

zebra — Writing Revision Toolkit*

Ruini Xue[†]

v1.9.0 (2026/05/01)

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

1	Introduction	2
2	Installation	2
3	Using the package	2
	3.1 Package Options	2
	3.2 Notes Macros	3
	3.3 Two-column Support	5
	3.4 Limitations	6
4	Implementation	7
	4.1 Package options	7
	4.2 Moving-argument deduplication	9
	4.3 Main notes macros	13
	4.4 Print summary at end of the document	20
	4.5 Compatibility shim	21
	4.6 Two-column demo	21
	Change History	22

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

[†]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^AT_EX, 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^AT_EX 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.
- final**
- 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.
- pagelinks** Controls whether page numbers in the detailed note list are clickable. Default: **true**. Set **pagelinks=false** to disable these links. The complementary option **nopagelinks** is also accepted.
- font-expansion** Controls **microtype** font expansion. This usually improves the appearance of the document. Disable it if it conflicts with your engine or another package by setting **font-expansion=false**. The complementary option **nofont-expansion** is also accepted. Default: **true**. **microtype** remains loaded when expansion is disabled.
- 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 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].

2 

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>

```
\fixed      \fixed[⟨name⟩]{⟨text⟩}
\zebrafixed \zebrafixed[⟨name⟩]{⟨text⟩}
```


```
\placeholder  \placeholder[⟨name⟩]{⟨text⟩}
\zebraplaceholder \zebraplaceholder[⟨name⟩]{⟨text⟩}
```


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 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

3  [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.

```
\zebranewnote \zebranewnote{⟨note name⟩}{⟨xcolor name⟩}[⟨symbol⟩]
```

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^AT_EX, 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 🛠️ 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].

1 🛠️ 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 🛠️ 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].

2 🛠️ 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.

2 🛠️ 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
Total	11

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

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 \ProvidesExplPackage{zebra}
4   {2026/05/01}
5   {1.9.0}
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^AT_EX 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_emit_label_bool
9 \bool_new:N \l__zebra_microtype_expansion_bool
10 \bool_new:N \l__zebra_pagelinks_bool
11 \bool_new:N \l__zebra_sort_none_bool
12 \bool_new:N \l__zebra_unnumbered_bool
13 \seq_new:N \g__zebra_note_types_seq
14 \prop_new:N \g__zebra_note_colors_prop
15 \prop_new:N \g__zebra_note_public_alias_prop
16 \int_new:N \g__zebra_note_id_int
17 \tl_new:N \l__zebra_note_target_tl
18 \tl_new:N \l__zebra_note_color_tl
19 \tl_new:N \l__zebra_note_ref_type_tl
20 \tl_new:N \l__zebra_summary_rows_tl
21 \int_new:N \l__zebra_total_notes_int
22 \prop_new:N \g__zebra_note_symbols_prop
23 \tl_new:N \l__zebra_symbol_tl
24 \tl_set:Nn \l__zebra_symbol_tl { \textdbend }
25
26 \msg_new:nnn { zebra } { command-taken }
27   {
28     The-command-'\iow_char:N\#1'-is-already-defined.~
29     Use-'\iow_char:N\zebra#1'-instead.
30   }

```

```

31 \msg_new:nnn { zebra } { duplicate-note-label }
32   { Note-label~'#1'~used-by-a-different-note;~second-note-gets~
33     its-own-identity. }
34
35 \prg_new_conditional:Npnn \__zebra_if_package_loaded:n #1 { T , F , TF }
36   {
37     \cs_if_exist:cTF { ver@#1.sty }
38     { \prg_return_true: }
39     { \prg_return_false: }
40   }
41
42 \keys_define:nn { zebra }
43   {
44     draft .bool_set:N = \l__zebra_draft_bool,
45     draft .initial:n = true,
46     final .meta:n = { draft = false },
47     font-expansion .bool_set:N = \l__zebra_microtype_expansion_bool,
48     font-expansion .initial:n = true,
49     nofont-expansion .meta:n = { font-expansion = false },
50     pagelinks .bool_set:N = \l__zebra_pagelinks_bool,
51     pagelinks .initial:n = true,
52     nopagelinks .meta:n = { pagelinks = false },
53     sort .choice:,
54     sort / type .code:n = { \bool_set_false:N \l__zebra_sort_none_bool },
55     sort / none .code:n = { \bool_set_true:N \l__zebra_sort_none_bool },
56     sort .initial:n = none,
57     unnumbered .bool_set:N = \l__zebra_unnumbered_bool,
58     unnumbered .initial:n = false,
59   }
60 \ProcessKeyOptions [ zebra ]
61 \keys_define:nn { zebra-setup / color }
62   {
63     unknown .code:n =
64       { \prop_gput:NVn \g__zebra_note_colors_prop \l_keys_key_str {#1} }
65   }
66 \keys_define:nn { zebra-setup / symbol }
67   {
68     unknown .code:n =
69       { \prop_gput:NVn \g__zebra_note_symbols_prop \l_keys_key_str {#1} }
70   }
71 \sys_if_engine_xetex:T
72   { \bool_set_false:N \l__zebra_microtype_expansion_bool }
73
74 \cs_new_protected:Npn \__zebra_setup_microtype:
75   {
76     \__zebra_if_package_loaded:nTF { microtype }
77     {
78       \bool_if:NF \l__zebra_microtype_expansion_bool
79       { \microtypesetup { expansion = false } }
80     }
81     {
82       \bool_if:NTF \l__zebra_microtype_expansion_bool
83       { \RequirePackage{microtype} }
84       { \RequirePackage[expansion=false]{microtype} }

```



```

85     }
86   }
87   \__zebra_setup_microtype:
88   \__zebra_if_package_loaded:nF { hyperref }
89   {
90     \bool_if:NT \l__zebra_pagelinks_bool
91     {
92       \RequirePackage{hyperref}
93       \hypersetup { pdfborder = { 0~0~0 } }
94     }
95   }

```

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.

```

96 \RequirePackage{xcolor}
97 \RequirePackage{marginnote}
98 \cs_new_eq:NN \__zebra_kernel_label:n \label
99 %% -- dedup data structures --

```

```

100 %% Maps any key (instance, stable, or content-sig) to the allocation.
101 \prop_new:N \g__zebra_note_target_prop
102 \prop_new:N \g__zebra_note_display_prop
103 %% Content signatures of labeled notes from previous pass (.aux).
104 \prop_new:N \g__zebra_note_sig_known_prop
105 %% Content signatures written this pass (dedup aux writes).
106 \prop_new:N \g__zebra_note_sig_written_prop
107 %% Content signature stored per stable key (for label-conflict detection).
108 \prop_new:N \g__zebra_note_stable_sig_prop
109 %% Group-local set of targets whose \label has been emitted. Sbox
110 %% boundaries reset it (so caption replay re-emits after the sbox is
111 %% discarded), while a same-source-line repeat in body text sees the
112 %% prior emission and skips the redundant \label, avoiding
113 %% multiply-defined labels.
114 \prop_new:N \l__zebra_label_emitted_prop
115 \tl_new:N \l__zebra_note_display_tl
116 \tl_new:N \l__zebra_note_key_tl
117 %% Instance key: unique per source location.
118 \cs_new:Npn \__zebra_instance_key:nnn #1#2#3
119 {
120   \tl_to_str:n {#1}
121   | \tl_to_str:n {#2}
122   | \tl_to_str:n {#3}
123   | \int_eval:n { \tex_inputlineno:D }
124 }
125 %% Content signature: body stringified with ALL \label{...} stripped.
126 %% Matches across body (has labels), TOC (labels consumed by \edef),
127 %% and running heads (which may uppercase the author/body text).
128 %% The optional \protect prefix covers marks and \protected@write paths.
129 %% Uses replace_all so that multiple labels are all stripped.
130 \cs_new_protected:Npn \__zebra_content_sig:nnnN #1#2#3#4
131 {
132   \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#3} }
133   \regex_replace_all:nnN
134   { (?: \protect\s* )? \\label\s* \{ [^{}]* \} { } \l_tmpa_tl
135   \tl_set:Nx #4
136   {
137     \tl_to_str:n {#1}
138     | \str_lowercase:f { \tl_to_str:n {#2} }
139     | \str_lowercase:f { \l_tmpa_tl }
140   }
141 }
142 %% Extract the last \label name from the stringified body
143 %% (the greedy \A.* consumes up to the rightmost \label). Both the
144 %% original encounter and any replay run through this same regex,
145 %% so they agree on which label to use as the stable key.
146 %% Sets #2 to the label name, or clears it if none found.
147 %% The optional \protect prefix covers marks and \protected@write paths.
148 \cs_new_protected:Npn \__zebra_extract_label:nN #1#2
149 {
150   \tl_set:Nx \l_tmpb_tl { \tl_to_str:n {#1} }
151   \tl_set_eq:NN \l_tmpc_tl \l_tmpb_tl
152   \regex_replace_once:nnN
153   { \A .* (?: \protect\s* )? \\label\s* \{ ([^{}]*) \} .* \Z }

```

```

154     { \1 } \l_tmpb_tl
155     \tl_if_eq:NNTF \l_tmpb_tl \l_tmpc_tl
156     { \tl_clear:N #2 }
157     { \tl_set_eq:NN #2 \l_tmpb_tl }
158   }
159   %% Stable key for labeled notes.
160   %% #2 is expected to be already stringified (from regex extraction),
161   %% so no \tl_to_str is applied - otherwise an unexpanded variable
162   %% token would be stringified instead of its value.
163   \cs_new:Npn \__zebra_stable_key:nn #1#2
164     { \tl_to_str:n {#1} | label | #2 }
165   %% Allocate a fresh note: increment the type counter, generate
166   %% a unique hypertarget name, and record the note in the list body.
167   \cs_new_protected:Npn \__zebra_allocate_note:nnn #1#2#3
168     {
169       \int_gincr:c { g__zebra_note_count_#1_int }
170       \tl_set:Nx \l__zebra_note_display_tl { \__zebra_note_count:n {#1} }
171       \int_gincr:N \g__zebra_note_id_int
172       \tl_set:Nx \l__zebra_note_target_tl
173         { zebranote.\int_use:N \g__zebra_note_id_int }
174       \__zebra_record_note:nnnnn
175         {#1}
176         { \l__zebra_note_display_tl }
177         {#2}
178         {#3}
179         { \l__zebra_note_target_tl }
180     }
181   %% Aux-file interface: record a content signature together with the
182   %% originating instance key. A later encounter whose instance key
183   %% differs from the stored one is a replay and is suppressed.
184   %% Re-stringify for catcode normalisation.
185   \cs_new_protected:Npn \zebra@sig #1#2
186     {
187       \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#1} }
188       \tl_set:Nx \l_tmpb_tl { \tl_to_str:n {#2} }
189       \prop_gput:NVV \g__zebra_note_sig_known_prop \l_tmpa_tl \l_tmpb_tl
190     }
191   \cs_new_protected:Npn \__zebra_write_sig:NN #1#2
192     {
193       \prop_if_in:NVF \g__zebra_note_sig_written_prop #1
194       {
195         \immediate\write \@auxout
196           { \string\zebra@sig { \tl_use:N #1 } { \tl_use:N #2 } }
197         \prop_gput:NVN \g__zebra_note_sig_written_prop #1 { 1 }
198       }
199     }
200   \cs_if_exist:NTF \dbend
201     {
202       \cs_set_eq:NN \__zebra_saved_dbend: \dbend
203       \cs_undefine:N \dbend
204       \RequirePackage{manfnt}
205       \cs_set_eq:NN \dbend \__zebra_saved_dbend:
206     }
207   { \RequirePackage{manfnt} }

```

```

208 \cs_new:Npn \__zebra_pdfstring_note:
209   #1
210   {
211     \str_if_eq:eeT { \tl_to_str:n {#1} } { [ ] }
212     { \__zebra_pdfstring_note_opt:w }
213   }
214 \cs_new:Npn \__zebra_pdfstring_note_opt:w #1 ] #2 { }
215 \cs_new:Npn \__zebra_target:nn #1#2 {#2}
216 \cs_new:Npn \__zebra_link:nn #1#2 {#2}
217 \cs_new:Npn \__zebra_pageref:n #1 { \pageref {#1} }
218 \cs_new:Npn \__zebra_zebra_label_name:n #1 { #1@zebra }
219 \cs_new:Npn \__zebra_zebra_label_type:n #1
220   {
221     \exp_after:wN \use_i:nn
222     \cs:w r@\__zebra_zebra_label_name:n {#1}\cs_end:
223     { }
224   }
225 \cs_new_protected:Npn \__zebra_write_zebra_label:n #1
226   {
227     \protected@write \@auxout { }
228     {
229       \string\newlabel{\__zebra_zebra_label_name:n {#1}}
230       {{\exp_not:V \l__zebra_note_ref_type_tl}{}}
231     }
232   }
233 \cs_new_protected:Npn \__zebra_note_label:n #1
234   {
235     \__zebra_kernel_label:n {#1}
236     \__zebra_write_zebra_label:n {#1}
237   }
238 \cs_new_protected:Npn \__zebra_zebra_ref:n #1
239   {
240     \cs_if_exist:cTF { r@\__zebra_zebra_label_name:n {#1} }
241     { \__zebra_zebra_label_type:n {#1}~\ref{#1} }
242     { \ref{#1} }
243   }
244 \NewDocumentCommand \zebraref { m }
245   {
246     \bool_if:NTF \l__zebra_unnumbered_bool
247     { \ref{#1} }
248     { \__zebra_zebra_ref:n {#1} }
249   }
250 \cs_new_protected:Npn \__zebra_apply_pdfstring_defs:
251   {
252     \pdfstringdefDisableCommands
253     {
254       \cs_set:Npn \zebraref ##1 { \ref{##1} }
255       \seq_map_inline:Nn \g__zebra_note_types_seq
256       {
257         \cs_set_eq:cN { zebra##1 } \__zebra_pdfstring_note:
258         \prop_if_in:NnT \g__zebra_note_public_alias_prop { ##1 }
259         { \cs_set_eq:cN { ##1 } \__zebra_pdfstring_note: }
260       }
261     }

```

```

262 }
263 \cs_new_protected:Npn \__zebra_setup_pagelinks:
264 {
265   \cs_set:Npn \__zebra_target:nn ##1##2 {##2}
266   \cs_set:Npn \__zebra_link:nn ##1##2 {##2}
267   \cs_set:Npn \__zebra_pageref:n ##1 { \pageref {##1} }
268   \__zebra_if_package_loaded:nT { hyperref }
269   {
270     \cs_set:Npn \__zebra_pageref:n ##1 { \pageref* {##1} }
271     \bool_if:NT \l__zebra_pagelinks_bool
272     {
273       \cs_set:Npn \__zebra_target:nn ##1##2 { \hypertarget{##1}{##2} }
274       \cs_set:Npn \__zebra_link:nn ##1##2 { \hyperlink{##1}{##2} }
275     }
276     \__zebra_apply_pdfstring_defs:
277   }
278 }
279 \hook_gput_code:nnn { begindocument } { zebra }
280 { \__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`.

```

281 \cs_new_protected:Npn \__zebra_margin_note:nn #1#2
282 {
283   \legacy_if:nTF { @twocolumn }
284   {
285     \bool_lazy_or:nnTF
286     { \mode_if_inner_p: }
287     {
288       \bool_lazy_or_p:nn
289       { \cs_if_exist_p:N \@capttype }
290       { \dim_compare_p:nNn { \hsize } > { \columnwidth } }
291     }
292     {
293       \marginnote
294       [ { \@parboxrestore \normalfont \normalsize #1 } ]
295       { \@parboxrestore \normalfont \normalsize #2 }
296     }
297     {
298       \marginpar
299       [ { \makebox[\marginparwidth][r]{#1} } ]

```

```

300         { \makebox[\marginparwidth][l]{#2} }
301     }
302 }
303 { \marginnote[#1]{#2} }
304 }
305 \cs_new:Npn \__zebra_prepend:nn #1#2
306 { \tl_if_blank:nTF {#2} {} {#1#2} }
307 \cs_new:Npn \__zebra_capitalize_type:n #1
308 { \text_uppercase:n { \tl_head:n {#1} } \tl_tail:n {#1} }
309 \cs_new:Npn \__zebra_note_count:n #1
310 { \int_use:c { g__zebra_note_count_#1_int } }
311 \cs_new:Npn \__zebra_note_color:n #1
312 { \prop_item:Nn \g__zebra_note_colors_prop {#1} }
313 \cs_new:Npn \__zebra_note_symbol:n #1
314 {
315     \prop_if_in:NnTF \g__zebra_note_symbols_prop {#1}
316     { \prop_item:Nn \g__zebra_note_symbols_prop {#1} }
317     { \l__zebra_symbol_tl }
318 }
319 \cs_new_protected:Npn \__zebra_new_listbody:n #1
320 { \tl_new:c { g__zebra_listbody_#1_tl } }
321 \tl_new:N \g__zebra_listbody_all_tl
322 \cs_new:Npn \__zebra_use_listbody:n #1
323 { \tl_use:c { g__zebra_listbody_#1_tl } }
324 \cs_new_protected:Npn \__zebra_record_note:nnnnn #1#2#3#4#5
325 {
326     \tl_gput_right:cx
327     {
328         \bool_if:NTF \l__zebra_sort_none_bool
329         { g__zebra_listbody_all_tl }
330         { g__zebra_listbody_#1_tl }
331     }
332     {
333         \exp_not:N \__zebra_list_entry:nnnnn
334         { \exp_not:n {#1} }
335         {#2}
336         { \exp_not:n {#3} }
337         { \exp_not:n {#4} }
338         {#5}
339     }
340 }
341 %% \__zebra_note_unnumbered:nnn {type}{author}{body}
342 %% Fast path for the \opt{unnumbered} option: skip dedup, .aux
343 %% signatures and list registration entirely. The counter is still
344 %% bumped on every call (so the summary table reflects rough counts;
345 %% notes inside captions/TOC/marks may be over-counted, which is the
346 %% accepted trade-off for cutting out the fragile bookkeeping).
347 \cs_new_protected:Npn \__zebra_note_unnumbered:nnn #1#2#3
348 {
349     \bool_if:NT \l__zebra_draft_bool
350     {
351         \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
352         \int_gincr:c { g__zebra_note_count_#1_int }
353         \__zebra_render_note:nnn {#1} {#2} {#3}

```

```

354     }
355   }
356   %% \__zebra_note:nnn {type}{author}{body}
357   %% Main entry point: dispatch to the lightweight unnumbered path or to
358   %% the numbered path with deduplication/bookkeeping.
359   \cs_new_protected:Npn \__zebra_note:nnn #1#2#3
360   {
361     \bool_if:NTF \l__zebra_unnumbered_bool
362     { \__zebra_note_unnumbered:nnn {#1} {#2} {#3} }
363     { \__zebra_note_numbered:nnn {#1} {#2} {#3} }
364   }
365   %% Numbered path. Four cases:
366   %% Case 1 - sbox reuse: instance_key found → reuse, render
367   %% Case 2 - stable key: label found, stable_key in prop → suppress
368   %% Case 3 - content sig: sig in .aux data → suppress
369   %% Case 4 - new note: allocate, render
370   %% Case 1 also catches an unlabeled same-source-line same-body pair
371   %% (\todo{x}\todo{x}); the second is collapsed. Two such notes are
372   %% indistinguishable at TeX level from a single note replayed by an
373   %% sbox/moving argument, so this collision is documented and resolved
374   %% by adding \label or splitting the line.
375   \cs_new_protected:Npn \__zebra_note_numbered:nnn #1#2#3
376   {
377     \bool_if:NT \l__zebra_draft_bool
378     {
379       \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
380       %% Case 1: sbox reuse (same \inputlineno)
381       \tl_set:Nx \l__zebra_note_key_tl
382       { \__zebra_instance_key:nnn {#1} {#2} {#3} }
383       \prop_get:NVNTF \g__zebra_note_target_prop \l__zebra_note_key_tl
384       \l__zebra_note_target_tl
385       {
386         \prop_get:NVN \g__zebra_note_display_prop \l__zebra_note_key_tl
387         \l__zebra_note_display_tl
388         \__zebra_render_note:nnn {#1} {#2} {#3}
389       }
390     }
391     {
392       %% Extract label and compute content signature
393       \__zebra_extract_label:nN {#3} \l_tmpb_tl
394       \__zebra_content_sig:nnnN {#1} {#2} {#3} \l_tmpa_tl
395       %% Case 2: stable-key suppress (labeled, marks/headers).
396       %% If the stable key exists AND the content signature
397       %% matches, this encounter is a replay → suppress.
398       %% Different content signature = label reuse → warn and
399       %% let Case 4 allocate independently.
400       \bool_set_false:N \l_tmpa_bool
401       \tl_if_empty:NF \l_tmpb_tl
402       {
403         \tl_set:Nx \l__zebra_note_key_tl
404         { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
405         \prop_get:NVNT \g__zebra_note_stable_sig_prop
406         \l__zebra_note_key_tl \l_tmpc_tl
407         {
408           \tl_if_eq:NNTF \l_tmpa_tl \l_tmpc_tl

```

```

408         { \bool_set_true:N \l_tmpa_bool }
409         {
410             \msg_warning:nnV { zebra }
411             { duplicate-note-label } \l_tmpb_tl
412         }
413     }
414 }
415 %% Case 3: content-sig suppress (TOC/LOF/marks replay).
416 %% Only unlabeled current encounters may be suppressed by
417 %% content signature; labeled encounters already have a
418 %% strong identity via the stable key. Suppress only if
419 %% the stored instance key differs from the current one -
420 %% same key means it is the original note, not a replay.
421 \bool_if:NF \l_tmpa_bool
422 {
423     \tl_if_empty:NT \l_tmpb_tl
424     {
425         \prop_get:NVNT \g__zebra_note_sig_known_prop
426         \l_tmpa_tl \l_tmpc_tl
427         {
428             \tl_set:Nx \l_tmpd_tl
429             { \__zebra_instance_key:nnn {#1} {#2} {#3} }
430             \tl_if_eq:NNF \l_tmpc_tl \l_tmpd_tl
431             { \bool_set_true:N \l_tmpa_bool }
432         }
433     }
434 }
435 \bool_if:NF \l_tmpa_bool
436 {
437     %% Case 4: new note - allocate and render
438     \tl_set:Nx \l__zebra_note_key_tl
439     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
440     \__zebra_allocate_note:nnn {#1} {#2} {#3}
441     \prop_gput:NVV \g__zebra_note_target_prop
442     \l__zebra_note_key_tl \l__zebra_note_target_tl
443     \prop_gput:NVV \g__zebra_note_display_prop
444     \l__zebra_note_key_tl \l__zebra_note_display_tl
445     %% For labeled notes: register stable key + write sig,
446     %% but only if the stable key is not already claimed by
447     %% an earlier note (label-conflict case).
448     \tl_if_empty:NF \l_tmpb_tl
449     {
450         \tl_set:Nx \l__zebra_note_key_tl
451         { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
452         \prop_if_in:NVF \g__zebra_note_stable_sig_prop
453         \l__zebra_note_key_tl
454         {
455             \prop_gput:NVV \g__zebra_note_target_prop
456             \l__zebra_note_key_tl \l__zebra_note_target_tl
457             \prop_gput:NVV \g__zebra_note_display_prop
458             \l__zebra_note_key_tl \l__zebra_note_display_tl
459             \prop_gput:NVV \g__zebra_note_stable_sig_prop
460             \l__zebra_note_key_tl \l_tmpa_tl
461             \tl_set:Nx \l_tmpc_tl

```



```

462         { \_zebra_instance_key:nnn {#1} {#2} {#3} }
463         \_zebra_write_sig:NN \l_tmpa_tl \l_tmpc_tl
464     }
465 }
466     \_zebra_render_note:nnn {#1} {#2} {#3}
467 }
468 %% Cases 2-3: suppress - no output
469 }
470 }
471 }
472 %% Full render: hypertarget, target label, margin note, inline text.
473 %% In \opt{unnumbered} mode the label/hypertarget setup is skipped so a
474 %% \cs{label} inside the note body falls through to the surrounding
475 %% \cs{@currentlabel} (e.g.\ the section number), and \cs{zebraref}
476 %% degrades cleanly to \cs{ref}.
477 \cs_new_protected:Npn \_zebra_render_note:nnn #1#2#3
478 {
479     %% Decide outside render's group whether to emit the kernel label.
480     %% The flag lives in the caller's group: sbox/parbox boundaries reset
481     %% it (so caption replay re-emits), but a same-line repeat in body
482     %% text sees the prior emission and skips the redundant \label.
483     \bool_set_false:N \l__zebra_emit_label_bool
484     \bool_if:NF \l__zebra_unnumbered_bool
485     {
486         \prop_if_in:NVF \l__zebra_label_emitted_prop \l__zebra_note_target_tl
487         {
488             \bool_set_true:N \l__zebra_emit_label_bool
489             \prop_put:NVn \l__zebra_label_emitted_prop
490                 \l__zebra_note_target_tl { 1 }
491         }
492     }
493     \group_begin:
494     \bool_if:NF \l__zebra_unnumbered_bool
495     {
496         \protected@edef \@currentlabel { \l__zebra_note_display_tl }
497         \_zebra_if_package_loaded:nT { hyperref }
498         { \tl_set:Nx \@currentHref { \l__zebra_note_target_tl } }
499         \tl_set:Nx \l__zebra_note_ref_type_tl
500             { \_zebra_capitalize_type:n {#1} }
501         \_zebra_target:nn { \l__zebra_note_target_tl } {}
502         \bool_if:NT \l__zebra_emit_label_bool
503         { \exp_args:NV \_zebra_kernel_label:n \l__zebra_note_target_tl }
504     }
505     \_zebra_margin_note:nn
506     { \textcolor{\l__zebra_note_color_tl}{%
507         \bool_if:NF \l__zebra_unnumbered_bool
508         { {\bfseries\l__zebra_note_display_tl}\kern1pt }%
509         \_zebra_note_symbol:n {#1}} }
510     { \textcolor{\l__zebra_note_color_tl}{%
511         \_zebra_note_symbol:n {#1}%
512         \bool_if:NF \l__zebra_unnumbered_bool
513         { \kern1pt {\bfseries\l__zebra_note_display_tl} } } }%
514     \bool_if:NF \l__zebra_unnumbered_bool
515     { \cs_set_eq:NN \label \_zebra_note_label:n }

```

```

516     \textcolor{\l__zebra_note_color_tl}{[\colorbox[gray]{0.97}{%
517       \textcolor{\l__zebra_note_color_tl !70!black}{%
518         \textsc{\MakeLowercase{\MakeUppercase#1}}}%
519         \bool_if:NF \l__zebra_unnumbered_bool
520         { ~\l__zebra_note_display_tl }%
521         \texttt{\__zebra_prepend:nn {0}{#2}:}} #3}}%
522   \group_end:
523 }
524 \cs_new_protected:Npn \__zebra_new_note_type:nn #1#2
525 { \__zebra_new_note_type:nnn {#1} {#2} {} }
526 \cs_new_protected:Npn \__zebra_new_note_type:nnn #1#2#3
527 {
528   \seq_gput_right:Nn \g__zebra_note_types_seq {#1}
529   \prop_if_in:NnF \g__zebra_note_colors_prop {#1}
530   { \prop_gput:Nnn \g__zebra_note_colors_prop {#1} {#2} }
531   \tl_if_blank:nF {#3}
532   {
533     \prop_if_in:NnF \g__zebra_note_symbols_prop {#1}
534     { \prop_gput:Nnn \g__zebra_note_symbols_prop {#1} {#3} }
535   }
536   \int_new:c { g__zebra_note_count_#1_int }
537   \__zebra_new_listbody:n {#1}
538   \exp_args:Nc \NewDocumentCommand { zebra#1 } { 0{ } m }
539   { \__zebra_note:nnn {#1}{##1}{##2} }
540   \cs_if_exist:cTF {#1}
541   { \msg_warning:nnn { zebra } { command-taken } {#1} }
542   {
543     \cs_set_eq:cc {#1} {zebra#1}
544     \prop_gput:Nnn \g__zebra_note_public_alias_prop {#1} { true }
545   }
546   \__zebra_if_package_loaded:nT { hyperref }
547   { \__zebra_apply_pdfstring_defs: }
548 }
549 \cs_new_protected:Npn \__zebra_list_entry:nnnnn #1#2#3#4#5
550 {
551   \par\noindent
552   \textcolor{\__zebra_note_color:n {#1}}{%
553     \textbf{\__zebra_capitalize_type:n {#1}~#2}%
554     \tl_if_blank:nF {#3} { \enspace \texttt{\__zebra_prepend:nn {0}{#3}} }}%
555   \nobreak\dotfill
556   \__zebra_link:nn {#5} { \__zebra_pageref:n {#5} }%
557   \par
558   \begingroup
559     \leftskip=2em
560     \rightskip=2em
561     \parindent=0pt
562     \cs_set_eq:NN \label \use_none:n
563     #4\par
564   \endgroup
565 }
566 \cs_new_protected:Npn \__zebra_print_note_group:n #1
567 {
568   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
569   {

```

```

570     \par\medskip
571     \_zebra_use_listbody:n {#1}
572   }
573 }
574 \cs_new_protected:Npn \_zebra_print_notes_inorder:
575 {
576   \tl_if_empty:NF \g_zebra_listbody_all_tl
577   { \par\medskip \tl_use:N \g_zebra_listbody_all_tl }
578 }
579 \cs_new_protected:Npn \_zebra_summary_row:n #1
580 {
581   \int_compare:nNnT { \_zebra_note_count:n {#1} } > { 0 }
582   {
583     \int_add:Nn \l_zebra_total_notes_int { \_zebra_note_count:n {#1} }
584     \tl_put_right:Nx \l_zebra_summary_rows_tl
585     {
586       \exp_not:N \textcolor
587       { \_zebra_note_color:n {#1} }
588       {#1}
589       \exp_not:N &
590       \_zebra_note_count:n {#1}
591       \exp_not:N \\
592     }
593   }
594 }
595 \cs_new_protected:Npn \_zebra_print_notes:
596 {
597   \tl_clear:N \l_zebra_summary_rows_tl
598   \int_zero:N \l_zebra_total_notes_int
599   \seq_map_inline:Nn \g_zebra_note_types_seq
600   { \_zebra_summary_row:n {##1} }
601   \tl_if_empty:NF \l_zebra_summary_rows_tl
602   {
603     \par\nobreak
604     \noindent\dotfill\par\medskip
605     \nobreak
606     \noindent\textbf{\Large Zebra~Notes}
607     \par \medskip
608     \begin{center}
609       \begin{tabular}{lr}
610         \hline
611         \textbf{Type} & \textbf{Count} \\ \hline
612         \tl_use:N \l_zebra_summary_rows_tl
613         \hline
614         \textbf{Total} & \textbf{\int_use:N \l_zebra_total_notes_int} \\ \hline
615         \hline
616       \end{tabular}
617     \end{center}
618     \bool_if:NF \l_zebra_unnumbered_bool
619     {
620       \par \medskip
621       \begin{group}
622         \small
623         \noindent{\bfseries List-of~notes}\par

```

```

624         \nobreak
625         \bool_if:NTF \l__zebra_sort_none_bool
626           { \__zebra_print_notes_inorder: }
627         {
628           \seq_map_inline:Nn \g__zebra_note_types_seq
629             { \__zebra_print_note_group:n {##1} }
630         }
631       \endgroup
632     }
633   }
634 }

```

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

```

635 \NewDocumentCommand \zebranewnote { m m O{} }
636   { \__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.

```

637 \NewDocumentCommand \zebrasetup { m }
638   { \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
\fixed
\comment
\placeholder

```

639 \zebranewnote{todo}{purple}
640 \colorlet{zebra@fixed@color}{green!50!black}
641 \zebranewnote{fixed}{zebra@fixed@color}
642 \zebranewnote{comment}{blue}
643 \zebranewnote{note}{violet}
644 \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.

```

645 %% At end of document: print the note summary and list.
646 %% Content signatures are written to .aux inline (at allocation time),
647 %% so no additional end-of-document aux writes are needed.
648 \hook_gput_code:nnn { enddocument } { zebra }
649 {
650   \bool_if:NT \l__zebra_draft_bool
651     { \__zebra_print_notes: }
652 }
653 \ExplSyntaxOff
654 \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.

```
655 <*compat>
656 \NeedsTeXFormat{LaTeX2e}
657 \ProvidesPackage{zebra-goodies}
658   [2026/05/01 v1.9.0 Deprecated: use zebra instead]
659 \PackageWarningNoLine{zebra-goodies}
660   {Package 'zebra-goodies' is deprecated.\MessageBreak
661   Use \string\usepackage{zebra} instead}
662 \RequirePackageWithOptions{zebra}
663 </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.

```
664 <*demo-twocol>
665 \documentclass[twocolumn]{article}
666 \usepackage[paperwidth=21cm,paperheight=15cm,margin=1.1cm]{geometry}
667 \usepackage{zebra}
668 \zebrasetup{symbol/comment=$\clubsuit$}
669 \pagestyle{empty}
670 \begin{document}
671 \section{Demo name\comment{revise the name}}
672 This draft still needs work
673 \todo[alice]{\label{zebra:intro}revise the introduction}. The
674 opening paragraph should also explain the main goal more plainly.
675 Add one more citation here \note[bob]{support this claim}. A
676 brief roadmap sentence would also make the structure easier to
677 scan.
678
679 The issue raised in Todo-\ref{zebra:intro} still applies in
680 the conclusion. The table now looks fine
681 \fixed[carol]{alignment corrected}, but one figure is still
682 missing \placeholder[eve]{insert the overview figure}. A short
683 transition would also help the flow. The middle section should
684 probably end with a clearer summary sentence before the
685 discussion begins. That summary can stay compact, but it should
686 signal why the next section matters.
687
688 Please verify the totals \note[frank]{check the numbers} and
689 confirm the wording in the last paragraph
690 \comment[tom]{is this sentence too strong?}. A small typo has
691 already been fixed \fixed[heidi]{typo corrected}. The ending
692 should stay short. The final sentence should return to the main
693 claim rather than repeat background material. You can place
694 \todo[judy]{summarise the findings} anywhere once the narrative
695 is stable.
696
697 One more short paragraph is enough to show how \placeholder{wow,
698 so great!} the markers stay readable in a compact two-column layout.
```

```

699 The example is intentionally small, but it should still look like
700 a realistic revision pass.\comment{Bye}
701 \end{document}
702 \demo-twocol)

```

Change History

v0.1.0	General: Initial public release	1	v1.2.0	General: Simplify key architecture.	1
v0.2.0	General: Fix <code>xcolor</code> conflict	1	v1.3.0	General: Rename package to <code>zebra</code>	1
v0.3.0	General: Detect command conflicts	1		Rename the microtype expansion option to <code>font-expansion</code>	1
v0.4.0	General: Show note number for easy reference	1		Rename the page-link option to <code>pagelinks/nopagelinks</code>	1
v0.5.0	General: Use darker color for label	1	v1.4.0	General: Support note labels via <code>\label</code> , <code>\ref</code> , and <code>\zebraref</code>	1
v0.6.0	General: Use gray background for label	1	v1.5.0	General: Fix notes numbering in moving arguments.	1
v0.7.0	General: Move to <code>docstrip</code>	2	v1.6.0	General: Numbering in moving arguments is hard.	1
v0.8.0	General: Fix new note demo	4		General: Robust margin notes in twocolumn captions, titles and headings.	1
	<code>\zebranewnote</code> : Fix on <code>\global</code> for examples	20	v1.7.0	General: Add <code>unnumbered</code> option for lightweight mode.	1
v0.8.1	General: Fix doc	4		Fix margin symbol distance.	1
v0.9.0	General: Fix legacy bugs and improve implementation	1		Stabilise labeled notes replayed through uppercased running heads.	1
v0.9.1	General: Beautify the numbers.	1	v1.8.0	General: Drop note bodies from PDF bookmarks.	1
v0.9.2	General: Faster.	1		Fix multiply-defined labels for repeated unlabeled notes.	1
v1.0.0	General: <code>expl3</code> , list of notes and compatibility.	1		Keep labeled notes with shared content distinct.	1
v1.1.0	General: Customisable margin symbol, accurate page numbers, code cleanup.	1		Silence pass-1 warning from <code>\zebraref</code>	1
v1.1.1	General: Per-type <code>color/symbol</code> keys, <code>\zebrasetup</code>	1	v1.8.1	General: Document known limitations.	1
			v1.9.0		

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.

Symbols		
<code>\\</code>	28, 29, 134, 153, 591, 611, 614	<code>\cs_set:Npn</code>
<code>\{</code>	134, 153 254, 265, 266, 267, 270, 273, 274
<code>\}</code>	134, 153	<code>\cs_set_eq:NN</code>
<code>_</code>	475 202, 205, 257, 259, 515, 543, 562
		<code>\cs_undefine:N</code>
		203
Numbers		D
<code>\1</code>	154	<code>\dbend</code>
		200, 202, 203, 205
A		<code>\definecolor</code>
<code>\A</code>	143, 153	4
		dim commands:
B		<code>\dim_compare_p:nNn</code>
<code>\begin</code>	608, 609, 670	290
<code>\begingroup</code>	558, 621	<code>\documentclass</code>
<code>\bfseries</code>	508, 513, 623	665
bool commands:		<code>\dotfill</code>
<code>\bool_if:NTF</code> 78, 82, 90, 246, 271, 328,		555, 604
349, 361, 377, 421, 435, 484, 494,		<code>draft</code> (option)
502, 507, 512, 514, 519, 618, 625, 650		2
<code>\bool_lazy_or:nmTF</code>	285	E
<code>\bool_lazy_or_p:nn</code>	288	<code>\edef</code>
<code>\bool_new:N</code>	7, 8, 9, 10, 11, 12	9, 126
<code>\bool_set_false:N</code> ...	54, 72, 399, 483	<code>\end</code>
<code>\bool_set_true:N</code> ...	55, 408, 431, 488	616, 617, 701
<code>\l_tmpa_bool</code> ..	399, 408, 421, 431, 435	<code>\endgroup</code>
		564, 631
		<code>\enspace</code>
		554
		exp commands:
C		<code>\exp_after:wN</code>
<code>\caption</code>	3, 6, 9	221
<code>\clubsuit</code>	668	<code>\exp_args:Nc</code>
<code>\cmd</code>	3	538
<code>\colorbox</code>	516	<code>\exp_args:NV</code>
<code>\colorlet</code>	4, 640	503
<code>\columnwidth</code>	13, 290	<code>\exp_not:N</code>
<code>\comment</code>	1, 3, 4, <u>639</u> , 671, 690, 700	333, 586, 589, 591
<code>\cs</code>	474, 475, 476	<code>\exp_not:n</code>
cs commands:		230, 334, 336, 337
<code>\cs:w</code>	222	<code>\ExplSyntaxOff</code>
<code>\cs_end:</code>	222	653
<code>\cs_if_exist:NTF</code>	37, 200, 240, 540	F
<code>\cs_if_exist_p:N</code>	289	<code>\fbox</code>
<code>\cs_new:Npn</code>		9
118, 163, 208, 214, 215, 216, 217,		<code>final</code> (option)
218, 219, 305, 307, 309, 311, 313, 322		2
<code>\cs_new_eq:NN</code>	98	<code>\fixed</code>
<code>\cs_new_protected:Npn</code> 74, 130, 148,		1, 4, <u>639</u> , 681, 691
167, 185, 191, 225, 233, 238, 250,		<code>font-expansion</code> (option)
263, 281, 319, 324, 347, 359, 375,		2
477, 524, 526, 549, 566, 574, 579, 595		G
		<code>\global</code>
		22
		group commands:
		<code>\group_begin:</code>
		493
		<code>\group_end:</code>
		522
		H
		<code>\hline</code>
		610, 611, 613, 615
		hook commands:
		<code>\hook_gput_code:nmn</code>
		279, 648
		<code>\hsize</code>
		13, 290
		<code>\Huge</code>
		13
		<code>\hyperlink</code>
		274
		<code>\hypersetup</code>
		93
		<code>\hypertarget</code>
		273

I		O	
<code>\immediate</code>	195	<code>\opt</code>	342, 473
<code>\inputlineno</code>	9, 380	options:	
int commands:		<code>draft</code>	2
<code>\int_add:Nn</code>	583	<code>final</code>	2
<code>\int_compare:nNnTF</code>	568, 581	<code>font-expansion</code>	2
<code>\int_eval:n</code>	123	<code>pagelinks</code>	2
<code>\int_gincr:N</code>	169, 171, 352	<code>sort</code>	2
<code>\int_new:N</code>	16, 21, 536	<code>unnumbered</code>	2
<code>\int_use:N</code>	173, 310, 614		
<code>\int_zero:N</code>	598	P	
iow commands:		<code>\PackageWarningNoLine</code>	659
<code>\iow_char:N</code>	28, 29	<code>pagelinks (option)</code>	2
K		<code>\pageref</code>	217, 267, 270
<code>\kern</code>	508, 513	<code>\pagestyle</code>	669
keys commands:		<code>\par</code>	551, 557, 563, 570, 577, 603, 604, 607, 620, 623
<code>\keys_define:nn</code>	42, 61, 66	<code>\parbox</code>	9
<code>\l_keys_key_str</code>	64, 69	<code>\parindent</code>	561
<code>\keys_set:nn</code>	638	<code>\pdfstringdefDisableCommands</code>	252
L		<code>\placeholder</code>	1, 4, 639, 682, 697
<code>\label</code>	3, 5–7, 9, 22, 98, 109, 112, 125, 142, 143, 374, 482, 515, 562, 673	prg commands:	
<code>\Large</code>	606	<code>\prg_new_conditional:Npnn</code>	35
<code>\leftskip</code>	559	<code>\prg_return_false:</code>	39
legacy commands:		<code>\prg_return_true:</code>	38
<code>\legacy_if:nTF</code>	283	<code>\ProcessKeyOptions</code>	60
M		prop commands:	
<code>\makebox</code>	299, 300	<code>\prop_get:NnN</code>	386
<code>\MakeLowercase</code>	518	<code>\prop_get:NnNTF</code>	383, 404, 425
<code>\MakeUppercase</code>	518	<code>\prop_gput:Nnn</code> ..	64, 69, 189, 197, 441, 443, 455, 457, 459, 530, 534, 544
<code>\marginnote</code>	13, 293, 303	<code>\prop_if_in:NnTF</code>	
<code>\marginpar</code>	13, 298	193, 258, 315, 452, 486, 529, 533
<code>\marginparwidth</code>	299, 300	<code>\prop_item:Nn</code>	312, 316
<code>\medskip</code>	570, 577, 604, 607, 620	<code>\prop_new:N</code>	14, 15, 22, 101, 102, 104, 106, 108, 114
<code>\MessageBreak</code>	660	<code>\prop_put:Nnn</code>	489
<code>\microtypesetup</code>	79	<code>\protect</code>	128, 147
mode commands:		<code>\ProvidesExplPackage</code>	3
<code>\mode_if_inner_p:</code>	286	<code>\ProvidesPackage</code>	657
msg commands:		Q	
<code>\msg_new:nnn</code>	26, 31	<code>\question</code>	4
<code>\msg_warning:nnn</code>	410, 541	R	
N		<code>\ref</code>	2, 5, 22, 241, 242, 247, 254, 679
<code>\NeedsTeXFormat</code>	656	regex commands:	
<code>\NewDocumentCommand</code> ..	244, 538, 635, 637	<code>\regex_replace_all:nnN</code>	133
<code>\newlabel</code>	229	<code>\regex_replace_once:nnN</code>	152
<code>\nobreak</code>	555, 603, 605, 624	<code>\RequirePackage</code> 83, 84, 92, 96, 97, 204, 207	
<code>\noindent</code>	551, 604, 606, 623	<code>\RequirePackageWithOptions</code>	662
<code>\normalfont</code>	13, 294, 295	<code>\rightskip</code>	560
<code>\normalsize</code>	13, 294, 295	S	
<code>\note</code>	1, 3, 639, 675, 688	<code>\s</code>	134, 153

<code>\sbox</code>	9	<code>\tl_to_str</code>	161
<code>\section</code>	3, 6, 9, 671	<code>\tl_to_str:n</code>	120, 121, 122, 132, 137, 138, 150, 164, 187, 188, 211
seq commands:		<code>\tl_use:N</code>	196, 323, 577, 612
<code>\seq_gput_right:Nn</code>	528	<code>\l_tmpa_tl</code>	132, 134, 139, 187, 189, 393, 407, 426, 460, 463
<code>\seq_map_inline:Nn</code>	255, 599, 628	<code>\l_tmpb_tl</code>	150, 151, 154, 155, 157, 188, 189, 392, 400, 403, 411, 423, 448, 451
<code>\seq_new:N</code>	13	tmpc commands:	
<code>\small</code>	622	<code>\l_tmpc_tl</code>	151, 155, 405, 407, 426, 430, 461, 463
sort (option)	2	tmpd commands:	
str commands:		<code>\l_tmpd_tl</code>	428, 430
<code>\str_if_eq:nnTF</code>	211	<code>\todo</code>	1, 3, 4, 9, 13, 371, 639, 673, 694
<code>\str_lowercase:n</code>	138, 139		
<code>\string</code>	196, 229, 661		
sys commands:			
<code>\sys_if_engine_xetex:TF</code>	71		
T			
T _E X and L ^A T _E X 2 _ε commands:			
<code>\@auxout</code>	195, 227	U	
<code>\@capytype</code>	13, 289	unnumbered (option)	2
<code>\@currentHref</code>	498	use commands:	
<code>\@currentlabel</code>	496	<code>\use_i:nn</code>	221
<code>\@makecaption</code>	9	<code>\use_none:n</code>	562
<code>\@marginparreset</code>	13	<code>\usepackage</code>	661, 666, 667
<code>\@parboxrestore</code>	13, 294, 295		
<code>\@topnewpage</code>	13	V	
<code>\if@firstcolumn</code>	13	<code>\vbox</code>	13
<code>\protected@edef</code>	496	W	
<code>\protected@write</code>	9, 128, 147, 227	<code>\write</code>	195
<code>\zebra@sig</code>	185, 196	Z	
tex commands:		<code>\Z</code>	153
<code>\tex_inputlineno:D</code>	123	zebra internal commands:	
text commands:		<code>_zebra_allocate_note:nnn</code> ..	167, 440
<code>\text_uppercase:n</code>	308	<code>_zebra_apply_pdfstring_defs:..</code>	250, 276, 547
<code>\textbf</code>	553, 606, 611, 614	<code>_zebra_capitalize_type:n</code>	307, 500, 553
<code>\textcolor</code>	506, 510, 516, 517, 552, 586	<code>_zebra_content_sig:nnnN</code> ..	130, 393
<code>\textdbend</code>	1, 4, 24	<code>\l_zebra_draft_bool</code> 7, 44, 349, 377, 650	
<code>\textsc</code>	518	<code>\l_zebra_emit_label_bool</code>	8, 483, 488, 502
<code>\texttt</code>	521, 554	<code>_zebra_extract_label:nN</code> ..	148, 392
<code>\title</code>	13	<code>_zebra_if_package_loaded:n</code> ..	35
tl commands:		<code>_zebra_if_package_loaded:nTF</code> ..	76, 88, 268, 497, 546
<code>\tl_clear:N</code>	156, 597	<code>_zebra_instance_key:nnn</code>	118, 382, 429, 439, 462
<code>\tl_gput_right:Nn</code>	326	<code>_zebra_kernel_label:n</code> ..	98, 235, 503
<code>\tl_head:n</code>	308	<code>\l_zebra_label_emitted_prop</code> ..	114, 486, 489
<code>\tl_if_blank:nTF</code>	306, 531, 554	<code>_zebra_link:nn</code> ..	216, 266, 274, 556
<code>\tl_if_empty:NTF</code> 400, 423, 448, 576, 601		<code>_zebra_list_entry:nnnnn</code> ..	333, 549
<code>\tl_if_eq:NNTF</code>	155, 407, 430	<code>\g_zebra_listbody_all_tl</code>	321, 576, 577
<code>\tl_new:N</code>	17, 18, 19, 20, 23, 115, 116, 320, 321	<code>_zebra_margin_note:nn</code>	281, 505
<code>\tl_put_right:Nn</code>	584		
<code>\tl_set:Nn</code>	24, 132, 135, 150, 170, 172, 187, 188, 351, 379, 381, 402, 428, 438, 450, 461, 498, 499		
<code>\tl_set_eq:NN</code>	151, 157		
<code>\tl_tail:n</code>	308		

\l_zebra_microtype_expansion_	\l_zebra_pagelinks_bool
bool 10, 50, 90, 271
_zebra_new_listbody:n	_zebra_pageref:n
..... 319, 537	. 217, 267, 270, 556
_zebra_new_note_type:nn	_zebra_pdfstring_note:
..... 524	208, 257, 259
_zebra_new_note_type:nnn	_zebra_pdfstring_note_opt:w
..... 525, 526, 636 212, 214
_zebra_note:nnn	_zebra_prepend:nn
..... 356, 359, 539	... 305, 521, 554
_zebra_note_color:n	_zebra_print_note_group:n
..... 311, 351, 379, 552, 587	566, 629
\l_zebra_note_color_tl	_zebra_print_notes:
..... 18, 351, 379, 506, 510, 516, 517 595, 651
\g_zebra_note_colors_prop	_zebra_print_notes_inorder:
..... 14, 64, 312, 529, 530 574, 626
_zebra_note_count:n	_zebra_record_note:nnnnn
..... 170, 309, 568, 581, 583, 590	. 174, 324
\g_zebra_note_display_prop	_zebra_render_note:nnn
..... 102, 386, 443, 457 353, 388, 466, 477
\l_zebra_note_display_tl	_zebra_saved_dbend:
115, 170, 176, 387, 444, 458, 496, 508, 513, 520 202, 205
\g_zebra_note_id_int	_zebra_setup_microtype:
. 16, 171, 173	... 74, 87
\l_zebra_note_key_tl	_zebra_setup_pagelinks:
..... 116, 381, 383, 386, 402, 405, 438, 442, 444, 450, 453, 456, 458, 460	. 263, 280
_zebra_note_label:n	\l_zebra_sort_none_bool
..... 233, 515 11, 54, 55, 328, 625
_zebra_note_numbered:nnn	_zebra_stable_key:nn
. 363, 375	. 163, 403, 451
\g_zebra_note_public_alias_prop	_zebra_summary_row:n
..... 15, 258, 544 579, 600
\l_zebra_note_ref_type_tl	\l_zebra_summary_rows_tl
..... 19, 230, 499 20, 584, 597, 601, 612
\g_zebra_note_sig_known_prop	\l_zebra_symbol_tl
..... 104, 189, 425 23, 24, 317
\g_zebra_note_sig_written_prop	_zebra_target:nn
..... 106, 193, 197	. 215, 265, 273, 501
\g_zebra_note_stable_sig_prop	\l_zebra_total_notes_int
..... 108, 404, 452, 459 21, 583, 598, 614
_zebra_note_symbol:n	\l_zebra_unnumbered_bool
. 313, 509, 511 12, 57, 246, 361, 484, 494, 507, 512, 514, 519, 618
\g_zebra_note_symbols_prop	_zebra_use_listbody:n
..... 22, 69, 315, 316, 533, 534 322, 571
\g_zebra_note_target_prop	_zebra_write_sig:NN
..... 101, 383, 441, 455 191, 463
\l_zebra_note_target_tl	_zebra_write_zebra_label:n
..... 17, 172, 179, 384, 442, 456, 486, 490, 498, 501, 503	225, 236
\g_zebra_note_types_seq	_zebra_zebra_label_name:n
..... 13, 255, 528, 599, 628 218, 222, 229, 240
_zebra_note_unnumbered:nnn	_zebra_zebra_label_type:n
..... 341, 347, 362	219, 241
	_zebra_zebra_ref:n
 238, 248
	\zebracomment
 3, 4
	\zebrafixed
 4
	\zebranewnote
 1, 4, 7, 20, 635, 639, 641, 642, 643, 644
	\zebranote
 3
	\zebraplaceholder
 4
	\zebraquestion
 4
	\zebraref
 2, 5, 22, 244, 254
	\zebrasetup
 5, 7, 22, 637, 668
	\zebratodo
 3

Zebra Notes

Type	Count
todo	4
fixed	1
comment	1
note	3
placeholder	1
question	1
Total	11

List of notes

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