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.\footnote{Actually, latexmk zebra-goodies.dtx is much better.}

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.

\begin{verbatim}
\usepackage[<options>]{zebra-goodies}
\end{verbatim}

3.1 Notes Macros

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 \textit{note content} of the todo, and an optional argument, the \textit{name} who should be responsible for this note. For example, the author wants \textit{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.
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.

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 \texttt{2@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 \texttt{3@lucy,jessie}: prepare the foo].

Since \texttt{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{\texttt{name}{note text}}
\comment{\texttt{name}{comment text}}
\fixed{\texttt{name}{fixed text}}
\placeholder{\texttt{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, \texttt{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 \texttt{zebra-comment}{tom}{you should buy the ticket}. As you noticed, \cs{zebra-comment} 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}.  

3
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\{\text{name}\}\{\text{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\{\text{name}\}\{\text{note text}\}

\zebracomment \zebracomment\{\text{name}\}\{\text{comment text}\}

\zebrafixed \zebrafixed\{\text{name}\}\{\text{fixed text}\}

\zebraplaceholder \zebraplaceholder\{\text{name}\}\{\text{placeholder text}\}

\zebranewnote \zebranewnote\{\text{note name}\}\{\text{xcolor,ame}\}
You will eventually need a new note type for whatever reasons. \zebranewnote needs two arguments to create a new note type. \text{name} will be used as the note’s macro as \todo, while the \text{xcolor,ame} defines in what color the new note show be, which should be a name instead of \text{xcolor} spec. So, it is necessary to define the color with \text{definecolor} or \text{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. \text{question}[who]{what’s this?}

3.2 Package Options

draft final
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.

microtype
Enable the great \text{microtype} package or not. Default: true. I personally love this package very much.
4 Predefined Colors

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

- \texttt{zebrablue}
- \texttt{zebrared}
- \texttt{zebrayellow}
- \texttt{zebrapurple}
- \texttt{zebragreen}

5 Implementation

\begin{verbatim}
\langle *package \rangle
\langle @@=zebra_goodies \rangle

Version data to start with.
\ProvidesPackage{zebra-goodies}
\[2019/07/04 0.8.0 Easy Notes Taking\]
%\ProvidesExplPackage{zebra-goodies}{2019/07/04}{0.8.0}{Easy Notes Taking}

5.1 Package options

Two package options \texttt{draft} and \texttt{microtype} are created.
\ \RequirePackage{kvoptions}
\ \SetupKeyvalOptions{
  family=zebra,\n  prefix=zebr@,\n  setkeys=kvsetkeys}
\DeclareBoolOption{draft}
\DeclareComplementaryOption{final}{draft}
\DeclareBoolOption{microtype}
\setkeys{zebra}{draft,microtype}
\ProcessKeyvalOptions{zebra}
\ifzebr@microtype
\RequirePackage{microtype}
\fi
\RequirePackage{xcolor}
\RequirePackage{tikzpagenodes}
\RequirePackage{marginnote}
\@ifundefined{dbend}{\RequirePackage{manfnt}}{}

5.2 Main notes macros

Various helper macros are defined before reaching out to the \texttt{todo} commands.

First, the margin note code is from \texttt{StackExchange.com} with a bit modification.
\newlength{\zebr@tempdima}
\newcount{\zebr@tikzpage}
\newcommand*{\zebr@marginnote}[1]{% 
\begin{tikzpicture}[remember picture, overlay]%
Then, two meta macros are used to create new types of note.

\def\zebr@note{%
\ifzebr@draft\expandafter\zebr@note@\else\expandafter\@gobblefour\fi}
\zebr@note@

\newcommand{\zebranewnote}{%
\allnote types are created with \zebranewnote.
\zebranewnote{todo}{purple}
\zebranewnote{fixed}{teal}
\zebranewnote{comment}{blue}
\zebranewnote{note}{violet}
\zebranewnote{placeholder}{gray}
(End definition for \zebranewnote. This function is documented on page 4.)

(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.

\def\zebr@listnotes{%
  \ifzebr@draft\zebr@listnotes@\fi
\def\zebr@listnotes@{%
  \def\temp@a{}
  \edef\temp@noteslist{\expandafter\@gobble\zebr@noteslist}
  \@for\temp@b:=\temp@noteslist\do{%
    \expandafter\ifnum\expandafter\value\expandafter{zebr@num@\temp@b} > 0%
      \def\temp@c{\expandafter\csname thezebr@num@\temp@b\endcsname}
      \edef\temp@d{\textcolor{\expandafter\csname zebr@color@\temp@b\endcsname}{\temp@b} & \temp@c \\}
      \expandafter\g@addto@macro\expandafter\temp@a\expandafter{\temp@d}%
    \fi
  }%
  \ifx\temp@a\@empty\else%
    \noindent\dotfill\par
    \section*{Zebra Notes}
    \par \medskip
    \begin{center}
      \begin{tabular}{lr}
        \hline
        \textbf{Type} & \textbf{Number} \\
        \hline
        \temp@a
        \hline
      \end{tabular}
    \end{center}
  \fi
}\AtEndDocument{\zebr@listnotes}

5.4 Personal Colors

Several colors are defined for plots.

\definecolor{zebrablue}{HTML}{4F81BD}
\definecolor{zebrared}{HTML}{C0504D}
\definecolor{zebragreen}{HTML}{9BBB00}
\definecolor{zebrapurple}{HTML}{9F4C7C}
\definecolor{zebrayellow}{HTML}{D9CD2E}
\definecolor{zebragreen2}{HTML}{00E000}

⟨/package⟩

Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1.0</td>
<td>General: Initial public release</td>
<td>1</td>
</tr>
<tr>
<td>v0.2.0</td>
<td>General: Fix xcolor conflict</td>
<td>1</td>
</tr>
<tr>
<td>v0.3.0</td>
<td>General: Detect command conflicts</td>
<td>1</td>
</tr>
<tr>
<td>v0.4.0</td>
<td>General: Show note number for easy</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>\fixed</td>
<td>1, 3, 58</td>
</tr>
<tr>
<td>G</td>
<td>51</td>
</tr>
<tr>
<td>global</td>
<td>8, 28, 29</td>
</tr>
<tr>
<td>H</td>
<td>82, 83, 85</td>
</tr>
<tr>
<td>I</td>
<td>32</td>
</tr>
<tr>
<td>\iff</td>
<td>69</td>
</tr>
<tr>
<td>\ifx</td>
<td>46, 76</td>
</tr>
<tr>
<td>K</td>
<td>10</td>
</tr>
<tr>
<td>\kvsetkeys</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>43</td>
</tr>
<tr>
<td>\MakeLowercase</td>
<td>43</td>
</tr>
<tr>
<td>\MakeUppercase</td>
<td>33</td>
</tr>
<tr>
<td>\marginnote</td>
<td></td>
</tr>
<tr>
<td>\medskip</td>
<td>4</td>
</tr>
<tr>
<td>\microtype</td>
<td>4</td>
</tr>
<tr>
<td>newcommand</td>
<td>25, 38, 45, 52</td>
</tr>
<tr>
<td>newcount</td>
<td>24</td>
</tr>
<tr>
<td>newcounter</td>
<td>50</td>
</tr>
<tr>
<td>\newlength</td>
<td>23</td>
</tr>
<tr>
<td>noexpand</td>
<td>71, 73</td>
</tr>
<tr>
<td>noindent</td>
<td>77</td>
</tr>
<tr>
<td>\note</td>
<td>1, 3, 58</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
<tr>
<td>options:</td>
<td></td>
</tr>
<tr>
<td>draft</td>
<td>4</td>
</tr>
<tr>
<td>final</td>
<td>4</td>
</tr>
<tr>
<td>\microtype</td>
<td>4</td>
</tr>
<tr>
<td>reference</td>
<td>1</td>
</tr>
<tr>
<td>\v0.5.0</td>
<td>1</td>
</tr>
<tr>
<td>General: Use darker color for label</td>
<td>1</td>
</tr>
<tr>
<td>\v0.6.0</td>
<td>1</td>
</tr>
<tr>
<td>General: Use gray background for label</td>
<td>1</td>
</tr>
<tr>
<td>General: Move to docstrip</td>
<td>2</td>
</tr>
<tr>
<td>\v0.7.0</td>
<td></td>
</tr>
<tr>
<td>General: Fix new note demo</td>
<td>4</td>
</tr>
<tr>
<td>\zebranewnote: Fix on \global for examples</td>
<td>6</td>
</tr>
</tbody>
</table>
\section*{Zebra Notes}

\begin{tabular}{|l|c|}
\hline
\textbf{Type} & \textbf{Number} \\
\hline
todo & 4 \\
fixed & 1 \\
comment & 1 \\
not & 2 \\
placeholder & 1 \\
question & 1 \\
\hline
\end{tabular}