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

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 wriggling 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

* This is a reimplementation (using expl3, the LaTeX3 programming language) of a package originally written by Luca Merciadri in 2010.

The initial 'E.' in Donald E. Knuth's name stands for _____(1). The well-known answer to the Ultimate Question _____ is 42 according to _____ _____(2). The first edition of _____(3) celebrates its silver anniversary in 2019. Historically speaking, `expl3` stands for _____ ____(4) 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

"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.

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.

`\uline` `\uuline` `\uwave` `\dashuline` `\dotuline`

`\uline{`⟨*simple-text*⟩`}` `\uwave{`⟨*simple-text*⟩`}` ...

This means that by loading dashundergaps the ulem commands such as `\uline`, `\uwave` 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 ulem. This way we get underlined gaps of the right size.

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

`\gap`

`\gap*[`⟨*style*⟩`]{`⟨*text*⟩`}`

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.

Possible ⟨*style*⟩s:
`u` = `\uline`
`d` = `\uuline`
`w` = `\uwave`
`b` = ⟨*blank*⟩
`-` = `\dashuline`
`.` = `\dotuline`

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?).

`\TeacherModeOn` `\TeacherModeOn % show gap material`
`\TeacherModeOff` `\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` `\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 LaTeX code (e.g., `gap-font`, which expects a font declaration command) then this will not work due to some limitations in the current LaTeX 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 LaTeX 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 LaTeX3. 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* (5). The well-known answer to the Ultimate Question *of Life, the Universe, and Everything* is 42 according to *Douglas Adams* (6). The first edition of *The LaTeX Companion* (7) celebrates its silver anniversary in 2019. Historically speaking, `expl3` stands for ***EX**perimental **P**rogramming **L**anguage **3*** (8) 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.

5 The implementation

5.1 Loading and fixing/changing ulem

The first thing to do is to load ulem without changing \emph or \em:

```
⟨*package⟩
\RequirePackage[normalem]{ulem}
```

The code in this section follows LaTeX 2ε conventions, i.e., models the commands as they look in the ulem package.

\dotuline The dots produced by \dotuline depend on the current font, which is a somewhat questionable design — if you underline a text part with a single bold word somewhere inside it will change the shape of the dot line. So we always use the \normalfont dot (this is not done in the original definition).

```
\def\dotuline{\bgroup
  \UL@setULdepth
  \markoverwith{\begingroup
     \advance\ULdepth0.08ex
     \lower\ULdepth\hbox{\normalfont \kern.1em .\kern.04em}%
     \endgroup}%
  \ULon}
\MakeRobust\dotuline
```

(*End definition for* \dotuline*. This function is documented on page 264.*)

\uwave The original \uwave used a hard-wired value of 3.5pt for the lowering. We change that to be based on the current value of \ULdepth so that the user (or this package here) can change the placement.

```
\def\uwave{\bgroup
  \UL@setULdepth
  \advance\ULdepth 0.6\p@
  \markoverwith{\lower\ULdepth\hbox{\sixly \char58}}\ULon}
\MakeRobust\uwave
```

(*End definition for* \uwave*. This function is documented on page 264.*)

\fmdug@ublank \fmdug@ublank underlines with blanks. Normally not especially useful (which is why we make it internal), but if we want to have ulem acting, but without actually visibly underlining, this is the command to use.

```
\def\fmdug@ublank{\bgroup\let\UL@leadtype\@empty\ULon}
```

(*End definition for* \fmdug@ublank*.*)

\UL@dischyp We need to do a little patching to ensure that nothing is output by the ulem commands
\UL@putbox if we don't want it to. So the next two commands are from ulem with \box replaced by \fmdug@box so that we can change the behavior.

```
\def\UL@dischyp{\global\setbox\UL@hyphenbox\hbox
   {\ifnum \hyphenchar\font<\z@ \string-\else \char\hyphenchar\font \fi}%
   \kern\wd\UL@hyphenbox \LA@penalty\@M
   \UL@stop \kern-\wd\UL@hyphenbox
   \discretionary{\fmdug@box\UL@hyphenbox}{}{}\UL@start}
\def\UL@putbox{\ifx\UL@start\@empty \else % not inner
  \vrule\@width\z@ \LA@penalty\@M
  {\UL@skip\wd\UL@box \UL@leaders \kern-\UL@skip}%
  \fmdug@box\UL@box  \fi}
```

(*End definition for* \UL@dischyp *and* \UL@putbox*.*)

\fmdug@box By default we output the box in the commands above, but when we don't want to output anything visible we change the definition to generate a box with empty content but the right size.

```
\let\fmdug@box\box
```

(*End definition for* `\fmdug@box`.)

5.2 The main implementation part

The rest of the package is written in `expl3`. We use `fmdug` as our internal prefix.

```
⟨@@=fmdug⟩
```

We need the package `xparse` for specifying the document-level interface commands and `l3keys2e` to use the `expl3` key value methods within LaTeX 2_ε. These packages automatically require `expl3` so there is no need to load that explicitly.

```
\RequirePackage{xparse,l3keys2e}
```

Here we introduce the package and specify its version number:

```
\ProvidesExplPackage{dashundergaps}
                    {2018/11/09}
                    {v2.0d}
                    {Dashing and underlining phantom text}
```

5.2.1 User interface commands

\gap The `\gap` command parses for a star, optional and mandatory argument and then calls `\__fmdug_gap:nnn` to do the work.

```
\DeclareDocumentCommand \gap { som } { \__fmdug_gap:nnn {#1}{#2}{#3} }
```

(*End definition for* `\gap`. *This function is documented on page 264.*)

\dashundergapssetup Change options anywhere.

```
\NewDocumentCommand \dashundergapssetup { m }
  { \keys_set:nn {fmdug} {#1} \ignorespaces }
```

(*End definition for* `\dashundergapssetup`. *This function is documented on page 265.*)

\TeacherModeOn We provide shortcuts for turning teacher mode on or off.
\TeacherModeOff

```
\DeclareDocumentCommand \TeacherModeOn  {}
                        { \bool_set_true:N \l__fmdug_teacher_bool }
\DeclareDocumentCommand \TeacherModeOff {}
                        { \bool_set_false:N \l__fmdug_teacher_bool }
```

(*End definition for* `\TeacherModeOn` *and* `\TeacherModeOff`. *These functions are documented on page 265.*)

5.2.2 Counters

\c@gapnumber We have one user-level counter which is referenceable and holds the gap number of the current gap. It can be reset to 0 to restart counting.

```
\newcounter{gapnumber}
```

(*End definition for* `\c@gapnumber`.)

\c@totalgapnumber We also keep track of all gaps ever made using another user-level counter. Since this one is supposed to keep track of the total number of gaps, it makes little sense to modify it at the document level. However, there may be use cases even for that and more importantly, by making it a user-level counter it is possible to refer to the total

number of gaps easily, e.g., via \thetotalgapnumber.

```
\newcounter{totalgapnumber}
```

(*End definition for* \c@totalgapnumber.)

\l__fmdug_extend_dim A help register to calculate the gap width later on.

```
\dim_new:N \l__fmdug_extend_dim
```

(*End definition for* \l__fmdug_extend_dim.)

\l__fmdug_extra_left_gap_tl
\l__fmdug_extra_right_gap_tl Two scratch token lists to enlarge the gap on the left or right side.

```
\tl_new:N \l__fmdug_extra_left_gap_tl
\tl_new:N \l__fmdug_extra_right_gap_tl
```

(*End definition for* \l__fmdug_extra_left_gap_tl *and* \l__fmdug_extra_right_gap_tl.)

\l__fmdug_gap_format_tl
\l__fmdug_teacher_gap_format_tl The gap formatting is normally handled by a ulem command; which one depends on the options used. To record the choice we store it in a token list (one for normal and one for teacher mode).

```
\tl_new:N \l__fmdug_gap_format_tl
\tl_new:N \l__fmdug_teacher_gap_format_tl
```

(*End definition for* \l__fmdug_gap_format_tl *and* \l__fmdug_teacher_gap_format_tl.)

5.2.3 Messages

```
\msg_new:nnn {dashundergaps} {deprecated}
  { The~ #1~ `#2'~ you~ used~ \msg_line_context: \ is~ deprecated~ and~
    there~ is~ no~ replacement.~ Since~ I~ will~ not~ guarantee~ that~
    #1~ `#2'~ will~ be~ kept~ forever~ I~ strongly~ encourage~ you~
    to~ remove~ it~ from~ your~ document. }
\msg_new:nnnn {dashundergaps} {nested}
  { The~ \gap command~ can't~ be~ nested! }
  { Nesting~ doesn't~ make~ much~ sense~ as~ the~ inner~ one~
    wouldn't~ be~ visible.~ ~ To~ allow~ further~ processing~ it~ is~
    handled~ as~ if~ it~ hasn't~ been~ asked~ for. }
\msg_new:nnnn {dashundergaps} {gap-format-value}
  { Unknown~ value~ for~ key~ '#1 gap-format'! }
  { Supported~ values~ are~ 'underline',~ 'double-underline',\\
    'dash',~ 'dot',~ 'wave'~ or~ 'blank'. }
```

5.2.4 Option handling

Here we define all the possible option keys for use either as package options or inside \dashundergapssetup. These are all straightforward assignments to variables. These internal variables are declared by the key declarations if unknown, so they are not separately declared beforehand.

```
\keys_define:nn {fmdug}
    {
      % ====================================
     ,teacher-mode      .bool_set:N = \l__fmdug_teacher_bool
     ,teacher-mode      .default:n  = true
     ,teacher-mode      .initial:n  = false
      % ------------------
     ,gap-mode  .bool_set_inverse:N = \l__fmdug_teacher_bool
      % ====================================
     ,gap-format
        .choice:
```

In the case of dashes and even more so in the case of dots, it looks fairly ugly if they are below the baseline as if there were text above. We therefore raise them up a bit if the option `gap-format-adjust` is given (which is the default).

In the case of dots we undo exactly the amount by which they are lowered in ulem so that they end up precisely at the baseline, in case they are followed by a real dot. In other cases we stay a bit below the baseline.

The same is done below when the optional argument is evaluated. But we don't do this in teacher mode since there we *will* have text above and we don't want to bump into that.

```
      ,gap-format / underline
         .code:n = \tl_set:Nn \l__fmdug_gap_format_tl
                              { \__fmdug_gap_format_adjust:n{.4pt} \uline }
      ,gap-format / double-underline
         .code:n = \tl_set:Nn \l__fmdug_gap_format_tl
                              { \__fmdug_gap_format_adjust:n{2pt} \uuline }
      ,gap-format / dash
         .code:n = \tl_set:Nn \l__fmdug_gap_format_tl
                           { \__fmdug_gap_format_adjust:n{0pt} \dashuline }
      ,gap-format / dot
         .code:n = \tl_set:Nn \l__fmdug_gap_format_tl
                         { \__fmdug_gap_format_adjust:n{-.08ex} \dotuline }
      ,gap-format / wave
         .code:n = \tl_set:Nn \l__fmdug_gap_format_tl
                              { \__fmdug_gap_format_adjust:n{1pt} \uwave }
      ,gap-format / blank
         .code:n = \tl_set:Nn \l__fmdug_gap_format_tl { \fmdug@ublank  }
      ,gap-format / unknown
         .code:n = \msg_error:nnn{dashundergaps}{gap-format-value}{}
      ,gap-format
         .initial:n = underline
       % ====================================
```

This controls the raising of the gap underline by some amount. We implement it as a `.choice` even though it looks like a boolean.

```
       ,gap-format-adjust
         .choice:
       ,gap-format-adjust / true
         .code:n = \cs_set:Npn \__fmdug_gap_format_adjust:n ##1
                               { \setlength\ULdepth {##1} }
       ,gap-format-adjust / false
         .code:n = \cs_set_eq:NN \__fmdug_gap_format_adjust:n \use_none:n
      ,gap-format-adjust
         .default:n = true
      ,gap-format-adjust
         .initial:n = true
      ,adjust .meta:n = { gap-format-adjust }
       % ====================================
      ,teacher-gap-format
         .choice:
     ,teacher-gap-format / underline
         .code:n = \tl_set:Nn \l__fmdug_teacher_gap_format_tl { \uline }
      ,teacher-gap-format / double-underline
         .code:n = \tl_set:Nn \l__fmdug_teacher_gap_format_tl { \uuline }
      ,teacher-gap-format / dash
         .code:n = \tl_set:Nn \l__fmdug_teacher_gap_format_tl { \dashuline }
      ,teacher-gap-format / dot
         .code:n = \tl_set:Nn \l__fmdug_teacher_gap_format_tl { \dotuline }
```

```
    ,teacher-gap-format / wave
       .code:n = \tl_set:Nn \l__fmdug_teacher_gap_format_tl { \uwave }
    ,teacher-gap-format / blank
       .code:n = \tl_set:Nn \l__fmdug_teacher_gap_format_tl { \fmdug@ublank }
    ,teacher-gap-format / unknown
       .code:n = \msg_error:nnn{dashundergaps}{gap-format-value}{teacher-}
    ,teacher-gap-format
       .initial:n = blank
     % ====================================
    ,gap-widen         .bool_set:N = \l__fmdug_gap_widen_bool
    ,gap-widen         .default:n  = true
    ,gap-widen         .initial:n  = false
     % ------------------
    ,widen                  .meta:n = { gap-widen }
     % ------------------
    ,gap-extend-minimum .dim_set:N = \l__fmdug_gap_min_dim
    ,gap-extend-minimum .initial:n = 20pt
     % ------------------
    ,gap-extend-percent .tl_set:N  = \l__fmdug_gap_percent_tl
    ,gap-extend-percent .initial:n = 20
     % ====================================
    ,gap-numbers        .bool_set:N = \l__fmdug_number_bool
    ,gap-numbers        .default:n  = true
    ,gap-numbers        .initial:n  = true
     % ------------------
    ,numbers                .meta:n = { gap-numbers }
     % ------------------
    ,gap-number-format .tl_set:N    = \l__fmdug_gapnum_format_tl
    ,gap-number-format .initial:n   = \textnormal{\space (\thegapnumber)}
     % ====================================
    ,display-total-gaps .bool_gset:N = \g__fmdug_display_total_gaps_bool
    ,display-total-gaps   .default:n = true
    ,display-total-gaps   .initial:n = false
     % ====================================
    ,gap-font    .tl_set:N = \l__fmdug_font_tl
    ,gap-font   .default:n =
    ,gap-font   .initial:n = \bfseries
```

And finally the original options, now as aliases:

```
     % ====================================
    ,teachermode   .meta:n =  { teacher-mode }
    ,dash          .meta:n =  { gap-format = dash }
    ,dot           .meta:n =  { gap-format = dot }
    ,displaynbgaps .meta:n =  { display-total-gaps }
     % ------------------
    ,phantomtext
       .code:n = \msg_warning:nnnn{dashundergaps}{deprecated}
                                  {option}{phantomtext}
     % ====================================
  }
```

`\__fmdug_gap:nnn` At last, here comes the action. `\__fmdug_gap:nn` expects two arguments: `#1` indicates what kind of "underlining" is wanted (anything not recognized is ignored, in particular "–NoValue–" if `\gap` was used without an optional argument) and `#2` is the material to produce a gap for.

```
\cs_new:Npn\__fmdug_gap:nnn #1#2#3 {
  \group_begin:
```

Define the font used inside the gap. We need to do this up front since we want to measure the text (and that needs the correct font already).

```
\l__fmdug_font_tl
```

Nesting is not supported so inside the gap we redefine `\__fmdug_gap:nnn` to raise an error and just return the third argument if it is encountered again.

```
\cs_set:Npn \__fmdug_gap:nnn ##1##2##3
  {
    \msg_error:nn{dashundergaps}{nested}
    ##3
  }
```

We always increment the counter for the total number of gaps, but increment the `gapnumber` only if we are displaying it. For the latter one we use `\refstepcounter` to make it referenceable.

```
\stepcounter{totalgapnumber}
\bool_xor:nnT { #1 } { \l__fmdug_number_bool }
  { \refstepcounter{gapnumber} }
```

Next we prepare for widening if that is being asked for: Measure the width of the text and then set `\l__fmdug_extend_dim` to be the requested percentage divided by two of that width (since we add it later on both sides).

```
\bool_if:NTF \l__fmdug_gap_widen_bool
  {
    \settowidth \l__fmdug_extend_dim {#3}
    \dim_set:Nn \l__fmdug_extend_dim
              { \l__fmdug_gap_percent_tl \l__fmdug_extend_dim / 200 }
```

Then compare it to the minimum / 2 and choose whatever is larger.

```
     \dim_compare:nNnT \l__fmdug_extend_dim < { .5\l__fmdug_gap_min_dim }
        { \dim_set:Nn \l__fmdug_extend_dim { .5\l__fmdug_gap_min_dim } }
```

Now we prepare what needs to go to the left and the right of the gap.

```
     \tl_set:Nn \l__fmdug_extra_left_gap_tl
        { \hbox_to_wd:nn\l__fmdug_extend_dim{} \allowbreak }
     \tl_set:Nn \l__fmdug_extra_right_gap_tl
        { \allowbreak \hbox_to_wd:nn\l__fmdug_extend_dim{} }
  }
```

And if no widening is asked for we clear these two token lists so they don't do anything.

```
  {
    \tl_clear:N \l__fmdug_extra_left_gap_tl
    \tl_clear:N \l__fmdug_extra_right_gap_tl
  }
```

Next comes deciding the gap format. If in teacher mode it will be whatever is in `\l__fmdug_teacher_gap_tl`. Otherwise, either it is based on the content of the optional argument or, if that is not given or unknown, it will be `\l__fmdug_gap_format_tl`.

```
\bool_if:NTF \l__fmdug_teacher_bool
  { \l__fmdug_teacher_gap_format_tl }
  {
```

But before we execute any of the ulem commands we make sure that they do not output text.

```
    \cs_set:Npn   \fmdug@box ##1 {\hbox_to_wd:nn{\box_wd:N ##1}{}}
    \str_case:nnF {#2}
        {
          {u} { \__fmdug_gap_format_adjust:n{.4pt}   \uline  }
```

```
              {d} { \__fmdug_gap_format_adjust:n{2pt}    \uuline }
              {w} { \__fmdug_gap_format_adjust:n{1pt}    \uwave  }
              {b} { \fmdug@ublank }
              {.} { \__fmdug_gap_format_adjust:n{-.08ex} \dotuline }
              {-} { \__fmdug_gap_format_adjust:n{0pt}   \dashuline }
            }
            { \l__fmdug_gap_format_tl }
      }
```

Whatever was decided as the gap format, it needs one argument, i.e., the material (with possible gap extension on both sides).

```
    {\l__fmdug_extra_left_gap_tl #3 \l__fmdug_extra_right_gap_tl }
```

Finally we typeset the gap number if that was requested.

```
    \bool_xor:nnT { #1 } { \l__fmdug_number_bool }
                  { \l__fmdug_gapnum_format_tl }
```

Close the group from above to keep any of the redefinitions confined.

```
  \group_end:
}
```

(*End definition for* `\__fmdug_gap:nnn`.)

`\__fmdug_display_total_gaps:` This command will display the total number of gaps if requested. The hard-wired formatting comes from the first version of the package.

```
\cs_new:Npn \__fmdug_display_total_gaps: {
  \vfill \centering
  \bfseries Total~ Gaps:~ \thetotalgapnumber
}
```

(*End definition for* `\__fmdug_display_total_gaps:`.)

5.2.5 Closing shop

At the end of the document we typeset the total number of gaps if requested.

```
\AtEndDocument{
  \bool_if:NT \g__fmdug_display_total_gaps_bool
              \__fmdug_display_total_gaps:
}
```

So what remains to be done is executing all options passed to the package via `\usepackage`.

```
\ProcessKeysPackageOptions{fmdug}
⟨*package⟩
```

References

[1] Frank Mittelbach. The widows-and-orphans package. *TUG*boat 39:3, 252–262, 2018. https://ctan.org/pkg/widows-and-orphans

[2] LaTeX3 Project Team. A collection of articles on expl3. https://latex-project.org/publications/indexbytopic/l3-expl3/

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