

zebra — Writing Revision Toolkit*

Ruini Xue[†]

v1.5.0 (2026/04/15)

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	17
	4.5 Compatibility shim	17
	4.6 Two-column demo	17
	Change History	18
	Index	19

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

[†]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

The package is supplied in `dtx` format and as a pre-extracted zip file. The latter is the most convenient option for most users: simply unzip it into your local `texmf` directory and run `texhash` to update the file-name database, or unzip the files directly into your working directory. To unpack the `dtx` yourself, run `tex zebra.dtx` to extract the package, or `pdflatex zebra.dtx` to extract it and typeset the documentation at the same time.¹

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.

¹Running `latexmk zebra.dtx` is even more convenient as it handles multiple compilation passes automatically.


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.
```

1  The motivation section still feels too vague [TODO 1: 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 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}.
```

2  The related work section needs more references [TODO 2@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 3@bob, carol: verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards].

3 

Because notes are set as inline text, they can appear inside moving arguments such as `\section` and `\caption`. Adding a `\label` inside the note gives it a stable identity across contexts:

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

A labeled note in a heading is counted exactly once; the table of contents and the note list link to the correct page after two compilations. Notes without a label still work but may be counted separately in each context (TOC, running headers).

<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

4  [TODO 4: 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 1: revise the introduction before submission**].
The same issue appears again later [**NOTE 3: see Todo 1 (that is, Todo 1) 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=16cm,paperheight=13cm,margin=1.2cm]{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 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.

`\end{document}`

The code above produces the following output:

1 Demo

1 ♣ **name** [**Comment 1: revise the name**]

1 ⚠ [**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.

1 ⚠ [**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 ⚠ [**NOTE 2@frank: check the numbers**]

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

One more short paragraph is enough to show how 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.

.....

Zebra Notes

Type	Count
todo	2
fixed	2
comment	2
note	2
placeholder	1
Total	9

4 Implementation

```

1 <*package>
2 <@@=zebra>
   Version data to start with.
3 \ProvidesExplPackage{zebra}
4   {2026/04/15}
5   {1.5.0}
6   {Writing Revision Toolkit}

```

4.1 Package options

Package options `draft`, `pagelinks`, `font-expansion`, and `sort` 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 \seq_new:N \g__zebra_note_types_seq
12 \prop_new:N \g__zebra_note_colors_prop
13 \prop_new:N \g__zebra_note_public_alias_prop
14 \int_new:N \g__zebra_note_id_int
15 \tl_new:N \l__zebra_note_target_tl
16 \tl_new:N \l__zebra_note_color_tl
17 \tl_new:N \l__zebra_note_ref_type_tl
18 \tl_new:N \l__zebra_summary_rows_tl
19 \int_new:N \l__zebra_total_notes_int
20 \prop_new:N \g__zebra_note_symbols_prop
21 \tl_new:N \l__zebra_symbol_tl
22 \tl_set:Nn \l__zebra_symbol_tl { \textdbend }
23
24 \msg_new:nnn { zebra } { command-taken }
25 {
26   The~command~'\iow_char:N\|#1'~is~already~defined.~
27   Use~'\iow_char:N\zebra#1'~instead.
28 }
29 \msg_new:nnn { zebra } { invalid-note-label }
30 { Label~'#1'~is~not~a~zebra~note~label. }
31
32 \prg_new_conditional:Npnn \__zebra_if_package_loaded:n #1 { T , F , TF }
33 {
34   \cs_if_exist:cTF { ver@#1.sty }
35     { \prg_return_true: }
36     { \prg_return_false: }
37 }
38
39 \keys_define:nn { zebra }
40 {
41   draft .bool_set:N = \l__zebra_draft_bool,
42   draft .initial:n = true,
43   final .meta:n = { draft = false },
44   font-expansion .bool_set:N = \l__zebra_microtype_expansion_bool,
45   font-expansion .initial:n = true,

```

```

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

```

4.2 Moving-argument deduplication

Notes inside moving arguments (`\section`, `\caption`, etc.) may be processed more than once per compilation pass: the original location plus each secondary context (TOC, LOF, running headers, `\sbox` measurement in `\@makecaption`). The deduplication layer ensures each unique note is allocated exactly once and rendered at most once per pass.

Dedup key. The key is $\langle type \rangle | \langle author \rangle | \langle sanitised body \rangle$. The body is stringified and the first $\backslash label\{...\}$ (if any) is stripped by a single regex pass. This makes the key identical whether the body still contains $\backslash label$ (body, marks) or has had it consumed by $\backslash protected@write$'s $\backslash edef$ (TOC, LOF entries).

Encounter counting. A global prop tracks how many times each key is encountered during the current pass. At $\backslash enddocument$ the counts are written to the $.aux$ file via $\backslash zebra@notecount$. On the next pass the expected total N is known from the $.aux$ data. Two render modes are used:

- *Full render* (encounter = N , $N > 0$): the last encounter, typically the body location, performs all writes (hypertarget, target label, user labels via $\backslash label$ inside the note).
- *Light render* (all other cases, including $N = 0$ on the first pass): margin note and inline text are shown but no $\backslash hypertarget$ or $\backslash label$ writes are emitted.

Every encounter produces visible output so that the page layout — in particular the caption width measured by $\backslash @makecaption$'s $\backslash hbox$ — stays stable across passes and $latexmk$ converges (typically in two to three passes).

```

91 \RequirePackage{xcolor}
92 \RequirePackage{marginnote}
93 \cs_new_eq:NN \__zebra_kernel_label:n \label
94 %% -- dedup data structures --
95 % key -> hypertarget name (e.g. zebranote.3)
96 \prop_new:N \g__zebra_note_target_prop
97 % key -> display number (e.g. 2)
98 \prop_new:N \g__zebra_note_display_prop
99 % key -> encounter count this pass
100 \prop_new:N \g__zebra_note_encounter_prop
101 % key -> expected total N from previous pass (.aux)
102 \prop_new:N \g__zebra_note_total_prop
103 \tl_new:N \l__zebra_note_display_tl
104 \tl_new:N \l__zebra_note_key_tl
105 %% Compute the dedup key. The body is stringified and the first
106 %% \label{...} is stripped so that the key matches across contexts
107 %% where \protected@write's \edef consumes the \label token.
108 \cs_new_protected:Npn \__zebra_note_dedup_key:nnnN #1#2#3#4
109 {
110   \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#3} }
111   \regex_replace_once:nnN
112     { \backslash label \s* \{ [^{}]* \} } { } \l_tmpa_tl
113   \tl_set:Nx #4
114     {
115       \tl_to_str:n {#1}
116       | \tl_to_str:n {#2}
117       | \l_tmpa_tl
118     }
119 }
120 %% Allocate a fresh note: increment the type counter, generate
121 %% a unique hypertarget name, and record the note in the list body.
122 \cs_new_protected:Npn \__zebra_allocate_note:nnn #1#2#3
123 {

```

```

124 \int_gincr:c { g__zebra_note_count_#1_int }
125 \tl_set:Nx \l__zebra_note_display_tl { \__zebra_note_count:n {#1} }
126 \int_gincr:N \g__zebra_note_id_int
127 \tl_set:Nx \l__zebra_note_target_tl
128   { zebernote.\int_use:N \g__zebra_note_id_int }
129 \__zebra_record_note:nnnnn
130   {#1}
131   { \l__zebra_note_display_tl }
132   {#2}
133   {#3}
134   { \l__zebra_note_target_tl }
135 }
136 %% Aux-file interface: read encounter counts from the previous pass.
137 %% The key is re-stringified to normalise catcodes: during .aux
138 %% reading \makeatletter is active (@ = catcode 11), but at note
139 %% time the key is built with \tl_to_str (@ = catcode 12).
140 \cs_new_protected:Npn \zebra@notecount #1#2
141   {
142     \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#1} }
143     \prop_gput:NVn \g__zebra_note_total_prop \l_tmpa_tl {#2}
144   }
145 %% Write one key's encounter count to .aux (called at \enddocument).
146 %% Uses \immediate\write so the data reaches the .aux regardless of
147 %% deferred-write timing at end of document.
148 \cs_new_protected:Npn \__zebra_write_one_encounter:nn #1#2
149   {
150     \immediate\write \@auxout
151       { \string\zebra@notecount {#1} {#2} }
152   }
153 \cs_if_exist:NTF \dbend
154   {
155     \cs_set_eq:NN \__zebra_saved_dbend: \dbend
156     \cs_undefine:N \dbend
157     \RequirePackage{manfnt}
158     \cs_set_eq:NN \dbend \__zebra_saved_dbend:
159   }
160   { \RequirePackage{manfnt} }
161 \cs_new_protected:Npn \__zebra_pdfstring_note:
162   { \@ifnextchar [ { \__zebra_pdfstring_note_opt:w } { \use_none:n } } }
163 \cs_new_protected:Npn \__zebra_pdfstring_note_opt:w [#1] #2 { }
164 \cs_new:Npn \__zebra_target:nn #1#2 {#2}
165 \cs_new:Npn \__zebra_link:nn #1#2 {#2}
166 \cs_new:Npn \__zebra_pageref:n #1 { \pageref {#1} }
167 \cs_new:Npn \__zebra_zebra_label_name:n #1 { #1@zebra }
168 \cs_new:Npn \__zebra_zebra_label_type:n #1
169   {
170     \exp_after:wN \use_i:nn
171     \cs:w r@\__zebra_zebra_label_name:n {#1}\cs_end:
172     { }
173   }
174 \cs_new_protected:Npn \__zebra_write_zebra_label:n #1
175   {
176     \protected@write \@auxout { }
177     {

```

```

178     \string\newlabel{\_zebra_zebra_label_name:n {#1}}
179     {{\exp_not:V \l_zebra_note_ref_type_tl}{}}
180   }
181 }
182 \cs_new_protected:Npn \_zebra_note_label:n #1
183 {
184   \_zebra_kernel_label:n {#1}
185   \_zebra_write_zebra_label:n {#1}
186 }
187 \cs_new_protected:Npn \_zebra_zebra_ref:n #1
188 {
189   \cs_if_exist:cTF { r@\_zebra_zebra_label_name:n {#1} }
190   { \_zebra_zebra_label_type:n {#1}~\ref{#1} }
191   {
192     \msg_warning:nnn { zebra } { invalid-note-label } {#1}
193     ??
194   }
195 }
196 \NewDocumentCommand \zebraref { m }
197 { \_zebra_zebra_ref:n {#1} }
198 \cs_new_protected:Npn \_zebra_apply_pdfstring_defs:
199 {
200   \pdfstringdefDisableCommands
201   {
202     \cs_set:Npn \zebraref ##1 { \ref{##1} }
203     \seq_map_inline:Nn \g_zebra_note_types_seq
204     {
205       \cs_set_eq:cN { zebra##1 } \_zebra_pdfstring_note:
206       \prop_if_in:NnT \g_zebra_note_public_alias_prop { ##1 }
207       { \cs_set_eq:cN { ##1 } \_zebra_pdfstring_note: }
208     }
209   }
210 }
211 \cs_new_protected:Npn \_zebra_setup_pagelinks:
212 {
213   \cs_set:Npn \_zebra_target:nn ##1##2 {##2}
214   \cs_set:Npn \_zebra_link:nn ##1##2 {##2}
215   \cs_set:Npn \_zebra_pageref:n ##1 { \pageref {##1} }
216   \_zebra_if_package_loaded:nT { hyperref }
217   {
218     \cs_set:Npn \_zebra_pageref:n ##1 { \pageref* {##1} }
219     \bool_if:NT \l_zebra_pagelinks_bool
220     {
221       \cs_set:Npn \_zebra_target:nn ##1##2 { \hypertarget{##1}{##2} }
222       \cs_set:Npn \_zebra_link:nn ##1##2 { \hyperlink{##1}{##2} }
223     }
224     \_zebra_apply_pdfstring_defs:
225   }
226 }
227 \hook_gput_code:nnn { begindocument } { zebra }
228 { \_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 twocolumn mode, `\marginpar`'s optional argument selects the left-margin variant automatically. In single-column mode, `\marginnote` is used with the right-margin variant as default.

```

229 \cs_new_protected:Npn \__zebra_margin_note:nn #1#2
230 {
231   \legacy_if:nTF { @twocolumn }
232   {
233     \mode_if_inner:TF
234     { \marginnote{#2} }
235     {
236       \marginpar
237       [ { \makebox[\marginparwidth]{#1} } ]
238       { \makebox[\marginparwidth]{#2} }
239     }
240   }
241   { \marginnote[#1]{#2} }
242 }
243 \cs_new:Npn \__zebra_prepend:nn #1#2
244 { \tl_if_blank:nTF {#2} {} {#1#2} }
245 \cs_new:Npn \__zebra_capitalize_type:n #1
246 { \text_uppercase:n { \tl_head:n {#1} } \tl_tail:n {#1} }
247 \cs_new:Npn \__zebra_note_count:n #1
248 { \int_use:c { g__zebra_note_count_#1_int } }
249 \cs_new:Npn \__zebra_note_color:n #1
250 { \prop_item:Nn \g__zebra_note_colors_prop {#1} }
251 \cs_new:Npn \__zebra_note_symbol:n #1
252 {
253   \prop_if_in:NnTF \g__zebra_note_symbols_prop {#1}
254   { \prop_item:Nn \g__zebra_note_symbols_prop {#1} }
255   { \l__zebra_symbol_tl }
256 }
257 \cs_new_protected:Npn \__zebra_new_listbody:n #1
258 { \tl_new:c { g__zebra_listbody_#1_tl } }
259 \tl_new:N \g__zebra_listbody_all_tl
260 \cs_new:Npn \__zebra_use_listbody:n #1
261 { \tl_use:c { g__zebra_listbody_#1_tl } }
262 \cs_new_protected:Npn \__zebra_record_note:nnnn #1#2#3#4#5
263 {
264   \tl_gput_right:cx
265   {
266     \bool_if:NTF \l__zebra_sort_none_bool
267     { g__zebra_listbody_all_tl }
268     { g__zebra_listbody_#1_tl }
269   }
270   {
271     \exp_not:N \__zebra_list_entry:nnnn
272     { \exp_not:n {#1} }
273     {#2}
274     { \exp_not:n {#3} }

```

```

275     { \exp_not:n {#4} }
276     {#5}
277   }
278 }
279 %% \__zebra_note:nnn {type}{author}{body}
280 %% Main entry point for every note. Two render modes:
281 %% full render (enc=N, N>0) -- writes hypertarget + labels
282 %% light render (otherwise) -- visible, no writes
283 \cs_new_protected:Npn \__zebra_note:nnn #1#2#3
284 {
285   \bool_if:NT \l__zebra_draft_bool
286   {
287     \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
288     \__zebra_note_dedup_key:nnnN {#1} {#2} {#3} \l__zebra_note_key_tl
289     %% -- encounter counting (global, survives output routine) --
290     \prop_get:NVNTF \g__zebra_note_encounter_prop \l__zebra_note_key_tl \l_tmpa_tl
291     { \tl_set:Nx \l_tmpa_tl { \int_eval:n { \l_tmpa_tl + 1 } } }
292     { \tl_set:Nn \l_tmpa_tl { 1 } }
293     \prop_gput:NVV \g__zebra_note_encounter_prop \l__zebra_note_key_tl \l_tmpa_tl
294     %% -- allocate once per key --
295     \prop_get:NVNTF \g__zebra_note_target_prop \l__zebra_note_key_tl
296     \l__zebra_note_target_tl
297     {
298       \prop_get:NVN \g__zebra_note_display_prop \l__zebra_note_key_tl
299       \l__zebra_note_display_tl
300     }
301     {
302       \__zebra_allocate_note:nnn {#1} {#2} {#3}
303       \prop_gput:NVV \g__zebra_note_target_prop
304       \l__zebra_note_key_tl \l__zebra_note_target_tl
305       \prop_gput:NVV \g__zebra_note_display_prop
306       \l__zebra_note_key_tl \l__zebra_note_display_tl
307     }
308     %% -- render decision --
309     %% l_tmpa_tl = current encounter number
310     %% l_tmpb_tl = N (expected total from .aux; 0 = unknown)
311     %% Two modes only (no suppress -- suppressing would change the
312     %% caption width inside \sbox, altering the encounter count and
313     %% preventing latexmk convergence):
314     %% full render (encounter=N, N>0) -- writes hypertarget + labels
315     %% light render (otherwise) -- visible, no writes
316     \prop_get:NVNF \g__zebra_note_total_prop \l__zebra_note_key_tl \l_tmpb_tl
317     { \tl_set:Nn \l_tmpb_tl { 0 } }
318     \group_begin:
319     %% Default to light render: \label inside the body is
320     %% silently consumed (no writes).
321     \cs_set_eq:NN \label \use_none:n
322     \bool_if:nT
323     {
324       \int_compare_p:nNn { \l_tmpb_tl } > { 0 } &&
325       \int_compare_p:nNn { \l_tmpa_tl } = { \l_tmpb_tl }
326     }
327     {
328       %% Full render: encounter = N. Set up cross-reference

```

```

329         %% data, write hypertarget + labels, and override \label.
330         \protected@edef \@currentlabel
331         { \l__zebra_note_display_tl }
332         \__zebra_if_package_loaded:nT { hyperref }
333         {
334             \tl_set:Nx \@currentHref
335             { \l__zebra_note_target_tl }
336         }
337         \tl_set:Nx \l__zebra_note_ref_type_tl
338         { \__zebra_capitalize_type:n {#1} }
339         \__zebra_target:nn { \l__zebra_note_target_tl } {}
340         \exp_args:NV \__zebra_kernel_label:n
341         \l__zebra_note_target_tl
342         \cs_set_eq:NN \label \__zebra_note_label:n
343     }
344     %% Margin note + inline text (shared by both modes).
345     \__zebra_margin_note:nn
346     {\textcolor{\l__zebra_note_color_tl}{%
347         {\bfseries\l__zebra_note_display_tl}\kern1pt
348         \__zebra_note_symbol:n {#1}}}
349     {\textcolor{\l__zebra_note_color_tl}{%
350         \__zebra_note_symbol:n {#1}\kern1pt
351         {\bfseries\l__zebra_note_display_tl}}}
352     \textcolor{\l__zebra_note_color_tl}{[\colorbox[gray]{0.97}{%
353         \textcolor{\l__zebra_note_color_tl !70!black}{%
354             \textsc{\MakeLowercase{\MakeUppercase#1}}~%
355             \l__zebra_note_display_tl
356             \texttt{\__zebra_prepend:nn {@}{#2}:} } #3]}%
357     \group_end:
358 }
359 }
360 \cs_new_protected:Npn \__zebra_new_note_type:nn #1#2
361 { \__zebra_new_note_type:nnn {#1} {#2} {} }
362 \cs_new_protected:Npn \__zebra_new_note_type:nnn #1#2#3
363 {
364     \seq_gput_right:Nn \g__zebra_note_types_seq {#1}
365     \prop_if_in:NnF \g__zebra_note_colors_prop {#1}
366     { \prop_gput:Nnn \g__zebra_note_colors_prop {#1} {#2} }
367     \tl_if_blank:nF {#3}
368     {
369         \prop_if_in:NnF \g__zebra_note_symbols_prop {#1}
370         { \prop_gput:Nnn \g__zebra_note_symbols_prop {#1} {#3} }
371     }
372     \int_new:c { g__zebra_note_count_#1_int }
373     \__zebra_new_listbody:n {#1}
374     \exp_args:Nc \NewDocumentCommand { zebra#1 } { 0 } m }
375     { \__zebra_note:nnn {#1}{##1}{##2} }
376     \__zebra_if_package_loaded:nT { hyperref }
377     { \__zebra_apply_pdfstring_defs: }
378     \cs_if_exist:cTF {#1}
379     { \msg_warning:nnn { zebra } { command-taken } {#1} }
380     {
381         \cs_set_eq:cc {#1} {zebra#1}
382         \prop_gput:Nnn \g__zebra_note_public_alias_prop {#1} { true }

```

```

383     }
384   }
385 \cs_new_protected:Npn \__zebra_list_entry:nnnnn #1#2#3#4#5
386 {
387   \par\noindent
388   \textcolor{\__zebra_note_color:n {#1}}{%
389     \textbf{\__zebra_capitalize_type:n {#1}~#2}%
390     \tl_if_blank:nF {#3} { \enspace \texttt{\__zebra_prepend:nn {#3}} }}%
391   \nobreak\dotfill
392   \__zebra_link:nn {#5} { \__zebra_pageref:n {#5} }%
393   \par
394   \begingroup
395     \leftskip=2em
396     \rightskip=2em
397     \parindent=0pt
398     \cs_set_eq:NN \label \use_none:n
399     #4\par
400   \endgroup
401 }
402 \cs_new_protected:Npn \__zebra_print_note_group:n #1
403 {
404   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
405   {
406     \par\medskip
407     \__zebra_use_listbody:n {#1}
408   }
409 }
410 \cs_new_protected:Npn \__zebra_print_notes_inorder:
411 {
412   \tl_if_empty:NF \g__zebra_listbody_all_tl
413   { \par\medskip \tl_use:N \g__zebra_listbody_all_tl }
414 }
415 \cs_new_protected:Npn \__zebra_summary_row:n #1
416 {
417   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
418   {
419     \int_add:Nn \l__zebra_total_notes_int { \__zebra_note_count:n {#1} }
420     \tl_put_right:Nx \l__zebra_summary_rows_tl
421     {
422       \exp_not:N \textcolor
423       { \__zebra_note_color:n {#1} }
424       {#1}
425       \exp_not:N &
426       \__zebra_note_count:n {#1}
427       \exp_not:N \\
428     }
429   }
430 }
431 \cs_new_protected:Npn \__zebra_print_notes:
432 {
433   \tl_clear:N \l__zebra_summary_rows_tl
434   \int_zero:N \l__zebra_total_notes_int
435   \seq_map_inline:Nn \g__zebra_note_types_seq
436   { \__zebra_summary_row:n {##1} }

```

```

437 \tl_if_empty:NF \l__zebra_summary_rows_tl
438 {
439   \par\nobreak
440   \noindent\dotfill\par\medskip
441   \nobreak
442   \noindent\textbf{\Large Zebra~Notes}
443   \par \medskip
444   \begin{center}
445     \begin{tabular}{lr}
446       \hline
447       \textbf{Type} & \textbf{Count} \\ \hline
448       \tl_use:N \l__zebra_summary_rows_tl
449       \hline
450       \textbf{Total} & \textbf{\int_use:N \l__zebra_total_notes_int} \\ \hline
451       \hline
452     \end{tabular}
453   \end{center}
454   \legacy_if:nTF { @twocolumn }
455   {
456     \clearpage
457     \onecolumn
458   }
459   { \par \medskip }
460   \begin{group}
461     \small
462     \noindent{\bfseries List~of~notes}\par
463     \nobreak
464     \bool_if:NTF \l__zebra_sort_none_bool
465     { \__zebra_print_notes_inorder: }
466     {
467       \seq_map_inline:Nn \g__zebra_note_types_seq
468       { \__zebra_print_note_group:n {##1} }
469     }
470   \end{group}
471 }
472 }

```

\zebranewnote All note types are created with \zebranewnote.

```

473 \NewDocumentCommand \zebranewnote { m m 0{} }
474 { \__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.

```

475 \NewDocumentCommand \zebrasetup { m }
476 { \keys_set:nn { zebra-setup } {#1} }

```

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

\todo Built-in note types, defined with \zebranewnote.

\note `\zebranewnote{todo}{purple}`

\fixed `\colorlet{zebra@fixed@color}{green!50!black}`

\comment `\zebranewnote{fixed}{zebra@fixed@color}`

\placeholder `\zebranewnote{comment}{blue}`


```

481 \zebranewnote{note}{violet}
482 \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.

```

483 %% At end of document: print the note summary and list, then
484 %% write encounter counts to .aux for the next pass.
485 \hook_gput_code:nnn { enddocument } { zebra }
486 {
487   \bool_if:NT \l__zebra_draft_bool
488   {
489     \__zebra_print_notes:
490     \prop_map_function:NN \g__zebra_note_encounter_prop
491     \__zebra_write_one_encounter:nn
492   }
493 }
494 \ExplSyntaxOff
495 \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.

```

496 \compat
497 \NeedsTeXFormat{LaTeX2e}
498 \ProvidesPackage{zebra-goodies}
499   [2026/04/15 v1.5.0 Deprecated: use zebra instead]
500 \PackageWarningNoLine{zebra-goodies}
501   {Package 'zebra-goodies' is deprecated.\MessageBreak
502   Use \string\usepackage{zebra} instead}
503 \RequirePackageWithOptions{zebra}
504 \endcompat

```

4.6 Two-column demo

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

```

505 \demo-twocol
506 \documentclass[twocolumn]{article}
507 \usepackage[paperwidth=16cm,paperheight=13cm,margin=1.2cm]{geometry}
508 \usepackage{zebra}
509 \zebrasetup{symbol/comment=\clubsuit}
510 \pagestyle{empty}
511 \begin{document}
512 \section{Demo name\comment{revise the name}}
513 This draft still needs work
514 \todo[alice]{\label{zebra:intro}revise the introduction}. The

```

515 opening paragraph should also explain the main goal more plainly.
516 Add one more citation here `\note[bob]{support this claim}`. A
517 brief roadmap sentence would also make the structure easier to
518 scan.

519

520 The issue raised in `Todo~\ref{zebra:intro}` still applies in
521 the conclusion. The table now looks fine
522 `\fixed[carol]{alignment corrected}`, but one figure is still
523 missing `\placeholder[eve]{insert the overview figure}`. A short
524 transition would also help the flow. The middle section should
525 probably end with a clearer summary sentence before the
526 discussion begins. That summary can stay compact, but it should
527 signal why the next section matters.

528

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

537

538 One more short paragraph is enough to show how the markers stay
539 readable in a compact two-column layout. The example is
540 intentionally small, but it should still look like a realistic
541 revision pass.

542 `\end{document}`
543 `</demo-twocol>`

Change History

v0.1.0		<code>\zebranewnote</code> : Fix on <code>\global</code> for examples	16
General: Initial public release	1		
v0.2.0		v0.8.1	
General: Fix <code>xcolor</code> conflict	1	General: Fix doc	4
v0.3.0		v0.9.0	
General: Detect command conflicts	1	General: Fix legacy bugs and improve implementation	1
v0.4.0		v0.9.1	
General: Show note number for easy reference	1	General: Beautify the numbers.	1
v0.5.0		v0.9.2	
General: Use darker color for label	1	General: Faster.	1
v0.6.0		v1.0.0	
General: Use gray background for label	1	General: <code>expl3</code> , list of notes and compatibility.	1
v0.7.0		v1.1.0	
General: Move to <code>docstrip</code>	2	General: Customisable margin symbol, accurate page numbers, code cleanup.	1
v0.8.0			
General: Fix new note demo	4		

v1.1.1	General: Per-type color/symbol keys, \zebrasetup.	1	Rename the page-link option to pagelinks/nopagelinks.	1	
v1.2.0	General: Simplify key architecture. . . .	1	v1.4.0	General: Support note labels via \label, \ref, and \zebraref.	1
v1.3.0	General: Rename package to zebra. . . .	1	v1.5.0	General: Fix notes numbering in moving arguments.	1
	Rename the microtype expansion option to font-expansion.	1			

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
\	26, 27, 112, 427, 447, 450
\{	112
\}	112
B	
\begin	444, 445, 511
\begingroup	394, 460
\bfseries	347, 351, 462
bool commands:	
\bool_if:NTF	73, 77, 85, 219, 266, 285, 464, 487
\bool_if:nTF	322
\bool_new:N	7, 8, 9, 10
\bool_set_false:N	51, 67
\bool_set_true:N	52
C	
\caption	3, 8
\clearpage	456
\clubsuit	509
\cmd	3
\colorbox	352
\colorlet	4, 478
\comment	1, 3, 4, 477, 512, 531
cs commands:	
\cs:w	171
\cs_end:	171
\cs_if_exist:NTF	34, 153, 189, 378
\cs_new:Npn	164, 165, 166, 167, 168, 243, 245, 247, 249, 251, 260
\cs_new_eq:NN	93
\cs_new_protected:Npn	69, 108, 122, 140, 148, 161, 163, 174, 182, 187, 198, 211, 229, 257, 262, 283, 360, 362, 385, 402, 410, 415, 431
\cs_set:Npn	202, 213, 214, 215, 218, 221, 222
\cs_set_eq:NN	155, 158, 205, 207, 321, 342, 381, 398
\cs_undefine:N	156
D	
\dbend	153, 155, 156, 158
\definecolor	4
\documentclass	506
\dotfill	391, 440
draft (option)	2
E	
\edef	9, 107
\end	452, 453, 542
\enddocument	9, 145
\endgroup	400, 470
\enspace	390
exp commands:	
\exp_after:wN	170
\exp_args:Nc	374
\exp_args:NV	340
\exp_not:N	271, 422, 425, 427
\exp_not:n	179, 272, 274, 275
\ExplSyntaxOff	494
F	
final (option)	2
\fixed	1, 4, 477, 522, 532
font-expansion (option)	2
G	
\global	18
group commands:	
\group_begin:	318
\group_end:	357

H		N	
<code>\hline</code>	446, 447, 449, 451	<code>\NeedsTeXFormat</code>	497
hook commands:		<code>\NewDocumentCommand</code> ...	196, 374, 473, 475
<code>\hook_gput_code:nnn</code>	227, 485	<code>\NewLabel</code>	178
<code>\hyperlink</code>	222	<code>\nobreak</code>	391, 439, 441, 463
<code>\hypersetup</code>	88	<code>\noindent</code>	387, 440, 442, 462
<code>\hypertarget</code>	9, 221	<code>\note</code>	1, 3, 477, 516, 529
I		O	
<code>\immediate</code>	146, 150	<code>\onecolumn</code>	457
int commands:		options:	
<code>\int_add:Nn</code>	419	<code>draft</code>	2
<code>\int_compare:nNnTF</code>	404, 417	<code>final</code>	2
<code>\int_compare_p:nNn</code>	324, 325	<code>font-expansion</code>	2
<code>\int_eval:n</code>	291	<code>pagelinks</code>	2
<code>\int_gincr:N</code>	124, 126	<code>sort</code>	2
<code>\int_new:N</code>	14, 19, 372		
<code>\int_use:N</code>	128, 248, 450	P	
<code>\int_zero:N</code>	434	<code>\PackageWarningNoLine</code>	500
iow commands:		<code>pagelinks (option)</code>	2
<code>\iow_char:N</code>	26, 27	<code>\pageref</code>	166, 215, 218
K		<code>\pagestyle</code>	510
<code>\kern</code>	347, 350	<code>\par</code>	387, 393,
keys commands:		399, 406, 413, 439, 440, 443, 459, 462	
<code>\keys_define:nn</code>	39, 56, 61	<code>\parindent</code>	397
<code>\l_keys_key_str</code>	59, 64	<code>\pdfstringdefDisableCommands</code>	200
<code>\keys_set:nn</code>	476	<code>\placeholder</code>	1, 4, 477, 523
L		prg commands:	
<code>\label</code>	3, 5, 9, 19, 93,	<code>\prg_new_conditional:Npnn</code>	32
106, 107, 319, 321, 329, 342, 398, 514		<code>\prg_return_false:</code>	36
<code>\Large</code>	442	<code>\prg_return_true:</code>	35
<code>\leftskip</code>	395	<code>\ProcessKeyOptions</code>	55
legacy commands:		prop commands:	
<code>\legacy_if:nTF</code>	231, 454	<code>\prop_get:NnN</code>	298
M		<code>\prop_get:NnNTF</code>	290, 295, 316
<code>\makeatletter</code>	138	<code>\prop_gput:Nnn</code>	59,
<code>\makebox</code>	237, 238	64, 143, 293, 303, 305, 366, 370, 382	
<code>\MakeLowercase</code>	354	<code>\prop_if_in:NnTF</code> ...	206, 253, 365, 369
<code>\MakeUppercase</code>	354	<code>\prop_item:Nn</code>	250, 254
<code>\marginnote</code>	12, 234, 241	<code>\prop_map_function:NN</code>	490
<code>\marginpar</code>	12, 236	<code>\prop_new:N</code> 12, 13, 20, 96, 98, 100, 102	
<code>\marginparwidth</code>	237, 238	<code>\ProvidesExplPackage</code>	3
<code>\medskip</code>	406, 413, 440, 443, 459	<code>\ProvidesPackage</code>	498
<code>\MessageBreak</code>	501		
<code>\microtypesetup</code>	74	Q	
mode commands:		<code>\question</code>	4
<code>\mode_if_inner:TF</code>	233	R	
msg commands:		<code>\ref</code>	5, 19, 190, 202, 520
<code>\msg_new:nnn</code>	24, 29	regex commands:	
<code>\msg_warning:nnn</code>	192, 379	<code>\regex_replace_once:nnN</code>	111
		<code>\RequirePackage</code> 78, 79, 87, 91, 92, 157, 160	
		<code>\RequirePackageWithOptions</code>	503
		<code>\rightskip</code>	396

S	
<code>\s</code>	112
<code>\sbox</code>	8, 9, 312
<code>\section</code>	3, 8, 512
seq commands:	
<code>\seq_gput_right:Nn</code>	364
<code>\seq_map_inline:Nn</code>	203, 435, 467
<code>\seq_new:N</code>	11
<code>\small</code>	461
<code>sort</code> (option)	2
<code>\string</code>	151, 178, 502
sys commands:	
<code>\sys_if_engine_xetex:TF</code>	66
T	
TeX and L ^A T _E X 2 _ε commands:	
<code>\@auxout</code>	150, 176
<code>\@currentHref</code>	334
<code>\@currentlabel</code>	330
<code>\@ifnextchar</code>	162
<code>\@makecaption</code>	8, 9
<code>\protected@edef</code>	330
<code>\protected@write</code>	9, 107, 176
<code>\zebra@notecount</code>	9, 140, 151
text commands:	
<code>\text_uppercase:n</code>	246
<code>\textbf</code>	389, 442, 447, 450
<code>\textcolor</code>	346, 349, 352, 353, 388, 422
<code>\textdbend</code>	1, 4, 22
<code>\textsc</code>	354
<code>\texttt</code>	356, 390
tl commands:	
<code>\tl_clear:N</code>	433
<code>\tl_gput_right:Nn</code>	264
<code>\tl_head:n</code>	246
<code>\tl_if_blank:nTF</code>	244, 367, 390
<code>\tl_if_empty:NTF</code>	412, 437
<code>\tl_new:N</code>	15, 16, 17, 18, 21, 103, 104, 258, 259
<code>\tl_put_right:Nn</code>	420
<code>\tl_set:Nn</code>	22, 110, 113, 125, 127, 142, 287, 291, 292, 317, 334, 337
<code>\tl_tail:n</code>	246
<code>\tl_to_str</code>	139
<code>\tl_to_str:n</code>	110, 115, 116, 142
<code>\tl_use:N</code>	261, 413, 448
<code>\l_tmpa_tl</code>	110, 112, 117, 142, 143, 290, 291, 292, 293, 325
<code>\l_tmpb_tl</code>	316, 317, 324, 325
<code>\todo</code>	1, 3, 4, 12, 477, 514, 535
U	
use commands:	
<code>\use_i:nn</code>	170
<code>\use_none:n</code>	162, 321, 398
<code>\usepackage</code>	502, 507, 508
W	
<code>\write</code>	146, 150
Z	
zebra internal commands:	
<code>__zebra_allocate_note:nnn</code> .	122, 302
<code>__zebra_apply_pdfstring_defs:</code> ..	198, 224, 377
<code>__zebra_capitalize_type:n</code>	245, 338, 389
<code>\l_zebra_draft_bool</code> .	7, 41, 285, 487
<code>__zebra_if_package_loaded:n</code> ...	32
<code>__zebra_if_package_loaded:nTF</code> ..	71, 83, 216, 332, 376
<code>__zebra_kernel_label:n</code> .	93, 184, 340
<code>__zebra_link:nn</code> ...	165, 214, 222, 392
<code>__zebra_list_entry:nnnnn</code> ..	271, 385
<code>\g_zebra_listbody_all_tl</code>	259, 412, 413
<code>_zebra_margin_note:nn</code>	229, 345
<code>\l_zebra_microtype_expansion_</code> -	
<code>bool</code>	8, 44, 67, 73, 77
<code>__zebra_new_listbody:n</code>	257, 373
<code>__zebra_new_note_type:nn</code>	360
<code>__zebra_new_note_type:nnn</code>	361, 362, 474
<code>_zebra_note:nnn</code>	279, 283, 375
<code>_zebra_note_color:n</code> 249, 287, 388, 423	
<code>\l_zebra_note_color_tl</code>	16, 287, 346, 349, 352, 353
<code>\g_zebra_note_colors_prop</code>	12, 59, 250, 365, 366
<code>__zebra_note_count:n</code>	125, 247, 404, 417, 419, 426
<code>_zebra_note_dedup_key:nnnN</code> 108, 288	
<code>\g_zebra_note_display_prop</code>	98, 298, 305
<code>\l_zebra_note_display_tl</code> ..	103, 125, 131, 299, 306, 331, 347, 351, 355
<code>\g_zebra_note_encounter_prop</code> ...	100, 290, 293, 490
<code>\g_zebra_note_id_int</code> ..	14, 126, 128
<code>\l_zebra_note_key_tl</code>	104, 288, 290, 293, 295, 298, 304, 306, 316
<code>__zebra_note_label:n</code>	182, 342
<code>\g_zebra_note_public_alias_prop</code> ..	13, 206, 382
<code>\l_zebra_note_ref_type_tl</code>	17, 179, 337
<code>_zebra_note_symbol:n</code> .	251, 348, 350

<code>\g_zebra_note_symbols_prop</code>	<code>_zebra_summary_row:n</code>
. 20, 64, 253, 254, 369, 370	415, 436
<code>\g_zebra_note_target_prop</code>	<code>\l_zebra_summary_rows_tl</code>
. 96, 295, 303 18, 420, 433, 437, 448
<code>\l_zebra_note_target_tl</code>	<code>\l_zebra_symbol_tl</code> 21, 22, 255
. 15, 127, 134, 296, 304, 335, 339, 341	<code>_zebra_target:nn</code> 164, 213, 221, 339
<code>\g_zebra_note_total_prop</code>	<code>\l_zebra_total_notes_int</code>
. 102, 143, 316 19, 419, 434, 450
<code>\g_zebra_note_types_seq</code>	<code>_zebra_use_listbody:n</code> 260, 407
. 11, 203, 364, 435, 467	<code>_zebra_write_one_encounter:nn</code>
<code>\l_zebra_pagelinks_bool</code> 9, 47, 85, 219 148, 491
<code>_zebra_pageref:n</code> 166, 215, 218, 392	<code>_zebra_write_zebra_label:n</code> 174, 185
<code>_zebra_pdfstring_note:</code> 161, 205, 207	<code>_zebra_zebra_label_name:n</code>
<code>_zebra_pdfstring_note_opt:w</code> 167, 171, 178, 189
. 162, 163	<code>_zebra_zebra_label_type:n</code> 168, 190
<code>_zebra_prepend:mn</code> 243, 356, 390	<code>_zebra_zebra_ref:n</code> 187, 197
<code>_zebra_print_note_group:n</code> 402, 468	<code>\zebracomment</code> 3, 4
<code>_zebra_print_notes:</code> 431, 489	<code>\zebrafixed</code> 4
<code>_zebra_print_notes_inorder:</code>	<code>\zebranewnote</code> 1,
. 410, 465	4, 16, 473, 477, 479, 480, 481, 482
<code>_zebra_record_note:nnmn</code> 129, 262	<code>\zebranote</code> 3
<code>_zebra_saved_dbend:</code> 155, 158	<code>\zebraplaceholder</code> 4
<code>_zebra_setup_microtype:</code> 69, 82	<code>\zebraquestion</code> 4
<code>_zebra_setup_pagelinks:</code> 211, 228	<code>\zebraref</code> 5, 19, 196, 202
<code>\l_zebra_sort_none_bool</code>	<code>\zebrasetup</code> 5, 7, 19, 475, 509
. 10, 51, 52, 266, 464	<code>\zebratodo</code> 3

Zebra Notes

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

List of notes

Todo 1	5
revise the introduction before submission	
Todo 2 @alice	3
add two or three citations from the latest survey	
Todo 3 @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 4	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?	
Note 3	5
see Todo 1 (that is, Todo 1) on p. 5	