

zebra-goodies — Easy Notes Taking*

Ruini Xue[†]

Released 2019/07/04

Abstract

The `zebra-goodies` package defines a collection of handy macros for taking notes during document writing. It is designed to be very simple and good enough for writing common documents independently or collaboratively. Several commands, `\todo`, `\comment`, `\note`, `\fixed` and `\placeholder`, are provided for quick start. It is also possible to add new note taking commands with `\zebranewnote`.

Contents

1	Introduction	2
2	Installation	2
3	Using the package	2
	3.1 Notes Macros	2
	3.2 Package Options	4
4	Predefined Colors	5
5	Implementation	5
	5.1 Package options	5
	5.2 Main notes macros	5
	5.3 Print summary at end of the document	7
	5.4 Personal Colors	7
	Change History	7
	Index	8

*This file describes version 0.8.0, last revised 2019/07/04.

[†]E-mail: xueruini@gmail.com

1 Introduction

There are quite a few note taking and todo packages available. However, I do not like the interfaces they provided. Some of them create a huge, complex and complicated feature set trying to fit all use cases, while others abuse the margin par to show too many colorful boxes and long text, even worse, some packages link the annotations and the original text with arrows, which results in a really unreadable document.

So, I need a simple and good enough note taking package. With “simple”, I want the commands to be intuitive and equipped with just necessary arguments. With “good enough”, the document’s appearance could be changed a bit to present the notes for easy reading but not too much. Additionally, all notes should be addressed once the file is in final state. As a kind reminder, a summary table presents the notes statistics according to note types at the end of the document.

With these objectives in mind, **zebra-goodies** is created and improved along with my writing activities. Apparently, this package is designed with a heavy personal bias. It is not supposed to be general and perfect.

2 Installation

The package is supplied in `dtx` format and as a pre-extracted zip file. The later is most convenient for most users: simply unzip this in your local `texmf` directory and run `texhash` to update the database of file locations, or just unzip the files into your working directory. If you want to unpack the `dtx` yourself, running `tex zebra-goodies.dtx` will extract the package whereas `latex zebra-goodies.dtx` will extract it and also typeset the documentation.¹

3 Using the package

The package should be loaded as normal in the preamble. The package recognises a number of options, which can also be used in the document body. These are described later in this document.

```
\usepackage[<options>]{zebra-goodies}
```

3.1 Notes Macros

```
\todo[<name>]{<note content>}
```

The basic function provided by **zebra-goodies** is the `\todo` macro. This is used to insert a piece of `todo` note right here, taking a mandatory argument, the `<note content>` of the `todo`, and an optional argument, the `<name>` who should be responsible for this note. For example, the author wants `<name>` to address this todo. This is very helpful during collaboration. The todo note will be typesetted in a predefined color and quoted in the current paragraph along with the famous symbol in margin par.

¹Actually, `latexmk zebra-goodies.dtx` is much better.

A very simple example of a todo note `\todo{how should we go to the lake?}`, and the result will be presented right below the code snippet. Please compare the code and the result.



A very simple example of a todo note [`TODO 1: how should we go to the lake?`], and the result will be presented right below the code snippet. Please compare the code and the result.

And, the following example shows how to set the assignees:

Another one with an assignee is like `\todo[tom]{you should buy the ticket}`, and you will find the same type of notes are numbered. And the assignees are being `\texttt{@}`-ed right before the text. And another one with many assignees `\todo[lucy,jessie]{prepare the foo}`.



Another one with an assignee is like [`TODO 2@tom: you should buy the ticket`], and you will find the same type of notes are numbered. And the assignees are being @-ed right before the text. And another one with many assignees [`TODO 3@lucy,jessie: prepare the foo`].



Since `zebra-goodies` uses running texts, so `\todo` also works in title macros like `\section` and `\caption`.

```
\section{Introduction\todo[jerry]{please figure out a better name}}
\begin{figure}
  \centering
  \caption{Speed vs distance. \todo[need to insert the figure]}
\end{figure}
```

`\note` `\note[name]{note text}`

`\comment` `\comment[name]{comment text}`

`\fixed` `\fixed[name]{fixed text}`

`\placeholder` `\placeholder[name]{placeholder text}`

All these commands share the same syntax as `\todo`, so do the behaviour. They are defined just for convenience for different cases. Nothing magic are behind.

If any of them does not work, it is very likely that they are defined by other packages. In order to avoid unnecessary confusion, `\zebra-goodies` will not `\renewcommand` them in case they are defined. So, it comes to their internal counterparts.

A very simple example of a todo note `\note{how should we go to the lake?}` and another one with an assignee is like `\zebracomment[tom]{you should buy the ticket}`. As you noticed, `\cs{zebracomment}` is used instead `\cs{comment}` here is because `\cs{comment}` has been defined by `\pkg{l3doc}`. Let's move on to other macros. and another one with many assignees `\placeholder[lucy,jessie]{prepare the foo}` and the issue has been `\fixed[John]{good job!}`. We'd like to add more lines to separate the bend symbols. Finally, let's try to use the full version like this, I am going to add a `\todo{a new todo}` and a extended note to be put in another line with useless words as you known like this, this and that and eventually we come to the `\note[who]{note}`.



A very simple example of a todo note [NOTE 1: how should we go to the lake?] and another one with an assignee is like [COMMENT 1@tom: you should buy the ticket]. As you noticed, `\zebracomment` is used instead `\comment` here is because `\comment` has been defined by `l3doc`. Let's move on to other macros. and another one with many assignees [PLACEHOLDER 1@lucy,jessie: prepare the foo] and the issue has been [FIXED 1@John: good job!]. We'd like to add more lines to separate the bend symbols. Finally, let's try to use the full version like this, I am going to add a [TODO 4: a new todo] and a extended note to be put in another line with useless words as you known like this, this and that and eventually we come to the [NOTE 2@who: note].

With these macros, it is very simple to good enough for note taking.

```
\zebratodo \zebratodo[<name>]{<note text>}
```

`\todo` will not be defined if it has been defined by other packages, please use its internal alias, `\zebratodo`. This is how the other notes macros are defined.

```
\zebranote \zebranote[<name>]{<note text>}
```

```
\zebracomment \zebracomment[<name>]{<comment text>}
```

```
\zebrafixed \zebrafixed[<name>]{<fixed text>}
```

```
\zebraplaceholder \zebraplaceholder[<name>]{<placeholder text>}
```

```
\zebranewnote \zebranewnote{<note name>}{<xcolor_name>}
```

You will eventually need a new note type for whatever reasons. `\zebranewnote` needs two arguments to create a new note type. `<note name>` will be used as the note's macro as `\todo`, while the `<xcolor_name>` defines in what color the new note show be, which should be a name instead of xcolor spec. So, it is necessary to define the color with `\definecolor` or `\colorlet` in advance as in the following example.

```
\colorlet{mycyan}{cyan}
\zebranewnote{question}{mycyan}
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?]

3.2 Package Options

`draft`
`final`

`microtype`

These two options are complementary to each other. When no explicit mode is given, `draft` is `true`, in which case all notes will be typesetted as well as the the summary table at end of the document. By setting `draft` to `false` or `final`, notes will not be presented.

Enable the great `microtype` package or not. Default: `true`. I personally love this package very much.

4 Predefined Colors

This is for personal use. I often use `pgfplots` to create plots, however, I do not like the default color. So several are define:

- `zebrablue`
- `zebrared`
- `zebrayellow`
- `zebrapurple`
- `zebragreen`

5 Implementation

```
1 (*package)
2 (@@=zebra_goodies)
   Version data to start with.
3 \ProvidesPackage{zebra-goodies}
4 [2019/07/04 0.8.0 Easy Notes Taking]
5 %\ProvidesExplPackage{zebra-goodies}{2019/07/04}{0.8.0}{Easy Notes Taking}
```

5.1 Package options

Two package options `draft` and `microtype` are created.

```
6 \RequirePackage{kvoptions}
7 \SetupKeyvalOptions{
8   family=zebra,
9   prefix=zebr@,
10  setkeys=\kvsetkeys}
11 \DeclareBoolOption{draft}
12 \DeclareComplementaryOption{final}{draft}
13 \DeclareBoolOption{microtype}
14 \setkeys{zebra}{draft,microtype}
15 \ProcessKeyvalOptions{zebra}
16 \ifzebr@microtype
17   \RequirePackage{microtype}
18 \fi
19 \RequirePackage{xcolor}
20 \RequirePackage{tikzpagenodes}
21 \RequirePackage{marginnote}
22 \ifundefined{dbend}{\RequirePackage{manfnt}}{}
```

5.2 Main notes macros

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

First, the margin note code is from [StackExchange.com](https://stackoverflow.com) with a bit modification.

```
23 \newlength{\zebr@tempdima}
24 \newcount{\zebr@tikzpage}
25 \newcommand*{\zebr@marginnote}[1]{%
26   \begin{tikzpicture}[remember picture, overlay]%
```

```

27 \pgfextractx{\zebr@tempdima}{\pgfpointanchor{current page text area}{center}}%
28 \global\zebr@tempdima=\zebr@tempdima
29 \global\zebr@tikzpage=\oddp@page
30 \end{tikzpicture}%
31 \begingroup
32 \ifdim\zebr@tempdima>0pt \reversemarginpar\fi
33 \marginnote{#1}%
34 \endgroup% resets \reversemarginpar
35 }

```

Then, two meta macros are used to create new types of note.

```

36 \def\zebr@note{%
37 \ifzebr@draft\expandafter\zebr@note@\else\expandafter\@gobblefour\fi}
\zebr@note@ is the core macro to typeset the note. The syntax is:
\zebr@note@{<note name>}{<color>}{<assignee>}{<note text>}
38 \newcommand{\zebr@note@}[4]{%
39 \zebr@marginnote{\textcolor{#2}{\dbend}}%
40 \expandafter\stepcounter{zebr@num@#1}%
41 \textcolor{#2}{[\colorbox[gray]{0.97}{%
42 \textcolor{#2!70!black}{%
43 \textsc{\MakeLowercase{\MakeUppercase#1}}
44 {\expandafter\csname thezebr@num@#1\endcsname}\texttt{#3}:}} #4]}}
45 \newcommand{\zebr@prepend}[2]{%
46 \ifx\relax#2\relax\relax\else#1#2\fi}

```

\zebr@newnote All note types are created with `\zebr@newnote`.

```

47 \def\zebr@noteslist{}
48 \def\zebr@newnote#1#2{%
49 \g@addto@macro\zebr@noteslist{,#1}%
50 \expandafter\newcounter\expandafter{zebr@num@#1}
51 \expandafter\gdef\csname zebr@color@#1\endcsname{#2}
52 \expandafter\newcommand\csname zebra#1\endcsname[2][]{%
53 \zebr@note{#1}{#2}{\zebr@prepend{@}{##1}{##2}}
54 \@ifundefined{#1}{%
55 \expandafter\def\csname #1\endcsname{\csname zebra#1\endcsname}}{%
56 \PackageWarning{zebra}{'\textbackslash{#1}' has been taken.
57 Use '\textbackslash{zebra#1}' instead.}}

```

(End definition for \zebr@newnote. This function is documented on page 4.)

\todo They are defined with `\zebr@newnote`. The colors are picked up from the default ones from `xcolor`.

\note

\fixed

\comment

\placeholder

```

58 \zebr@newnote{todo}{purple}
59 \zebr@newnote{fixed}{teal}
60 \zebr@newnote{comment}{blue}
61 \zebr@newnote{note}{violet}
62 \zebr@newnote{placeholder}{gray}

```

(End definition for \todo and others. These functions are documented on page 2.)

5.3 Print summary at end of the document

A summary table will be inserted automatically at end of the document for easy checking. Each note type is present in a line with its associated color.

```
63 \def\zebr@listnotes{%
64   \ifzebr@draft\zebr@listnotes@fi}
65 \def\zebr@listnotes@{%
66   \def\temp@a{}
67   \edef\temp@noteslist{\expandafter@gobble\zebr@noteslist}
68   \@for\temp@b:=\temp@noteslist\do{%
69     \expandafter\ifnum\expandafter\value\expandafter{zebr@num@\temp@b} > 0%
70     \def\temp@c{\expandafter\csname thezebr@num@\temp@b\endcsname}
71     \edef\temp@d{\noexpand\textcolor{%
72       \expandafter\csname zebr@color@\temp@b\endcsname}{\temp@b}%
73       \noexpand & \temp@c\noexpand}}
74     \expandafter\g@addto@macro\expandafter\temp@a\expandafter{\temp@d}%
75   \fi}
76 \ifx\temp@a\@empty\else%
77   \noindent\dotfill\par
78   \section*{Zebra Notes}
79   \par \medskip
80   \begin{center}
81     \begin{tabular}{lr}
82       \hline
83       \textbf{Type} & \textbf{Number} \\ \hline
84       \temp@a
85       \hline
86     \end{tabular}
87   \end{center}
88 \fi}
89 \AtEndDocument{\zebr@listnotes}
```

5.4 Personal Colors

Several colors are defined for plots.

```
90 \definecolor{zebrablue}{HTML}{4F81BD}
91 \definecolor{zebrared}{HTML}{C0504D}
92 \definecolor{zebragreen}{HTML}{9BBB00}
93 \definecolor{zebrapurple}{HTML}{9F4C7C}
94 \definecolor{zebrayellow}{HTML}{D9CD2E}
95 \definecolor{zebragreen2}{HTML}{00E000}
96 \end{package}
```

Change History

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

reference	1	v0.7.0	
v0.5.0		General: Move to docstrip	2
General: Use darker color for label	1	v0.8.0	
v0.6.0		General: Fix new note demo	4
General: Use gray background for label	1	\zebranewnote: Fix on \global for examples	6

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		\fixed	1, 3, <u>58</u>
\\	73, 83	G	
A		\gdef	51
\AtEndDocument	89	\global	8, 28, 29
B		H	
\begin	26, 80, 81	\hline	82, 83, 85
\begingroup	31	I	
C		\ifdim	32
\caption	3	\ifnum	69
\colorbox	41	\ifx	46, 76
\colorlet	4	K	
\comment	1, 3, 4, <u>58</u>	\kvsetkeys	10
\csname	44, 51, 52, 55, 70, 72	M	
D		\MakeLowercase	43
\dbend	39	\MakeUppercase	43
\DeclareBoolOption	11, 13	\marginnote	33
\DeclareComplementaryOption	12	\medskip	79
\def	36, 47, 48, 55, 63, 65, 66, 70	microtype (option)	4
\definecolor	4, 90, 91, 92, 93, 94, 95	N	
\do	68	\newcommand	25, 38, 45, 52
\dotfill	77	\newcount	24
draft (option)	4	\newcounter	50
E		\newlength	23
\edef	67, 71	\noexpand	71, 73
\else	37, 46, 76	\noindent	77
\end	30, 86, 87	\note	1, 3, <u>58</u>
\endcsname	44, 51, 52, 55, 70, 72	O	
\endgroup	34	options:	
\expandafter	37, 40, 44, 50, 51, 52, 55, 67, 69, 70, 72, 74	draft	4
F		final	4
\fi	18, 32, 37, 46, 64, 75, 88	microtype	4
final (option)	4		

P	
<code>\PackageWarning</code>	56
<code>\par</code>	77, 79
<code>\pgfextractx</code>	27
<code>\pgfpointanchor</code>	27
<code>\placeholder</code>	1, 3, 58
<code>\ProcessKeyvalOptions</code>	15
<code>\ProvidesExplPackage</code>	5
<code>\ProvidesPackage</code>	3
R	
<code>\relax</code>	46
<code>\renewcommand</code>	3
<code>\RequirePackage</code>	6, 17, 19, 20, 21, 22
<code>\reversemarginpar</code>	32, 34
S	
<code>\section</code>	3, 78
<code>\setkeys</code>	14
<code>\SetupKeyvalOptions</code>	7
<code>\stepcounter</code>	40
T	
temp@b commands:	
<code>\temp@b:</code>	68
TeX and L ^A T _E X 2 _ε commands:	
<code>\@empty</code>	76
<code>\@for</code>	68
<code>\@gobble</code>	67
<code>\@gobblefour</code>	37
<code>\@ifundefined</code>	22, 54
<code>\g@addto@macro</code>	49, 74
<code>\ifzebr@draft</code>	37, 64
<code>\ifzebr@microtype</code>	16
<code>\oddp@page</code>	29
<code>\temp@a</code>	66, 74, 76, 84
<code>\temp@b</code>	69, 70, 72
<code>\temp@c</code>	70, 73
<code>\temp@d</code>	71, 74
<code>\temp@noteslist</code>	67, 68
<code>\zebr@listnotes</code>	63, 89
<code>\zebr@listnotes@</code>	64, 65
<code>\zebr@marginnote</code>	25, 39
<code>\zebr@note</code>	36, 53
<code>\zebr@note@</code>	6, 37, 38
<code>\zebr@noteslist</code>	47, 49, 67
<code>\zebr@prepend</code>	45, 53
<code>\zebr@tempdima</code>	23, 27, 28, 32
<code>\zebr@tikzpage</code>	24, 29
<code>\textbackslash</code>	56, 57
<code>\textbf</code>	83
<code>\textcolor</code>	39, 41, 42, 71
<code>\textsc</code>	43
<code>\texttt</code>	44
<code>\todo</code>	1, 2, 3, 3, 4, 4, 5, 58
V	
<code>\value</code>	69
Z	
<code>\zebra-goodies</code>	3
<code>\zebracomment</code>	4, 4
<code>\zebrafixed</code>	4
<code>\zebranewnote</code>	1, 4, 6, 6, 47, 58, 59, 60, 61, 62
<code>\zebranote</code>	4
<code>\zebrplaceholder</code>	4
<code>\zebratodo</code>	4

Zebra Notes

Type	Number
todo	4
fixed	1
comment	1
note	2
placeholder	1
question	1