# The l3galley package
# Galley code

The LaTeX Project*

Released 2023-11-01

## 1 Introduction

In LaTeX3 terminology a galley is a rectangular area which receives text and other material filling it from top. The vertically extend of a galley is normally not restricted: instead certain chunks are taken off the top of an already partially filled galley to form columns or similar areas on a page. This process is typically asynchronous but there are ways to control or change its behaviour.

Examples for galleys are "the main galley", where the continuous document data gets formatted into and from which columns and pages are constructed, and "vertical box galleys", such as the body of a minipage environment. The latter galleys are typically not split after formatting, though there can be exceptions.

## 2 Formatting layers

The present module is mainly concerned with the formatting of text in galleys. The mechanism by which this is achieved uses four (somewhat) distinct layers, some of which can be addressed using the templates provided here.

### 2.1 Layer one: external dimensions

The bottom layer of the system is the external dimensions of the galley. Normally only the horizontal dimension is fixed externally, while the vertical (filling) dimension is unspecified. The external dimensions are fixed when starting a new galley, and are therefore not modifiable within the galley.

There are no templates for setting this layer directly, although the external values are influenced by other parts of the system (for example when creating minipage environments).

*E-mail: latex-team@latex-project.org

### 2.2 Layer two: internal dimensions

The second layer is the internal dimensions of the galley: the *measure* used for paragraph text and the position of the paragraph relative to the edges of the galley.

This layer is normally accessed by higher-level templates *via* the object type `measure`. Changes made using level two templates will often extend for large parts of a document (up to and including the entire document).

### 2.3 Layer three: paragraph shape

The third layer defines the paragraph shape within the measure as provided by the second layer. In the absence of any specification for that layer the paragraph shape used will be that of a rectangular area of the width of the current measure.

There are some restrictions imposed on the shape of a paragraph by the underlying TEX mechanisms. For example, cut out sections in paragraphs can be specified from the top of the paragraph but not from the bottom.

### 2.4 Layer four: formatting inside the paragraph

The forth layer deals with the paragraph formatting aspects such as hyphenation and justification within the paragraph (this is sometimes referred to as "`h&j`" or "`hj`"). This layer is somewhat distinct from the galley as such, but is handled in the same place as there is, internally, interaction between the different layers.

## 3 Code interfaces

### 3.1 Galley layers

`\l_galley_width_dim` The total width of a galley, set either by the page geometry code for the main vertical galley or when creating an independent galley, such as a minipage.

`\galley_level:` `\galley_level:`

Sets up a vertical box to contain a new galley level. The box should include "wrapper" group (be "color safe"): this is automatically true for all LATEX3 boxes and coffins.

### 3.2 Measure

`\galley_margins_set_absolute:nn` `\galley_margins_set_absolute:nn {⟨left margin⟩} {⟨right margin⟩}`
`\galley_margins_set_relative:nn` `\galley_margins_set_relative:nn {⟨left margin⟩} {⟨right margin⟩}`

Sets the width of the measure to have the ⟨*left margin*⟩ and ⟨*right margin*⟩ specified by the arguments, both of which are ⟨*dimension expressions*⟩. The `relative` function will adjust the text width within any existing margins, whereas the `absolute` measure sets the margins based on the edges of the galley only. One or both of the ⟨*margins*⟩ may be negative, to specify and outdent.

### 3.3 Between paragraphs

`\g_galley_previous_par_lines_int`

The number of lines in the preceding conceptual paragraph. This may not correspond to the TEX `\prevgraf` primitive value as one conceptual paragraph may contain several TEX `\par` primitive tokens.

`\g_galley_restore_running_tl`

When galley settings need to be reset at the end of a paragraph, the appropriate detail should be added to this token list. It is inserted immediately before the start of each paragraph, and can therefore be used to clear otherwise global settings. The token list itself is also cleared as part of this process.

`\g_galley_no_break_next_bool`

Indicates that no page break should be allowed between the current paragraph and the next paragraph.

`\g_galley_omit_next_indent_bool`

Indicates that the indent should be omitted from the start of the next paragraph started.

`\l_galley_interpar_penalty_int`

The ⟨*penalty*⟩ for a break between paragraphs. The ⟨*penalty*⟩ should be in the range −10 000 to 10 000, where −10 000 forces a page break, 0 has no effect at all and 10 000 forbids a page break. Note that setting `\g_galley_no_break_next_bool` to `true` will override any setting of `\l_galley_interpar_penalty_int`.

`\l_galley_interpar_vspace_skip`

Stretchable space to be inserted between paragraphs, set at the design or template level.

`\galley_penalty_set_single:n` `\galley_penalty_set_single:n {⟨penalty⟩}`
`\galley_penalty_add_single:n {⟨penalty⟩}`

Sets the ⟨*penalty*⟩ for a break between the current and next paragraph on a one-off basis. This function is intended for user-level adjustments to design, and takes precedent over both settings from `\l_galley_interpar_penalty_int` and from `\galley_no_break_next:`. The `add` variant adds the penalty to any existing values.

`\galley_vspace_set_single:n` `\galley_vspace_set_single:n {⟨space⟩}`
`\galley_vspace_add_single:n`
`\galley_vspace_max_single:n`

Sets the ⟨*space*⟩ to be inserted between the current and next paragraph on a one-off basis. This function is intended for user-level adjustments to design. The `add` and `max` variants add to the existing spacing and set to the maximum of the existing value and the argument, respectively.

## 3.4 Paragraph shape

\galley_parshape_set_multi:nnnN
\galley_parshape_set_multi:nVVN
\galley_parshape_set_single:nnnN
\galley_parshape_set_single:nVVN

\galley_parshape_set_multi:nnnN {⟨*unaltered lines*⟩} {⟨*left indents*⟩} {⟨*right indents*⟩} ⟨*resume flag*⟩
\galley_parshape_set_single:nnnN {⟨*unaltered lines*⟩} {⟨*left indents*⟩} {⟨*right indents*⟩} ⟨*resume flag*⟩

Sets the current paragraph shape to create an arbitrary paragraph shape. The paragraph shape is set such that there are ⟨*unaltered lines*⟩ which have width and indent as set by the measure. The "altered" lines are then defined by the comma-separated lists of ⟨*left indents*⟩ and ⟨*right indents*⟩. These are both indents from the edge of the measure, and may be negative, and should both contain the same number of items. If the ⟨*resume flag*⟩ is `true`, after the last altered line the paragraph shape returns to that of the measure. On the other hand, if the flag is `false` then the shape of the last line is retained for the rest of the paragraph. For example,

```
\galley_parshape_set_multi:nnnN { 1 }
  { 2 pt , 4 pt , 6 pt } { 2 pt , 4 pt , 6 pt } \c_true_bool
```

would create a paragraph shape in which the first line is the full width of the measure, the second line is indented by 2 pt on each side, the third line by 4 pt and the fourth line and subsequent lines by 6 pt from the edge of the measure on each side.

The `single` version applies only to a single paragraph, while the `multi` function sets the paragraph shape on an ongoing basis within the scope of the current TEX group.

\galley_cutout_left:nn
\galley_cutout_right:nn

\galley_cutout_left:nn {⟨*unaltered lines*⟩} {⟨*indents*⟩}
\galley_cutout_right:nn {⟨*unaltered lines*⟩} {⟨*indents*⟩}

Adds a cutout section to the active paragraph shape, leaving ⟨*unaltered lines*⟩ unchanged and then applying the ⟨*indents*⟩ (a comma list). The cutout will be placed on the left or right as indicated by the function name, and will apply to exactly the number of lines specified (the total of the ⟨*unaltered lines*⟩ and the number of entries in the ⟨*indents*⟩ list). Several cutouts may be applied sequentially: these act in an additive sense.

## 3.5 Formatting inside the paragraph

The settings described here apply "inside" the paragraph, and so are active irrespective of any paragraph shape within the measure.

\l_galley_line_left_skip
\l_galley_line_right_skip

Stretchable space added to the appropriate side each line in a paragraph.

\l_galley_par_begin_skip
\l_galley_par_end_skip

Stretchable space added to the beginning of the first line and end of the last line of a paragraph, respectively.

\l_galley_par_indent_dim

Fixed space added to the start of each paragraph except for those where `\g_galley_omit_next_indent_bool` is `true`.

`\l_galley_last_line_fit_int`

Determines how the inter-word stretch is set for the last line of a paragraph when

1. the value of `\l_galley_par_end_skip` contains an infinite (`fil(l)`) component;
2. the values of `\l_galley_line_left_skip` and `\l_galley_line_right_skip` do *not* contain an infinite (`fil(l)`) component.

Under these circumstances, `\l_galley_last_line_fit_int` is active, and applies as follows:

- when set to 0, the last line of the paragraph is set with the inter-word spacing at natural width;
- when set to a 1000 (or above), the inter-word spacing in the last line is stretched by the same factor as that applied to the penultimate line;
- when set to $n$ between these extremes, the inter-word spacing in the last line is stretched by $n/1000$ times the factor used for the penultimate line.

`\galley_interword_spacing_set:N` `\galley_interword_spacing_set:N` ⟨*fixed spacing bool*⟩

Sets the inter-word spacing used based on the values supplied by the current font. If the ⟨*fixed spacing bool*⟩ flag is `true` then no stretch is permitted between words, otherwise the stretch specified by the font designer is used.

## 3.6 Display material

Material which is set in "display-style" require additional settings to control the relationship with the surrounding material.

`\galley_display_begin:`
`\galley_display_end:`

```
\galley_display_begin:
...
\galley_display_end:
```

Sets up a group to contain display-style material. Unlike an independent galley level, settings are inherited from the surroundings. However, the interaction of a display block with the paragraphs before and after it can be adjusted independent of the design of text.

## 3.7 Hyphenation

`\l_galley_hyphen_left_int` THIS IS A HACK: SEE CODE!

## 3.8 Line breaking

`\l_galley_binop_penalty_int`

Penalty charged if an inline math formula is broken at a binary operator.

`\l_galley_double_hyphen_demerits_int`

Extra demerit charge of two (or more) lines in succession end in a hyphen.

`\l_galley_emergency_stretch_skip`

Additional stretch assumed for each line if no better line breaking can be found without it. This stretch is not actually added to lines, so its use may result in underfull box warnings.

`\l_galley_final_hyphen_demerits_int`

Extra demerit charge if the second last line is hyphenated.

`\l_galley_linebreak_badness_int`

Boundary that if exceeded will cause TEX to report an underfull line.

`\l_galley_linebreak_fuzz_dim`

Boundary below which overfull lines are not reported.

`\l_galley_linebreak_penalty_int`

Extra penalty charged per line in the paragraph. By making this penalty higher TEX will try harder to produce compact paragraphs.

`\l_galley_linebreak_pretolerance_int`

Maximum tolerance allowed for individual lines to break the paragraph without attempting hyphenation.

`\l_galley_linebreak_tolerance_int`

Maximum tolerance allowed for individual lines when breaking a paragraph while attempting hyphenation (if this limit can't be met `\l_galley_emergency_stretch_skip` comes into play).

`\l_galley_mismatch_demerits_int`

Extra demerit charge if two visually incompatible lines follow each other.

`\l_galley_relation_penalty_int`

Penalty charged if an inline math formula is broken at a relational symbol.

`\galley_break_line:Nn` `\galley_break_line:Nn` ⟨*boolean*⟩ {⟨*dimexpr*⟩}

Breaks the current line, filling the remaining space with `fil` glue. If the ⟨*boolean*⟩ is `true` then a page break is possible after the broken line. Vertical space as given by the ⟨*dimexpr*⟩ will be inserted between the broken line and the next line.

## 3.9 Paragraph breaking

`\l_galley_parbreak_badness_int`

Boundary that if exceeded will cause TEX to report an underfull vertical box.

`\l_galley_parbreak_fuzz_dim`

Boundary below which overfull vertical boxes are not reported.

`\l_galley_broken_penalty_int`

Penalty for page breaking after a hyphenated line.

`\l_galley_pre_display_penalty_int`

Penalty for breaking between immediately before display math material.

`\l_galley_post_display_penalty_int`

Penalty for breaking between immediately after display math material.

`\galley_club_penalties_set:n` `\galley_club_penalties_set:n {⟨penalty list⟩}`
`\galley_club_penalties_set:(V|v)`
`\galley_display_club_penalties_set:n`
`\galley_display_club_penalties_set:(V|v)`
`\galley_display_widow_penalties_set:n`
`\galley_display_widow_penalties_set:(V|v)`
`\galley_widow_penalties_set:n`
`\galley_widow_penalties_set:(V|v)`

Set the penalties for breaking lines at the beginning and end of (partial) paragraphs. In each case, the ⟨*penalty list*⟩ is a comma-separated list of penalty values. The list applies as follows:

`club` Penalties for breaking after the first, second, third, *etc.* line of the paragraph.

`display_club` Penalties for breaking after the first, second, third, *etc.* line after a display math environment.

`display_club` Penalties for breaking before the last, penultimate, antepenultimate, *etc.* line before a display math environment.

`widow` Penalties for breaking before the last, penultimate, antepenultimate, *etc.* line of the paragraph.

In all cases, these penalties apply in addition to the general interline penalty or to any "special" line penalties.

`\galley_interline_penalty_set:n` `\galley_interline_penalty_set:n {⟨penalty⟩}`

Sets the standard interline penalty applied between lines of a paragraph. This value is added to any (display) club or widow penalty in force.

\galley_interline_penalties_set:n \galley_interline_penalties_set:n {⟨*penalty list*⟩}
\galley_interline_penalties_set:V

Sets "special" interline penalties to be used in place of the standard value, specified as a comma-separated ⟨*penalty list*⟩. The ⟨*penalties*⟩ apply to the first, second, third, *etc.* line of the paragraph.

\galley_save_club_penalties:N \galley_save_club_penalties:N {⟨*comma list*⟩}
\galley_save_display_club_penalties:N
\galley_save_display_widow_penalties:N
\galley_save_interline_penalties:N
\galley_save_widow_penalties:N

These functions save the current value of the appropriate penalty to the comma list specified, within the current TEX group.

\galley_interline_penalty: ⋆ \galley_interline_penalty:

Expands to the current interline penalty as a ⟨*integer denotation*⟩.

# 4 Hooks and insertion points

\g_galley_par_begin_hook_tl

Token list inserted at the beginning of every paragraph in horizontal mode. This is inserted after any paragraph indent but before any other horizontal mode material.

\g_galley_par_end_hook_tl Token list inserted at the end of every paragraph in horizontal mode.

\g_galley_par_reset_hook_tl

Token list inserted after each paragraph. This is used for resetting galley parameters, and is therefore cleared after use. It is inserted in vertical mode and must not contain horizontal mode material.

\g_galley_whatsit_next_tl Token list for whatsits to be inserted at the very beginning of the next paragraph started.

\g_galley_whatsit_previous_tl

Token list for whatsits to be inserted at the very end of the last paragraph started.

# 5 Paragraphs

\galley_par: \galley_par:

Finalises the material collected as the last paragraph, inserts tokens at the start of the new paragraph, setting paragraph shape and any special values as required.

\galley_par:n `\galley_par:n {⟨tokens⟩}`

Adds the ⟨*tokens*⟩ to the material collected for the last paragraph before finalising it in the usual way. This function should therefore be the *first* non-expandable entry used when a function needs to add tokens to the preceding paragraph.

## 6 Information variables

Some of the variables for the galley mechanism may be of interest to the programmer. These should all be treated as read-only values and accessed only through the defined interfaces described above.

`\l_galley_total_left_margin_dim`

The total margin between the left side of the galley and the left side of the text block. This may be negative if the measure is set to overlap the text beyond the edge of the galley.

`\l_galley_total_right_margin_dim`

The total margin between the right side of the galley and the right side of the text block. This may be negative if the measure is set to overlap the text beyond the edge of the galley.

`\l_galley_text_width_dim` The width of a line of text within the galley, taking account of any margins added. This may be larger than `\l_galley_width_dim` if the margins are negative.

# 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.

**T**