

---

## The `dashundergaps` package\*

Frank Mittelbach

### Abstract

The `dashundergaps` package offers the possibility to replace material in running text with white space in order to build up forms that can be filled in at a later time.

By default the gaps are underlined and followed by a gap number in parentheses, but many other designs are possible, e.g., dashes or dots instead of the underline, no gap numbers or a different format for them, gap widening for easier fill-in, etc.

There is also a teacher’s mode which shows the normally hidden text in a special (customizable) format.

This is another article in a series of *TUGboat* articles describing small packages to introduce coding practices using the `expl3` programming language. See [1] for the first article in the series. For more details on `expl3` refer to [2].

### Contents

1	Introduction	1
2	The user interface	2
2.1	Options to customize the gap display . . . . .	3
2.1.1	Gap modes . . . . .	3
2.1.2	Gap formatting . . . . .	3
2.1.3	Gap numbers . . . . .	4
2.1.4	Gap widening . . . . .	4
3	Differences from the original package	4
4	Solution to the puzzle	5
	Index	6

## 1 Introduction

The `dashundergaps` package provides a single command `\gap` which takes one argument and produces a gap of the width of that argument. To better mark this gap it is underlined in some form (could be a solid line, a dashed or dotted line or even a wiggling line). Furthermore, gaps can be numbered to be able to easily refer to them. Figure 1 shows an example in the form of a fill-in puzzle.

As you see there, some gaps are numbered with a superscript number (not the default setting) while others aren’t. How this is done and how to change the result is explained in the next section.

There also exists a “teacher mode” in which the gaps are filled with the text given in the argument. This can be used to show the correct answers of a test (as we do in Section 4) or to give a sample fill-in for a form, to help people fill it out correctly. The “teacher mode” produces the same line breaks because it ensures that the fill-ins take the same amount of space as the gaps.

Another important feature is the possibility to artificially widen the gaps, compared to the textual material in the argument. After all, when a form is filled by hand people typically need more space to write some text compared to the same text being typeset. So making the gaps simply as wide as the material likely results in too little space.

---

\* This is a reimplementaion (using `expl3`, the  $\text{\LaTeX}3$  programming language) of a package originally written by Luca Merciadri in 2010. The current package version is v2.0h dated 2021-03-05.

The initial ‘E.’ in Donald E. Knuth’s name stands for \_\_\_\_\_<sup>(1)</sup>. The well-known answer to the Ultimate Question \_\_\_\_\_ is 42 according to \_\_\_\_\_<sup>(2)</sup>. The first edition of \_\_\_\_\_<sup>(3)</sup> celebrates its silver anniversary in 2019. Historically speaking, `exp13` stands for \_\_\_\_\_<sup>(4)</sup> even though it is a production language these days.

And here are some hints for the puzzle if you want to fill it out:

1. If only everything would be that easy to answer.
2. The author of the book “Last Chance To See” and of a famous radio show.
3. Back then known as the doggie book.
4. Old names die hard.

The answers are given in Section 4, showing the gaps filled in using the so-called teacher mode, which can be activated or deactivated at any point in the document.

**Figure 1:** A fill-in puzzle using `dashundergaps`

## 2 The user interface

The `dashundergaps` package is built as a small application on top of the `ulem` package, a package that defines several commands for underlining *simple-text* in various ways.

<code>\uline</code>	<code>\uline{&lt;simple-text&gt;} \uwave{&lt;simple-text&gt;} ...</code>
<code>\uuline</code>	This means that by loading <code>dashundergaps</code> the <code>ulem</code> commands such as <code>\uline</code> , <code>\uwave</code> and so forth are automatically made available. These commands are used to do most of the work and the current package only makes sure that, instead of the words, empty boxes of the same width are used by <code>ulem</code> . This way we get underlined gaps of the right size.
<code>\uwave</code>	
<code>\dashuline</code>	
<code>\dotuline</code>	

By default, `ulem` changes `\emph` to underline text, so for this application, it is loaded with the option `normalem` to prevent that from happening.

<code>\gap</code>	<code>\gap*[&lt;style&gt;]{&lt;text&gt;}</code>
-------------------	---

Possible *<style>*s:

`u = \uline`  
`d = \uuline`  
`w = \uwave`  
`b = <blank>`  
`- = \dashuline`  
`. = \dotuline`

The main command provided by the package is `\gap` which expects a mandatory *<text>* argument containing the material that is used to produce the gap (and is normally invisible). By default the gap is underlined, though that can be changed.

The optional *<style>* argument explicitly defines a certain type of underlining: `u` stands for normal underlining (via `\uline`), `d` for double underlining (via `\uuline`), `w` for a wavy line (via `\uwave`), `b` for blank (i.e., no underlining whatsoever), “-” for a dash-line (via `\dashuline`) and finally “.” for underlining with dots (via `\dotuline`).

In the default configuration gaps are numbered using the counter `gapnumber` and this number is shown in parentheses after the gap. With the star form the generation of the number is toggled, i.e., if it would be produced because of the current option settings it will be suppressed; if it is suppressed through an option it will be typeset. This way one can select the most convenient setting via an option for the whole document and use `*` to toggle it as needed.

Since `\gap` uses `ulem`’s commands it inherits the limitations of these commands; notably, only simple text can be used in the *<text>* argument. For example, a `\footnote` couldn’t be used in the argument (but then that wouldn’t make much sense in a gap, would it?).

Another important (and sometimes annoying) restriction is that any brace group or command with arguments inside the `ulem` commands is set as if it is inside an `\mbox`, in particular it will not break across lines. For example `\gap{\emph{...}}` will show this behavior. The `dashundergaps` package tries to mediate as best as possible, e.g., `\gap{\mytext}` works, but in other cases you have to live with this limitation.

---

```
\TeacherModeOn % show gap material
\TeacherModeOff % do not show gap material
```

---

Also supported is a teacher mode in which the material for the gaps is visible. This can be used to show the expected answers in case `\gap` is used for preparing tests, or to show a sample fill-in of a form. The teacher mode can be turned on or off anywhere in the document using `\TeacherModeOn` or `\TeacherModeOff`, respectively. Alternatively, it can also be set via an option, as we will see below.

---

```
\dashundergapssetup{(comma-separated key-value list)}
```

---

The package can be loaded with a number of options (discussed in Section 2.1). A likely better approach is to set any options with the declaration `\dashundergapssetup` which is normally used in the preamble, but can be used throughout the document to change settings on the fly. It only changes explicitly given options so it can be used to overwrite some defaults but leave everything else unchanged.

## 2.1 Options to customize the gap display

All of the package options are implemented as key/value options. For boolean options one can give just the option name as a short form for setting the option to `true`. Most options can be specified during package loading in the optional argument of `\usepackage`. However if the value requires some L<sup>A</sup>T<sub>E</sub>X code (e.g., `gap-font`, which expects a font declaration command) then this will not work due to some limitations in the current L<sup>A</sup>T<sub>E</sub>X package loader. For such options use `\dashundergapssetup` instead, which will always work.

### 2.1.1 Gap modes

The general processing mode is defined through the following options:

**teacher-mode** Boolean that turns on teacher mode (i.e., the gap material will be visible if set to `true`). Its default is `false`.

**gap-mode** Boolean that is the inverse of `teacher-mode` and just provided for convenience, i.e., an abbreviation for `teacher-mode=false`.

**teachermode** Alternative name for `teacher-mode` because that is what it was called in the first package release.

### 2.1.2 Gap formatting

Formatting of the gaps is handled by the following six options:

**gap-format** A choice option defining how the gap is marked. It accepts the following values: `underline` (default), `double-underline`, `dash`, `dot`, `wave`, `blank`.

**gap-format-adjust** A boolean (default `true`). If set, the “line” below the gap is raised to be roughly at the baseline, which normally looks better when there is no text above the line.

**teacher-gap-format** Another choice option, with the same values as `gap-format`, used when we are in “teacher mode”, but this time the default is `blank` as normally the gap text is typeset in the bold font and is therefore already identifiable, with no need for additional underlining. However, depending on the circumstances it might be helpful to keep the underlining (or use a different kind of underlining) while in “teacher mode”.

**gap-font** This option expects a font directive as its value, e.g., `\bfseries` (which is also the default). Using this option without supplying a value is equivalent to supplying an empty value. It will be used to determine the font for the gap material regardless of the mode. This is important to ensure that the gaps always have the same width regardless of whether or not the material is shown.

For the example puzzle above it was set to `\itshape`, which you can see in the puzzle answer.

**dash** Short name for `gap-format=dash`.

**dot** Short name for `gap-format=dot`.

### 2.1.3 Gap numbers

Producing the gap numbers is handled by the following options:

**gap-numbers** Boolean that determines whether or not gap numbers are displayed. Default is `true`.

**gap-number-format** Code that is executed when a gap number is produced. Default is `\textnormal{\_(\thegapnumber)}`.

**numbers** Short name for `gap-numbers`.

There is also a way to control displaying the total number of gaps:

**display-total-gaps** Boolean to determine if the total number of gaps should be shown at the very end of the document. Default is `false`.

**displaynbgaps** This is just another name for the same boolean; it was used in the first version of the package.

### 2.1.4 Gap widening

Finally, for extending the gap width we have these options:

**gap-widen** Boolean that decides if the gaps should be made wider or not (default is `false` but mainly for historical reasons).

**gap-extend-minimum** Minimum of extra space that should be added to each gap if gap widening is active. Default is `20pt`, i.e., `10pt` on either side.

**gap-extend-percent** Percentage (as a number) by which the gap should be made wider if widening is active. The result is compared to `gap-extend-minimum` and the larger of the two is used. Default is `20`.

**widen** Short name for `gap-widen`.

## 3 Differences from the original package

The main user interface of the two versions is identical, so it is possible to use the new version as a drop-in replacement for the old. However, the feature set in form of key/value options has been greatly extended, offering functionality previously unavailable. Furthermore, a number of bugs have been corrected (and possibly new ones introduced).

- Stray spaces in the definition of `\gap` (that showed up in the output) have been eliminated.
- Various combinations of options that didn't work are now possible.
- Explicit hyphenations `\-` showed up in gap mode, now they can be used.
- Nesting isn't possible for obvious reasons, but the fact is now detected and catered to by ignoring the inner gap requests after generating an error.
- Option names have been normalized (though the original names are still available).
- The option `phantomtext` is no longer necessary, though still supported (with a warning) as a no-op.

- The names of the L<sup>A</sup>T<sub>E</sub>X counters used have changed, so if you directly addressed them that would need changing.
- The font used in teacher mode (by default boldface) is now also used if gap mode is chosen, to ensure that the output in all modes produces identical line breaks; for the same reason, the `ulem` machinery is always used, even if not underlining (or dashing, etc.).
- The gaps can be extended by a percentage or by a minimum amount to ensure that there is enough space to fill in the text (given that hand-written text is typically wider than typeset material); the values are adjustable.
- `\gap` now has an optional argument through which you can explicitly request the type of underlining you want to use.
- `\gap` also supports a star form which toggles the setting of gap numbers.
- The use of `\label` within the `\gap` command argument allows for later reference to that gap by its number (provided a gap number is typeset).
- The implementation is done with `expl3`, the programming language for L<sup>A</sup>T<sub>E</sub>X3. Although invisible to the user, in some sense that was the main purpose of the exercise: to see how easy it is to convert a package and use the extended features of `expl3`.

#### 4 Solution to the puzzle

Here we repeat the puzzle from above with `\TeacherModeOn`.

The initial ‘E.’ in Donald E. Knuth’s name stands for Ervin <sup>(5)</sup>. The well-known answer to the Ultimate Question of Life, the Universe, and Everything is 42 according to Douglas Adams <sup>(6)</sup>. The first edition of The L<sup>A</sup>T<sub>E</sub>X Companion <sup>(7)</sup> celebrates its silver anniversary in 2019. Historically speaking, `expl3` stands for EXperimental Programming Language 3 <sup>(8)</sup> even though it is a production language these days.

This was produced using the following changes to the defaults:

```
\dashundergapssetup{
  ,gap-number-format = \,\textsuperscript{\normalfont
                                (\thegapnumber)}
  ,gap-font          = \itshape
  ,teacher-gap-format = underline
  ,gap-widen
}
```

As you can see we use `\itshape` for the font (to be able to show the bold face in one of the answers) and also force underlining in teacher mode to better show the gap widening. The gap number is raised and we separate it a tiny bit from the gap material. We also use `\normalfont` in the formatting to ensure that the gap number is set upright and not in italic shape.

#### References

- [1] Frank Mittelbach. The widows-and-orphans package. *TUGboat* 39:3, 20018. <https://ctan.org/pkg/widows-and-orphans>
- [2] L<sup>A</sup>T<sub>E</sub>X3 Project Team. A collection of articles on `expl3`. <https://latex-project.org/publications/indexbytopic/l3-expl3/>

**Index**

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

	<b>D</b>			<b>M</b>
<code>\dashuline</code>	.....	<i>2, 2</i>	<code>\mbox</code>	..... <i>3</i>
<code>\dashundergapssetup</code>	.....	<i>3</i>		<b>N</b>
<code>\dotuline</code>	.....	<i>2, 2</i>	<code>\normalfont</code>	..... <i>5</i>
	<b>E</b>			<b>T</b>
<code>\emph</code>	.....	<i>2</i>	<code>\TeacherModeOff</code>	..... <i>3</i>
	<b>F</b>		<code>\TeacherModeOn</code>	..... <i>3, 5</i>
<code>\footnote</code>	.....	<i>2</i>		<b>U</b>
	<b>G</b>		<code>\uline</code>	..... <i>2, 2</i>
<code>\gap</code>	.....	<i>1, 2, 3, 4, 5</i>	<code>\usepackage</code>	..... <i>3</i>
	<b>L</b>		<code>\uuline</code>	..... <i>2, 2</i>
<code>\label</code>	.....	<i>5</i>	<code>\uwave</code>	..... <i>2, 2</i>

◇ Frank Mittelbach  
 Mainz, Germany  
<https://www.latex-project.org>  
<https://ctan.org/pkg/dashundergaps>