarity, `\CurrentFontStep` will be explained in detail. If the current font size equals the font size of any font size command from `\tiny` to `\Huge`, then `\CurrentFontStep` expands to the font step of that font size command (Table 1). If not and if the value of the key `typographic-scale` is `musical`, `\CurrentFontStep` expands to the font step calculated from the musical typographic scale (Equation 1). Otherwise, the current font step is undefined and `\CurrentFontStep` expands to nothing.

3.3 More font size commands

This package provides additional font size commands for use only in special cases. Users should prefer the standard L^AT_EX font size commands from `\tiny` to `\Huge` for typographic and syntactic consistency.

```
\SetFontStep <*> {<floating point expression>}
```

Sets the font size by setting the font step to the result of computing the *<floating point expression>*. Adding the optional star *<*>* instead sets the font step equal to the current font step + the result of computing the *<floating point expression>*. Sets the font baselineskip equal to the new font size \times the value of the key `baselineskip-size-ratio`.

Three exceptions:

- If the font step equals the font step of any font size command from `\tiny` to `\Huge` (Table 1), then that font size command will be used directly.⁴
- `\SetFontStep` will issue an error if the font step is undefined (as explained for `\CurrentFontStep` in §3.2). The font step must equal the font step of any font size command from `\tiny` to `\Huge` unless the value of the key `typographic-scale` is `musical`.⁵ `\SetFontStep*` will always issue an error if the current font step is undefined.
- Does nothing, except issue a warning, if used in math mode.

```
\SetFontScale <*> {<floating point expression>}
```

Sets the font size by setting the font scale to the result of computing the *<floating point expression>*. Adding the optional star *<*>* instead sets the font

4. The font step is rounded to 5 decimal places when testing equality due to floating point precision.

5. See Footnote 4.

scale equal to the current font scale + the result of computing the *floating point expression*. Sets the font baselineskip equal to the new font size \times the value of the key `baselineskip-size-ratio`. Does nothing, except issue a warning, if used in math mode.

```
\SetFontSize <*> {<dimen expression>}
```

Sets the font size to the value of *dimen expression*, appending a default unit of pt. Adding the optional star *<*>* instead sets the font size equal to the current font size + the result of *dimen expression*, appending a default unit of pt. Sets the font baselineskip equal to the new font size \times the value of the key `baselineskip-size-ratio`. Does nothing, except issue a warning, if used in math mode.

```
\ScaleFont {<floating point expression>}
```

Sets the font size equal to the current font size \times the result of computing the *floating point expression*. Sets the font baselineskip equal to the current font baselineskip \times the result of computing the *floating point expression*. The new font baselineskip has no stretch and shrink components. Does nothing, except issue a warning, if used in math mode. This command is intended as a more robust alternative to `\scalefont` from the `scalefont` package.

```
\SetFontSizeBaselineskip {<dimen expression>} {<skip expression>}
```

Sets the font size to the value of *dimen expression*, appending a default unit of pt. Sets the font baselineskip to the value of *skip expression*, appending a default unit of pt. Does nothing, except issue a warning, if used in math mode. This command is intended as a more robust alternative to `\fontsize + \selectfont`.

3.4 Setting only the font baselineskip

```
\SetFontBaselineskip <*> {<skip expression>}
```

Sets the font baselineskip to the value of *skip expression*, appending a default unit of pt. Adding the optional star *<*>* instead sets the font baselineskip equal to the current font baselineskip + the result of *skip expression*, appending a default unit of pt. Does not change the font size. Does nothing, except issue a warning, if used in math mode.

3.5 Testing and debugging

This subsection documents commands that are intended primarily for testing and debugging. I have used them often when developing this package so I think users will find them helpful.

```
\PrintFontSizeParameters
```

Prints the current font step, font scale, font size, and font baselineskip. The printed font step and font scale are rounded to 5 decimal places. To print with more decimal places, use `\CurrentFontStep` and `\CurrentFontScale` (§3.2).

The printed font step will be left blank if it is undefined (as explained for `\CurrentFontStep` in §3.2).

`\PrintAllFontSizeParameters`

Prints the font step, font scale, font size, and font baselineskip of each font size command from `\tiny` to `\Huge`. The printed font scale is rounded to 5 decimal places. To print with more decimal places, use `\CurrentFontScale` (§3.2). Cannot be used in math mode.

`\PrintSampleText` *{*}* *{text}*

Prints *text* in each font size ordered from `\tiny` to `\Huge` each followed by a paragraph break. *text* can contain `\par` tokens. Adding the optional star *{*}* reverses the order of the font sizes. Cannot be used in math mode. One useful way of printing sample text is `\PrintSampleText{\PrintFontSizeCommand:\PrintFontSizeParameters}`.

`\PrintFontSizeCommand`

Tests if the current font size equals the font size of any font size command from `\tiny` to `\Huge`. If so, prints the name of that font size command. If not, prints “\undefined”. Cannot be used in math mode.

4 Programming

This section documents the `expl3` programming support provided by the `fontscale` package.

4.1 Compatibility with `\text_purify:n`

`\text_purify:n` will correctly remove the formatting commands defined by this package. This includes the commands documented in §3.1, §3.3, and §3.4.

4.2 Public functions and variables

This package does not define any public `expl3` functions.

This package defines some public `expl3` variables. They are set either by `\fontscylesetup` or in the `selectfont` hook. They should never be modified directly. The syntax *font size command* represents the name of a font size command from `\tiny` to `\Huge`, omitting the backslash character.

```
\c_fontscale_<font size command>_step_fp  
\l_fontscale_<font size command>_scale_fp  
\l_fontscale_<font size command>_size_dim  
\l_fontscale_<font size command>_baselineskip_skip
```

Stores the font step, font scale, font size, and font baselineskip of each font size command from `\tiny` to `\Huge`. Exception: `\l_fontscale_normalsize_scale_fp` is not defined. The font scale of `\normalsize` is stored in `\c_fontscale_normalsize_scale_fp`. The local variables are set by `\fontscylesetup`.

```
\l_fontscale_step_fp  
\l_fontscale_scale_fp  
\l_fontscale_size_dim  
\l_fontscale_baselineskip_skip
```

Stores the current font step, font scale, font size, and font baselineskip.

`\l_fontscale_step_fp` equals `nan` (not a number) if the current font step is undefined (as explained for `\CurrentFontStep` in §3.2). These variables are set in the `selectfont` hook.