

zebra — Writing Revision Toolkit*

Ruini Xue[†]

v1.8.0 (2026/04/26)

Abstract

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



Contents

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
4	Implementation	6
	4.1 Package options	7
	4.2 Moving-argument deduplication	8
	4.3 Main notes macros	12
	4.4 Print summary at end of the document	19
	4.5 Compatibility shim	20
	4.6 Two-column demo	20
	Change History	21
	Index	21

*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 \LaTeX , 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 \LaTeX 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
- final** 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.
- 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}`.

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`].

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.
```

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.

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.

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.

```
\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?}
```

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]. 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]. 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

4 Implementation

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

Version data to start with.

```

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

```

4.1 Package options

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

```

```

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

```

4.2 Moving-argument deduplication

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

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

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

1. A *stable key* (`\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 (cosmetic duplicate); adding `\label` is the recommended fix.

```

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

```

```

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

```

```

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

```

```

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

```

4.3 Main notes macros

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

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

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

599             { \_zebra_print_note_group:n {##1} }
600           }
601         \endgroup
602       }
603     }
604   }

```

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

```

605 \NewDocumentCommand \zebranewnote { m m O{} }
606   { \_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.

```

607 \NewDocumentCommand \zebrasetup { m }
608   { \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 609 \zebranewnote{todo}{purple}
\fixed 610 \colorlet{zebra@fixed@color}{green!50!black}
\comment 611 \zebranewnote{fixed}{zebra@fixed@color}
\placeholder 612 \zebranewnote{comment}{blue}
613 \zebranewnote{note}{violet}
614 \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.

```

615 %% At end of document: print the note summary and list.
616 %% Content signatures are written to .aux inline (at allocation time),
617 %% so no additional end-of-document aux writes are needed.
618 \hook_gput_code:nnn { enddocument } { zebra }
619   {
620     \bool_if:NT \l__zebra_draft_bool
621       { \_zebra_print_notes: }
622   }
623 \ExplSyntaxOff
624 \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.

```
625 <*compat>
626 \NeedsTeXFormat{LaTeX2e}
627 \ProvidesPackage{zebra-goodies}
628 [2026/04/26 v1.8.0 Deprecated: use zebra instead]
629 \PackageWarningNoLine{zebra-goodies}
630 {Package 'zebra-goodies' is deprecated.\MessageBreak
631 Use \string\usepackage{zebra} instead}
632 \RequirePackageWithOptions{zebra}
633 </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.

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

```

669 The example is intentionally small, but it should still look like
670 a realistic revision pass.\comment{Bye}
671 \end{document}
672 \demo-twocol)

```

Change History

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

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>	129, 145
<code>\</code>		27, 28, 129, 145, 561, 581, 584

keys commands:	
\keys_define:nn	43, 62, 67
\l_keys_key_str	65, 70
\keys_set:nn	608
L	
\label	3, 5, 9, 21, 99, 120, 137, 485, 532, 643
\Large	576
\leftskip	529
legacy commands:	
\legacy_if:nTF	274
M	
\makebox	290, 291
\MakeLowercase	488
\MakeUppercase	488
\marginnote	12, 284, 295
\marginpar	12, 289
\marginparwidth	290, 291
\medskip	540, 547, 574, 577, 590
\MessageBreak	630
\microtypesetup	80
mode commands:	
\mode_if_inner_p:	277
msg commands:	
\msg_new:nnn	25, 30, 32
\msg_warning:nnn	231, 400, 513
N	
\NeedsTeXFormat	626
\NewDocumentCommand	235, 508, 605, 607
\newlabel	217
\nobreak	525, 573, 575, 594
\noindent	521, 574, 576, 593
\normalfont	12, 285, 286, 296, 297
\normalsize	12, 285, 286, 296, 297
\note	1, 3, 609, 645, 658
O	
\opt	337, 458
options:	
draft	2
final	2
font-expansion	2
pagelinks	2
sort	2
unnumbered	2
P	
\PackageWarningNoLine	629
pagelinks (option)	2
\pageref	205, 258, 261
\pagestyle	639
\par	521, 527, 533, 540, 547, 573, 574, 577, 590, 593
\parindent	531
\pdfstringdefDisableCommands	243
\placeholder	1, 4, 609, 652, 667
prg commands:	
\prg_new_conditional:Npnn	36
\prg_return_false:	40
\prg_return_true:	39
\ProcessKeyOptions	61
prop commands:	
\prop_get:NnN	376
\prop_get:NnNTF	373, 394, 411
\prop_gput:Nnn	65, 70, 181, 189, 426, 428, 440, 442, 444, 500, 504, 516
\prop_if_in:NnTF	
	185, 249, 310, 437, 499, 503
\prop_item:Nn	307, 311
\prop_new:N	
	13, 14, 21, 102, 103, 105, 107, 109
\protect	123, 139
\ProvidesExplPackage	3
\ProvidesPackage	627
Q	
\question	4
R	
\ref	2, 5, 21, 229, 238, 245, 649
regex commands:	
\regex_replace_all:mnN	128
\regex_replace_once:nnN	144
\RequirePackage	84, 85, 93, 97, 98, 196, 199
\RequirePackageWithOptions	632
\rightskip	530
S	
\s	129, 145
\sbox	9
\section	3, 8, 641
seq commands:	
\seq_gput_right:Nn	498
\seq_map_inline:Nn	246, 569, 598
\seq_new:N	12
\small	592
sort (option)	2
str commands:	
\str_lowercase:n	133, 134
\string	188, 217, 631
sys commands:	
\sys_if_engine_xetex:TF	72
T	
TeX and L ^A T _E X 2 _ε commands:	
\auxout	187, 215
\capttype	12, 280
\currentHref	469

<code>\@currentlabel</code>	467	<code>\use_none:n</code>	201, 532
<code>\@ifnextchar</code>	201	<code>\usepackage</code>	631, 636, 637
<code>\@makecaption</code>	9	V	
<code>\@marginparreset</code>	12	<code>\vbox</code>	12
<code>\@parboxrestore</code>	12, 285, 286, 296, 297	W	
<code>\@topnewpage</code>	12	<code>\write</code>	187
<code>\if@firstcolumn</code>	12	Z	
<code>\protected@edef</code>	467	<code>\Z</code>	145
<code>\protected@write</code>	9, 123, 139, 215	zebra internal commands:	
<code>\zebra@sig</code>	177, 188	<code>__zebra_allocate_note:nmn</code> ..	159, 425
tex commands:		<code>__zebra_apply_pdfstring_defs:..</code>	241, 267, 511
<code>\tex_inputlineno:D</code>	118	<code>__zebra_capitalize_type:n</code>	302, 471, 523
text commands:		<code>__zebra_content_sig:nmnN</code> ..	125, 383
<code>\textuppercase:n</code>	303	<code>\l__zebra_draft_bool</code> 7, 45, 344, 367, 620	
<code>\textbf</code>	523, 576, 581, 584	<code>__zebra_extract_label:nN</code> ..	140, 382
<code>\textcolor</code>	476, 480, 486, 487, 522, 556	<code>__zebra_if_package_loaded:n</code> ...	36
<code>\textdbend</code>	1, 4, 23	<code>__zebra_if_package_loaded:nTF</code> ..	77, 89, 259, 468, 510
<code>\textsc</code>	488	<code>__zebra_instance_key:nmn</code>	113, 372, 415, 424, 447
<code>\texttt</code>	491, 524	<code>__zebra_kernel_label:n</code> ..	99, 223, 473
<code>\title</code>	12	<code>__zebra_link:nm</code> ...	204, 257, 265, 526
tl commands:		<code>__zebra_list_entry:nmnmn</code> ..	328, 519
<code>\tl_clear:N</code>	148, 567	<code>\g__zebra_listbody_all_tl</code>	316, 546, 547
<code>\tl_gput_right:Nn</code>	321	<code>__zebra_margin_note:nm</code>	272, 475
<code>\tl_head:n</code>	303	<code>\l__zebra_microtype_expansion_</code>	
<code>\tl_if_blank:nTF</code>	301, 501, 524	<code>bool</code>	8, 48, 73, 79, 83
<code>\tl_if_empty:NTF</code> ...	390, 433, 546, 571	<code>__zebra_new_listbody:n</code> ...	314, 507
<code>\tl_if_eq:NNTF</code>	147, 397, 416	<code>__zebra_new_note_type:nm</code>	494
<code>\tl_new:N</code>	16, 17, 18, 19, 22, 110, 111, 315, 316	<code>__zebra_new_note_type:nmn</code>	495, 496, 606
<code>\tl_put_right:Nn</code>	554	<code>__zebra_note:nmn</code>	351, 354, 509
<code>\tl_set:Nn</code>	23, 127, 130, 142, 162, 164, 179, 180, 346, 369, 371, 392, 414, 423, 435, 446, 469, 470	<code>__zebra_note_color:n</code>	306, 346, 369, 522, 557
<code>\tl_set_eq:NN</code>	143, 149	<code>\l__zebra_note_color_tl</code>	17, 346, 369, 476, 480, 486, 487
<code>\tl_tail:n</code>	303	<code>\g__zebra_note_colors_prop</code>	13, 65, 307, 499, 500
<code>\tl_to_str</code>	153	<code>__zebra_note_count:n</code>	162, 304, 538, 551, 553, 560
<code>\tl_to_str:n</code>	115, 116, 117, 127, 132, 133, 142, 156, 179, 180	<code>\g__zebra_note_display_prop</code>	103, 376, 428, 442
<code>\tl_use:N</code>	188, 318, 547, 582	<code>\l__zebra_note_display_tl</code> 110, 162, 168, 377, 429, 443, 467, 478, 483, 490	
<code>\l_tmpa_tl</code>	127, 129, 134, 179, 181, 383, 397, 412, 445, 448	<code>\g__zebra_note_id_int</code> ..	15, 163, 165
<code>\l_tmpb_tl</code> ..	142, 143, 146, 147, 149, 180, 181, 382, 390, 393, 401, 433, 436	<code>\l__zebra_note_key_tl</code>	111, 371, 373, 376, 392, 395, 423, 427, 429, 435, 438, 441, 443, 445
tmpc commands:			
<code>\l_tmpc_tl</code>	143, 147, 395, 397, 412, 416, 446, 448		
tmpd commands:			
<code>\l_tmpd_tl</code>	414, 416		
<code>\todo</code>	1, 3, 4, 12, 609, 643, 664		
U			
unnumbered (option)	2		
use commands:			
<code>\use_i:nm</code>	209		

_zebra_note_label:n	221, 485	_zebra_render_note:nnn	348, 378, 451, 462
_zebra_note_numbered:nnn	358, 365	_zebra_saved_dbend:	194, 197
\g_zebra_note_public_alias_prop	14, 249, 516	_zebra_setup_microtype:	75, 88
\l_zebra_note_ref_type_tl	18, 218, 470	_zebra_setup_pagelinks:	254, 271
\g_zebra_note_sig_known_prop	105, 181, 411	\l_zebra_sort_none_bool	10, 55, 56, 323, 595
\g_zebra_note_sig_written_prop	107, 185, 189	_zebra_stable_key:nn	155, 393, 436
\g_zebra_note_stable_sig_prop	109, 394, 437, 444	_zebra_summary_row:n	549, 570
_zebra_note_symbol:n	308, 479, 481	\l_zebra_summary_rows_tl	19, 554, 567, 571, 582
\g_zebra_note_symbols_prop	21, 70, 310, 311, 503, 504	\l_zebra_symbol_tl	22, 23, 312
\g_zebra_note_target_prop	102, 373, 426, 440	_zebra_target:nn	203, 256, 264, 472
\l_zebra_note_target_tl	16, 164, 171, 374, 427, 441, 469, 472, 473	\l_zebra_total_notes_int	20, 553, 568, 584
\g_zebra_note_types_seq	12, 246, 498, 569, 598	\l_zebra_unnumbered_bool	11, 58, 237, 356, 465, 477, 482, 484, 489, 588
_zebra_note_unnumbered:nnn	336, 342, 357	_zebra_use_listbody:n	317, 541
\l_zebra_pagelinks_bool	9, 51, 91, 262	_zebra_write_sig:NN	183, 448
_zebra_pageref:n	205, 258, 261, 526	_zebra_write_zebra_label:n	213, 224
_zebra_pdfstring_note	200, 248, 250	_zebra_zebra_label_name:n	206, 210, 217, 228
_zebra_pdfstring_note_opt:w	201, 202	_zebra_zebra_label_type:n	207, 229
_zebra_prepend:nn	300, 491, 524	_zebra_zebra_ref:n	226, 239
_zebra_print_note_group:n	536, 599	\zebracomment	3, 4
_zebra_print_notes:	565, 621	\zebrafixed	4
_zebra_print_notes_inorder:	544, 596	\zebranewnote	1, 4, 19, 605, 609, 611, 612, 613, 614
_zebra_record_note:nnnnn	166, 319	\zebranote	3
		\zebraplaceholder	4
		\zebraquestion	4
		\zebraref	2, 5, 21, 235, 245
		\zebrasetup	5, 7, 21, 607, 638
		\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	