

# zebra — Writing Revision Toolkit\*

Ruini Xue<sup>†</sup>

v1.8.0 (2026/04/26)

## Abstract

The **zebra** package is a writing revision toolkit. The current release focuses on inline note-taking, with a lightweight set of macros designed to be simple and practical for both solo and collaborative workflows. Five built-in commands—`\todo`, `\note`, `\comment`, `\fixed`, and `\placeholder`—cover common use cases out of the box, and `\zebranewnote` lets you define additional note types as needed. Notes are automatically numbered per type, marked with a customisable symbol (default: `\textdbend`) in the nearest margin, and summarised with a summary table plus a detailed note list at the end of the document. Passing the `final` option suppresses all notes for production output.



## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>Using the package</b>	<b>2</b>
3.1	Package Options	2
3.2	Notes Macros	3
3.3	Two-column Support	5
<b>4</b>	<b>Implementation</b>	<b>6</b>
4.1	Package options	7
4.2	Moving-argument deduplication	8
4.3	Main notes macros	12
4.4	Print summary at end of the document	19
4.5	Compatibility shim	19
4.6	Two-column demo	20
	<b>Change History</b>	<b>20</b>
	<b>Index</b>	<b>21</b>

---

\*This package was previously distributed as **zebra-goodies**. The old name still works but will print a deprecation warning. Please update to **zebra**.

<sup>†</sup>Email: xueruini@gmail.com

# 1 Introduction

**zebra** is a writing revision toolkit. The current release focuses on inline note-taking. Many note-taking and to-do packages exist for L<sup>A</sup>T<sub>E</sub>X, but most fall into one of two traps: they either offer an overwhelming feature set that tries to cover every conceivable use case, or they clutter the margins with oversized colourful boxes and arrows that make the document hard to read.

**zebra** takes a different approach. It aims to be *simple*—intuitive commands with only the arguments you actually need—and *good enough*—notes appear inline with a small visual cue in the margin, keeping the document readable while still making annotations easy to spot. Each note type is automatically numbered, and a summary table plus a detailed note list at the end of the document serve as a gentle reminder to address them before the final version.

# 2 Installation

**zebra** is available on CTAN; install it through your L<sup>A</sup>T<sub>E</sub>X distribution’s package manager. To build from source, run `latexmk zebra.dtx` to extract the package and typeset the documentation in one step.

# 3 Using the package

Load the package in the preamble with any desired options.

```
\usepackage[<options>]{zebra} % was zebra-goodies
```

## 3.1 Package Options

- |                       |   |
|-----------------------|---|
| <b>draft</b>          | These two options are complementary. Default: <b>true</b> (draft mode). All notes are typeset   |
| <b>final</b>          | inline and a summary table plus a detailed note list are appended at the end of the document. Setting <b>final</b> (or <b>draft=false</b> ) suppresses all notes and the generated lists, producing clean output ready for distribution.  |
| <b>sort</b>           | Controls the order of the detailed note list printed at the end of the document. Default: <b>none</b> (document order). <b>sort=type</b> groups them by note type.  |
| <b>pagelinks</b>      | Controls whether page numbers in the detailed note list are clickable. Default: <b>true</b> . Set <b>pagelinks=false</b> to disable these links. The complementary option <b>nopagelinks</b> is also accepted.  |
| <b>font-expansion</b> | Controls <b>microtype</b> font expansion. This usually improves the appearance of the document. Disable it if it conflicts with your engine or another package by setting <b>font-expansion=false</b> . The complementary option <b>nofont-expansion</b> is also accepted. Default: <b>true</b> . <b>microtype</b> remains loaded when expansion is disabled.   |
| <b>unnumbered</b>     | Turns off note numbering. Inline markers become <code>[todo: ...]</code> (no number), the margin symbol carries no number, the end-of-document detailed note list is omitted (the per-type summary table remains), and <code>\zebraref</code> falls back to <code>\ref</code> . Useful as a workaround if numbered notes interact badly with a particular class; the summary table count is then an upper-bound estimate. Default: <b>false</b> . |

## 3.2 Notes Macros

All note commands share the syntax `\cmd[⟨name⟩]{⟨text⟩}`. Each also has a prefixed alias (e.g. `\zebratodo`) that is always available, regardless of name conflicts. If a short name clashes with another loaded package, **zebra** will *not* overwrite the existing definition; use the prefixed form instead.

---

<code>\todo</code>	<code>\todo[⟨name⟩]{⟨text⟩}</code>
<code>\zebratodo</code>	<code>\zebratodo[⟨name⟩]{⟨text⟩}</code>



---

The primary command provided by **zebra** is `\todo`. It inserts an inline note in the current paragraph, typeset in a predefined colour and marked with a symbol in the nearest margin. The mandatory `⟨text⟩` describes the task; the optional `⟨name⟩` specifies who is responsible for addressing it, which is particularly useful during collaborative writing.

The motivation section still feels too vague `\todo{revise the introduction before submission}` and could benefit from a concrete running example to guide the reader through the key ideas step by step.  
The motivation section still feels too vague and could benefit from a concrete running example to guide the reader through the key ideas step by step.

The optional argument assigns one or more people to the note. Assignees appear prefixed with `@`, and notes of the same type are numbered sequentially.

The related work section needs more references `\todo[alice]{add two or three citations from the latest survey}` to recent advances in the field. We should also double-check the experimental setup before the camera-ready deadline `\todo[bob, carol]{verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards}`.

1  The related work section needs more references [ `TODO 1@alice: add two or three citations from the latest survey`] to recent advances in the field. We should also double-check the experimental  
2  setup before the camera-ready deadline [ `TODO 2@bob, carol: verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards`].

Notes can appear inside moving arguments such as `\section` and `\caption`. To ensure stable numbering and cross-references, add a `\label` inside the note:

```
\section{Introduction\todo[jerry]{\label{zebra:heading}fix the name}}
\begin{figure}
  \caption{Speed vs distance. \todo{need to insert the figure}}
\end{figure}
```

With a `\label`, the note is counted once regardless of how many times the heading appears (table of contents, running headers, etc.). Notes without a `\label` in moving arguments are still safe but may receive a separate number in each context.

---

<code>\note</code>	<code>\note[⟨name⟩]{⟨text⟩}</code>
<code>\zebranote</code>	<code>\zebranote[⟨name⟩]{⟨text⟩}</code>

---



---

<code>\comment</code>	<code>\comment[⟨name⟩]{⟨text⟩}</code>
<code>\zebracomment</code>	<code>\zebracomment[⟨name⟩]{⟨text⟩}</code>

---

---

<code>\fixed</code>	<code>\fixed[⟨name⟩]{⟨text⟩}</code>
<code>\zebrafixed</code>	<code>\zebrafixed[⟨name⟩]{⟨text⟩}</code>

---



---

<code>\placeholder</code>	<code>\placeholder[⟨name⟩]{⟨text⟩}</code>
<code>\zebraplaceholder</code>	<code>\zebraplaceholder[⟨name⟩]{⟨text⟩}</code>


---


These commands share the same syntax and behaviour as `\todo`; they differ only in name and colour, providing semantic distinction for different annotation purposes. Note that `\zebracomment` is used in the example below because `\comment` is already defined by `l3doc`.


We may want to reorganise `\note{how should we structure the intro?}` this part before the final submission. The experimental setup in Section~2 has already been reviewed by a collaborator `\zebracomment[tom]{the setup description looks clear now}`. Results are presented in the following tables and figures, but some of them are still missing.


The discussion has been revised `\placeholder[lucy, tom]{good job!}` and the related work comparison strengthened with two additional references. The list of references still needs a second pass `\todo{check bibliography entries for formatting}` before we can finalize the submission.


With those items addressed, the conclusion has been rewritten so the argument flows more naturally from the results. `\fixed[John]{updated the conclusion}` The overall structure now matches the revised outline we agreed on last week. `\note[who]{anything else?}` If not, the draft should be fine.


1  We may want to reorganise [ NOTE 1: how should we structure the intro?] this part before the final submission. The experimental setup in Section 2 has already been reviewed by a collaborator


1  [ COMMENT 1@tom: the setup description looks clear now]. Results are presented in the following

1  tables and figures, but some of them are still missing.

1  The discussion has been revised [ PLACEHOLDER 1@lucy, tom: good job!] and the related work comparison strengthened with two additional references. The list of references still needs a second

3  pass [ TODO 3: check bibliography entries for formatting] before we can finalize the submission.

1  With those items addressed, the conclusion has been rewritten so the argument flows more naturally from the results. [ FIXED 1@John: updated the conclusion] The overall structure now

2  matches the revised outline we agreed on last week. [ NOTE 2@who: anything else?] If not, the draft should be fine.

---

<code>\zebranewnote</code>	<code>\zebranewnote{⟨note name⟩}{⟨xcolor name⟩}[⟨symbol⟩]</code>
----------------------------	--

---

Creates a new note type. The `⟨note name⟩` becomes the command name (e.g. passing `question` creates `\question` and `\zebraquestion`), and `⟨xcolor name⟩` sets its colour. The colour must be a named colour already known to `xcolor`; define it with `\definecolor` or `\colorlet` beforehand if needed. The optional `⟨symbol⟩` overrides the default margin symbol (`\textdbend`) for this note type only. Per-type symbols can also be changed after loading via `\zebrasetup{symbol/⟨type⟩=⟨symbol⟩}`.

```
\colorlet{mycyan}{cyan!80!black}
\zebranewnote{question}{mycyan}[\faQuestionCircle] % \usepackage{fontawesome}
```

When it moves to the next step, we should be fine. `\question[who]{what's this?}`

1  When it moves to the next step, we should be fine. [ QUESTION 1@who: what's this?]

---

`\zebraref` `\zebraref{<label>}`

---

Labels may be placed inside note bodies with the usual `\label` command. Standard `\ref` returns the note number, while `\zebraref` prints the note type together with the number.

```
The motivation section still feels too vague \todo{\label{zebra:intro}revise
the introduction before submission}. The same issue appears again later
\note{see Todo~\ref{zebra:intro} (that is, \zebraref{zebra:intro}) on
p.~\pageref{zebra:intro}}.
```



The motivation section still feels too vague [ TODO 4: revise the introduction before submission].  
The same issue appears again later [ NOTE 3: see Todo 4 (that is, Todo 4) on p. 5].

As in standard L<sup>A</sup>T<sub>E</sub>X, labels inside notes are unavailable in `final` mode because the notes themselves are suppressed.

---

`\zebrasetup` `\zebrasetup{<key=value list>}`

---

Configures note appearance after loading. Accepted keys:

- `color/<type>=<colour>` — override the colour of a note type.
- `symbol/<type>=<symbol>` — override the margin symbol of a note type.

For example:

```
\zebrasetup{symbol/fixed=\manerrarrow} % like this doc
\zebrasetup{color/todo=red}
```

### 3.3 Two-column Support

In `twocolumn` documents, the margin symbol is automatically placed on the nearest margin: left margin for the left column, right margin for the right column. No special configuration is needed. This also works correctly in combination with the `twoside` option.

```
\usepackage[paperwidth=21cm,paperheight=15cm,margin=1.1cm]{geometry}
\usepackage{zebra}
\zebrasetup{symbol/comment=$\clubsuit$}
\pagestyle{empty}
\begin{document}
\section{Demo name\comment{revise the name}}
This draft still needs work
\todo[alice]{\label{zebra:intro}revise the introduction}. The
opening paragraph should also explain the main goal more plainly.
Add one more citation here \note[bob]{support this claim}. A
brief roadmap sentence would also make the structure easier to
scan.

The issue raised in Todo~\ref{zebra:intro} still applies in
the conclusion. The table now looks fine
\fixed[carol]{alignment corrected}, but one figure is still
missing \placeholder[eve]{insert the overview figure}. A short
transition would also help the flow. The middle section should
```

probably end with a clearer summary sentence before the discussion begins. That summary can stay compact, but it should signal why the next section matters.

Please verify the totals `\note[frank]{check the numbers}` and confirm the wording in the last paragraph `\comment[tom]{is this sentence too strong?}`. A small typo has already been fixed `\fixed[heidi]{typo corrected}`. The ending should stay short. The final sentence should return to the main claim rather than repeat background material. You can place `\todo[judy]{summarise the findings}` anywhere once the narrative is stable.

One more short paragraph is enough to show how `\placeholder{wow, so great!}` the markers stay readable in a compact two-column layout. The example is intentionally small, but it should still look like a realistic revision pass. `\comment{Bye}`  
`\end{document}`

The code above produces the following output:

1 ♣

1 Demo

name[Comment 1: revise the name]

pass.[COMMENT 3: Bye]

♣3

1 ♣

1

This draft still needs work [TODO 1@alice: revise the introduction]. The opening paragraph should also explain the main goal more plainly. Add one more citation here [NOTE 1@bob: support this claim]. A brief roadmap sentence would also make the structure easier to scan.

The issue raised in Todo 1 still applies in the conclusion. The table now looks fine [FIXED 1@carol: alignment corrected], but one figure is still missing [PLACEHOLDER 1@eve: insert the overview figure]. A short transition would also help the flow. The middle section should probably end with a clearer summary sentence before the discussion begins. That summary can stay compact, but it should signal why the next section matters.

2 ♣

2

Please verify the totals [NOTE 2@frank: check the numbers] and confirm the wording in the last paragraph [COMMENT 2@tom: is this sentence too strong?]. A small typo has already been fixed [FIXED 2@heidi: typo corrected]. The ending should stay short. The final sentence should return to the main claim rather than repeat background material. You can place [TODO 2@judy: summarise the findings] anywhere once the narrative is stable.

One more short paragraph is enough to show how [PLACEHOLDER 2: wow, so great!] the markers stay readable in a compact two-column layout. The example is intentionally small, but it should still look like a realistic revision

Zebra Notes

Type	Count
todo	2
fixed	2
comment	3
note	2
placeholder	2
<b>Total</b>	<b>11</b>

List of notes

Comment 1.....1

revise the name

Todo 1 @alice.....1

revise the introduction

Note 1 @bob.....1

support this claim

Fixed 1 @carol.....1

alignment corrected

Placeholder 1 @eve.....1

insert the overview figure

Note 2 @frank.....1

check the numbers

Comment 2 @tom.....1

is this sentence too strong?

Fixed 2 @heidi.....1

typo corrected

Todo 2 @judy.....1

## 4 Implementation

- 1 `\package`
- 2 `\@@=zebra`

Version data to start with.

6

```

3 \ProvidesExplPackage{zebra}
4   {2026/04/26}
5   {1.8.0}
6   {Writing Revision Toolkit}

```

## 4.1 Package options

Package options `draft`, `pagelinks`, `font-expansion`, `sort`, and `unnumbered` are created using the kernel key–value interface available since L<sup>A</sup>T<sub>E</sub>X 2022-06-01. Post-load configuration (`\zebrasetup`) uses a separate `zebra-setup` key family with `color/⟨type⟩` and `symbol/⟨type⟩` sub-families.

```

7 \bool_new:N \l__zebra_draft_bool
8 \bool_new:N \l__zebra_microtype_expansion_bool
9 \bool_new:N \l__zebra_pagelinks_bool
10 \bool_new:N \l__zebra_sort_none_bool
11 \bool_new:N \l__zebra_unnumbered_bool
12 \seq_new:N \g__zebra_note_types_seq
13 \prop_new:N \g__zebra_note_colors_prop
14 \prop_new:N \g__zebra_note_public_alias_prop
15 \int_new:N \g__zebra_note_id_int
16 \tl_new:N \l__zebra_note_target_tl
17 \tl_new:N \l__zebra_note_color_tl
18 \tl_new:N \l__zebra_note_ref_type_tl
19 \tl_new:N \l__zebra_summary_rows_tl
20 \int_new:N \l__zebra_total_notes_int
21 \prop_new:N \g__zebra_note_symbols_prop
22 \tl_new:N \l__zebra_symbol_tl
23 \tl_set:Nn \l__zebra_symbol_tl { \textdbend }
24
25 \msg_new:nnn { zebra } { command-taken }
26   {
27     The~command~'\iow_char:N\|#1'~is~already~defined.~
28     Use~'\iow_char:N\zebra#1'~instead.
29   }
30 \msg_new:nnn { zebra } { invalid-note-label }
31   { Label~'#1'~is~not~a~zebra~note~label. }
32 \msg_new:nnn { zebra } { duplicate-note-label }
33   { Note~label~'#1'~used~by~a~different~note;~second~note~gets~
34     its~own~identity. }
35
36 \prg_new_conditional:Npnn \__zebra_if_package_loaded:n #1 { T , F , TF }
37   {
38     \cs_if_exist:cTF { ver@#1.sty }
39       { \prg_return_true: }
40       { \prg_return_false: }
41   }
42
43 \keys_define:nn { zebra }
44   {
45     draft .bool_set:N = \l__zebra_draft_bool,
46     draft .initial:n = true,
47     final .meta:n = { draft = false },
48     font-expansion .bool_set:N = \l__zebra_microtype_expansion_bool,
49     font-expansion .initial:n = true,

```

```

50   nofont-expansion .meta:n = { font-expansion = false },
51   pagelinks .bool_set:N = \l__zebra_pagelinks_bool,
52   pagelinks .initial:n = true,
53   nopagelinks .meta:n = { pagelinks = false },
54   sort .choice:,
55   sort / type .code:n = { \bool_set_false:N \l__zebra_sort_none_bool },
56   sort / none .code:n = { \bool_set_true:N \l__zebra_sort_none_bool },
57   sort .initial:n = none,
58   unnumbered .bool_set:N = \l__zebra_unnumbered_bool,
59   unnumbered .initial:n = false,
60 }
61 \ProcessKeyOptions [ zebra ]
62 \keys_define:nn { zebra-setup / color }
63 {
64   unknown .code:n =
65     { \prop_gput:NVn \g__zebra_note_colors_prop \l_keys_key_str {#1} }
66 }
67 \keys_define:nn { zebra-setup / symbol }
68 {
69   unknown .code:n =
70     { \prop_gput:NVn \g__zebra_note_symbols_prop \l_keys_key_str {#1} }
71 }
72 \sys_if_engine_xetex:T
73 { \bool_set_false:N \l__zebra_microtype_expansion_bool }
74
75 \cs_new_protected:Npn \__zebra_setup_microtype:
76 {
77   \__zebra_if_package_loaded:nTF { microtype }
78   {
79     \bool_if:NF \l__zebra_microtype_expansion_bool
80     { \microtypesetup { expansion = false } }
81   }
82   {
83     \bool_if:NTF \l__zebra_microtype_expansion_bool
84     { \RequirePackage{microtype} }
85     { \RequirePackage[expansion=false]{microtype} }
86   }
87 }
88 \__zebra_setup_microtype:
89 \__zebra_if_package_loaded:nF { hyperref }
90 {
91   \bool_if:NT \l__zebra_pagelinks_bool
92   {
93     \RequirePackage{hyperref}
94     \hypersetup { pdfborder = { 0~0~0 } }
95   }
96 }

```

## 4.2 Moving-argument deduplication

Notes inside moving arguments (`\section`, `\caption`, etc.) may be processed more than once per compilation pass. Two separate problems are handled independently:



**Problem A — \sbox re-measurement.** `\@makecaption` typesets the caption in an `\sbox` for width measurement, then typesets it again if it is long. Both executions share the same `\inputlineno`, so the *instance key* (`\type`|`\author`|`\body`|`\inputlineno`) catches the replay. The second execution reuses the first’s allocation and re-renders, so writes that were lost inside the discarded `\sbox` are re-emitted by the actual typesetting pass.

**Problem B — TOC/LOF/header replay.** The note token is written verbatim to `.toc/.lof/marks` and re-executed in a secondary context with a different `\inputlineno`. For *labeled* notes the replay is caught by two mechanisms:

1. A *stable key* (`\type`|`\label name`) stored alongside the allocation; a later encounter from marks/headers that still carries the `\label` finds this key and suppresses.
2. A *content signature* (`\type`|`\author`|`\sanitised body`) written to the `.aux` file; on the next pass, TOC/LOF encounters whose `\label` was consumed by `\protected@write`’s `\edef` match the signature and suppress.

Unlabeled notes in moving arguments receive independent allocations (cosmetic duplicate); adding `\label` is the recommended fix.

```

97 \RequirePackage{xcolor}
98 \RequirePackage{marginnote}
99 \cs_new_eq:NN \__zebra_kernel_label:n \label
100 %% -- dedup data structures --
101 %% Maps any key (instance, stable, or content-sig) to the allocation.
102 \prop_new:N \g__zebra_note_target_prop
103 \prop_new:N \g__zebra_note_display_prop
104 %% Content signatures of labeled notes from previous pass (.aux).
105 \prop_new:N \g__zebra_note_sig_known_prop
106 %% Content signatures written this pass (dedup aux writes).
107 \prop_new:N \g__zebra_note_sig_written_prop
108 %% Content signature stored per stable key (for label-conflict detection).
109 \prop_new:N \g__zebra_note_stable_sig_prop
110 \tl_new:N \l__zebra_note_display_tl
111 \tl_new:N \l__zebra_note_key_tl
112 %% Instance key: unique per source location.
113 \cs_new:Npn \__zebra_instance_key:nnn #1#2#3
114 {
115   \tl_to_str:n {#1}
116   | \tl_to_str:n {#2}
117   | \tl_to_str:n {#3}
118   | \int_eval:n { \tex_inputlineno:D }
119 }
120 %% Content signature: body stringified with ALL \label{...} stripped.
121 %% Matches across body (has labels), TOC (labels consumed by \edef),
122 %% and running heads (which may uppercase the author/body text).
123 %% The optional \protect prefix covers marks and \protected@write paths.
124 %% Uses replace_all so that multiple labels are all stripped.
125 \cs_new_protected:Npn \__zebra_content_sig:nnnN #1#2#3#4
126 {
127   \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#3} }
128   \regex_replace_all:nnN

```

```

129     { (?: \\protect \\s* )? \\label \\s* \\{ [^{}]* \\} } { } \\l_tmpa_tl
130     \\tl_set:Nx #4
131     {
132         \\tl_to_str:n {#1}
133         | \\str_lowercase:f { \\tl_to_str:n {#2} }
134         | \\str_lowercase:f { \\l_tmpa_tl }
135     }
136 }
137 %% Extract the first \\label name from the stringified body.
138 %% Sets #2 to the label name, or clears it if none found.
139 %% The optional \\protect prefix covers marks and \\protected@write paths.
140 \\cs_new_protected:Npn \\__zebra_extract_label:nN #1#2
141 {
142     \\tl_set:Nx \\l_tmpb_tl { \\tl_to_str:n {#1} }
143     \\tl_set_eq:NN \\l_tmpc_tl \\l_tmpb_tl
144     \\regex_replace_once:nnN
145     { \\A .* (?: \\protect \\s* )? \\label \\s* \\{ ([^{}]* ) \\} .* \\Z }
146     { \\1 } \\l_tmpb_tl
147     \\tl_if_eq:NNTF \\l_tmpb_tl \\l_tmpc_tl
148     { \\tl_clear:N #2 }
149     { \\tl_set_eq:NN #2 \\l_tmpb_tl }
150 }
151 %% Stable key for labeled notes.
152 %% #2 is expected to be already stringified (from regex extraction),
153 %% so no \\tl_to_str is applied - otherwise an unexpanded variable
154 %% token would be stringified instead of its value.
155 \\cs_new:Npn \\__zebra_stable_key:nn #1#2
156 { \\tl_to_str:n {#1} | label | #2 }
157 %% Allocate a fresh note: increment the type counter, generate
158 %% a unique hypertarget name, and record the note in the list body.
159 \\cs_new_protected:Npn \\__zebra_allocate_note:nnn #1#2#3
160 {
161     \\int_gincr:c { g__zebra_note_count_#1_int }
162     \\tl_set:Nx \\l__zebra_note_display_tl { \\__zebra_note_count:n {#1} }
163     \\int_gincr:N \\g__zebra_note_id_int
164     \\tl_set:Nx \\l__zebra_note_target_tl
165     { zebranote.\\int_use:N \\g__zebra_note_id_int }
166     \\__zebra_record_note:nnnnn
167     {#1}
168     { \\l__zebra_note_display_tl }
169     {#2}
170     {#3}
171     { \\l__zebra_note_target_tl }
172 }
173 %% Aux-file interface: record a content signature together with the
174 %% originating instance key. A later encounter whose instance key
175 %% differs from the stored one is a replay and is suppressed.
176 %% Re-stringify for catcode normalisation.
177 \\cs_new_protected:Npn \\zebra@sig #1#2
178 {
179     \\tl_set:Nx \\l_tmpa_tl { \\tl_to_str:n {#1} }
180     \\tl_set:Nx \\l_tmpb_tl { \\tl_to_str:n {#2} }
181     \\prop_gput:NVV \\g__zebra_note_sig_known_prop \\l_tmpa_tl \\l_tmpb_tl
182 }

```

```

183 \cs_new_protected:Npn \__zebra_write_sig:NN #1#2
184 {
185   \prop_if_in:NVF \g__zebra_note_sig_written_prop #1
186   {
187     \immediate\write \@auxout
188     { \string\zebra@sig { \tl_use:N #1 } { \tl_use:N #2 } }
189     \prop_gput:NVn \g__zebra_note_sig_written_prop #1 { 1 }
190   }
191 }
192 \cs_if_exist:NTF \dbend
193 {
194   \cs_set_eq:NN \__zebra_saved_dbend: \dbend
195   \cs_undefine:N \dbend
196   \RequirePackage{manfnt}
197   \cs_set_eq:NN \dbend \__zebra_saved_dbend:
198 }
199 { \RequirePackage{manfnt} }
200 \cs_new_protected:Npn \__zebra_pdfstring_note:
201 { \@ifnextchar [ { \__zebra_pdfstring_note_opt:w } { \use_none:n } }
202 \cs_new_protected:Npn \__zebra_pdfstring_note_opt:w [#1] #2 { }
203 \cs_new:Npn \__zebra_target:nn #1#2 {#2}
204 \cs_new:Npn \__zebra_link:nn #1#2 {#2}
205 \cs_new:Npn \__zebra_pageref:n #1 { \pageref {#1} }
206 \cs_new:Npn \__zebra_zebra_label_name:n #1 { #1@zebra }
207 \cs_new:Npn \__zebra_zebra_label_type:n #1
208 {
209   \exp_after:wN \use_i:nn
210   \cs:w r@\__zebra_zebra_label_name:n {#1}\cs_end:
211   { }
212 }
213 \cs_new_protected:Npn \__zebra_write_zebra_label:n #1
214 {
215   \protected@write \@auxout { }
216   {
217     \string\newlabel{\__zebra_zebra_label_name:n {#1}}
218     {{\exp_not:V \l__zebra_note_ref_type_tl}{}}
219   }
220 }
221 \cs_new_protected:Npn \__zebra_note_label:n #1
222 {
223   \__zebra_kernel_label:n {#1}
224   \__zebra_write_zebra_label:n {#1}
225 }
226 \cs_new_protected:Npn \__zebra_zebra_ref:n #1
227 {
228   \cs_if_exist:cTF { r@\__zebra_zebra_label_name:n {#1} }
229   { \__zebra_zebra_label_type:n {#1}~\ref{#1} }
230   {
231     \msg_warning:nnn { zebra } { invalid-note-label } {#1}
232     ??
233   }
234 }
235 \NewDocumentCommand \zebraref { m }
236 {

```

```

237 \bool_if:NTF \l__zebra_unnumbered_bool
238 { \ref{#1} }
239 { \__zebra_zebra_ref:n {#1} }
240 }
241 \cs_new_protected:Npn \__zebra_apply_pdfstring_defs:
242 {
243   \pdfstringdefDisableCommands
244   {
245     \cs_set:Npn \zebraref ##1 { \ref{##1} }
246     \seq_map_inline:Nn \g__zebra_note_types_seq
247     {
248       \cs_set_eq:cN { zebra##1 } \__zebra_pdfstring_note:
249       \prop_if_in:NnT \g__zebra_note_public_alias_prop { ##1 }
250       { \cs_set_eq:cN { ##1 } \__zebra_pdfstring_note: }
251     }
252   }
253 }
254 \cs_new_protected:Npn \__zebra_setup_pagelinks:
255 {
256   \cs_set:Npn \__zebra_target:nn ##1##2 {##2}
257   \cs_set:Npn \__zebra_link:nn ##1##2 {##2}
258   \cs_set:Npn \__zebra_pageref:n ##1 { \pageref {##1} }
259   \__zebra_if_package_loaded:nT { hyperref }
260   {
261     \cs_set:Npn \__zebra_pageref:n ##1 { \pageref* {##1} }
262     \bool_if:NNT \l__zebra_pagelinks_bool
263     {
264       \cs_set:Npn \__zebra_target:nn ##1##2 { \hypertarget{##1}{##2} }
265       \cs_set:Npn \__zebra_link:nn ##1##2 { \hyperlink{##1}{##2} }
266     }
267     \__zebra_apply_pdfstring_defs:
268   }
269 }
270 \hook_gput_code:nnn { begindocument } { { zebra }
271 { \__zebra_setup_pagelinks: }

```

### 4.3 Main notes macros

Various helper macros are defined before reaching out to the `\todo` commands.

Place the margin note on the nearest margin. Takes two arguments: `#1` for the left margin (number then symbol) and `#2` for the right margin (symbol then number), so the symbol always sits closest to the text column. In single-column mode `\marginnote` is used with the right-margin variant as default. In twocolumn mode `\marginpar` positions the symbol close to the text column reliably (it picks side from `\if@firstcolumn`, not from a `.aux` round trip), so we keep that path for the body. We must, however, route around the contexts where `\marginpar` would crash with “Float(s) lost”: caption sboxes (inner mode), caption parboxes (signalled by `\@capttype`), and the wide `\vbox` that `\@topnewpage` builds for `\title` (signalled by `\hsize > \columnwidth`). Those contexts fall back to `\marginnote`, where we additionally replicate `\@marginparreset` (`\@parboxrestore + \normalfont\normalsize`) so the marker does not inherit a `\Huge` font from `\title`.

```

272 \cs_new_protected:Npn \__zebra_margin_note:nn #1#2
273 {
274   \legacy_if:NTF { @twocolumn }

```

```

275 {
276   \bool_lazy_or:nnTF
277   { \mode_if_inner_p: }
278   {
279     \bool_lazy_or_p:nn
280     { \cs_if_exist_p:N \@capytype }
281     { \dim_compare_p:nNn { \hspace } > { \columnwidth } }
282   }
283   {
284     \marginnote
285     [ { \@parboxrestore \normalfont \normalsize #1 } ]
286     { \@parboxrestore \normalfont \normalsize #2 }
287   }
288   {
289     \marginpar
290     [ { \makebox[\marginparwidth][r]{#1} } ]
291     { \makebox[\marginparwidth][l]{#2} }
292   }
293 }
294 { \marginnote[#1]{#2} }
295 }
296 \cs_new:Npn \__zebra_prepend:nn #1#2
297 { \tl_if_blank:nTF {#2} {} {#1#2} }
298 \cs_new:Npn \__zebra_capitalize_type:n #1
299 { \text_uppercase:n { \tl_head:n {#1} } \tl_tail:n {#1} }
300 \cs_new:Npn \__zebra_note_count:n #1
301 { \int_use:c { g__zebra_note_count_#1_int } }
302 \cs_new:Npn \__zebra_note_color:n #1
303 { \prop_item:Nn \g__zebra_note_colors_prop {#1} }
304 \cs_new:Npn \__zebra_note_symbol:n #1
305 {
306   \prop_if_in:NnTF \g__zebra_note_symbols_prop {#1}
307   { \prop_item:Nn \g__zebra_note_symbols_prop {#1} }
308   { \l__zebra_symbol_tl }
309 }
310 \cs_new_protected:Npn \__zebra_new_listbody:n #1
311 { \tl_new:c { g__zebra_listbody_#1_tl } }
312 \tl_new:N \g__zebra_listbody_all_tl
313 \cs_new:Npn \__zebra_use_listbody:n #1
314 { \tl_use:c { g__zebra_listbody_#1_tl } }
315 \cs_new_protected:Npn \__zebra_record_note:nnnnn #1#2#3#4#5
316 {
317   \tl_gput_right:cx
318   {
319     \bool_if:NTF \l__zebra_sort_none_bool
320     { g__zebra_listbody_all_tl }
321     { g__zebra_listbody_#1_tl }
322   }
323   {
324     \exp_not:N \__zebra_list_entry:nnnnn
325     { \exp_not:n {#1} }
326     {#2}
327     { \exp_not:n {#3} }
328     { \exp_not:n {#4} }

```

```

329         {#5}
330     }
331 }
332 %% \__zebra_note_unnumbered:nnn {type}{author}{body}
333 %% Fast path for the \opt{unnumbered} option: skip dedup, .aux
334 %% signatures and list registration entirely. The counter is still
335 %% bumped on every call (so the summary table reflects rough counts;
336 %% notes inside captions/TOC/marks may be over-counted, which is the
337 %% accepted trade-off for cutting out the fragile bookkeeping).
338 \cs_new_protected:Npn \__zebra_note_unnumbered:nnn #1#2#3
339 {
340     \bool_if:NT \l__zebra_draft_bool
341     {
342         \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
343         \int_gincr:c { g__zebra_note_count_#1_int }
344         \__zebra_render_note:nnn {#1} {#2} {#3}
345     }
346 }
347 %% \__zebra_note:nnn {type}{author}{body}
348 %% Main entry point: dispatch to the lightweight unnumbered path or to
349 %% the numbered path with deduplication/bookkeeping.
350 \cs_new_protected:Npn \__zebra_note:nnn #1#2#3
351 {
352     \bool_if:NTF \l__zebra_unnumbered_bool
353     { \__zebra_note_unnumbered:nnn {#1} {#2} {#3} }
354     { \__zebra_note_numbered:nnn {#1} {#2} {#3} }
355 }
356 %% Numbered path. Four cases:
357 %% Case 1 - sbox reuse: instance_key found → reuse, render
358 %% Case 2 - stable key: label found, stable_key in prop → suppress
359 %% Case 3 - content sig: sig in .aux data → suppress
360 %% Case 4 - new note: allocate, render
361 \cs_new_protected:Npn \__zebra_note_numbered:nnn #1#2#3
362 {
363     \bool_if:NT \l__zebra_draft_bool
364     {
365         \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
366         %% Case 1: sbox reuse (same \inputlineno)
367         \tl_set:Nx \l__zebra_note_key_tl
368         { \__zebra_instance_key:nnn {#1} {#2} {#3} }
369         \prop_get:NVNTF \g__zebra_note_target_prop \l__zebra_note_key_tl
370         \l__zebra_note_target_tl
371         {
372             \prop_get:NVN \g__zebra_note_display_prop \l__zebra_note_key_tl
373             \l__zebra_note_display_tl
374             \__zebra_render_note:nnn {#1} {#2} {#3}
375         }
376     }
377     {
378         %% Extract label and compute content signature
379         \__zebra_extract_label:nN {#3} \l_tmpb_tl
380         \__zebra_content_sig:nnnN {#1} {#2} {#3} \l_tmpa_tl
381         %% Case 2: stable-key suppress (labeled, marks/headers).
382         %% If the stable key exists AND the content signature
383         %% matches, this encounter is a replay → suppress.

```

```

383 %% Different content signature = label reuse → warn and
384 %% let Case 4 allocate independently.
385 \bool_set_false:N \l_tmpa_bool
386 \tl_if_empty:NF \l_tmpb_tl
387 {
388   \tl_set:Nx \l__zebra_note_key_tl
389     { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
390   \prop_get:NVNT \g__zebra_note_stable_sig_prop
391     \l__zebra_note_key_tl \l_tmpc_tl
392   {
393     \tl_if_eq:NNTF \l_tmpa_tl \l_tmpc_tl
394     { \bool_set_true:N \l_tmpa_bool }
395     {
396       \msg_warning:nnV { zebra }
397         { duplicate-note-label } \l_tmpb_tl
398     }
399   }
400 }
401 %% Case 3: content-sig suppress (labeled, TOC replay).
402 %% Only suppress if the stored instance key differs from
403 %% the current one - same key means it is the original
404 %% note, not a replay.
405 \bool_if:NF \l_tmpa_bool
406 {
407   \prop_get:NVNT \g__zebra_note_sig_known_prop
408     \l_tmpa_tl \l_tmpc_tl
409   {
410     \tl_set:Nx \l_tmpd_tl
411       { \__zebra_instance_key:nnn {#1} {#2} {#3} }
412     \tl_if_eq:NNTF \l_tmpc_tl \l_tmpd_tl
413     { \bool_set_true:N \l_tmpa_bool }
414   }
415 }
416 \bool_if:NF \l_tmpa_bool
417 {
418   %% Case 4: new note - allocate and render
419   \tl_set:Nx \l__zebra_note_key_tl
420     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
421   \__zebra_allocate_note:nnn {#1} {#2} {#3}
422   \prop_gput:NVV \g__zebra_note_target_prop
423     \l__zebra_note_key_tl \l__zebra_note_target_tl
424   \prop_gput:NVV \g__zebra_note_display_prop
425     \l__zebra_note_key_tl \l__zebra_note_display_tl
426   %% For labeled notes: register stable key + write sig,
427   %% but only if the stable key is not already claimed by
428   %% an earlier note (label-conflict case).
429   \tl_if_empty:NF \l_tmpb_tl
430   {
431     \tl_set:Nx \l__zebra_note_key_tl
432       { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
433     \prop_if_in:NVF \g__zebra_note_stable_sig_prop
434       \l__zebra_note_key_tl
435     {
436       \prop_gput:NVV \g__zebra_note_target_prop

```

```

437         \l__zebra_note_key_tl \l__zebra_note_target_tl
438         \prop_gput:NVV \g__zebra_note_display_prop
439         \l__zebra_note_key_tl \l__zebra_note_display_tl
440         \prop_gput:NVV \g__zebra_note_stable_sig_prop
441         \l__zebra_note_key_tl \l_tmpa_tl
442         \tl_set:Nx \l_tmpc_tl
443         { \__zebra_instance_key:nnn {#1} {#2} {#3} }
444         \__zebra_write_sig:NN \l_tmpa_tl \l_tmpc_tl
445     }
446 }
447     \__zebra_render_note:nnn {#1} {#2} {#3}
448 }
449 %% Cases 2-3: suppress - no output
450 }
451 }
452 }
453 %% Full render: hypertarget, target label, margin note, inline text.
454 %% In \opt{unnumbered} mode the label/hypertarget setup is skipped so a
455 %% \cs{label} inside the note body falls through to the surrounding
456 %% \cs{@currentlabel} (e.g. \ the section number), and \cs{zebraref}
457 %% degrades cleanly to \cs{ref}.
458 \cs_new_protected:Npn \__zebra_render_note:nnn #1#2#3
459 {
460     \group_begin:
461     \bool_if:NF \l__zebra_unnumbered_bool
462     {
463         \protected@edef \@currentlabel { \l__zebra_note_display_tl }
464         \__zebra_if_package_loaded:nT { hyperref }
465         { \tl_set:Nx \@currentHref { \l__zebra_note_target_tl } }
466         \tl_set:Nx \l__zebra_note_ref_type_tl
467         { \__zebra_capitalize_type:n {#1} }
468         \__zebra_target:nn { \l__zebra_note_target_tl } {}
469         \exp_args:NV \__zebra_kernel_label:n \l__zebra_note_target_tl
470     }
471     \__zebra_margin_note:nn
472     { \textcolor{\l__zebra_note_color_tl}{%
473         \bool_if:NF \l__zebra_unnumbered_bool
474         { {\bfseries\l__zebra_note_display_tl}\kern1pt }%
475         \__zebra_note_symbol:n {#1}}}%
476     { \textcolor{\l__zebra_note_color_tl}{%
477         \__zebra_note_symbol:n {#1}%
478         \bool_if:NF \l__zebra_unnumbered_bool
479         { \kern1pt {\bfseries\l__zebra_note_display_tl} }}}}%
480     \bool_if:NF \l__zebra_unnumbered_bool
481     { \cs_set_eq:NN \label \__zebra_note_label:n }
482     \textcolor{\l__zebra_note_color_tl}{[\colorbox[gray]{0.97}{%
483         \textcolor{\l__zebra_note_color_tl}{!70!black}{%
484             \textsc{\MakeLowercase{\MakeUppercase#1}}}%
485         \bool_if:NF \l__zebra_unnumbered_bool
486         { ~\l__zebra_note_display_tl }%
487         \texttt{\__zebra_prepend:nn {@}{#2}:}} #3}}}%
488     \group_end:
489 }
490 \cs_new_protected:Npn \__zebra_new_note_type:nn #1#2

```



```

491 { \__zebra_new_note_type:nnn {#1} {#2} {} }
492 \cs_new_protected:Npn \__zebra_new_note_type:nnn #1#2#3
493 {
494   \seq_gput_right:Nn \g__zebra_note_types_seq {#1}
495   \prop_if_in:NnF \g__zebra_note_colors_prop {#1}
496   { \prop_gput:Nnn \g__zebra_note_colors_prop {#1} {#2} }
497   \tl_if_blank:nF {#3}
498   {
499     \prop_if_in:NnF \g__zebra_note_symbols_prop {#1}
500     { \prop_gput:Nnn \g__zebra_note_symbols_prop {#1} {#3} }
501   }
502   \int_new:c { g__zebra_note_count_#1_int }
503   \__zebra_new_listbody:n {#1}
504   \exp_args:Nc \NewDocumentCommand { zebra#1 } { 0{ } m }
505   { \__zebra_note:nnn {#1}{##1}{##2} }
506   \__zebra_if_package_loaded:nT { hyperref }
507   { \__zebra_apply_pdfstring_defs: }
508   \cs_if_exist:cTF {#1}
509   { \msg_warning:nnn { zebra } { command-taken } {#1} }
510   {
511     \cs_set_eq:cc {#1} {zebra#1}
512     \prop_gput:Nnn \g__zebra_note_public_alias_prop {#1} { true }
513   }
514 }
515 \cs_new_protected:Npn \__zebra_list_entry:nnnnn #1#2#3#4#5
516 {
517   \par\noindent
518   \textcolor{\__zebra_note_color:n {#1}}{%
519     \textbf{\__zebra_capitalize_type:n {#1}~#2}%
520     \tl_if_blank:nF {#3} { \enspace \texttt{\__zebra_prepend:nn {#1}{#3}} } }%
521   \nobreak\dotfill
522   \__zebra_link:nn {#5} { \__zebra_pageref:n {#5} }%
523   \par
524   \beginngroup
525     \leftskip=2em
526     \rightskip=2em
527     \parindent=0pt
528     \cs_set_eq:NN \label \use_none:n
529     #4\par
530   \endgroup
531 }
532 \cs_new_protected:Npn \__zebra_print_note_group:n #1
533 {
534   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
535   {
536     \par\medskip
537     \__zebra_use_listbody:n {#1}
538   }
539 }
540 \cs_new_protected:Npn \__zebra_print_notes_inorder:
541 {
542   \tl_if_empty:NF \g__zebra_listbody_all_tl
543   { \par\medskip \tl_use:N \g__zebra_listbody_all_tl }
544 }

```

```

545 \cs_new_protected:Npn \__zebra_summary_row:n #1
546 {
547   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
548   {
549     \int_add:Nn \l__zebra_total_notes_int { \__zebra_note_count:n {#1} }
550     \tl_put_right:Nx \l__zebra_summary_rows_tl
551     {
552       \exp_not:N \textcolor
553       { \__zebra_note_color:n {#1} }
554       {#1}
555       \exp_not:N &
556       \__zebra_note_count:n {#1}
557       \exp_not:N \\
558     }
559   }
560 }
561 \cs_new_protected:Npn \__zebra_print_notes:
562 {
563   \tl_clear:N \l__zebra_summary_rows_tl
564   \int_zero:N \l__zebra_total_notes_int
565   \seq_map_inline:Nn \g__zebra_note_types_seq
566   { \__zebra_summary_row:n {##1} }
567   \tl_if_empty:NF \l__zebra_summary_rows_tl
568   {
569     \par\nobreak
570     \noindent\dotfill\par\medskip
571     \nobreak
572     \noindent\textbf{\Large Zebra~Notes}
573     \par \medskip
574     \begin{center}
575       \begin{tabular}{lr}
576         \hline
577         \textbf{Type} & \textbf{Count} \\ \hline
578         \tl_use:N \l__zebra_summary_rows_tl
579         \hline
580         \textbf{Total} & \int_use:N \l__zebra_total_notes_int \\ \hline
581       \end{tabular}
582     \end{center}
583     \bool_if:NF \l__zebra_unnumbered_bool
584     {
585       \par \medskip
586       \begin{group}
587         \small
588         \noindent{\bfseries List~of~notes}\par
589         \nobreak
590         \bool_if:NTF \l__zebra_sort_none_bool
591         { \__zebra_print_notes_inorder: }
592         {
593           \seq_map_inline:Nn \g__zebra_note_types_seq
594           { \__zebra_print_note_group:n {##1} }
595         }
596       \end{group}
597     }
598   }

```

```

599     }
600 }

\zebranewnote All note types are created with \zebranewnote.
601 \NewDocumentCommand \zebranewnote { m m O{} }
602 { \_zebra_new_note_type:nnn {#1} {#2} {#3} }

(End of definition for \zebranewnote. This function is documented on page 4.)

\zebrasetup Applies configuration keys after loading using the zebra-setup key family.
603 \NewDocumentCommand \zebrasetup { m }
604 { \keys_set:nn { zebra-setup } {#1} }

(End of definition for \zebrasetup. This function is documented on page 5.)

\todo Built-in note types, defined with \zebranewnote.
\note \zebranewnote{todo}{purple}
\fixed \colorlet{zebra@fixed@color}{green!50!black}
\comment \zebranewnote{fixed}{zebra@fixed@color}
\placeholder \zebranewnote{comment}{blue}
608 \zebranewnote{note}{violet}
609 \zebranewnote{placeholder}{gray}
610

(End of definition for \todo and others. These functions are documented on page 3.)

```

## 4.4 Print summary at end of the document

A summary table and a detailed note list are inserted automatically at the end of the document. Each note type with at least one instance is listed with its colour and count, followed by notes in document order or grouped by type.

```

611 %% At end of document: print the note summary and list.
612 %% Content signatures are written to .aux inline (at allocation time),
613 %% so no additional end-of-document aux writes are needed.
614 \hook_gput_code:nnn { enddocument } { zebra }
615 {
616     \bool_if:NT \l_zebra_draft_bool
617     { \_zebra_print_notes: }
618 }
619 \ExplSyntaxOff
620 \endpackage

```

## 4.5 Compatibility shim

The old package name `zebra-goodies` is supported via a thin wrapper that loads `zebra` and prints a deprecation warning.

```

621 \compat
622 \NeedsTeXFormat{LaTeX2e}
623 \ProvidesPackage{zebra-goodies}
624 [2026/04/26 v1.8.0 Deprecated: use zebra instead]
625 \PackageWarningNoLine{zebra-goodies}
626 {Package 'zebra-goodies' is deprecated.\MessageBreak
627 Use \string\usepackage{zebra} instead}
628 \RequirePackageWithOptions{zebra}
629 \compat

```

## 4.6 Two-column demo

A standalone two-column document used to generate the demo figure included in the documentation. It is extracted automatically by docstrip and compiled during the build.

```

630 <*demo-twocol>
631 \documentclass[twocolumn]{article}
632 \usepackage[paperwidth=21cm,paperheight=15cm,margin=1.1cm]{geometry}
633 \usepackage{zebra}
634 \zebrasetup{symbol/comment=$\clubsuit$}
635 \pagestyle{empty}
636 \begin{document}
637 \section{Demo name\comment{revise the name}}
638 This draft still needs work
639 \todo[alice]{\label{zebra:intro}revise the introduction}. The
640 opening paragraph should also explain the main goal more plainly.
641 Add one more citation here \note[bob]{support this claim}. A
642 brief roadmap sentence would also make the structure easier to
643 scan.
644
645 The issue raised in Todo~\ref{zebra:intro} still applies in
646 the conclusion. The table now looks fine
647 \fixed[carol]{alignment corrected}, but one figure is still
648 missing \placeholder[eve]{insert the overview figure}. A short
649 transition would also help the flow. The middle section should
650 probably end with a clearer summary sentence before the
651 discussion begins. That summary can stay compact, but it should
652 signal why the next section matters.
653
654 Please verify the totals \note[frank]{check the numbers} and
655 confirm the wording in the last paragraph
656 \comment[tom]{is this sentence too strong?}. A small typo has
657 already been fixed \fixed[heidi]{typo corrected}. The ending
658 should stay short. The final sentence should return to the main
659 claim rather than repeat background material. You can place
660 \todo[judy]{summarise the findings} anywhere once the narrative
661 is stable.
662
663 One more short paragraph is enough to show how \placeholder{wow,
664 so great!} the markers stay readable in a compact two-column layout.
665 The example is intentionally small, but it should still look like
666 a realistic revision pass.\comment{Bye}
667 \end{document}
668 </demo-twocol>

```

## Change History

v0.1.0		v0.3.0
General: Initial public release . . . . .	1	General: Detect command conflicts . . . 1
v0.2.0		v0.4.0
General: Fix xcolor conflict . . . . .	1	General: Show note number for easy

reference	1	v1.2.0	
v0.5.0		General: Simplify key architecture.	1
General: Use darker color for label	1	v1.3.0	
v0.6.0		General: Rename package to <b>zebra</b> .	1
General: Use gray background for label	1	Rename the microtype expansion option to <b>font-expansion</b> .	1
v0.7.0		Rename the page-link option to <b>pagelinks/nopagelinks</b> .	1
General: Move to docstrip	2	v1.4.0	
v0.8.0		General: Support note labels via <b>\label</b> , <b>\ref</b> , and <b>\zebraref</b> .	1
General: Fix new note demo	4	v1.5.0	
<b>\zebranewnote</b> : Fix on <b>\global</b> for examples	19	General: Fix notes numbering in moving arguments.	1
v0.8.1		v1.6.0	
General: Fix doc	4	General: Numbering in moving arguments is hard.	1
v0.9.0		v1.7.0	
General: Fix legacy bugs and improve implementation	1	General: Robust margin notes in twocolumn captions, titles and headings.	1
v0.9.1		v1.8.0	
General: Beautify the numbers.	1	General: Add <b>unnumbered</b> option for lightweight mode.	1
v0.9.2		Fix margin symbol distance.	1
General: Faster.	1	Stabilise labeled notes replayed through uppercased running heads.	1
v1.0.0			
General: <b>expl3</b> , list of notes and compatibility.	1		
v1.1.0			
General: Customisable margin symbol, accurate page numbers, code cleanup.	1		
v1.1.1			
General: Per-type <b>color/symbol</b> keys, <b>\zebrasetup</b> .	1		

## 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>Symbols</b>		<b>\bfseries</b>	474, 479, 589
<b>\</b>	27, 28, 129, 145, 557, 577, 580	<b>bool</b> commands:	
<b>\{</b>	129, 145	<b>\bool_if:NTF</b>	79, 83, 91, 237, 262, 319, 340, 352, 363, 405, 416, 461, 473, 478, 480, 485, 584, 591, 616
<b>\}</b>	129, 145	<b>\bool_lazy_or:nnTF</b>	276
<b>\_</b>	456	<b>\bool_lazy_or_p:nn</b>	279
<b>Numbers</b>		<b>\bool_new:N</b>	7, 8, 9, 10, 11
<b>\1</b>	146	<b>\bool_set_false:N</b>	55, 73, 385
<b>A</b>		<b>\bool_set_true:N</b>	56, 394, 413
<b>\A</b>	145	<b>\l_tmpa_bool</b>	385, 394, 405, 413, 416
<b>B</b>		<b>C</b>	
<b>\begin</b>	574, 575, 636	<b>\caption</b>	3, 8
<b>\begingroup</b>	524, 587	<b>\clubsuit</b>	634





<code>\tl_if_empty:NTF</code> . . .	386, 429, 542, 567	<code>\__zebra_link:nn</code> . . .	204, 257, 265, 522
<code>\tl_if_eq:NNTF</code> . . . . .	147, 393, 412	<code>\__zebra_list_entry:nnnnn</code> . .	324, 515
<code>\tl_new:N</code> . . . . .	16, 17, 18, 19, 22, 110, 111, 311, 312	<code>\g_zebra_listbody_all_tl</code> . . . . .	312, 542, 543
<code>\tl_put_right:Nn</code> . . . . .	550	<code>\_zebra_margin_note:nn</code> . . . .	272, 471
<code>\tl_set:Nn</code> . . . . .	23, 127, 130, 142, 162, 164, 179, 180, 342, 365, 367, 388, 410, 419, 431, 442, 465, 466	<code>\l_zebra_microtype_expansion_</code> <code>bool</code> . . . . .	8, 48, 73, 79, 83
<code>\tl_set_eq:NN</code> . . . . .	143, 149	<code>\__zebra_new_listbody:n</code> . . . .	310, 503
<code>\tl_tail:n</code> . . . . .	299	<code>\__zebra_new_note_type:nn</code> . . . .	490
<code>\tl_to_str</code> . . . . .	153	<code>\__zebra_new_note_type:nnn</code> . . . . .	491, 492, 602
<code>\tl_to_str:n</code> . . . . .	115, 116, 117, 127, 132, 133, 142, 156, 179, 180	<code>\_zebra_note:nnn</code> . . . . .	347, 350, 505
<code>\tl_use:N</code> . . . . .	188, 314, 543, 578	<code>\__zebra_note_color:n</code> . . . . .	302, 342, 365, 518, 553
<code>\l_tmpa_tl</code> . . . . .	127, 129, 134, 179, 181, 379, 393, 408, 441, 444	<code>\l_zebra_note_color_tl</code> . . . . .	17, 342, 365, 472, 476, 482, 483
<code>\l_tmpb_tl</code> . . . . .	142, 143, 146, 147, 149, 180, 181, 378, 386, 389, 397, 429, 432	<code>\g_zebra_note_colors_prop</code> . . . . .	13, 65, 303, 495, 496
tmpc commands:		<code>\_zebra_note_count:n</code> . . . . .	162, 300, 534, 547, 549, 556
<code>\l_tmpc_tl</code> . . . . .	143, 147, 391, 393, 408, 412, 442, 444	<code>\g_zebra_note_display_prop</code> . . . . .	103, 372, 424, 438
tmpd commands:		<code>\l_zebra_note_display_tl</code> . . . . .	110, 162, 168, 373, 425, 439, 463, 474, 479, 486
<code>\l_tmpd_tl</code> . . . . .	410, 412	<code>\g_zebra_note_id_int</code> . . . . .	15, 163, 165
<code>\todo</code> . . . . .	1, 3, 4, 12, 605, 639, 660	<code>\l_zebra_note_key_tl</code> . . . . .	111, 367, 369, 372, 388, 391, 419, 423, 425, 431, 434, 437, 439, 441
<b>U</b>			
unnumbered (option) . . . . .	2	<code>\_zebra_note_label:n</code> . . . . .	221, 481
use commands:		<code>\_zebra_note_numbered:nnn</code> . . . .	354, 361
<code>\use_i:nn</code> . . . . .	209	<code>\g_zebra_note_public_alias_prop</code> . . . . .	14, 249, 512
<code>\use_none:n</code> . . . . .	201, 528	<code>\l_zebra_note_ref_type_tl</code> . . . . .	18, 218, 466
<code>\usepackage</code> . . . . .	627, 632, 633	<code>\g_zebra_note_sig_known_prop</code> . . . . .	105, 181, 407
<b>V</b>			
<code>\vbox</code> . . . . .	12	<code>\g_zebra_note_sig_written_prop</code> . . . . .	107, 185, 189
<b>W</b>			
<code>\write</code> . . . . .	187	<code>\g_zebra_note_stable_sig_prop</code> . . . . .	109, 390, 433, 440
<b>Z</b>			
<code>\Z</code> . . . . .	145	<code>\_zebra_note_symbol:n</code> . . . . .	304, 475, 477
zebra internal commands:		<code>\g_zebra_note_symbols_prop</code> . . . . .	21, 70, 306, 307, 499, 500
<code>\__zebra_allocate_note:nnn</code> . .	159, 421	<code>\g_zebra_note_target_prop</code> . . . . .	102, 369, 422, 436
<code>\__zebra_apply_pdfstring_defs:</code> . .	241, 267, 507	<code>\l_zebra_note_target_tl</code> . . . . .	16, 164, 171, 370, 423, 437, 465, 468, 469
<code>\__zebra_capitalize_type:n</code> . . . . .	298, 467, 519	<code>\g_zebra_note_types_seq</code> . . . . .	12, 246, 494, 565, 594
<code>\__zebra_content_sig:nnnN</code> . .	125, 379	<code>\_zebra_note_unnumbered:nnn</code> . . . . .	332, 338, 353
<code>\l_zebra_draft_bool</code> . . . . .	7, 45, 340, 363, 616	<code>\l_zebra_pagelinks_bool</code> . . . . .	9, 51, 91, 262
<code>\__zebra_extract_label:nN</code> . .	140, 378	<code>\_zebra_pageref:n</code> . . . . .	205, 258, 261, 522
<code>\_zebra_if_package_loaded:n</code> . . . .	36	<code>\_zebra_pdfstring_note:</code> . . . . .	200, 248, 250
<code>\_zebra_if_package_loaded:NTF</code> . . . . .	77, 89, 259, 464, 506		
<code>\_zebra_instance_key:nnn</code> . . . . .	113, 368, 411, 420, 443		
<code>\_zebra_kernel_label:n</code> . . . . .	99, 223, 469		



\_zebra_pdfstring_note_opt:w . . .	\l_zebra_total_notes_int . . . . .
. . . . . 201, 202	. . . . . 20, 549, 564, 580
\_zebra_prepend:nn . . . 296, 487, 520	\l_zebra_unnumbered_bool 11, 58,
\_zebra_print_note_group:n 532, 595	237, 352, 461, 473, 478, 480, 485, 584
\_zebra_print_notes: . . . . 561, 617	\_zebra_use_listbody:n . . . . 313, 537
\_zebra_print_notes_inorder: . . .	\_zebra_write_sig:NN . . . . 183, 444
. . . . . 540, 592	\_zebra_write_zebra_label:n 213, 224
\_zebra_record_note:nnnnn . 166, 315	\_zebra_zebra_label_name:n . . . .
\_zebra_render_note:nnn . . . . .	. . . . . 206, 210, 217, 228
. . . . . 344, 374, 447, 458	\_zebra_zebra_label_type:n 207, 229
\_zebra_saved_dbend: . . . . 194, 197	\_zebra_zebra_ref:n . . . . . 226, 239
\_zebra_setup_microtype: . . . 75, 88	\zebracomment . . . . . 3, 4
\_zebra_setup_pagelinks: . . 254, 271	\zebrafixed . . . . . 4
\l_zebra_sort_none_bool . . . . .	\zebranewnote . . . . . 1,
. . . . . 10, 55, 56, 319, 591	4, 19, 601, 605, 607, 608, 609, 610
\_zebra_stable_key:nn . 155, 389, 432	\zebranote . . . . . 3
\_zebra_summary_row:n . . . . 545, 566	\zebraplaceholder . . . . . 4
\l_zebra_summary_rows_tl . . . . .	\zebraquestion . . . . . 4
. . . . . 19, 550, 563, 567, 578	\zebraref . . . . . 2, 5, 21, 235, 245
\l_zebra_symbol_tl . . . . . 22, 23, 308	\zebrasetup . . . . . 5, 7, 21, 603, 634
\_zebra_target:nn . 203, 256, 264, 468	\zebratodo . . . . . 3

## Zebra Notes

Type	Count
todo	4
fixed	1
comment	1
note	3
placeholder	1
question	1
<b>Total</b>	<b>11</b>

### List of notes

<b>Todo 1</b> @alice . . . . .	3
add two or three citations from the latest survey	
<b>Todo 2</b> @bob, carol . . . . .	3
verify the hyperparameter table against the source code and update any outdated entries	
and let's check afterwards	
<b>Note 1</b> . . . . .	4
how should we structure the intro?	
<b>Comment 1</b> @tom . . . . .	4
the setup description looks clear now	
Placeholder 1 @lucy, tom . . . . .	4
good job!	
<b>Todo 3</b> . . . . .	4
check bibliography entries for formatting	
<b>Fixed 1</b> @John . . . . .	4
updated the conclusion	
<b>Note 2</b> @who . . . . .	4

anything else?	
<b>Question 1</b> @who .....	4
what's this?	
<b>Todo 4</b> .....	5
revise the introduction before submission	
<b>Note 3</b> .....	5
see Todo 4 (that is, Todo 4) on p. 5	