

zebra — Writing Revision Toolkit*

Ruini Xue[†]

v1.6.0 (2026/04/19)

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	18
4.5	Compatibility shim	18
4.6	Two-column demo	19
	Change History	19
	Index	20

*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 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.
The motivation section still feels too vague and could benefit from a concrete running example to guide the reader through the key ideas step by step.

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

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the camera-ready deadline `\todo[bob, carol]{verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards}`.

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the camera-ready deadline [TODO 2@bob, carol: verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards].

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

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

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

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

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

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


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


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


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


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


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


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


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


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


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

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

1  With those items addressed, the conclusion has been rewritten so the argument flows more

1  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

2  draft should be fine.

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

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

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

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

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

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

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

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



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

As in standard L^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 ⚡

This draft still needs work

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 ⚡

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

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 ⚡

Please verify the totals

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   {2026/04/19}
5   {1.6.0}
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 \msg_new:nnn { zebra } { duplicate-note-label }
32 { Note~label~'#1'~used~by~a~different~note;~second~note~gets~
33   its~own~identity. }
34
35 \prg_new_conditional:Npnn \__zebra_if_package_loaded:n #1 { T , F , TF }
36 {
37   \cs_if_exist:cTF { ver@#1.sty }
38     { \prg_return_true: }
39     { \prg_return_false: }
40 }
41
42 \keys_define:nn { zebra }
43 {
44   draft .bool_set:N = \l__zebra_draft_bool,
45   draft .initial:n = true,

```

```

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

```

4.2 Moving-argument deduplication

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

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

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

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

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

```

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

```

```

126 \tl_set:Nx #4
127 {
128   \tl_to_str:n {#1}
129   | \tl_to_str:n {#2}
130   | \l_tmpa_tl
131 }
132 }
133 %% Extract the first \label name from the stringified body.
134 %% Sets #2 to the label name, or clears it if none found.
135 %% The optional \protect prefix covers marks and \protected@write paths.
136 \cs_new_protected:Npn \__zebra_extract_label:nN #1#2
137 {
138   \tl_set:Nx \l_tmpb_tl { \tl_to_str:n {#1} }
139   \tl_set_eq:NN \l_tmpc_tl \l_tmpb_tl
140   \regex_replace_once:nnN
141     { \A .* (? \protect \s* )? \\label \s* \{ ([^}]*) \} .* \Z }
142     { \1 } \l_tmpb_tl
143   \tl_if_eq:NNTF \l_tmpb_tl \l_tmpc_tl
144     { \tl_clear:N #2 }
145     { \tl_set_eq:NN #2 \l_tmpb_tl }
146 }
147 %% Stable key for labeled notes.
148 %% #2 is expected to be already stringified (from regex extraction),
149 %% so no \tl_to_str is applied - otherwise an unexpanded variable
150 %% token would be stringified instead of its value.
151 \cs_new:Npn \__zebra_stable_key:nn #1#2
152 { \tl_to_str:n {#1} | label | #2 }
153 %% Allocate a fresh note: increment the type counter, generate
154 %% a unique hypertarget name, and record the note in the list body.
155 \cs_new_protected:Npn \__zebra_allocate_note:nnn #1#2#3
156 {
157   \int_gincr:c { g__zebra_note_count_#1_int }
158   \tl_set:Nx \l__zebra_note_display_tl { \__zebra_note_count:n {#1} }
159   \int_gincr:N \g__zebra_note_id_int
160   \tl_set:Nx \l__zebra_note_target_tl
161     { zebranote.\int_use:N \g__zebra_note_id_int }
162   \__zebra_record_note:nnnnn
163     {#1}
164     { \l__zebra_note_display_tl }
165     {#2}
166     {#3}
167     { \l__zebra_note_target_tl }
168 }
169 %% Aux-file interface: record a content signature together with the
170 %% originating instance key. A later encounter whose instance key
171 %% differs from the stored one is a replay and is suppressed.
172 %% Re-stringify for catcode normalisation.
173 \cs_new_protected:Npn \zebra@sig #1#2
174 {
175   \tl_set:Nx \l_tmpa_tl { \tl_to_str:n {#1} }
176   \tl_set:Nx \l_tmpb_tl { \tl_to_str:n {#2} }
177   \prop_gput:NVV \g__zebra_note_sig_known_prop \l_tmpa_tl \l_tmpb_tl
178 }
179 \cs_new_protected:Npn \__zebra_write_sig:NN #1#2

```

```

180 {
181   \prop_if_in:NVF \g__zebra_note_sig_written_prop #1
182   {
183     \immediate\write \@auxout
184     { \string\zebra@sig { \tl_use:N #1 } { \tl_use:N #2 } }
185     \prop_gput:NVn \g__zebra_note_sig_written_prop #1 { 1 }
186   }
187 }
188 \cs_if_exist:NTF \dbend
189 {
190   \cs_set_eq:NN \__zebra_saved_dbend: \dbend
191   \cs_undefine:N \dbend
192   \RequirePackage{manfnt}
193   \cs_set_eq:NN \dbend \__zebra_saved_dbend:
194 }
195 { \RequirePackage{manfnt} }
196 \cs_new_protected:Npn \__zebra_pdfstring_note:
197 { \@ifnextchar [ { \__zebra_pdfstring_note_opt:w } { \use_none:n } }
198 \cs_new_protected:Npn \__zebra_pdfstring_note_opt:w [#1] #2 { }
199 \cs_new:Npn \__zebra_target:nn #1#2 {#2}
200 \cs_new:Npn \__zebra_link:nn #1#2 {#2}
201 \cs_new:Npn \__zebra_pageref:n #1 { \pageref {#1} }
202 \cs_new:Npn \__zebra_zebra_label_name:n #1 { #1@zebra }
203 \cs_new:Npn \__zebra_zebra_label_type:n #1
204 {
205   \exp_after:wN \use_i:nn
206   \cs:w r@\__zebra_zebra_label_name:n {#1}\cs_end:
207   { }
208 }
209 \cs_new_protected:Npn \__zebra_write_zebra_label:n #1
210 {
211   \protected@write \@auxout { }
212   {
213     \string\newlabel{\__zebra_zebra_label_name:n {#1}}
214     {\{\exp_not:V \l__zebra_note_ref_type_tl\}\}}
215   }
216 }
217 \cs_new_protected:Npn \__zebra_note_label:n #1
218 {
219   \__zebra_kernel_label:n {#1}
220   \__zebra_write_zebra_label:n {#1}
221 }
222 \cs_new_protected:Npn \__zebra_zebra_ref:n #1
223 {
224   \cs_if_exist:cTF { r@\__zebra_zebra_label_name:n {#1} }
225   { \__zebra_zebra_label_type:n {#1}~\ref{#1} }
226   {
227     \msg_warning:nnn { zebra } { invalid-note-label } {#1}
228     ??
229   }
230 }
231 \NewDocumentCommand \zebraref { m }
232 { \__zebra_zebra_ref:n {#1} }
233 \cs_new_protected:Npn \__zebra_apply_pdfstring_defs:

```

```

234 {
235   \pdfstringdefDisableCommands
236   {
237     \cs_set:Npn \zebraref ##1 { \ref{##1} }
238     \seq_map_inline:Nn \g__zebra_note_types_seq
239     {
240       \cs_set_eq:cN { zebra##1 } \__zebra_pdfstring_note:
241       \prop_if_in:NnT \g__zebra_note_public_alias_prop { ##1 }
242       { \cs_set_eq:cN { ##1 } \__zebra_pdfstring_note: }
243     }
244   }
245 }
246 \cs_new_protected:Npn \__zebra_setup_pagelinks:
247 {
248   \cs_set:Npn \__zebra_target:nn ##1##2 {##2}
249   \cs_set:Npn \__zebra_link:nn ##1##2 {##2}
250   \cs_set:Npn \__zebra_pageref:n ##1 { \pageref {##1} }
251   \__zebra_if_package_loaded:nT { hyperref }
252   {
253     \cs_set:Npn \__zebra_pageref:n ##1 { \pageref* {##1} }
254     \bool_if:NT \l__zebra_pagelinks_bool
255     {
256       \cs_set:Npn \__zebra_target:nn ##1##2 { \hypertarget{##1}{##2} }
257       \cs_set:Npn \__zebra_link:nn ##1##2 { \hyperlink{##1}{##2} }
258     }
259     \__zebra_apply_pdfstring_defs:
260   }
261 }
262 \hook_gput_code:nnn { begindocument } { zebra }
263 { \__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.

```

264 \cs_new_protected:Npn \__zebra_margin_note:nn #1#2
265 {
266   \legacy_if:nTF { @twocolumn }
267   {
268     \mode_if_inner:TF
269     { \marginnote{#2} }
270     {
271       \marginpar
272       [ { \makebox[\marginparwidth]{#1} } ]
273       { \makebox[\marginparwidth]{#2} }
274     }
275   }
276   { \marginnote[#1]{#2} }
277 }

```

```

278 \cs_new:Npn \__zebra_prepend:nn #1#2
279 { \tl_if_blank:nTF {#2} {} {#1#2} }
280 \cs_new:Npn \__zebra_capitalize_type:n #1
281 { \text_uppercase:n { \tl_head:n {#1} } \tl_tail:n {#1} }
282 \cs_new:Npn \__zebra_note_count:n #1
283 { \int_use:c { g__zebra_note_count_#1_int } }
284 \cs_new:Npn \__zebra_note_color:n #1
285 { \prop_item:Nn \g__zebra_note_colors_prop {#1} }
286 \cs_new:Npn \__zebra_note_symbol:n #1
287 {
288   \prop_if_in:NnTF \g__zebra_note_symbols_prop {#1}
289   { \prop_item:Nn \g__zebra_note_symbols_prop {#1} }
290   { \l__zebra_symbol_tl }
291 }
292 \cs_new_protected:Npn \__zebra_new_listbody:n #1
293 { \tl_new:c { g__zebra_listbody_#1_tl } }
294 \tl_new:N \g__zebra_listbody_all_tl
295 \cs_new:Npn \__zebra_use_listbody:n #1
296 { \tl_use:c { g__zebra_listbody_#1_tl } }
297 \cs_new_protected:Npn \__zebra_record_note:nnnnn #1#2#3#4#5
298 {
299   \tl_gput_right:cx
300   {
301     \bool_if:NnTF \l__zebra_sort_none_bool
302     { g__zebra_listbody_all_tl }
303     { g__zebra_listbody_#1_tl }
304   }
305   {
306     \exp_not:N \__zebra_list_entry:nnnnn
307     { \exp_not:n {#1} }
308     {#2}
309     { \exp_not:n {#3} }
310     { \exp_not:n {#4} }
311     {#5}
312   }
313 }
314 %% \__zebra_note:nnn {type}{author}{body}
315 %% Main entry point. Four cases:
316 %% Case 1 - sbox reuse: instance_key found → reuse, render
317 %% Case 2 - stable key: label found, stable_key in prop → suppress
318 %% Case 3 - content sig: sig in .aux data → suppress
319 %% Case 4 - new note: allocate, render
320 \cs_new_protected:Npn \__zebra_note:nnn #1#2#3
321 {
322   \bool_if:NT \l__zebra_draft_bool
323   {
324     \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
325     %% Case 1: sbox reuse (same \inputlineno)
326     \tl_set:Nx \l__zebra_note_key_tl
327     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
328     \prop_get:NVNTF \g__zebra_note_target_prop \l__zebra_note_key_tl
329     \l__zebra_note_target_tl
330     {
331       \prop_get:NVN \g__zebra_note_display_prop \l__zebra_note_key_tl

```

```

332     \l__zebra_note_display_tl
333     \__zebra_render_note:nnn {#1} {#2} {#3}
334 }
335 {
336     %% Extract label and compute content signature
337     \__zebra_extract_label:nN {#3} \l_tmpb_tl
338     \__zebra_content_sig:nnnN {#1} {#2} {#3} \l_tmpa_tl
339     %% Case 2: stable-key suppress (labeled, marks/headers).
340     %% If the stable key exists AND the content signature
341     %% matches, this encounter is a replay → suppress.
342     %% Different content signature = label reuse → warn and
343     %% let Case 4 allocate independently.
344     \bool_set_false:N \l_tmpa_bool
345     \tl_if_empty:NF \l_tmpb_tl
346     {
347         \tl_set:Nx \l__zebra_note_key_tl
348         { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
349         \prop_get:NVNT \g__zebra_note_stable_sig_prop
350         \l__zebra_note_key_tl \l_tmpc_tl
351         {
352             \tl_if_eq:NNTF \l_tmpa_tl \l_tmpc_tl
353             { \bool_set_true:N \l_tmpa_bool }
354             {
355                 \msg_warning:nnV { zebra }
356                 { duplicate-note-label } \l_tmpb_tl
357             }
358         }
359     }
360     %% Case 3: content-sig suppress (labeled, TOC replay).
361     %% Only suppress if the stored instance key differs from
362     %% the current one - same key means it is the original
363     %% note, not a replay.
364     \bool_if:NF \l_tmpa_bool
365     {
366         \prop_get:NVNT \g__zebra_note_sig_known_prop
367         \l_tmpa_tl \l_tmpc_tl
368         {
369             \tl_set:Nx \l_tmpd_tl
370             { \__zebra_instance_key:nnn {#1} {#2} {#3} }
371             \tl_if_eq:NNTF \l_tmpc_tl \l_tmpd_tl
372             { \bool_set_true:N \l_tmpa_bool }
373         }
374     }
375     \bool_if:NF \l_tmpa_bool
376     {
377         %% Case 4: new note - allocate and render
378         \tl_set:Nx \l__zebra_note_key_tl
379         { \__zebra_instance_key:nnn {#1} {#2} {#3} }
380         \__zebra_allocate_note:nnn {#1} {#2} {#3}
381         \prop_gput:NVV \g__zebra_note_target_prop
382         \l__zebra_note_key_tl \l__zebra_note_target_tl
383         \prop_gput:NVV \g__zebra_note_display_prop
384         \l__zebra_note_key_tl \l__zebra_note_display_tl
385         %% For labeled notes: register stable key + write sig,

```

```

386             %% but only if the stable key is not already claimed by
387             %% an earlier note (label-conflict case).
388             \tl_if_empty:NF \l_tmpb_tl
389             {
390                 \tl_set:Nx \l__zebra_note_key_tl
391                 { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
392                 \prop_if_in:NVF \g__zebra_note_stable_sig_prop
393                 \l__zebra_note_key_tl
394                 {
395                     \prop_gput:NVV \g__zebra_note_target_prop
396                     \l__zebra_note_key_tl \l__zebra_note_target_tl
397                     \prop_gput:NVV \g__zebra_note_display_prop
398                     \l__zebra_note_key_tl \l__zebra_note_display_tl
399                     \prop_gput:NVV \g__zebra_note_stable_sig_prop
400                     \l__zebra_note_key_tl \l_tmpa_tl
401                     \tl_set:Nx \l_tmpc_tl
402                     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
403                     \__zebra_write_sig:NN \l_tmpa_tl \l_tmpc_tl
404                 }
405             }
406             \__zebra_render_note:nnn {#1} {#2} {#3}
407         }
408         %% Cases 2-3: suppress - no output
409     }
410 }
411 }
412 %% Full render: hypertarget, target label, margin note, inline text.
413 \cs_new_protected:Npn \__zebra_render_note:nnn #1#2#3
414 {
415     \group_begin:
416     \protected@edef \@currentlabel { \l__zebra_note_display_tl }
417     \__zebra_if_package_loaded:nT { hyperref }
418     { \tl_set:Nx \@currentHref { \l__zebra_note_target_tl } }
419     \tl_set:Nx \l__zebra_note_ref_type_tl
420     { \__zebra_capitalize_type:n {#1} }
421     \__zebra_target:nn { \l__zebra_note_target_tl } {}
422     \exp_args:NV \__zebra_kernel_label:n \l__zebra_note_target_tl
423     \__zebra_margin_note:nn
424     { \textcolor{\l__zebra_note_color_tl}{%
425         { \bfseries \l__zebra_note_display_tl } \kern1pt
426         \__zebra_note_symbol:n {#1} } }
427     { \textcolor{\l__zebra_note_color_tl}{%
428         \__zebra_note_symbol:n {#1} \kern1pt
429         { \bfseries \l__zebra_note_display_tl } } }
430     \cs_set_eq:NN \label \__zebra_note_label:n
431     \textcolor{\l__zebra_note_color_tl}{[ \colorbox[gray]{0.97}{%
432         \textcolor{\l__zebra_note_color_tl}{!70!black}{%
433             \textsc{\MakeLowercase{\MakeUppercase{#1}}}~%
434             \l__zebra_note_display_tl
435             \texttt{\__zebra_prepend:nn {#1}{#2}}:} } #3} }
436     \group_end:
437 }
438 \cs_new_protected:Npn \__zebra_new_note_type:nn #1#2
439 { \__zebra_new_note_type:nnn {#1} {#2} {} }

```

```

440 \cs_new_protected:Npn \__zebra_new_note_type:nnn #1#2#3
441 {
442   \seq_gput_right:Nn \g__zebra_note_types_seq {#1}
443   \prop_if_in:NnF \g__zebra_note_colors_prop {#1}
444   { \prop_gput:Nnn \g__zebra_note_colors_prop {#1} {#2} }
445   \tl_if_blank:nF {#3}
446   {
447     \prop_if_in:NnF \g__zebra_note_symbols_prop {#1}
448     { \prop_gput:Nnn \g__zebra_note_symbols_prop {#1} {#3} }
449   }
450   \int_new:c { g__zebra_note_count_#1_int }
451   \__zebra_new_listbody:n {#1}
452   \exp_args:Nc \NewDocumentCommand { zebra#1 } { 0{ } m }
453   { \__zebra_note:nnn {#1}{##1}{##2} }
454   \__zebra_if_package_loaded:nT { hyperref }
455   { \__zebra_apply_pdfstring_defs: }
456   \cs_if_exist:cTF {#1}
457   { \msg_warning:nnn { zebra } { command-taken } {#1} }
458   {
459     \cs_set_eq:cc {#1} {zebra#1}
460     \prop_gput:Nnn \g__zebra_note_public_alias_prop {#1} { true }
461   }
462 }
463 \cs_new_protected:Npn \__zebra_list_entry:nnnnn #1#2#3#4#5
464 {
465   \par\noindent
466   \textcolor{\__zebra_note_color:n {#1}}{
467     \textbf{\__zebra_capitalize_type:n {#1}~#2}%
468     \tl_if_blank:nF {#3} { \enspace \texttt{\__zebra_prepend:nn {0}{#3}} }}%
469   \nobreak\dotfill
470   \__zebra_link:nn {#5} { \__zebra_pageref:n {#5} }%
471   \par
472   \begingroup
473     \leftskip=2em
474     \rightskip=2em
475     \parindent=0pt
476     \cs_set_eq:NN \label \use_none:n
477     #4\par
478   \endgroup
479 }
480 \cs_new_protected:Npn \__zebra_print_note_group:n #1
481 {
482   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
483   {
484     \par\medskip
485     \__zebra_use_listbody:n {#1}
486   }
487 }
488 \cs_new_protected:Npn \__zebra_print_notes_inorder:
489 {
490   \tl_if_empty:NF \g__zebra_listbody_all_tl
491   { \par\medskip \tl_use:N \g__zebra_listbody_all_tl }
492 }
493 \cs_new_protected:Npn \__zebra_summary_row:n #1

```



```

494 {
495   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
496   {
497     \int_add:Nn \l__zebra_total_notes_int { \__zebra_note_count:n {#1} }
498     \tl_put_right:Nx \l__zebra_summary_rows_tl
499     {
500       \exp_not:N \textcolor
501       { \__zebra_note_color:n {#1} }
502       {#1}
503       \exp_not:N &
504       \__zebra_note_count:n {#1}
505       \exp_not:N \\
506     }
507   }
508 }
509 \cs_new_protected:Npn \__zebra_print_notes:
510 {
511   \tl_clear:N \l__zebra_summary_rows_tl
512   \int_zero:N \l__zebra_total_notes_int
513   \seq_map_inline:Nn \g__zebra_note_types_seq
514   { \__zebra_summary_row:n {##1} }
515   \tl_if_empty:NF \l__zebra_summary_rows_tl
516   {
517     \par\nobreak
518     \noindent\dotfill\par\medskip
519     \nobreak
520     \noindent\textbf{\Large Zebra~Notes}
521     \par \medskip
522     \begin{center}
523       \begin{tabular}{lr}
524         \hline
525         \textbf{Type} & \textbf{Count} \\ \hline
526         \tl_use:N \l__zebra_summary_rows_tl
527         \hline
528         \textbf{Total} & \int_use:N \l__zebra_total_notes_int \\ \hline
529       \end{tabular}
530     \end{center}
531     \legacy_if:nTF { @twocolumn }
532     {
533       \clearpage
534       \onecolumn
535     }
536     { \par \medskip }
537   }
538   \begin{group}
539     \small
540     \noindent{\bfseries List~of~notes}\par
541     \nobreak
542     \bool_if:NTF \l__zebra_sort_none_bool
543     { \__zebra_print_notes_inorder: }
544     {
545       \seq_map_inline:Nn \g__zebra_note_types_seq
546       { \__zebra_print_note_group:n {##1} }
547     }

```

```

548         \endgroup
549     }
550 }

\zebranewnote All note types are created with \zebranewnote.
551 \NewDocumentCommand \zebranewnote { m m 0{} }
552 { \_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.
553 \NewDocumentCommand \zebrasetup { m }
554 { \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 555 \zebranewnote{todo}{purple}
\fixed 556 \colorlet{zebra@fixed@color}{green!50!black}
\comment 557 \zebranewnote{fixed}{zebra@fixed@color}
\placeholder 558 \zebranewnote{comment}{blue}
559 \zebranewnote{note}{violet}
560 \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.

```

561 %% At end of document: print the note summary and list.
562 %% Content signatures are written to .aux inline (at allocation time),
563 %% so no additional end-of-document aux writes are needed.
564 \hook_gput_code:nnn { enddocument } { zebra }
565 {
566     \bool_if:NT \l__zebra_draft_bool
567     { \__zebra_print_notes: }
568 }
569 \ExplSyntaxOff
570 \</package>

```

4.5 Compatibility shim

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

```

571 \<compat>
572 \NeedsTeXFormat{LaTeX2e}
573 \ProvidesPackage{zebra-goodies}
574 [2026/04/19 v1.6.0 Deprecated: use zebra instead]
575 \PackageWarningNoLine{zebra-goodies}
576 {Package 'zebra-goodies' is deprecated.\MessageBreak
577 Use \string\usepackage{zebra} instead}
578 \RequirePackageWithOptions{zebra}
579 \</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.

```

580 <*demo-twocol>
581 \documentclass[twocolumn]{article}
582 \usepackage[paperwidth=16cm,paperheight=13cm,margin=1.2cm]{geometry}
583 \usepackage{zebra}
584 \zebrasetup{symbol/comment=\clubsuit$}
585 \pagestyle{empty}
586 \begin{document}
587 \section{Demo name\comment{revise the name}}
588 This draft still needs work
589 \todo[alice]{\label{zebra:intro}revise the introduction}. The
590 opening paragraph should also explain the main goal more plainly.
591 Add one more citation here \note[bob]{support this claim}. A
592 brief roadmap sentence would also make the structure easier to
593 scan.
594
595 The issue raised in Todo~\ref{zebra:intro} still applies in
596 the conclusion. The table now looks fine
597 \fixed[carol]{alignment corrected}, but one figure is still
598 missing \placeholder[eve]{insert the overview figure}. A short
599 transition would also help the flow. The middle section should
600 probably end with a clearer summary sentence before the
601 discussion begins. That summary can stay compact, but it should
602 signal why the next section matters.
603
604 Please verify the totals \note[frank]{check the numbers} and
605 confirm the wording in the last paragraph
606 \comment[tom]{is this sentence too strong?}. A small typo has
607 already been fixed \fixed[heidi]{typo corrected}. The ending
608 should stay short. The final sentence should return to the main
609 claim rather than repeat background material. You can place
610 \todo[judy]{summarise the findings} anywhere once the narrative
611 is stable.
612
613 One more short paragraph is enough to show how the markers stay
614 readable in a compact two-column layout. The example is
615 intentionally small, but it should still look like a realistic
616 revision pass.
617 \end{document}
618 </demo-twocol>

```

Change History

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

<code>\cs_set:Npn</code>	237, 248, 249, 250, 253, 256, 257	<code>\int_zero:N</code>	512		
<code>\cs_set_eq:NN</code>	190, 193, 240, 242, 430, 459, 476	iow commands:			
<code>\cs_undefine:N</code>	191	<code>\iow_char:N</code>	26, 27		
D					
<code>\dbend</code>	188, 190, 191, 193	K			
<code>\definecolor</code>	4	<code>\kern</code>	425, 428		
<code>\documentclass</code>	581	keys commands:			
<code>\dotfill</code>	469, 518	<code>\keys_define:nn</code>	42, 59, 64		
<code>draft (option)</code>	2	<code>\l_keys_key_str</code>	62, 67		
E				<code>\keys_set:nn</code>	554
<code>\edef</code>	9, 118	L			
<code>\end</code>	530, 531, 617	<code>\label</code> 3, 5, 9, 20, 96, 117, 133, 430, 476, 589			
<code>\endgroup</code>	478, 548	<code>\Large</code>	520		
<code>\enspace</code>	468	<code>\leftskip</code>	473		
exp commands:		legacy commands:			
<code>\exp_after:wN</code>	205	<code>\legacy_if:nTF</code>	266, 532		
<code>\exp_args:Nc</code>	452	M			
<code>\exp_args:NV</code>	422	<code>\makebox</code>	272, 273		
<code>\exp_not:N</code>	306, 500, 503, 505	<code>\MakeLowercase</code>	433		
<code>\exp_not:n</code>	214, 307, 309, 310	<code>\MakeUppercase</code>	433		
<code>\ExplSyntaxOff</code>	569	<code>\marginnote</code>	12, 269, 276		
F				<code>\marginpar</code>	12, 271
<code>final (option)</code>	2	<code>\marginparwidth</code>	272, 273		
<code>\fixed</code>	1, 4, 555, 597, 607	<code>\medskip</code>	484, 491, 518, 521, 537		
<code>font-expansion (option)</code>	2	<code>\MessageBreak</code>	576		
G				<code>\microtypesetup</code>	77
<code>\global</code>	20	mode commands:			
group commands:		<code>\mode_if_inner:TF</code>	268		
<code>\group_begin:</code>	415	msg commands:			
<code>\group_end:</code>	436	<code>\msg_new:nnn</code>	24, 29, 31		
H				<code>\msg_warning:nnn</code>	227, 355, 457
<code>\hline</code>	524, 525, 527, 529	N			
hook commands:		<code>\NeedsTeXFormat</code>	572		
<code>\hook_gput_code:nnn</code>	262, 564	<code>\NewDocumentCommand</code> ...	231, 452, 551, 553		
<code>\hyperlink</code>	257	<code>\newlabel</code>	213		
<code>\hypersetup</code>	91	<code>\nobreak</code>	469, 517, 519, 541		
<code>\hypertarget</code>	256	<code>\noindent</code>	465, 518, 520, 540		
I				<code>\note</code>	1, 3, 555, 591, 604
<code>\immediate</code>	183	O			
<code>\inputlineno</code>	9, 325	<code>\onecolumn</code>	535		
int commands:		options:			
<code>\int_add:Nn</code>	497	<code>draft</code>	2		
<code>\int_compare:nNnTF</code>	482, 495	<code>final</code>	2		
<code>\int_eval:n</code>	115	<code>font-expansion</code>	2		
<code>\int_gincr:N</code>	157, 159	<code>pagelinks</code>	2		
<code>\int_new:N</code>	14, 19, 450	<code>sort</code>	2		
<code>\int_use:N</code>	161, 283, 528	P			
\int_zero:N				512	
iow commands:					
\iow_char:N				26, 27	
K					
\kern				425, 428	
keys commands:					
\keys_define:nn				42, 59, 64	
\l_keys_key_str				62, 67	
\keys_set:nn				554	
L					
\label 3, 5, 9, 20, 96, 117, 133, 430, 476, 589					
\Large				520	
\leftskip				473	
legacy commands:					
\legacy_if:nTF				266, 532	
M					
\makebox				272, 273	
\MakeLowercase				433	
\MakeUppercase				433	
\marginnote				12, 269, 276	
\marginpar				12, 271	
\marginparwidth				272, 273	
\medskip				484, 491, 518, 521, 537	
\MessageBreak				576	
\microtypesetup				77	
mode commands:					
\mode_if_inner:TF				268	
msg commands:					
\msg_new:nnn				24, 29, 31	
\msg_warning:nnn				227, 355, 457	
N					
\NeedsTeXFormat				572	
\NewDocumentCommand ...				231, 452, 551, 553	
\newlabel				213	
\nobreak				469, 517, 519, 541	
\noindent				465, 518, 520, 540	
\note				1, 3, 555, 591, 604	
O					
\onecolumn				535	
options:					
draft				2	
final				2	
font-expansion				2	
pagelinks				2	
sort				2	
P					
\PackageWarningNoLine				575	
pagelinks (option)				2	
\pageref				201, 250, 253	
\pagestyle				585	

zebra internal commands:

__zebra_allocate_note:nnn .	155, 380
__zebra_apply_pdfstring_defs:	233, 259, 455
__zebra_capitalize_type:n	280, 420, 467
__zebra_content_sig:nnnN . .	121, 338
\l_zebra_draft_bool . .	7, 44, 322, 566
__zebra_extract_label:nN . .	136, 337
__zebra_if_package_loaded:n . . .	35
__zebra_if_package_loaded:nTF	74, 86, 251, 417, 454
__zebra_instance_key:nnn	110, 327, 370, 379, 402
__zebra_kernel_label:n . .	96, 219, 422
__zebra_link:nn . . .	200, 249, 257, 470
__zebra_list_entry:nnnnn . .	306, 463
\g_zebra_listbody_all_tl	294, 490, 491
__zebra_margin_note:nn	264, 423
\l_zebra_microtype_expansion_bool	8, 47, 70, 76, 80
__zebra_new_listbody:n	292, 451
__zebra_new_note_type:nn	438
__zebra_new_note_type:nnn	439, 440, 552
__zebra_note:nnn	314, 320, 453
__zebra_note_color:n 284, 324, 466, 501	
\l_zebra_note_color_tl	16, 324, 424, 427, 431, 432
\g_zebra_note_colors_prop	12, 62, 285, 443, 444
__zebra_note_count:n	158, 282, 482, 495, 497, 504
\g_zebra_note_display_prop	100, 331, 383, 397
\l_zebra_note_display_tl 107, 158, 164, 332, 384, 398, 416, 425, 429, 434	
\g_zebra_note_id_int . .	14, 159, 161
\l_zebra_note_key_tl	108, 326, 328, 331, 347, 350, 378, 382, 384, 390, 393, 396, 398, 400
__zebra_note_label:n	217, 430
\g_zebra_note_public_alias_prop	13, 241, 460
\l_zebra_note_ref_type_tl	17, 214, 419
\g_zebra_note_sig_known_prop	102, 177, 366
\g_zebra_note_sig_written_prop	104, 181, 185
\g_zebra_note_stable_sig_prop	106, 349, 392, 399
__zebra_note_symbol:n . .	286, 426, 428
\g_zebra_note_symbols_prop	20, 67, 288, 289, 447, 448
\g_zebra_note_target_prop	99, 328, 381, 395
\l_zebra_note_target_tl	15, 160, 167, 329, 382, 396, 418, 421, 422
\g_zebra_note_types_seq	11, 238, 442, 513, 545
\l_zebra_pagelinks_bool 9, 50, 88, 254	
__zebra_pageref:n . .	201, 250, 253, 470
__zebra_pdfstring_note: 196, 240, 242	
__zebra_pdfstring_note_opt:w	197, 198
__zebra_prepend:nn . . .	278, 435, 468
__zebra_print_note_group:n 480, 546	
__zebra_print_notes:	509, 567
__zebra_print_notes_inorder:	488, 543
__zebra_record_note:nnnnn .	162, 297
__zebra_render_note:nnn 333, 406, 413	
__zebra_saved_dbend:	190, 193
__zebra_setup_microtype: . . .	72, 85
__zebra_setup_pagelinks: . .	246, 263
\l_zebra_sort_none_bool	10, 54, 55, 301, 542
__zebra_stable_key:nn . .	151, 348, 391
__zebra_summary_row:n	493, 514
\l_zebra_summary_rows_tl	18, 498, 511, 515, 526
\l_zebra_symbol_tl	21, 22, 290
__zebra_target:nn . .	199, 248, 256, 421
\l_zebra_total_notes_int	19, 497, 512, 528
__zebra_use_listbody:n	295, 485
__zebra_write_sig:NN	179, 403
__zebra_write_zebra_label:n 209, 220	
__zebra_zebra_label_name:n	202, 206, 213, 224
__zebra_zebra_label_type:n 203, 225	
__zebra_zebra_ref:n	222, 232
\zebracomment	3, 4
\zebrafixed	4
\zebranewnote	1, 4, 18, 551, 555, 557, 558, 559, 560
\zebranote	3
\zebraplaceholder	4
\zebraquestion	4
\zebraref	5, 20, 231, 237
\zebrasetup	5, 7, 20, 553, 584
\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	