

# zebra — Writing Revision Toolkit\*

Ruini Xue<sup>†</sup>

v2.0.0 (2026/05/20)

## 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. If `hyperref` is loaded, the detailed list links back to the notes. 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
3.4	Limitations . . . . .	6
<b>4</b>	<b>Implementation</b>	<b>7</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 . . . . .	20
4.5	Compatibility shim . . . . .	20
4.6	Two-column demo . . . . .	20

<b>Change History</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

- draft** These two options are complementary. Default: **true** (draft mode). All notes are typeset inline and a summary table plus a detailed note list are appended at the end of the document. Setting **final** (or **draft=false**) suppresses all notes and the generated lists, producing clean output ready for distribution.
- sort** Controls the order of the detailed note list printed at the end of the document. Default: **none** (document order). **sort=type** groups them by note type.  
If **hyperref** is loaded by the document, page numbers in the detailed note list link back to the corresponding notes. **zebra** does not load **hyperref** itself; load it explicitly if links are desired. Without **hyperref**, the list uses ordinary page references.
- unnumbered** Turns off note numbering. Inline markers become `[todo: ...]` (no number), the margin symbol carries no number, the end-of-document detailed note list is omitted (the per-type summary table remains), and `\zebraref` falls back to `\ref`. Useful as a workaround if numbered notes interact badly with a particular class; the summary table count is then an upper-bound estimate. Default: **false**.

## 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 [ COMMENT 1@tom: the setup description looks clear now ]. Results are presented in the following 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 pass [ TODO 3: check bibliography entries for formatting ] before we can finalize the submission.

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

2 

---

<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]

1 🚧

1 🚧

1 🌱

1 🚧

2 🚧

2 ♣

2 🌱

2 🚧

2 🚧

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.

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

pass.[ COMMENT 3: Bye]

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

revis the name

Todo 1 @alice.....1

revis 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

### 3.4 Limitations

- Notes in moving arguments such as \section and \caption should use \label for stable numbering. Unlabelled notes there may be counted again when replayed in contents lists or running heads.

- A labelled note and an unlabelled note with the same type, author, and body may be treated as the same replay. Add a `\label`, vary the text, or avoid identical note bodies.
- Identical unlabelled notes on the same source line, such as `\todo{x}\todo{x}`, collapse to one note.
- In `final` mode, labels inside notes are unavailable. In `unnumbered` mode, the detailed list is omitted and summary counts are only an upper-bound estimate.
- `\zebranewnote` expects command-name-safe note names and colours already known to `xcolor`; invalid values are left to `TEX` or `xcolor` errors.

## 4 Implementation

```

1 <*package>
2 <@@=zebra>
   Version data to start with.
3 \NeedsTeXFormat{LaTeX2e}[2022-06-01]
4 \ProvidesExplPackage{zebra}
5   {2026/05/20}
6   {2.0.0}
7   {Writing Revision Toolkit}

```

### 4.1 Package options

Package options `draft`, `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.

```

8 \bool_new:N \l__zebra_draft_bool
9 \bool_new:N \l__zebra_emit_label_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 } { duplicate-note-label }
31 { Note~label~'#1'~used~by~a~different~note;~second~note~gets~

```

```

32     its-own-identity. }
33
34 \prg_new_conditional:Npnn \__zebra_if_package_loaded:n #1 { T , F , TF }
35 {
36     \cs_if_exist:cTF { ver@#1.sty }
37     { \prg_return_true: }
38     { \prg_return_false: }
39 }
40
41 \keys_define:nn { zebra }
42 {
43     draft .bool_set:N = \l__zebra_draft_bool,
44     draft .initial:n = true,
45     final .meta:n = { draft = false },
46     sort .choice:,
47     sort / type .code:n = { \bool_set_false:N \l__zebra_sort_none_bool },
48     sort / none .code:n = { \bool_set_true:N \l__zebra_sort_none_bool },
49     sort .initial:n = none,
50     unnumbered .bool_set:N = \l__zebra_unnumbered_bool,
51     unnumbered .initial:n = false,
52 }
53 \ProcessKeyOptions [ zebra ]
54 \keys_define:nn { zebra-setup / color }
55 {
56     unknown .code:n =
57         { \prop_gput:NVn \g__zebra_note_colors_prop \l_keys_key_str {#1} }
58 }
59 \keys_define:nn { zebra-setup / symbol }
60 {
61     unknown .code:n =
62         { \prop_gput:NVn \g__zebra_note_symbols_prop \l_keys_key_str {#1} }
63 }

```

## 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* (`\langle type \rangle | \langle author \rangle | \langle body \rangle | \inputlineno`) catches the replay for both labeled and unlabeled notes. 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* (`\langle type \rangle | \langle label name \rangle`) stored alongside the allocation; a later encounter from marks/headers that still carries the `\label` finds this key and suppresses.



2. A *content signature* ( $\langle type \rangle | \langle author \rangle | \langle sanitised body \rangle$ ) 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 on each replay (cosmetic duplicate on TOC/LOF/marks); adding \label is the recommended fix when stable replay behaviour is required.

**Limitation — same-line same-body unlabeled.** Two unlabeled notes with identical body on the same source line (e.g. \todo{x}\todo{x}) share an instance key with a single \sbox replay of the same source location and are indistinguishable from it at the TeX level. Case 1 collapses the second occurrence into the first. Workarounds: split across lines, vary the body, or add \label. Notes inside \fbox, tabular cells, \parbox, minipage, footnotes, and other inner-mode containers are unaffected — a single source encounter records normally regardless of the surrounding mode.

```

64 \RequirePackage{xcolor}
65 \RequirePackage{marginnote}
66 \cs_new_eq:NN \__zebra_kernel_label:n \label
67 %% -- dedup data structures --
68 %% Maps any key (instance, stable, or content-sig) to the allocation.
69 \prop_new:N \g__zebra_note_target_prop
70 \prop_new:N \g__zebra_note_display_prop
71 %% Content signatures of labeled notes from previous pass (.aux).
72 \prop_new:N \g__zebra_note_sig_known_prop
73 %% Content signatures written this pass (dedup aux writes).
74 \prop_new:N \g__zebra_note_sig_written_prop
75 %% Content signature stored per stable key (for label-conflict detection).
76 \prop_new:N \g__zebra_note_stable_sig_prop
77 %% Group-local set of targets whose \label has been emitted. Sbox
78 %% boundaries reset it (so caption replay re-emits after the sbox is
79 %% discarded), while a same-source-line repeat in body text sees the
80 %% prior emission and skips the redundant \label, avoiding
81 %% multiply-defined labels.
82 \prop_new:N \l__zebra_label_emitted_prop
83 \tl_new:N \l__zebra_note_display_tl
84 \tl_new:N \l__zebra_note_key_tl
85 %% Instance key: unique per source location.
86 \cs_new:Npn \__zebra_instance_key:nnn #1#2#3
87 {
88   \tl_to_str:n {#1}
89   | \tl_to_str:n {#2}
90   | \tl_to_str:n {#3}
91   | \int_eval:n { \tex_inputlineno:D }
92 }
93 %% Content signature: body stringified with ALL \label{...} stripped.
94 %% Matches across body (has labels), TOC (labels consumed by \edef),
95 %% and running heads (which may uppercase the author/body text).
96 %% The optional \protect prefix covers marks and \protected@write paths.
97 %% Uses replace_all so that multiple labels are all stripped.
98 \cs_new_protected:Npn \__zebra_content_sig:nnnN #1#2#3#4
99 {
100   \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#3} }
101   \regex_replace_all:nnN

```

```

102     { (? : \protect\s* )? \\label\s* \{ [^{}]* \} } { } \l_tmpa_tl
103 \tl_set:Nx #4
104 {
105     \tl_to_str:n {#1}
106     | \str_lowercase:f { \tl_to_str:n {#2} }
107     | \str_lowercase:f { \l_tmpa_tl }
108 }
109 }
110 %% Extract the last \label name from the stringified body
111 %% (the greedy \A.* consumes up to the rightmost \label). Both the
112 %% original encounter and any replay run through this same regex,
113 %% so they agree on which label to use as the stable key.
114 %% Sets #2 to the label name, or clears it if none found.
115 %% The optional \protect prefix covers marks and \protected@write paths.
116 \cs_new_protected:Npn \__zebra_extract_label:nN #1#2
117 {
118     \tl_set:Nx \l_tmpb_tl { \tl_to_str:n {#1} }
119     \tl_set_eq:NN \l_tmpc_tl \l_tmpb_tl
120     \regex_replace_once:nnN
121     { \A .* (? : \protect\s* )? \\label\s* \{ ([^{}]* ) \} .* \Z }
122     { \1 } \l_tmpb_tl
123     \tl_if_eq:NNTF \l_tmpb_tl \l_tmpc_tl
124     { \tl_clear:N #2 }
125     { \tl_set_eq:NN #2 \l_tmpb_tl }
126 }
127 %% Stable key for labeled notes.
128 %% #2 is expected to be already stringified (from regex extraction),
129 %% so no \tl_to_str is applied - otherwise an unexpanded variable
130 %% token would be stringified instead of its value.
131 \cs_new:Npn \__zebra_stable_key:nn #1#2
132 { \tl_to_str:n {#1} | label | #2 }
133 %% Allocate a fresh note: increment the type counter, generate
134 %% a unique hypertarget name, and record the note in the list body.
135 \cs_new_protected:Npn \__zebra_allocate_note:nnn #1#2#3
136 {
137     \int_gincr:c { g__zebra_note_count_#1_int }
138     \tl_set:Nx \l__zebra_note_display_tl { \__zebra_note_count:n {#1} }
139     \int_gincr:N \g__zebra_note_id_int
140     \tl_set:Nx \l__zebra_note_target_tl
141     { zebranote.\int_use:N \g__zebra_note_id_int }
142     \__zebra_record_note:nnnnn
143     {#1}
144     { \l__zebra_note_display_tl }
145     {#2}
146     {#3}
147     { \l__zebra_note_target_tl }
148 }
149 %% Aux-file interface: record a content signature together with the
150 %% originating instance key. A later encounter whose instance key
151 %% differs from the stored one is a replay and is suppressed.
152 %% Re-stringify for catcode normalisation.
153 \cs_new_protected:Npn \zebra@sig #1#2
154 {
155     \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#1} }

```

```

156 \tl_set:Nx \l_tmpb_tl { \tl_to_str:n {#2} }
157 \prop_gput:NVV \g__zebra_note_sig_known_prop \l_tmpa_tl \l_tmpb_tl
158 }
159 \cs_new_protected:Npn \__zebra_write_sig:NN #1#2
160 {
161   \prop_if_in:NVF \g__zebra_note_sig_written_prop #1
162   {
163     \immediate\write \@auxout
164     { \string\zebra@sig { \tl_use:N #1 } { \tl_use:N #2 } }
165     \prop_gput:NVN \g__zebra_note_sig_written_prop #1 { 1 }
166   }
167 }
168 \cs_if_exist:NTF \dbend
169 {
170   \cs_set_eq:NN \__zebra_saved_dbend: \dbend
171   \cs_undefine:N \dbend
172   \RequirePackage{manfnt}
173   \cs_set_eq:NN \dbend \__zebra_saved_dbend:
174 }
175 { \RequirePackage{manfnt} }
176 \cs_new:Npn \__zebra_pdfstring_note:
177 #1
178 {
179   \str_if_eq:eeT { \tl_to_str:n {#1} } { [ ] }
180   { \__zebra_pdfstring_note_opt:w }
181 }
182 \cs_new:Npn \__zebra_pdfstring_note_opt:w #1 ] #2 { }
183 \cs_new:Npn \__zebra_target:nn #1#2 {#2}
184 \cs_new:Npn \__zebra_link:nn #1#2 {#2}
185 \cs_new:Npn \__zebra_pageref:n #1 { \pageref {#1} }
186 \cs_new:Npn \__zebra_zebra_label_name:n #1 { #1@zebra }
187 \cs_new:Npn \__zebra_zebra_label_type:n #1
188 {
189   \exp_after:wN \use_i:nn
190   \cs:w r@\__zebra_zebra_label_name:n {#1}\cs_end:
191   { }
192 }
193 \cs_new_protected:Npn \__zebra_write_zebra_label:n #1
194 {
195   \protected@write \@auxout { }
196   {
197     \string\newlabel{\__zebra_zebra_label_name:n {#1}}
198     {{\exp_not:V \l__zebra_note_ref_type_tl}{}}
199   }
200 }
201 \cs_new_protected:Npn \__zebra_note_label:n #1
202 {
203   \__zebra_kernel_label:n {#1}
204   \__zebra_write_zebra_label:n {#1}
205 }
206 \cs_new_protected:Npn \__zebra_zebra_ref:n #1
207 {
208   \cs_if_exist:cTF { r@\__zebra_zebra_label_name:n {#1} }
209   { \__zebra_zebra_label_type:n {#1}~\ref{#1} }

```

```

210     { \ref{#1} }
211   }
212 \NewDocumentCommand \zebraref { m }
213 {
214   \bool_if:NTF \l__zebra_unnumbered_bool
215     { \ref{#1} }
216     { \__zebra_zebra_ref:n {#1} }
217 }
218 \cs_new_protected:Npn \__zebra_apply_pdfstring_defs:
219 {
220   \pdfstringdefDisableCommands
221   {
222     \cs_set:Npn \zebraref ##1 { \ref{##1} }
223     \seq_map_inline:Nn \g__zebra_note_types_seq
224     {
225       \cs_set_eq:cN { zebra##1 } \__zebra_pdfstring_note:
226       \prop_if_in:NnT \g__zebra_note_public_alias_prop { ##1 }
227         { \cs_set_eq:cN { ##1 } \__zebra_pdfstring_note: }
228     }
229   }
230 }
231 \cs_new_protected:Npn \__zebra_setup_hyperref:
232 {
233   \cs_set:Npn \__zebra_target:nn ##1##2 {##2}
234   \cs_set:Npn \__zebra_link:nn ##1##2 {##2}
235   \cs_set:Npn \__zebra_pageref:n ##1 { \pageref {##1} }
236   \__zebra_if_package_loaded:nT { hyperref }
237   {
238     \cs_set:Npn \__zebra_pageref:n ##1 { \pageref* {##1} }
239     \cs_set:Npn \__zebra_target:nn ##1##2 { \hypertarget{##1}{##2} }
240     \cs_set:Npn \__zebra_link:nn ##1##2 { \hyperlink{##1}{##2} }
241     \__zebra_apply_pdfstring_defs:
242   }
243 }
244 \hook_gput_code:nnn { begindocument } { { zebra }
245   { \__zebra_setup_hyperref: }

```

### 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 `\@capytype`), 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`.

```

246 \cs_new_protected:Npn \__zebra_margin_note:nn #1#2
247 {
248   \legacy_if:nTF { @twocolumn }
249   {
250     \bool_lazy_or:nnTF
251     { \mode_if_inner_p: }
252     {
253       \bool_lazy_or_p:nn
254       { \cs_if_exist_p:N \@capttype }
255       { \dim_compare_p:nNn { \hspace } > { \columnwidth } }
256     }
257     {
258       \marginnote
259       [ { \@parboxrestore \normalfont \normalsize #1 } ]
260       { \@parboxrestore \normalfont \normalsize #2 }
261     }
262     {
263       \marginpar
264       [ { \makebox[\marginparwidth][r]{#1} } ]
265       { \makebox[\marginparwidth][l]{#2} }
266     }
267   }
268   { \marginnote[#1]{#2} }
269 }
270 \cs_new:Npn \__zebra_prepend:nn #1#2
271 { \tl_if_blank:nTF {#2} {} {#1#2} }
272 \cs_new:Npn \__zebra_capitalize_type:n #1
273 { \text_uppercase:n { \tl_head:n {#1} } \tl_tail:n {#1} }
274 \cs_new:Npn \__zebra_note_count:n #1
275 { \int_use:c { g__zebra_note_count_#1_int } }
276 \cs_new:Npn \__zebra_note_color:n #1
277 { \prop_item:Nn \g__zebra_note_colors_prop {#1} }
278 \cs_new:Npn \__zebra_note_symbol:n #1
279 {
280   \prop_if_in:NnTF \g__zebra_note_symbols_prop {#1}
281   { \prop_item:Nn \g__zebra_note_symbols_prop {#1} }
282   { \l__zebra_symbol_tl }
283 }
284 \cs_new_protected:Npn \__zebra_new_listbody:n #1
285 { \tl_new:c { g__zebra_listbody_#1_tl } }
286 \tl_new:N \g__zebra_listbody_all_tl
287 \cs_new:Npn \__zebra_use_listbody:n #1
288 { \tl_use:c { g__zebra_listbody_#1_tl } }
289 \cs_new_protected:Npn \__zebra_record_note:nnnnn #1#2#3#4#5
290 {
291   \tl_gput_right:cx
292   {
293     \bool_if:NTF \l__zebra_sort_none_bool
294     { g__zebra_listbody_all_tl }
295     { g__zebra_listbody_#1_tl }
296   }
297   {
298     \exp_not:N \__zebra_list_entry:nnnnn
299     { \exp_not:n {#1} }

```

```

300     {#2}
301     { \exp_not:n {#3} }
302     { \exp_not:n {#4} }
303     {#5}
304   }
305 }
306 %% \__zebra_note_unnumbered:nnn {type}{author}{body}
307 %% Fast path for the \opt{unnumbered} option: skip dedup, .aux
308 %% signatures and list registration entirely. The counter is still
309 %% bumped on every call (so the summary table reflects rough counts;
310 %% notes inside captions/TOC/marks may be over-counted, which is the
311 %% accepted trade-off for cutting out the fragile bookkeeping).
312 \cs_new_protected:Npn \__zebra_note_unnumbered:nnn #1#2#3
313 {
314   \bool_if:NT \l__zebra_draft_bool
315   {
316     \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
317     \int_gincr:c { g__zebra_note_count_#1_int }
318     \__zebra_render_note:nnn {#1} {#2} {#3}
319   }
320 }
321 %% \__zebra_note:nnn {type}{author}{body}
322 %% Main entry point: dispatch to the lightweight unnumbered path or to
323 %% the numbered path with deduplication/bookkeeping.
324 \cs_new_protected:Npn \__zebra_note:nnn #1#2#3
325 {
326   \bool_if:NTF \l__zebra_unnumbered_bool
327   { \__zebra_note_unnumbered:nnn {#1} {#2} {#3} }
328   { \__zebra_note_numbered:nnn {#1} {#2} {#3} }
329 }
330 %% Numbered path. Four cases:
331 %% Case 1 - sbbox reuse: instance_key found → reuse, render
332 %% Case 2 - stable key: label found, stable_key in prop → suppress
333 %% Case 3 - content sig: sig in .aux data → suppress
334 %% Case 4 - new note: allocate, render
335 %% Case 1 also catches an unlabeled same-source-line same-body pair
336 %% (\todo{x}\todo{x}); the second is collapsed. Two such notes are
337 %% indistinguishable at TeX level from a single note replayed by an
338 %% sbbox/moving argument, so this collision is documented and resolved
339 %% by adding \label or splitting the line.
340 \cs_new_protected:Npn \__zebra_note_numbered:nnn #1#2#3
341 {
342   \bool_if:NT \l__zebra_draft_bool
343   {
344     \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
345     %% Case 1: sbbox reuse (same \inputlineno)
346     \tl_set:Nx \l__zebra_note_key_tl
347     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
348     \prop_get:NVNTF \g__zebra_note_target_prop \l__zebra_note_key_tl
349     \l__zebra_note_target_tl
350     {
351       \prop_get:NVN \g__zebra_note_display_prop \l__zebra_note_key_tl
352       \l__zebra_note_display_tl
353       \__zebra_render_note:nnn {#1} {#2} {#3}

```

```

354 }
355 {
356     %% Extract label and compute content signature
357     \__zebra_extract_label:nN {#3} \l_tmpb_tl
358     \__zebra_content_sig:nnnN {#1} {#2} {#3} \l_tmpa_tl
359     %% Case 2: stable-key suppress (labeled, marks/headers).
360     %% If the stable key exists AND the content signature
361     %% matches, this encounter is a replay → suppress.
362     %% Different content signature = label reuse → warn and
363     %% let Case 4 allocate independently.
364     \bool_set_false:N \l_tmpa_bool
365     \tl_if_empty:NF \l_tmpb_tl
366     {
367         \tl_set:Nx \l__zebra_note_key_tl
368         { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
369         \prop_get:NVNT \g__zebra_note_stable_sig_prop
370         \l__zebra_note_key_tl \l_tmpc_tl
371         {
372             \tl_if_eq:NNTF \l_tmpa_tl \l_tmpc_tl
373             { \bool_set_true:N \l_tmpa_bool }
374             {
375                 \msg_warning:nnV { zebra }
376                 { duplicate-note-label } \l_tmpb_tl
377             }
378         }
379     }
380     %% Case 3: content-sig suppress (TOC/LOF/marks replay).
381     %% Only unlabeled current encounters may be suppressed by
382     %% content signature; labeled encounters already have a
383     %% strong identity via the stable key. Suppress only if
384     %% the stored instance key differs from the current one -
385     %% same key means it is the original note, not a replay.
386     \bool_if:NF \l_tmpa_bool
387     {
388         \tl_if_empty:NT \l_tmpb_tl
389         {
390             \prop_get:NVNT \g__zebra_note_sig_known_prop
391             \l_tmpa_tl \l_tmpc_tl
392             {
393                 \tl_set:Nx \l_tmpd_tl
394                 { \__zebra_instance_key:nnn {#1} {#2} {#3} }
395                 \tl_if_eq:NNTF \l_tmpc_tl \l_tmpd_tl
396                 { \bool_set_true:N \l_tmpa_bool }
397             }
398         }
399     }
400     \bool_if:NF \l_tmpa_bool
401     {
402         %% Case 4: new note - allocate and render
403         \tl_set:Nx \l__zebra_note_key_tl
404         { \__zebra_instance_key:nnn {#1} {#2} {#3} }
405         \__zebra_allocate_note:nnn {#1} {#2} {#3}
406         \prop_gput:NVV \g__zebra_note_target_prop
407         \l__zebra_note_key_tl \l__zebra_note_target_tl

```

```

408 \prop_gput:NVV \g__zebra_note_display_prop
409 \l__zebra_note_key_tl \l__zebra_note_display_tl
410 %% For labeled notes: register stable key + write sig,
411 %% but only if the stable key is not already claimed by
412 %% an earlier note (label-conflict case).
413 \tl_if_empty:NF \l_tmpb_tl
414 {
415   \tl_set:Nx \l__zebra_note_key_tl
416   { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
417   \prop_if_in:NVF \g__zebra_note_stable_sig_prop
418   \l__zebra_note_key_tl
419   {
420     \prop_gput:NVV \g__zebra_note_target_prop
421     \l__zebra_note_key_tl \l__zebra_note_target_tl
422     \prop_gput:NVV \g__zebra_note_display_prop
423     \l__zebra_note_key_tl \l__zebra_note_display_tl
424     \prop_gput:NVV \g__zebra_note_stable_sig_prop
425     \l__zebra_note_key_tl \l_tmpa_tl
426     \tl_set:Nx \l_tmpc_tl
427     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
428     \__zebra_write_sig:NN \l_tmpa_tl \l_tmpc_tl
429   }
430 }
431 \__zebra_render_note:nnn {#1} {#2} {#3}
432 }
433 %% Cases 2-3: suppress - no output
434 }
435 }
436 }
437 %% Full render: hypertarget, target label, margin note, inline text.
438 %% In \opt{unnumbered} mode the label/hypertarget setup is skipped so a
439 %% \cs{label} inside the note body falls through to the surrounding
440 %% \cs{@currentlabel} (e.g.\ the section number), and \cs{zebraref}
441 %% degrades cleanly to \cs{ref}.
442 \cs_new_protected:Npn \__zebra_render_note:nnn #1#2#3
443 {
444   %% Decide outside render's group whether to emit the kernel label.
445   %% The flag lives in the caller's group: sbox/parbox boundaries reset
446   %% it (so caption replay re-emits), but a same-line repeat in body
447   %% text sees the prior emission and skips the redundant \label.
448   \bool_set_false:N \l__zebra_emit_label_bool
449   \bool_if:NF \l__zebra_unnumbered_bool
450   {
451     \prop_if_in:NVF \l__zebra_label_emitted_prop \l__zebra_note_target_tl
452     {
453       \bool_set_true:N \l__zebra_emit_label_bool
454       \prop_put:NVn \l__zebra_label_emitted_prop
455       \l__zebra_note_target_tl { 1 }
456     }
457   }
458   \group_begin:
459   \bool_if:NF \l__zebra_unnumbered_bool
460   {
461     \protected@edef \@currentlabel { \l__zebra_note_display_tl }

```



```

462     \__zebra_if_package_loaded:nT { hyperref }
463     { \tl_set:Nx \@currentHref { \l__zebra_note_target_tl } }
464     \tl_set:Nx \l__zebra_note_ref_type_tl
465     { \__zebra_capitalize_type:n {#1} }
466     \__zebra_target:nn { \l__zebra_note_target_tl } {}
467     \bool_if:NT \l__zebra_emit_label_bool
468     { \exp_args:NV \__zebra_kernel_label:n \l__zebra_note_target_tl }
469   }
470   \__zebra_margin_note:nn
471   { \textcolor{\l__zebra_note_color_tl}{%
472     \bool_if:NF \l__zebra_unnumbered_bool
473     { {\bfseries\l__zebra_note_display_tl}\kern1pt }%
474     \__zebra_note_symbol:n {#1}} }
475   { \textcolor{\l__zebra_note_color_tl}{%
476     \__zebra_note_symbol:n {#1}%
477     \bool_if:NF \l__zebra_unnumbered_bool
478     { \kern1pt {\bfseries\l__zebra_note_display_tl} } } }%
479   \bool_if:NF \l__zebra_unnumbered_bool
480   { \cs_set_eq:NN \label \__zebra_note_label:n }
481   \textcolor{\l__zebra_note_color_tl}{[\colorbox[gray]{0.97}{%
482     \textcolor{\l__zebra_note_color_tl}{!70!black}{%
483       \textsc{\MakeLowercase\MakeUppercase#1}}}%
484     \bool_if:NF \l__zebra_unnumbered_bool
485     { ~\l__zebra_note_display_tl } }%
486     \texttt{\__zebra_prepend:nn {#1}{#2}}:}} #3}}%
487   \group_end:
488 }
489 \cs_new_protected:Npn \__zebra_new_note_type:nn #1#2
490 { \__zebra_new_note_type:nnn {#1} {#2} {} }
491 \cs_new_protected:Npn \__zebra_new_note_type:nnn #1#2#3
492 {
493   \seq_gput_right:Nn \g__zebra_note_types_seq {#1}
494   \prop_if_in:NnF \g__zebra_note_colors_prop {#1}
495   { \prop_gput:Nnn \g__zebra_note_colors_prop {#1} {#2} }
496   \tl_if_blank:nF {#3}
497   {
498     \prop_if_in:NnF \g__zebra_note_symbols_prop {#1}
499     { \prop_gput:Nnn \g__zebra_note_symbols_prop {#1} {#3} }
500   }
501   \int_new:c { g__zebra_note_count_#1_int }
502   \__zebra_new_listbody:n {#1}
503   \exp_args:Nc \NewDocumentCommand { zebra#1 } { 0{} m }
504   { \__zebra_note:nnn {#1}{##1}{##2} }
505   \cs_if_exist:cTF {#1}
506   { \msg_warning:nnn { zebra } { command-taken } {#1} }
507   {
508     \cs_set_eq:cc {#1} {zebra#1}
509     \prop_gput:Nnn \g__zebra_note_public_alias_prop {#1} { true }
510   }
511   \__zebra_if_package_loaded:nT { hyperref }
512   { \__zebra_apply_pdfstring_defs: }
513 }
514 \cs_new_protected:Npn \__zebra_list_entry:nnnnn #1#2#3#4#5
515 {

```

```

516 \par\noindent
517 \textcolor{\_zebra_note_color:n {#1}}{%
518 \textbf{\_zebra_capitalize_type:n {#1}~#2}%
519 \tl_if_blank:nF {#3} { \enspace \texttt{\_zebra_prepend:nn {0}{#3}} }}%
520 \nobreak\dotfill
521 \_zebra_link:nn {#5} { \_zebra_pageref:n {#5} }%
522 \par
523 \beginingroup
524 \leftskip=2em
525 \rightskip=2em
526 \parindent=0pt
527 \cs_set_eq:NN \label \use_none:n
528 #4\par
529 \endgroup
530 }
531 \cs_new_protected:Npn \_zebra_print_note_group:n #1
532 {
533 \int_compare:nNnT { \_zebra_note_count:n {#1} } > { 0 }
534 {
535 \par\medskip
536 \_zebra_use_listbody:n {#1}
537 }
538 }
539 \cs_new_protected:Npn \_zebra_print_notes_inorder:
540 {
541 \tl_if_empty:NF \g_zebra_listbody_all_tl
542 { \par\medskip \tl_use:N \g_zebra_listbody_all_tl }
543 }
544 \cs_new_protected:Npn \_zebra_summary_row:n #1
545 {
546 \int_compare:nNnT { \_zebra_note_count:n {#1} } > { 0 }
547 {
548 \int_add:Nn \l_zebra_total_notes_int { \_zebra_note_count:n {#1} }
549 \tl_put_right:Nx \l_zebra_summary_rows_tl
550 {
551 \exp_not:N \textcolor
552 { \_zebra_note_color:n {#1} }
553 {#1}
554 \exp_not:N &
555 \_zebra_note_count:n {#1}
556 \exp_not:N \\
557 }
558 }
559 }
560 \cs_new_protected:Npn \_zebra_print_notes:
561 {
562 \tl_clear:N \l_zebra_summary_rows_tl
563 \int_zero:N \l_zebra_total_notes_int
564 \seq_map_inline:Nn \g_zebra_note_types_seq
565 { \_zebra_summary_row:n {##1} }
566 \tl_if_empty:NF \l_zebra_summary_rows_tl
567 {
568 \par\nobreak
569 \noindent\dotfill\par\medskip

```

```

570     \nobreak
571     \noindent\textbf{\Large Zebra-Notes}
572     \par \medskip
573     \begin{center}
574         \begin{tabular}{lr}
575             \hline
576             \textbf{Type} & \textbf{Count} \\ \hline
577             \tl_use:N \l__zebra_summary_rows_tl
578             \hline
579             \textbf{Total} & \textbf{\int_use:N \l__zebra_total_notes_int} \\ \hline
580             \end{tabular}
581     \end{center}
582     \bool_if:NF \l__zebra_unnumbered_bool
583     {
584         \par \medskip
585         \begin{group}
586             \small
587             \noindent{\bfseries List-of-notes}\par
588             \nobreak
589             \bool_if:NTF \l__zebra_sort_none_bool
590             { \__zebra_print_notes_inorder: }
591             {
592                 \seq_map_inline:Nn \g__zebra_note_types_seq
593                 { \__zebra_print_note_group:n {##1} }
594             }
595         \end{group}
596     }
597 }
598 }
599 }

```

**\zebranewnote** All note types are created with \zebranewnote.

```

600 \NewDocumentCommand \zebranewnote { m m O{} }
601 { \__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.

```

602 \NewDocumentCommand \zebrasetup { m }
603 { \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 604 \zebranewnote{todo}{purple}
\fixed 605 \colorlet{zebra@fixed@color}{green!50!black}
\comment 606 \zebranewnote{fixed}{zebra@fixed@color}
\placeholder 607 \zebranewnote{comment}{blue}
608 \zebranewnote{note}{violet}
609 \zebranewnote{placeholder}{gray}

```

(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.

```
610 %% At end of document: print the note summary and list.
611 %% Content signatures are written to .aux inline (at allocation time),
612 %% so no additional end-of-document aux writes are needed.
613 \hook_gput_code:nnn { enddocument } { zebra }
614 {
615     \bool_if:NT \l__zebra_draft_bool
616     { \__zebra_print_notes: }
617 }
618 \ExplSyntaxOff
619 \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.

```
620 \compat
621 \NeedsTeXFormat{LaTeX2e}
622 \ProvidesPackage{zebra-goodies}
623 [2026/05/20 v2.0.0 Deprecated: use zebra instead]
624 \PackageWarningNoLine{zebra-goodies}
625 {Package 'zebra-goodies' is deprecated.\MessageBreak
626 Use \string\usepackage{zebra} instead}
627 \RequirePackageWithOptions{zebra}
628 \endcompat
```

## 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.

```
629 \demo-twocol
630 \documentclass[twocolumn]{article}
631 \usepackage[paperwidth=21cm,paperheight=15cm,margin=1.1cm]{geometry}
632 \usepackage{zebra}
633 \zebrasetup{symbol/comment=$\clubsuit$}
634 \pagestyle{empty}
635 \begin{document}
636 \section{Demo name\comment{revise the name}}
637 This draft still needs work
638 \todo[alice]{\label{zebra:intro}revise the introduction}. The
639 opening paragraph should also explain the main goal more plainly.
640 Add one more citation here \note[bob]{support this claim}. A
641 brief roadmap sentence would also make the structure easier to
642 scan.
643
644 The issue raised in \todo{\ref{zebra:intro}} still applies in
645 the conclusion. The table now looks fine
646 \fixed[carol]{alignment corrected}, but one figure is still
```

```

647 missing \placeholder[eve]{insert the overview figure}. A short
648 transition would also help the flow. The middle section should
649 probably end with a clearer summary sentence before the
650 discussion begins. That summary can stay compact, but it should
651 signal why the next section matters.
652
653 Please verify the totals \note[frank]{check the numbers} and
654 confirm the wording in the last paragraph
655 \comment[tom]{is this sentence too strong?}. A small typo has
656 already been fixed \fixed[heidi]{typo corrected}. The ending
657 should stay short. The final sentence should return to the main
658 claim rather than repeat background material. You can place
659 \todo[judy]{summarise the findings} anywhere once the narrative
660 is stable.
661
662 One more short paragraph is enough to show how \placeholder{wow,
663 so great!} the markers stay readable in a compact two-column layout.
664 The example is intentionally small, but it should still look like
665 a realistic revision pass.\comment{Bye}
666 \end{document}
667 </demo-twocol>

```

## Change History

v0.1.0		v0.9.2	
General: Initial public release . . . . .	1	General: Faster. . . . .	1
v0.2.0		v1.0.0	
General: Fix xcolor conflict . . . . .	1	General: <code>expl3</code> , list of notes and compatibility. . . . .	1
v0.3.0		v1.1.0	
General: Detect command conflicts . . .	1	General: Customisable margin symbol, accurate page numbers, code cleanup. . . . .	1
v0.4.0		v1.1.1	
General: Show note number for easy reference . . . . .	1	General: Per-type <code>color/symbol</code> keys, <code>\zebrasetup</code> . . . . .	1
v0.5.0		v1.2.0	
General: Use darker color for label . . .	1	General: Simplify key architecture. . . .	1
v0.6.0		v1.3.0	
General: Use gray background for label	1	General: Rename package to <b>zebra</b> . . . .	1
v0.7.0		Rename the microtype expansion option to <code>font-expansion</code> . . . . .	1
General: Move to docstrip . . . . .	2	Rename the page-link option to <code>pagelinks/nopagelinks</code> . . . . .	1
v0.8.0		v1.4.0	
General: Fix new note demo . . . . .	4	General: Support note labels via <code>\label</code> , <code>\ref</code> , and <code>\zebraref</code> . . . .	1
<code>\zebranewnote</code> : Fix on <code>\global</code> for examples . . . . .	19	v1.5.0	
v0.8.1		General: Fix notes numbering in moving arguments. . . . .	1
General: Fix doc . . . . .	4		
v0.9.0			
General: Fix legacy bugs and improve implementation . . . . .	1		
v0.9.1			
General: Beautify the numbers. . . . .	1		

v1.6.0			
General: Numbering in moving arguments is hard. . . . .	1		
v1.7.0			
General: Robust margin notes in twocolumn captions, titles and headings. . . . .	1		
v1.8.0			
General: Add <code>unnumbered</code> option for lightweight mode. . . . .	1		
Fix margin symbol distance. . . . .	1		
Stabilise labeled notes replayed through uppercased running heads. . . . .	1		
v1.8.1			
General: Drop note bodies from PDF bookmarks. . . . .	1		
Fix multiply-defined labels for repeated unlabeled notes. . . . .	1		
Keep labeled notes with shared content distinct. . . . .	1		
Silence pass-1 warning from <code>\zebraref</code> . . . . .	1		
v1.9.0			
General: Document known limitations. . . . .	1		
v2.0.0			
General: Stop loading <code>hyperref</code> and <code>microtype</code> . . . . .	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>		
<code>\</code> . . . . .	27, 28, 102, 121, 556, 576, 579	
<code>\{</code> . . . . .	102, 121	
<code>\}</code> . . . . .	102, 121	
<code>\_</code> . . . . .	440	
<b>Numbers</b>		
<code>\1</code> . . . . .	122	
<b>A</b>		
<code>\A</code> . . . . .	111, 121	
<b>B</b>		
<code>\begin</code> . . . . .	573, 574, 635	
<code>\begingroup</code> . . . . .	523, 586	
<code>\bfseries</code> . . . . .	473, 478, 588	
bool commands:		
<code>\bool_if:NTF</code> . . . . .	214, 293, 314, 326, 342, 386, 400, 449, 459, 467, 472, 477, 479, 484, 583, 590, 615	
<code>\bool_lazy_or:nnTF</code> . . . . .	250	
<code>\bool_lazy_or_p:nn</code> . . . . .	253	
<code>\bool_new:N</code> . . . . .	8, 9, 10, 11	
<code>\bool_set_false:N</code> . . . . .	47, 364, 448	
<code>\bool_set_true:N</code> . . . . .	48, 373, 396, 453	
<code>\l_tmpa_bool</code> . . . . .	364, 373, 386, 396, 400	
<b>C</b>		
<code>\caption</code> . . . . .	3, 6, 8	
<code>\clubsuit</code> . . . . .	633	
<code>\cmd</code> . . . . .	3	
<code>\colorbox</code> . . . . .	481	
<code>\colorlet</code> . . . . .	4, 605	
<code>\columnwidth</code> . . . . .	12, 255	
<code>\comment</code> . . . . .	1, 3, 4, 604, 636, 655, 665	
<code>\cs</code> . . . . .	439, 440, 441	
cs commands:		
<code>\cs:w</code> . . . . .	190	
<code>\cs_end:</code> . . . . .	190	
<code>\cs_if_exist:NTF</code> . . . . .	36, 168, 208, 505	
<code>\cs_if_exist_p:N</code> . . . . .	254	
<code>\cs_new:Npn</code> . . . . .	86, 131, 176, 182, 183, 184, 185, 186, 187, 270, 272, 274, 276, 278, 287	
<code>\cs_new_eq:NN</code> . . . . .	66	
<code>\cs_new_protected:Npn</code> . . . . .	98, 116, 135, 153, 159, 193, 201, 206, 218, 231, 246, 284, 289, 312, 324, 340, 442, 489, 491, 514, 531, 539, 544, 560	
<code>\cs_set:Npn</code> . . . . .	222, 233, 234, 235, 238, 239, 240	
<code>\cs_set_eq:NN</code> . . . . .	170, 173, 225, 227, 480, 508, 527	
<code>\cs_undefine:N</code> . . . . .	171	
<b>D</b>		
<code>\dbend</code> . . . . .	168, 170, 171, 173	
<code>\definecolor</code> . . . . .	4	
dim commands:		
<code>\dim_compare_p:nNn</code> . . . . .	255	
<code>\documentclass</code> . . . . .	630	
<code>\dotfill</code> . . . . .	520, 569	

draft (option) . . . . .	2		
		<b>E</b>	
\edef . . . . .	9, 94		
\end . . . . .	581, 582, 666		
\endgroup . . . . .	529, 596		
\enspace . . . . .	519		
exp commands:			
\exp_after:wN . . . . .	189		
\exp_args:Nc . . . . .	503		
\exp_args:NV . . . . .	468		
\exp_not:N . . . . .	298, 551, 554, 556		
\exp_not:n . . . . .	198, 299, 301, 302		
\ExplSyntaxOff . . . . .	618		
		<b>F</b>	
\fbox . . . . .	9		
final (option) . . . . .	2		
\fixed . . . . .	1, 4, 604, 646, 656		
		<b>G</b>	
\global . . . . .	21		
group commands:			
\group_begin: . . . . .	458		
\group_end: . . . . .	487		
		<b>H</b>	
\hline . . . . .	575, 576, 578, 580		
hook commands:			
\hook_gput_code:nnn . . . . .	244, 613		
\hsize . . . . .	12, 255		
\Huge . . . . .	12		
\hyperlink . . . . .	240		
\hypertarget . . . . .	239		
		<b>I</b>	
\immediate . . . . .	163		
\inputlineno . . . . .	8, 345		
int commands:			
\int_add:Nn . . . . .	548		
\int_compare:nNnTF . . . . .	533, 546		
\int_eval:n . . . . .	91		
\int_gincr:N . . . . .	137, 139, 317		
\int_new:N . . . . .	15, 20, 501		
\int_use:N . . . . .	141, 275, 579		
\int_zero:N . . . . .	563		
iow commands:			
\iow_char:N . . . . .	27, 28		
		<b>K</b>	
\kern . . . . .	473, 478		
keys commands:			
\keys_define:nn . . . . .	41, 54, 59		
\l_keys_key_str . . . . .	57, 62		
\keys_set:nn . . . . .	603		
		<b>L</b>	
\label . . . . .	3, 5–9, 21, 66, 77, 80, 93, 110, 111, 339, 447, 480, 527, 638		
\Large . . . . .	571		
\leftskip . . . . .	524		
legacy commands:			
\legacy_if:nTF . . . . .	248		
		<b>M</b>	
\makebox . . . . .	264, 265		
\MakeLowercase . . . . .	483		
\MakeUppercase . . . . .	483		
\marginnote . . . . .	12, 258, 268		
\marginpar . . . . .	12, 263		
\marginparwidth . . . . .	264, 265		
\medskip . . . . .	535, 542, 569, 572, 585		
\MessageBreak . . . . .	625		
mode commands:			
\mode_if_inner_p: . . . . .	251		
msg commands:			
\msg_new:nnn . . . . .	25, 30		
\msg_warning:nnn . . . . .	375, 506		
		<b>N</b>	
\NeedsTeXFormat . . . . .	3, 621		
\NewDocumentCommand . . . . .	212, 503, 600, 602		
\newlabel . . . . .	197		
\nobreak . . . . .	520, 568, 570, 589		
\noindent . . . . .	516, 569, 571, 588		
\normalfont . . . . .	12, 259, 260		
\normalsize . . . . .	12, 259, 260		
\note . . . . .	1, 3, 604, 640, 653		
		<b>O</b>	
\opt . . . . .	307, 438		
options:			
draft . . . . .	2		
final . . . . .	2		
sort . . . . .	2		
unnumbered . . . . .	2		
		<b>P</b>	
\PackageWarningNoLine . . . . .	624		
\pageref . . . . .	185, 235, 238		
\pagestyle . . . . .	634		
\par . . . . .	516, 522, 528, 535, 542, 568, 569, 572, 585, 588		
\parbox . . . . .	9		
\parindent . . . . .	526		
\pdfstringdefDisableCommands . . . . .	220		
\placeholder . . . . .	1, 4, 604, 647, 662		
prg commands:			
\prg_new_conditional:Npnn . . . . .	34		
\prg_return_false: . . . . .	38		
\prg_return_true: . . . . .	37		





## Z

\Z .....	121	\g__zebra_note_sig_known_prop ....	72, 157, 390
zebra internal commands:		\g__zebra_note_sig_written_prop .	74, 161, 165
\_zebra_allocate_note:nnn .	135, 405	\g__zebra_note_stable_sig_prop ..	76, 369, 417, 424
\_zebra_apply_pdfstring_defs: ..	218, 241, 512	\__zebra_note_symbol:n .	278, 474, 476
\_zebra_capitalize_type:n .....	272, 465, 518	\g__zebra_note_symbols_prop .....	21, 62, 280, 281, 498, 499
\_zebra_content_sig:nnnN ...	98, 358	\g__zebra_note_target_prop .....	69, 348, 406, 420
\l_zebra_draft_bool .....	8, 43, 314, 342, 615	\l_zebra_note_target_tl .....	16, 140, 147,
\l_zebra_emit_label_bool .....	9, 448, 453, 467		349, 407, 421, 451, 455, 463, 466, 468
\_zebra_extract_label:nN ..	116, 357	\g__zebra_note_types_seq .....	12, 223, 493, 564, 593
\_zebra_if_package_loaded:n ...	34	\__zebra_note_unnumbered:nnn ...	306, 312, 327
\_zebra_if_package_loaded:nTF ..	236, 462, 511	\__zebra_pageref:n .	185, 235, 238, 521
\_zebra_instance_key:nnn .....	86, 347, 394, 404, 427	\__zebra_pdfstring_note: ..	176, 225, 227
\_zebra_kernel_label:n .	66, 203, 468	\__zebra_pdfstring_note_opt:w ...	180, 182
\l_zebra_label_emitted_prop ...	82, 451, 454	\__zebra_prepend:nn ...	270, 486, 519
\_zebra_link:nn ...	184, 234, 240, 521	\__zebra_print_note_group:n	531, 594
\_zebra_list_entry:nnnnn ..	298, 514	\__zebra_print_notes: .....	560, 616
\g__zebra_listbody_all_tl .....	286, 541, 542	\__zebra_print_notes_inorder: ...	539, 591
\_zebra_margin_note:nn .....	246, 470	\__zebra_record_note:nnnnn .	142, 289
\_zebra_new_listbody:n .....	284, 502	\__zebra_render_note:nnn .....	318, 353, 431, 442
\_zebra_new_note_type:nn .....	489	\__zebra_saved_dbend: .....	170, 173
.....	490, 491, 601	\__zebra_setup_hyperref: ...	231, 245
\_zebra_note:nnn .....	321, 324, 504	\l_zebra_sort_none_bool .....	10, 47, 48, 293, 590
\_zebra_note_color:n .....	276, 316, 344, 517, 552	\__zebra_stable_key:nn .	131, 368, 416
\l_zebra_note_color_tl .....	17, 316, 344, 471, 475, 481, 482	\__zebra_summary_row:n .....	544, 565
\g__zebra_note_colors_prop .....	13, 57, 277, 494, 495	\l_zebra_summary_rows_tl .....	19, 549, 562, 566, 577
\_zebra_note_count:n .....	138, 274, 533, 546, 548, 555	\l_zebra_symbol_tl .....	22, 23, 282
\g__zebra_note_display_prop .....	70, 351, 408, 422	\__zebra_target:nn .	183, 233, 239, 466
\l_zebra_note_display_tl	83, 138,	\l_zebra_total_notes_int .....	20, 548, 563, 579
144, 352, 409, 423, 461, 473, 478, 485		\l_zebra_unnumbered_bool .....	11, 50, 214,
\g__zebra_note_id_int ..	15, 139, 141		326, 449, 459, 472, 477, 479, 484, 583
\l_zebra_note_key_tl .....	84, 346, 348, 351, 367, 370,	\__zebra_use_listbody:n ...	287, 536
403, 407, 409, 415, 418, 421, 423, 425		\__zebra_write_sig:NN .....	159, 428
\_zebra_note_label:n .....	201, 480	\__zebra_write_zebra_label:n	193, 204
\_zebra_note_numbered:nnn .	328, 340	\__zebra_zebra_label_name:n ...	186, 190, 197, 208
\g__zebra_note_public_alias_prop	14, 226, 509	\__zebra_zebra_label_type:n	187, 209
\l_zebra_note_ref_type_tl .....	18, 198, 464	\__zebra_zebra_ref:n .....	206, 216
		\zebracomment .....	3, 4
		\zebrafixed .....	4

<code>\zebranewnote</code> .....	<a href="#">1</a> , <a href="#">4</a> , <a href="#">7</a> , <a href="#">19</a> , <a href="#">600</a> , <a href="#">604</a> , <a href="#">606</a> , <a href="#">607</a> , <a href="#">608</a> , <a href="#">609</a>	<code>\zebraquestion</code> .....	<a href="#">4</a>
<code>\zebranote</code> .....	<a href="#">3</a>	<code>\zebraref</code> .....	<a href="#">2</a> , <a href="#">5</a> , <a href="#">21</a> , <a href="#">22</a> , <a href="#">212</a> , <a href="#">222</a>
<code>\zebraplaceholder</code> .....	<a href="#">4</a>	<code>\zebrasetup</code> .....	<a href="#">5</a> , <a href="#">7</a> , <a href="#">21</a> , <a href="#">602</a> , <a href="#">633</a>
		<code>\zebratodo</code> .....	<a href="#">3</a>

## Zebra Notes

Type	Count
<a href="#">todo</a>	4
<a href="#">fixed</a>	1
<a href="#">comment</a>	1
<a href="#">note</a>	3
placeholder	1
<a href="#">question</a>	1
<b>Total</b>	<b>11</b>

### List of notes

<b>Todo 1</b> <a href="#">@alice</a> .....	<a href="#">3</a>
add two or three citations from the latest survey	
<b>Todo 2</b> <a href="#">@bob</a> , <a href="#">carol</a> .....	<a href="#">3</a>
verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards	
<b>Note 1</b> .....	<a href="#">4</a>
how should we structure the intro?	
<b>Comment 1</b> <a href="#">@tom</a> .....	<a href="#">4</a>
the setup description looks clear now	
Placeholder 1 <a href="#">@lucy</a> , <a href="#">tom</a> .....	<a href="#">4</a>
good job!	
<b>Todo 3</b> .....	<a href="#">4</a>
check bibliography entries for formatting	
<b>Fixed 1</b> <a href="#">@John</a> .....	<a href="#">4</a>
updated the conclusion	
<b>Note 2</b> <a href="#">@who</a> .....	<a href="#">4</a>
anything else?	
<b>Question 1</b> <a href="#">@who</a> .....	<a href="#">4</a>
what's this?	
<b>Todo 4</b> .....	<a href="#">5</a>
revise the introduction before submission	
<b>Note 3</b> .....	<a href="#">5</a>
see <a href="#">Todo 4</a> (that is, <a href="#">Todo 4</a> ) on p. <a href="#">5</a>	