

zebra — Writing Revision Toolkit*

Ruini Xue[†]

v1.7.0 (2026/04/25)

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	7
	4.1 Package options	7
	4.2 Moving-argument deduplication	9
	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`.

[†]Email: xueruini@gmail.com

1 Introduction

zebra is a writing revision toolkit. The current release focuses on inline note-taking. Many note-taking and to-do packages exist for L^AT_EX, but most fall into one of two traps: they either offer an overwhelming feature set that tries to cover every conceivable use case, or they clutter the margins with oversized colourful boxes and arrows that make the document hard to read.

zebra takes a different approach. It aims to be *simple*—intuitive commands with only the arguments you actually need—and *good enough*—notes appear inline with a small visual cue in the margin, keeping the document readable while still making annotations easy to spot. Each note type is automatically numbered, and a summary table plus a detailed note list at the end of the document serve as a gentle reminder to address them before the final version.

2 Installation

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.

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

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

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the camera-ready deadline `\todo[bob, carol]{verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards}`.

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the camera-ready deadline [`TODO 2@bob, carol: verify the hyperparameter table against the source code and update any outdated entries and let's check afterwards`].

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

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

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

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

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



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

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

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

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

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

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

We may want to reorganise [NOTE 1: how should we structure the intro?] this part before the final submission. The experimental setup in Section 2 has already been reviewed by a collaborator [COMMENT 1@tom: the setup description looks clear now]. Results are presented in the following tables and figures, but some of them are still missing.

The discussion has been revised [PLACEHOLDER 1@lucy, tom: good job!] and the related work comparison strengthened with two additional references. The list of references still needs a second pass [TODO 3: check bibliography entries for formatting] before we can finalize the submission.

With those items addressed, the conclusion has been rewritten so the argument flows more naturally from the results. [FIXED 1@John: updated the conclusion] The overall structure now matches the revised outline we agreed on last week. [NOTE 2@who: anything else?] If not, the draft should be fine.

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

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

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

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

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



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

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

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

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

As in standard L^AT_EX, labels inside notes are unavailable in `final` mode because the notes themselves are suppressed.

`\zebrasetup` `\zebrasetup{<key=value list>}`

Configures note appearance after loading. Accepted keys:

- `color/<type>=<colour>` — override the colour of a note type.
- `symbol/<type>=<symbol>` — override the margin symbol of a note type.

For example:

```
\zebrasetup{symbol/fixed=\manerrarrow} % like this doc
\zebrasetup{color/todo=red}
```

3.3 Two-column Support

In `twocolumn` documents, the margin symbol is automatically placed on the nearest margin: left margin for the left column, right margin for the right column. No special configuration is needed. This also works correctly in combination with the `twoside` option.

```
\usepackage[paperwidth=16cm,paperheight=15cm,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**]

This draft still needs work [**TODO 1@alice: revise the introduction**]. The opening paragraph should also 1 explain the main goal more plainly. Add one more citation here [**NOTE 1@bob: support this claim**]. A brief roadmap sentence would 1 also make the structure easier to scan.

The issue raised in **Todo 1** still applies in the conclusion. The table now looks fine [**FIXED 1@carol: alignment corrected**], but one figure is still missing 1 PLACEHOLDER 1@eve: insert the overview figure]. A short transition would also help 1 the flow. The middle section should probably end with a clearer summary sentence before the discussion begins. That summary can stay compact, but it should signal why the next section matters.

Please verify the totals [**NOTE 2@frank: check the numbers**] and confirm the wording in the last paragraph [**COMMENT 2@tom: is this sentence too strong?**]. A small typo has already been fixed [**FIXED 2@heidi: typo corrected**].

The ending should stay short. The final sentence should return to the main claim rather than repeat background material. You can place [**TODO 2@judy: summarise the findings**] anywhere once the narrative is stable. 2

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

List of notes

Comment 1.....1
revise the name

Todo 1 @alice.....1
revise the introduction

Note 1 @bob.....1
support this claim

4 Implementation

```
1 <*package>
2 <@@=zebra>
3
4   Version data to start with.
5 \ProvidesExplPackage{zebra}
6   {2026/04/25}
7   {1.7.0}
8   {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* (`\langle type \rangle | \langle author \rangle | \langle body \rangle | \inputlineno`) catches the replay. The second execution reuses the first's allocation and re-renders, so writes that were lost inside the discarded `\sbox` are re-emitted by the actual typesetting pass.

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

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

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

```
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. `\marginnote` (from the `marginnote` package) picks the appropriate side automatically in twocolumn mode and avoids the “Float(s) lost” errors that `\marginpar` triggers when notes appear inside captions, `\title`, or other restricted contexts that are typeset through a `\vbox/\parbox`. Unlike `\marginpar`, `\marginnote` does not reset paragraph parameters or the font, so we replicate `\@marginparreset` ourselves — `\@parboxrestore` normalises paragraph parameters and `\normalfont\normalize` drop oversized fonts inherited from `\title` or `\section`.

```

264 \cs_new_protected:Npn \__zebra_margin_note:nn #1#2
265 {
266   \marginnote
267   [ { \@parboxrestore \normalfont \normalsize #1 } ]
268   { \@parboxrestore \normalfont \normalsize #2 }

```

```

269 }
270 \cs_new:Npn \__zebra_prepend:nn #1#2
271 { \tl_if_blank:nTF {#2} {} {#1#2} }
272 \cs_new:Npn \__zebra_capitalize_type:n #1
273 { \text_uppercase:n { \tl_head:n {#1} } \tl_tail:n {#1} }
274 \cs_new:Npn \__zebra_note_count:n #1
275 { \int_use:c { g__zebra_note_count_#1_int } }
276 \cs_new:Npn \__zebra_note_color:n #1
277 { \prop_item:Nn \g__zebra_note_colors_prop {#1} }
278 \cs_new:Npn \__zebra_note_symbol:n #1
279 {
280   \prop_if_in:NnTF \g__zebra_note_symbols_prop {#1}
281   { \prop_item:Nn \g__zebra_note_symbols_prop {#1} }
282   { \l__zebra_symbol_tl }
283 }
284 \cs_new_protected:Npn \__zebra_new_listbody:n #1
285 { \tl_new:c { g__zebra_listbody_#1_tl } }
286 \tl_new:N \g__zebra_listbody_all_tl
287 \cs_new:Npn \__zebra_use_listbody:n #1
288 { \tl_use:c { g__zebra_listbody_#1_tl } }
289 \cs_new_protected:Npn \__zebra_record_note:nnnnn #1#2#3#4#5
290 {
291   \tl_gput_right:cx
292   {
293     \bool_if:NTF \l__zebra_sort_none_bool
294     { g__zebra_listbody_all_tl }
295     { g__zebra_listbody_#1_tl }
296   }
297   {
298     \exp_not:N \__zebra_list_entry:nnnnn
299     { \exp_not:n {#1} }
300     {#2}
301     { \exp_not:n {#3} }
302     { \exp_not:n {#4} }
303     {#5}
304   }
305 }
306 %% \__zebra_note:nnn {type}{author}{body}
307 %% Main entry point. Four cases:
308 %% Case 1 - sbox reuse: instance_key found → reuse, render
309 %% Case 2 - stable key: label found, stable_key in prop → suppress
310 %% Case 3 - content sig: sig in .aux data → suppress
311 %% Case 4 - new note: allocate, render
312 \cs_new_protected:Npn \__zebra_note:nnn #1#2#3
313 {
314   \bool_if:NT \l__zebra_draft_bool
315   {
316     \tl_set:Nx \l__zebra_note_color_tl { \__zebra_note_color:n {#1} }
317     %% Case 1: sbox reuse (same \inputlineno)
318     \tl_set:Nx \l__zebra_note_key_tl
319     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
320     \prop_get:NVNTF \g__zebra_note_target_prop \l__zebra_note_key_tl
321     \l__zebra_note_target_tl
322     {

```

```

323     \prop_get:NVN \g__zebra_note_display_prop \l__zebra_note_key_tl
324     \l__zebra_note_display_tl
325     \__zebra_render_note:nnn {#1} {#2} {#3}
326   }
327   {
328     %% Extract label and compute content signature
329     \__zebra_extract_label:nN {#3} \l_tmpb_tl
330     \__zebra_content_sig:nnnN {#1} {#2} {#3} \l_tmpa_tl
331     %% Case 2: stable-key suppress (labeled, marks/headers).
332     %% If the stable key exists AND the content signature
333     %% matches, this encounter is a replay → suppress.
334     %% Different content signature = label reuse → warn and
335     %% let Case 4 allocate independently.
336     \bool_set_false:N \l_tmpa_bool
337     \tl_if_empty:NF \l_tmpb_tl
338     {
339       \tl_set:Nx \l__zebra_note_key_tl
340         { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
341       \prop_get:NVNT \g__zebra_note_stable_sig_prop
342         \l__zebra_note_key_tl \l_tmpc_tl
343       {
344         \tl_if_eq:NNTF \l_tmpa_tl \l_tmpc_tl
345         { \bool_set_true:N \l_tmpa_bool }
346         {
347           \msg_warning:nnV { zebra }
348             { duplicate-note-label } \l_tmpb_tl
349         }
350       }
351     }
352     %% Case 3: content-sig suppress (labeled, TOC replay).
353     %% Only suppress if the stored instance key differs from
354     %% the current one - same key means it is the original
355     %% note, not a replay.
356     \bool_if:NF \l_tmpa_bool
357     {
358       \prop_get:NVNT \g__zebra_note_sig_known_prop
359         \l_tmpa_tl \l_tmpc_tl
360       {
361         \tl_set:Nx \l_tmpd_tl
362           { \__zebra_instance_key:nnn {#1} {#2} {#3} }
363         \tl_if_eq:NNTF \l_tmpc_tl \l_tmpd_tl
364         { \bool_set_true:N \l_tmpa_bool }
365       }
366     }
367     \bool_if:NF \l_tmpa_bool
368     {
369       %% Case 4: new note - allocate and render
370       \tl_set:Nx \l__zebra_note_key_tl
371         { \__zebra_instance_key:nnn {#1} {#2} {#3} }
372       \__zebra_allocate_note:nnn {#1} {#2} {#3}
373       \prop_gput:NVV \g__zebra_note_target_prop
374         \l__zebra_note_key_tl \l__zebra_note_target_tl
375       \prop_gput:NVV \g__zebra_note_display_prop
376         \l__zebra_note_key_tl \l__zebra_note_display_tl

```

```

377             %% For labeled notes: register stable key + write sig,
378             %% but only if the stable key is not already claimed by
379             %% an earlier note (label-conflict case).
380             \tl_if_empty:NF \l_tmpb_tl
381             {
382                 \tl_set:Nx \l__zebra_note_key_tl
383                 { \__zebra_stable_key:nn {#1} { \l_tmpb_tl } }
384                 \prop_if_in:NVF \g__zebra_note_stable_sig_prop
385                 \l__zebra_note_key_tl
386                 {
387                     \prop_gput:NVV \g__zebra_note_target_prop
388                     \l__zebra_note_key_tl \l__zebra_note_target_tl
389                     \prop_gput:NVV \g__zebra_note_display_prop
390                     \l__zebra_note_key_tl \l__zebra_note_display_tl
391                     \prop_gput:NVV \g__zebra_note_stable_sig_prop
392                     \l__zebra_note_key_tl \l_tmpa_tl
393                     \tl_set:Nx \l_tmpc_tl
394                     { \__zebra_instance_key:nnn {#1} {#2} {#3} }
395                     \__zebra_write_sig:NN \l_tmpa_tl \l_tmpc_tl
396                 }
397             }
398             \__zebra_render_note:nnn {#1} {#2} {#3}
399         }
400         %% Cases 2-3: suppress - no output
401     }
402 }
403 }
404 %% Full render: hypertarget, target label, margin note, inline text.
405 \cs_new_protected:Npn \__zebra_render_note:nnn #1#2#3
406 {
407     \group_begin:
408     \protected@edef \@currentlabel { \l__zebra_note_display_tl }
409     \__zebra_if_package_loaded:nT { hyperref }
410     { \tl_set:Nx \@currentHref { \l__zebra_note_target_tl } }
411     \tl_set:Nx \l__zebra_note_ref_type_tl
412     { \__zebra_capitalize_type:n {#1} }
413     \__zebra_target:nn { \l__zebra_note_target_tl } {}
414     \exp_args:NV \__zebra_kernel_label:n \l__zebra_note_target_tl
415     \__zebra_margin_note:nn
416     {\textcolor{\l__zebra_note_color_tl}{%
417         {\bfseries\l__zebra_note_display_tl}\kern1pt
418         \__zebra_note_symbol:n {#1}}}
419     {\textcolor{\l__zebra_note_color_tl}{%
420         \__zebra_note_symbol:n {#1}\kern1pt
421         {\bfseries\l__zebra_note_display_tl}}}%
422     \cs_set_eq:NN \label \__zebra_note_label:n
423     \textcolor{\l__zebra_note_color_tl}{[\colorbox[gray]{0.97}{%
424         \textcolor{\l__zebra_note_color_tl !70!black}{%
425             \textsc{MakeLowercase{MakeUppercase#1}}~%
426             \l__zebra_note_display_tl
427             \texttt{\__zebra_prepend:nn {@}{#2}}:}} #3}}%
428     \group_end:
429 }
430 \cs_new_protected:Npn \__zebra_new_note_type:nn #1#2

```

```

431 { \__zebra_new_note_type:nnn {#1} {#2} {} }
432 \cs_new_protected:Npn \__zebra_new_note_type:nnn #1#2#3
433 {
434   \seq_gput_right:Nn \g__zebra_note_types_seq {#1}
435   \prop_if_in:NnF \g__zebra_note_colors_prop {#1}
436     { \prop_gput:Nnn \g__zebra_note_colors_prop {#1} {#2} }
437   \tl_if_blank:nF {#3}
438     {
439       \prop_if_in:NnF \g__zebra_note_symbols_prop {#1}
440         { \prop_gput:Nnn \g__zebra_note_symbols_prop {#1} {#3} }
441     }
442   \int_new:c { g__zebra_note_count_#1_int }
443   \__zebra_new_listbody:n {#1}
444   \exp_args:Nc \NewDocumentCommand { zebra#1 } { 0{} m }
445     { \__zebra_note:nnn {#1}{##1}{##2} }
446   \__zebra_if_package_loaded:nT { hyperref }
447     { \__zebra_apply_pdfstring_defs: }
448   \cs_if_exist:cTF {#1}
449     { \msg_warning:nnn { zebra } { command-taken } {#1} }
450     {
451       \cs_set_eq:cc {#1} {zebra#1}
452       \prop_gput:Nnn \g__zebra_note_public_alias_prop {#1} { true }
453     }
454 }
455 \cs_new_protected:Npn \__zebra_list_entry:nnnnn #1#2#3#4#5
456 {
457   \par\noindent
458   \textcolor{\__zebra_note_color:n {#1}}{%
459     \textbf{\__zebra_capitalize_type:n {#1}~#2}%
460     \tl_if_blank:nF {#3} { \enspace \texttt{\__zebra_prepend:nn {#3}} }}%
461   \nobreak\dotfill
462   \__zebra_link:nn {#5} { \__zebra_pageref:n {#5} }%
463   \par
464   \begingroup
465     \leftskip=2em
466     \rightskip=2em
467     \parindent=0pt
468     \cs_set_eq:NN \label \use_none:n
469     #4\par
470   \endgroup
471 }
472 \cs_new_protected:Npn \__zebra_print_note_group:n #1
473 {
474   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
475     {
476       \par\medskip
477       \__zebra_use_listbody:n {#1}
478     }
479 }
480 \cs_new_protected:Npn \__zebra_print_notes_inorder:
481 {
482   \tl_if_empty:NF \g__zebra_listbody_all_tl
483     { \par\medskip \tl_use:N \g__zebra_listbody_all_tl }
484 }

```

```

485 \cs_new_protected:Npn \__zebra_summary_row:n #1
486 {
487   \int_compare:nNnT { \__zebra_note_count:n {#1} } > { 0 }
488   {
489     \int_add:Nn \l__zebra_total_notes_int { \__zebra_note_count:n {#1} }
490     \tl_put_right:Nx \l__zebra_summary_rows_tl
491     {
492       \exp_not:N \textcolor
493       { \__zebra_note_color:n {#1} }
494       {#1}
495       \exp_not:N &
496       \__zebra_note_count:n {#1}
497       \exp_not:N \\
498     }
499   }
500 }
501 \cs_new_protected:Npn \__zebra_print_notes:
502 {
503   \tl_clear:N \l__zebra_summary_rows_tl
504   \int_zero:N \l__zebra_total_notes_int
505   \seq_map_inline:Nn \g__zebra_note_types_seq
506   { \__zebra_summary_row:n {##1} }
507   \tl_if_empty:NF \l__zebra_summary_rows_tl
508   {
509     \par\nobreak
510     \noindent\dotfill\par\medskip
511     \nobreak
512     \noindent\textbf{\Large Zebra~Notes}
513     \par \medskip
514     \begin{center}
515       \begin{tabular}{lr}
516         \hline
517         \textbf{Type} & \textbf{Count} \\ \hline
518         \tl_use:N \l__zebra_summary_rows_tl
519         \hline
520         \textbf{Total} & \int_use:N \l__zebra_total_notes_int \\ \hline
521         \hline
522       \end{tabular}
523     \end{center}
524     \par \medskip
525     \begin{group}
526       \small
527       \noindent{\bfseries List~of~notes}\par
528       \nobreak
529       \bool_if:NTF \l__zebra_sort_none_bool
530       { \__zebra_print_notes_inorder: }
531       {
532         \seq_map_inline:Nn \g__zebra_note_types_seq
533         { \__zebra_print_note_group:n {##1} }
534       }
535     \end{group}
536   }
537 }

```

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

```

538 \NewDocumentCommand \zebranewnote { m m 0{} }
539   { \_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.

```

540 \NewDocumentCommand \zebrasetup { m }
541   { \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 542 \zebranewnote{todo}{purple}
\fixed 543 \colorlet{zebra@fixed@color}{green!50!black}
\comment 544 \zebranewnote{fixed}{zebra@fixed@color}
\placeholder 545 \zebranewnote{comment}{blue}
546 \zebranewnote{note}{violet}
547 \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.

```

548 %% At end of document: print the note summary and list.
549 %% Content signatures are written to .aux inline (at allocation time),
550 %% so no additional end-of-document aux writes are needed.
551 \hook_gput_code:nnn { enddocument } { zebra }
552   {
553     \bool_if:NT \l_zebra_draft_bool
554       { \_zebra_print_notes: }
555   }
556 \ExplSyntaxOff
557 </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.

```

558 <*compat>
559 \NeedsTeXFormat{LaTeX2e}
560 \ProvidesPackage{zebra-goodies}
561   [2026/04/25 v1.7.0 Deprecated: use zebra instead]
562 \PackageWarningNoLine{zebra-goodies}
563   {Package 'zebra-goodies' is deprecated.\MessageBreak
564     Use \string\usepackage{zebra} instead}
565 \RequirePackageWithOptions{zebra}
566 </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.

```
567 <*demo-twocol>
568 \documentclass[twocolumn]{article}
569 \usepackage[paperwidth=16cm,paperheight=15cm,margin=1.2cm]{geometry}
570 \usepackage{zebra}
571 \zebrasetup{symbol/comment=$\clubsuit$}
572 \pagestyle{empty}
573 \begin{document}
574 \section{Demo name\comment{revise the name}}
575 This draft still needs work
576 \todo[alice]{\label{zebra:intro}revise the introduction}. The
577 opening paragraph should also explain the main goal more plainly.
578 Add one more citation here \note[bob]{support this claim}. A
579 brief roadmap sentence would also make the structure easier to
580 scan.
581
582 The issue raised in Todo~\ref{zebra:intro} still applies in
583 the conclusion. The table now looks fine
584 \fixed[carol]{alignment corrected}, but one figure is still
585 missing \placeholder[eve]{insert the overview figure}. A short
586 transition would also help the flow. The middle section should
587 probably end with a clearer summary sentence before the
588 discussion begins. That summary can stay compact, but it should
589 signal why the next section matters.
590
591 Please verify the totals \note[frank]{check the numbers} and
592 confirm the wording in the last paragraph
593 \comment[tom]{is this sentence too strong?}. A small typo has
594 already been fixed \fixed[heidi]{typo corrected}. The ending
595 should stay short. The final sentence should return to the main
596 claim rather than repeat background material. You can place
597 \todo[judy]{summarise the findings} anywhere once the narrative
598 is stable.
599
600 One more short paragraph is enough to show how the markers stay
601 readable in a compact two-column layout. The example is
602 intentionally small, but it should still look like a realistic
603 revision pass.
604 \end{document}
605 </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	

reference	1	cleanup.	1
v0.5.0		v1.1.1	
General: Use darker color for label	1	General: Per-type color/symbol keys,	
v0.6.0		<code>\zebrasetup</code> .	1
General: Use gray background for label	1	v1.2.0	
v0.7.0		General: Simplify key architecture.	1
General: Move to docstrip	2	v1.3.0	
v0.8.0		General: Rename package to <code>zebra</code> .	1
General: Fix new note demo	4	Rename the microtype expansion	
<code>\zebranewnote</code> : Fix on <code>\global</code> for		option to <code>font-expansion</code> .	1
examples	18	Rename the page-link option to	
v0.8.1		<code>pagelinks/nopagelinks</code> .	1
General: Fix doc	4	v1.4.0	
v0.9.0		General: Support note labels via	
General: Fix legacy bugs and improve		<code>\label</code> , <code>\ref</code> , and <code>\zebraref</code> .	1
implementation	1	v1.5.0	
v0.9.1		General: Fix notes numbering in	
General: Beautify the numbers.	1	moving arguments.	1
v0.9.2		v1.6.0	
General: Faster.	1	General: Numbering in moving	
v1.0.0		arguments is hard.	1
General: <code>expl3</code> , list of notes and		v1.7.0	
compatibility.	1	General: Robust margin notes in	
v1.1.0		twocolumn captions, titles and	
General: Customisable margin symbol,		headings.	1
accurate page numbers, code			

Index

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

Symbols		<code>\bool_set_false:N</code>	54, 70, 336
<code>\</code>	26, 27, 125, 141, 497, 517, 520	<code>\bool_set_true:N</code>	55, 345, 364
<code>\{</code>	125, 141	<code>\l_tmpa_bool</code>	336, 345, 356, 364, 367
<code>\}</code>	125, 141	C	
Numbers		<code>\caption</code>	3, 9
<code>\1</code>	142	<code>\clubsuit</code>	571
A		<code>\cmd</code>	3
<code>\A</code>	141	<code>\colorbox</code>	423
B		<code>\colorlet</code>	4, 543
<code>\begin</code>	514, 515, 573	<code>\comment</code>	1, 3, 4, 542, 574, 593
<code>\begingroup</code>	464, 525	cs commands:	
<code>\bfseries</code>	417, 421, 527	<code>\cs:w</code>	206
bool commands:		<code>\cs_end:</code>	206
<code>\bool_if:NTF</code>	76,	<code>\cs_if_exist:NTF</code>	37, 188, 224, 448
80, 88, 254, 293, 314, 356, 367, 529, 553		<code>\cs_new:Npn</code>	110, 151, 199, 200, 201,
<code>\bool_new:N</code>	7, 8, 9, 10	202, 203, 270, 272, 274, 276, 278, 287	
		<code>\cs_new_eq:NN</code>	96

<code>\cs_new_protected:Npn</code>	72, 121, 136, 155, 173, 179, 196, 198, 209, 217, 222, 233, 246, 264, 284, 289, 312, 405, 430, 432, 455, 472, 480, 485, 501	<code>\int_eval:n</code>	115
<code>\cs_set:Npn</code>	237, 248, 249, 250, 253, 256, 257	<code>\int_gincr:N</code>	157, 159
<code>\cs_set_eq:NN</code>	190, 193, 240, 242, 422, 451, 468	<code>\int_new:N</code>	14, 19, 442
<code>\cs_undefine:N</code>	191	<code>\int_use:N</code>	161, 275, 520
		<code>\int_zero:N</code>	504
		iow commands:	
		<code>\iow_char:N</code>	26, 27
		K	
		<code>\kern</code>	417, 420
		keys commands:	
		<code>\keys_define:mn</code>	42, 59, 64
		<code>\l_keys_key_str</code>	62, 67
		<code>\keys_set:mn</code>	541
		L	
<code>\dbend</code>	188, 190, 191, 193	<code>\label</code>	3, 5, 9, 20, 96, 117, 133, 422, 468, 576
<code>\definecolor</code>	4	<code>\Large</code>	512
<code>\documentclass</code>	568	<code>\leftskip</code>	465
<code>\dotfill</code>	461, 510		
<code>draft</code> (option)	2		
		M	
		<code>\MakeLowercase</code>	425
		<code>\MakeUppercase</code>	425
		<code>\marginnote</code>	12, 266
		<code>\marginpar</code>	12
		<code>\medskip</code>	476, 483, 510, 513, 524
		<code>\MessageBreak</code>	563
		<code>\microtypesetup</code>	77
		msg commands:	
		<code>\msg_new:nnn</code>	24, 29, 31
		<code>\msg_warning:nnn</code>	227, 347, 449
		N	
		<code>\NeedsTeXFormat</code>	559
		<code>\NewDocumentCommand</code>	231, 444, 538, 540
		<code>\newlabel</code>	213
		<code>\nobreak</code>	461, 509, 511, 528
		<code>\noindent</code>	457, 510, 512, 527
		<code>\normalfont</code>	12, 267, 268
		<code>\normalsize</code>	12, 267, 268
		<code>\note</code>	1, 3, 542, 578, 591
		O	
		options:	
		<code>draft</code>	2
		<code>final</code>	2
		<code>font-expansion</code>	2
		<code>pagelinks</code>	2
		<code>sort</code>	2
		P	
		<code>\PackageWarningNoLine</code>	562
		<code>pagelinks</code> (option)	2
		<code>\pageref</code>	201, 250, 253
		<code>\pagestyle</code>	572
<code>\cs_new_protected:Npn</code>	72, 121, 136, 155, 173, 179, 196, 198, 209, 217, 222, 233, 246, 264, 284, 289, 312, 405, 430, 432, 455, 472, 480, 485, 501		
<code>\cs_set:Npn</code>	237, 248, 249, 250, 253, 256, 257		
<code>\cs_set_eq:NN</code>	190, 193, 240, 242, 422, 451, 468		
<code>\cs_undefine:N</code>	191		
D			
<code>\dbend</code>	188, 190, 191, 193		
<code>\definecolor</code>	4		
<code>\documentclass</code>	568		
<code>\dotfill</code>	461, 510		
<code>draft</code> (option)	2		
E			
<code>\edef</code>	9, 118		
<code>\end</code>	522, 523, 604		
<code>\endgroup</code>	470, 535		
<code>\enspace</code>	460		
exp commands:			
<code>\exp_after:wN</code>	205		
<code>\exp_args:Nc</code>	444		
<code>\exp_args:NV</code>	414		
<code>\exp_not:N</code>	298, 492, 495, 497		
<code>\exp_not:n</code>	214, 299, 301, 302		
<code>\ExplSyntaxOff</code>	556		
F			
<code>final</code> (option)	2		
<code>\fixed</code>	1, 4, 542, 584, 594		
<code>font-expansion</code> (option)	2		
G			
<code>\global</code>	20		
group commands:			
<code>\group_begin:</code>	407		
<code>\group_end:</code>	428		
H			
<code>\hline</code>	516, 517, 519, 521		
hook commands:			
<code>\hook_gput_code:nnn</code>	262, 551		
<code>\hyperlink</code>	257		
<code>\hypersetup</code>	91		
<code>\hypertarget</code>	256		
I			
<code>\immediate</code>	183		
<code>\inputlineno</code>	9, 317		
int commands:			
<code>\int_add:Nn</code>	489		
<code>\int_compare:nNnTF</code>	474, 487		

	V	
\vbox	12
	W	
\write	183
	Z	
\Z	141
zebra internal commands:		
_zebra_allocate_note:nnn	..	155, 372
_zebra_apply_pdfstring_defs:	..	233, 259, 447
_zebra_capitalize_type:n	272, 412, 459
_zebra_content_sig:nnnN	..	121, 330
\l_zebra_draft_bool	..	7, 44, 314, 553
_zebra_extract_label:nN	..	136, 329
_zebra_if_package_loaded:n	...	35
_zebra_if_package_loaded:nTF	..	74, 86, 251, 409, 446
_zebra_instance_key:nnn	110, 319, 362, 371, 394
_zebra_kernel_label:n	..	96, 219, 414
_zebra_link:nn	...	200, 249, 257, 462
_zebra_list_entry:nnnnn	..	298, 455
\g_zebra_listbody_all_tl	286, 482, 483
_zebra_margin_note:nn	264, 415
\l_zebra_microtype_expansion_		
bool	8, 47, 70, 76, 80
_zebra_new_listbody:n	284, 443
_zebra_new_note_type:nn	430
_zebra_new_note_type:nnn	431, 432, 539
_zebra_note:nnn	306, 312, 445
_zebra_note_color:n	276, 316, 458, 493	
\l_zebra_note_color_tl	16, 316, 416, 419, 423, 424
\g_zebra_note_colors_prop	12, 62, 277, 435, 436
_zebra_note_count:n	158, 274, 474, 487, 489, 496
\g_zebra_note_display_prop	100, 323, 375, 389
\l_zebra_note_display_tl	107, 158,	
	164, 324, 376, 390, 408, 417, 421, 426	
\g_zebra_note_id_int	..	14, 159, 161
\l_zebra_note_key_tl	108, 318, 320, 323, 339, 342,
	370, 374, 376, 382, 385, 388, 390, 392	
_zebra_note_label:n	217, 422
\g_zebra_note_public_alias_prop	13, 241, 452
\l_zebra_note_ref_type_tl	17, 214, 411
\g_zebra_note_sig_known_prop	102, 177, 358
\g_zebra_note_sig_written_prop	104, 181, 185
\g_zebra_note_stable_sig_prop	..	106, 341, 384, 391
_zebra_note_symbol:n	..	278, 418, 420
\g_zebra_note_symbols_prop	20, 67, 280, 281, 439, 440
\g_zebra_note_target_prop	99, 320, 373, 387
\l_zebra_note_target_tl	15,
	160, 167, 321, 374, 388, 410, 413, 414	
\g_zebra_note_types_seq	11, 238, 434, 505, 532
\l_zebra_pagelinks_bool	9, 50, 88, 254	
_zebra_pageref:n	..	201, 250, 253, 462
_zebra_pdfstring_note:	196, 240, 242	
_zebra_pdfstring_note_opt:w	197, 198
_zebra_prepend:nn	270, 427, 460
_zebra_print_note_group:n	472, 533	
_zebra_print_notes:	501, 554
_zebra_print_notes_inorder:	480, 530
_zebra_record_note:nnnnn	..	162, 289
_zebra_render_note:nnn	325, 398, 405	
_zebra_saved_dbend:	190, 193
_zebra_setup_microtype:	72, 85
_zebra_setup_pagelinks:	..	246, 263
\l_zebra_sort_none_bool	10, 54, 55, 293, 529
_zebra_stable_key:nn	..	151, 340, 383
_zebra_summary_row:n	485, 506
\l_zebra_summary_rows_tl	18, 490, 503, 507, 518
\l_zebra_symbol_tl	21, 22, 282
_zebra_target:nn	..	199, 248, 256, 413
\l_zebra_total_notes_int	19, 489, 504, 520
_zebra_use_listbody:n	287, 477
_zebra_write_sig:NN	179, 395
_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, 538, 542, 544, 545, 546, 547	
\zebranote	3

<code>\zebraplaceholder</code>	4	<code>\zebrasetup</code>	5, 7, 20, <u>540</u> , 571
<code>\zebraquestion</code>	4		
<code>\zebraref</code>	5, 20, 231, 237	<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 @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	