The *todonotes* package*

Henrik Skov Midtiby
henrikmidtiby@gmail.com
April 26, 2021

Abstract

The *todonotes* package allows you to insert to-do items in your document. At any point in the document a list of all the inserted to-do items can be listed with the \listoftodos command.

Contents

1 Introduction ........................................... 2
   1.1 Usage ............................................ 2
   1.2 Package options .................................. 3
   1.3 Options for the todo command ....................... 4
   1.4 Options for the missingfigure command .............. 7
   1.5 Options for the listoftodos command ................. 8
   1.6 Known issues ..................................... 8
   1.7 Things to improve .................................. 11
   1.8 Usage methods ................................... 12

2 Implementation ......................................... 18
   2.1 Declaration of options for the package ............... 18
   2.2 Options for the todo command ....................... 22
   2.3 The main code part ................................ 25

*This document corresponds to todonotes v1.1.4, dated 2021/04/26.
1 Introduction

The todonotes package makes three commands available to the user: \texttt{\todo{}}, \texttt{\missingfigure{}} and \texttt{\listoftodos{}}. \texttt{\todo{}} and \texttt{\missingfigure{}} makes it possible to insert notes in your document about things that has to be done later (todonotes . . . ). I developed the basic functionality of the package while I worked on my bachelor project.

Some alternatives for the todonotes package are:

- **easy-todo**
  - Depends on color, tocloft and ifthen, small feature set.

- **fixmetodonotes**
  - Depends on graphicx, color, transparent, watermark, fix-cm, ulem and tocloft, small feature set.

- **todo**
  - Depends on amssymb, medium feature set.

- **fixme**
  - Large package with a lot of features.

The main reason for considering other packages is that the todonotes package is quite large and relies heavily on tikz. This can slow down compilation of large documents significantly. The mentioned alternatives have a different feature set and does not rely on tikz, which makes them require fewer resources.

1.1 Usage

\texttt{\todo{}}

My most common usage of the todonotes package, is to insert an todonotes somewhere in a latex document. An example of this usage is the command

\texttt{\todo{Make a cake \ldots}}

which renders like. The \texttt{\todo{}} command has this structure: \texttt{\todo{[options]{}{(todo text)}}}. The \texttt{\todo{}} text is the text that will be shown in the todonote and in the list of todos. The optional argument \texttt{options}, allows the user to customize the appearance of the inserted todonotes. For a description of all the options see section 1.3.

\texttt{\missingfigure{}}

The \texttt{\missingfigure{}} command inserts an image containing an attention sign and the given text. The command takes only one argument \texttt{\missingfigure{(text)}}, a text string that could describe what the figure should consist of. An example of its usage could be

\texttt{\missingfigure{Make a sketch of the structure of a trebuchet.}}

which renders like.
Make a sketch of the structure of a trebuchet.

The \listoftodos command inserts a list of all the todos in the current document. \listoftodos takes no arguments. For this document the list of to–do’s looks like.

**Todo list**

- Make a cake .................................................. 2
- Figure: Make a sketch of the structure of a trebuchet. .................. 2
- And a green note .............................................. 4
- Anything but default colors ...................................... 5
- A note without a shadow ........................................ 5
- A note with a shadow ........................................... 5
- Test of option .................................................. 5
- A note with no line connecting it to the placement in the original text. 5
- A todonote placed in the text .................................. 5
- Fill those circles .............................................. 5
- A note with a large font size ................................... 5
- Note with very small font size .................................. 5
- Short note ...................................................... 6
- Short note with prepend ....................................... 6
- Short note with noprepend ..................................... 6
- Testing ......................................................... 6
- Testing author option ......................................... 6
- Testing author option ......................................... 6
- Testing the option inlinewidth ................................ 6
- Testing the option inlinepar .................................. 6
- Figure: Testing a long text string ................................ 7
- Figure: Testing a long text string ................................ 7
- Figure: Add an image ......................................... 7
- Figure: Testing .................................................. 8
- Figure: Testing figcolor ....................................... 8
- Test of newly defined command ................................ 12
- Test of newly defined command, requesting a green color ........... 12
- 1: A numbered todonote ....................................... 13
- 2: Another numbered todonote .................................. 13
Comment [HSM1]: Testing first time. 
Comment [HSM2]: Testing second time. 

Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard latex distributions. 

Some lines with a decreased line spacing. This is accomplished without using any special packages.

2do
Translation
Translation
1.8.9. A numbered todo.
1. Smart notes with links back to the list of todos.
fix text

The \todototoc command adds an entry to the table of contents for list of todos. The command should be placed right before the \listoftodos command.

1.2 Package options

disable
If the option disable is passed to the package, the macros usually defined by the package (\todo, \listoftodos and \missingfigure) are defined as macros with no effect, and thus all inserted notes are removed.

obeyDraft, obeyFinal
When the option obeyDraft is given, the package checks if the one of the options draft, draftcls or draftclanonfoot is given (this option is usually given to the documentclass). If the draft option is given, the functionality of the package is enabled and otherwise the effect of the package is disabled. The option obeyFinal does something similar, except that the todonotes package is only disabled if the final option given.

Use translations of the text strings ”List of todos” and ”Missing figure”. The default is to use none of these options, which results in english text strings. Currently the following languages are supported: catalan, croatian, danish, dutch, english, french, german, ngerman, italian, portuguese, spanish and swedish.

Adds a small colored square in front of all items in the Todo list. The color of the square is the same as the fill color of the inserted todonote. This can be useful if there are different types of todos (insert reference, explain in detail, ...) where the color of the inserted todonote marks the type of todo.

color
These options sets the default colors for the todo command. There are four colors that can be specified. The border color (default bordercolor=black) around the inserted text, the color behind the inserted text (default backgroundcolor=orange), the color of the inserted text (default textcolor=black) and the color of the line connecting the inserted textbox with the current location in the text (default linecolor=orange). Setting the color option to val passes this value on to the background and line color options. The specified colors must be valid according to the xcolor package.

tickmarkheight
tickmarkheight=length set the height of the tickmark at the location where
the todo is inserted, the default value is 0pt.

textwidth \textwidth=length sets the width of a todo item in the margin to \textwidth. The width of inline todonotes will always be the same as the current line width.

textsize \textsize=value sets the default text size of the inserted todonotes to the given value. Value is the "name" of the used font size, e.g. if the desired font size is \tiny use \textsize=tiny. The default value is \textsize=normalsize.

prependcaption The prependcaption option triggers a special behaviour of the \caption=val option for the todo command, where the given value val is inserted in the inserted todonote.

loadshadowlibrary If given, the \shadows tikz library will be loaded. This is required for adding shadows to the inserted todonotes.

shadow If the \shadow option is given, the inserted todonotes will by default be displayed with a gray shadow. For this to work, the package option loadshadowlibrary must be provided. I expect that the option will trigger problems with tikz versions prior to 2.0.

dvistyle When a document with todonotes is compiled with plain latex (to a dvi-file), there is an issue with the visual appearance\footnote{The problem is placement of text inside the colored boxes.}. The option dvistyle changes the appearance of the inserted todonotes to avoid this problem.

figwidth The \figwidth=length option sets the default width of the figure inserted by the \missingfigure command. The default value is \linewidth.

1.3 Options for the todo command

There are several options that can be given to the \todo command. All the options are described here and often I have included examples of the change in visual appearance. Default values for these options can be set using the setuptodonotes command.

\setuptodonotes{fancyline, color=blue!30}

disable The disable option can be given directly to the todo command. If given the command has no effect.

color These options set the color that is used in the current todo command. The color classes is the same as used in the color package options, see section 1.2.

backgroundcolor Default values can be set by the color options when the todonotes package is loaded. The todo notes inserted in this paragraph is created with the command

\todo[color=green!40]{And a green note}. The color of the inserted note could be used to mark different types of tasks (insert references, explain something in detail, ...) this could be streamlined by defining new commands like below.

\newcommand{\insertref}[1]{\todo[color=green!40]{#1}}
\newcommand{\explainindetail}[1]{\todo[color=red!40]{#1}}

An example that uses all of the color options is given below.

And a green note

Anything but default colors
If the package is loaded with the `shadow` option, a shadow is added to all notes, otherwise no shadows are shown. This global setting can be overwritten by the `shadow` and `noshadow` options to the todo command. To insert a note with and without shadows, use the `shadow` and `noshadow` options as shown here:

```latex
\todo[shadow]{A note with a shadow}
\todo[noshadow]{A note without a shadow}
```

`tickmarkheight` set the height of the tickmark at the location where the todo is inserted. An example of the option in use is shown here.

```latex
\todo[tickmarkheight=0.1cm]{Test of option.}
```

If you want to get rid of the line connecting the inserted note with the place in the text where the note occurs in the latex code, the option `noline` can be used.

```latex
\todo[noline]{A note with no line ...}
```

It is possible to place a todonote inside the text instead of placing it in the margin, this could be desirable if the text in the note has a considerable length.

```latex
\todo[inline]{A todonote placed in the text}
```

Another usage for the inline option is when you want to add a todonote to a figure caption.

```latex
\begin{wrapfigure}{r}[20mm]{40mm}
\begin{tikzpicture}
\draw[red] (0, 0) circle(0.45);
\draw[green] (1, 0) circle(0.45);
\draw[blue] (2, 0) circle(0.45);
\end{tikzpicture}
\caption{A text explaining the image.
\todo[inline]{Fill those circles \ldots}}
\end{wrapfigure}
```

`size=val` changes the size of the text inside the todonote. The commands used to create the notes below are

```latex
\todo[size=\Large]{A note with a large font size.} and
\todo[inline, size=\tiny]{Note with very small font size.}
```

When the option `nolist` is given, the todo item will not appear in the list of
todos.

The **caption** option enables the user to specify a short description of the todo note that are inserted in the list of todos instead of the full todonote text. A very long and tedious note that cannot be on one line in the list of todos.

\todo[\caption={Short note}]{A very long and tedious note that cannot be on one line in the list of todos.}

The effect of this option is altered with the package option **prependcaption** or the **prepend** / **noprepend** option for the todo command. The options **prepend** and **noprepend** can be used for setting whether a given caption should be prepended to the todonote or not. Globally this can be set using the **prependcaption** option for the package. Below is the effect of the option shown using the code:

\todo[\prepend, \caption={Short note with prepend}]{A very long and tedious note that cannot be on one line in the list of todos.}
\todo[\noprepend, \caption={Short note with noprepend}]{A very long and tedious note that cannot be on one line in the list of todos.}

The **fancyline** option inserts a curved arrow, pointing from the inserted note to the insertion point. The option is used like this:

\todo[\fancyline]{Testing.}

The **author** option takes a parameter, the name of the author. The given name is inserted in the todonote.

\todo[author=Xavier]{Testing author option.}
\todo[author=Xavier, inline]{Testing author option.}

The **inlinewidth** option makes it possible to alter the width of inline todos. Testing the option inlinewidth.

\todo[inline, inlinewidth=5cm]{Testing the option inlinewidth.}

The options **inlinepar** and **nolinepar** (default) determine whether the inserted inline todonote is surrounded with **par** macros that enforce new lines prior to and after inserted inline todonotes. Testing the option inlinepar. This is a line after the inserted todonote.

\todo[inline, inlinewidth=5cm, noinlinepar]{Testing the option inlinepar.}
1.4 Options for the \texttt{missingfigure} command

\texttt{figwidth}  The \texttt{figwidth=\textit{length}} option sets the width of the figure inserted by the \texttt{\textcolor{red}{missingfigure}} command. Length values below 6\textit{cm} might trigger some problems with the visual appearance. Try to compare the default of the missing figure command, when the option is given or not.

\begin{verbatim}
\texttt{\textcolor{red}{missingfigure}[figwidth=6cm]{Testing a long text string}}
\end{verbatim}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{Testing a long text string}
\end{figure}

Another usage of the option is when \texttt{\textcolor{red}{missingfigure}} is used in the \texttt{wrapfigure} environment.

\begin{verbatim}
\begin{wrapfigure}[1]{r}[2cm]{6cm}
\texttt{\textcolor{red}{missingfigure}[figwidth=6cm]{Add an image \ldots}}
\end{wrapfigure}
\end{verbatim}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Add an image \ldots}
\end{figure}

\texttt{figheight}  The \texttt{figheight=\textit{length}} option changes the height of the inserted missing figure. The default height is 4\textit{cm} and using values lower than this might cause the warning sign to pop out of the gray area.

8
The \texttt{\textbackslash missingfigure} command can be used to insert missing figures. The \texttt{figcolor=color} options sets the background color of inserted missing figures. The default color is \texttt{black!40}.

\begin{verbatim}
\missingfigure[figheight=6cm]{Testing a long text string}
\end{verbatim}

\begin{verbatim}
\missingfigure[figcolor=white]{Testing figcolor}
\end{verbatim}

\section*{1.5 Options for the listoftodos command}

The \texttt{\textbackslash listoftodos} command takes one optional argument, that defines the name of the inserted list of todos.

\begin{verbatim}
\listoftodos[I can be called anything]
\end{verbatim}

\section*{1.6 Known issues}

\subsection*{1.6.1 Package loading order}

The todonotes package requires the following packages.
When todonotes are loaded in the preamble, the package checks if these packages all are loaded. If that is not the case it loads the missing packages with no options given. If you want to give some specific options to some of these packages, you have to load them before the todonotes package, otherwise you will get an "Option clash" error when latex works on the document.

If both the menukeys and the xcolor (with the option table) package should be loaded, the following order must be used.

```latex
\usepackage[table]{xcolor}
\usepackage{todonotes}
\usepackage{menukeys}
```

The following packages must be loaded before the todonotes package:

- polyglossia

### 1.6.2 Wrapping of long lines in list of todos

When a document is compiled with latex (and not pdflatex) long items in the list of todos are not wrapped into several lines, and do instead continue to the right out of the page.

### 1.6.3 Conflicts with the amsart documentclass

The amsart document class redefines some internal commands that is used by the todonotes package, this will cause a malfunctioning \listoftodos command. The following code to circumvent the problem was given by Dan Luecking on comp.text.tex

```latex
\makeatletter
\providecommand\@dotsep{5}
\makeatother
\listoftodos\relax
```

NOT TESTED NOT TESTED NOT TESTED

Dominique suggests the following workaround.

```latex
\makeatletter
\providecommand\@dotsep{5}
\def\listtodoname{List of Todos}
\def\listoftodos\@starttoc{tqd}\listtodoname
\makeatother
```
1.6.4 Unknown option ”remember picture”

If latex throws the error

```
Package tikz Error: I do not know what to do with the option ‘‘remember picture’’.
```

It probably means that your latex installation is outdated, as only newer versions of latex driver for tikz supports the remember picture option. For additional info consult ”Section 9.2.2 Producing PDF Output” in the tikz manual.  [http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf](http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf)

1.6.5 Todonotes wrongly placed in the margin

When using some document classes or packages, the todonotes inserted in the page margin can be placed quite oddly. This is often caused by a wrong value of the \marginparwidth length. Try using the code below in your preamble to see if this cures the problem.

```
\setlength{\marginparwidth}{2cm}
```

If the todonotes are inserted in the wrong margin, the solution is the \reversemarginpar command. When this command is issued the following marginpars (which todonotes relies on) is inserted in the other margin.

1.6.6 Reduce number of warnings

If the width of the inserted todonotes is forced to be larger than the available space in the margin, a lot of warnings are issued. This can be reduced with the following code.

```
\usepackage[textwidth=3.7cm]{todonotes}
\setlength{\marginparwidth}{3.7cm}
```

1.6.7 Todonotes in footnotes

Placing todonotes in footnotes throws a lot of errors. Richard Stanton comes with the following work around.

```
\renewcommand{\marginpar}{\marginnote}
```

1.6.8 Trouble with classicthesis.sty

Problem description on tex.stackexchange.com.

Solution by Stefan Kottwitz. The problem is caused by the redefinition of \marginpar in classicthesis.sty. \marginpar is used by todonotes. It can be fixed by restoring the original meaning, if you don’t need the classicthesis marginpar style. Just add this to your document preamble: \let\marginpar\oldmarginpar
1.6.9 todos in math and floating environments

It is not possible to insert todonotes into math environments or floating environments like figure or table. By replacing the marginpar command with a marginnote this can be enabled. For todos inside math environments the todo must be placed in an mbox or \texttt{mbox} command.

\documentclass{article}
\usepackage{todonotes}
\usepackage{marginnote}
\let\marginpar\marginnote

\begin{document}
\begin{equation}
a^2=b^2+c^2 \texttt{\todo{Test}}
\end{equation}
\begin{figure}
\centering
\caption{Caption\todo{tests}.}
\label{fig}
\end{figure}

1.7 Things to improve

This is a list of things I consider to improve sometime in the future. It have not been done yet as I lack the time or skills to implement them. Patches with implementations of these tasks will be appreciated and might be included in the package if it will improve the package quality.

1.7.1 Owner information

Option for the todo command.
\todo[owner={Fabrice}]{Stuff}

Add info on who ”owns” the current todo. Idea: Fabrice Niessen

1.7.2 Due date

Option for the todo command.
\todo[due=2008-12-07]{Stuff}

Add info on when the current todo is due. Might be enhanced by a time line of the todos that have a due date assigned. Idea: Fabrice Niessen
1.7.3 Mark accomplished todos
\todo[done]{Stuff}

Idea: Fabrice Niessen

1.8 Usage methods
In this section I have collected some different methods to use the \texttt{todonotes} package.

1.8.1 Define new commands with fixed options
Often there is a need for marking different classes of things to do (add reference, rewrite, \ldots). One way to do this, is to define some new commands as shown below (idea from Florent B.).
\begin{verbatim}
\newcommand{\addref}{\todo[color=red!40]{Add reference.}}
\newcommand{\rewrite}[1]{\todo[color=green!40]{#1}}
\end{verbatim}

To distinguish between the different types of todos, the \texttt{todonotes} package can be loaded with the \texttt{colorinlistoftodos} option, which adds small colored squares to the list of todos.
\begin{verbatim}
\usepackage[colorinlistoftodos]{todonotes}
\end{verbatim}

1.8.2 Define new commands with arbitrary default options
If you do not like the default values of the standard todo command, it is possible to define a new command with the similar functionality of \texttt{\todo} with custom default values.
\begin{verbatim}
\newcommand{\todoredefined}[2]{\todo[color=red, #1]{#2}}
\end{verbatim}

The new command can now be used like shown below
\begin{verbatim}
\todoredefined{Test of newly defined command.}
\todoredefined[color=green]{Test of newly defined command, requesting a green color.}
\end{verbatim}

This can be done with all the accepted options for the \texttt{\todo} command.

1.8.3 Enumerate todonotes
If the inserted todonotes should be enumerated, it is possible to define a new command with the desired behaviour.
\begin{verbatim}
\newcounter{todocounter}
\newcommand{\todonum}[2]{\stepcounter{todocounter}\todo[#1]{\thetodocounter: #2}}
\end{verbatim}
The idea is to define a new counter `todocounter`, and insert the value of the counter in each todonote. The new command can be used like:

```
\todonum{A numbered todonote.}
\todonum{Another numbered todonote.}
```

### 1.8.4 Comments "a la Word"

Fabrice Niessen sent me the following use case. The idea is to define a new command `\mycomment` which adds a counter and optionally the initials of the author to the inserted todonote.

```latex
\newcounter{mycomment}
\newcommand{\mycomment}[2][{}]{
\refstepcounter{mycomment}\
\setstretch{0.7}\
\todo[color={red!100!green!33},size=small]{\bf Comment [\uppercase{#1}\mycomment]:}~#2}
```

The command `\mycomment[HSM]{Testing first time.}` is displayed like shown in the left margin, and another call of the command is added below `\mycomment[HSM]{Testing second time.}`.

### 1.8.5 Combination with the fixme package

Thomas Arildsen has mailed me this use case. Check the documentation for the `fixme` package, as the code below relies directly on it (the `\FDUser` command is augmented when `\begin{document}` is reached).

```latex
\usepackage[user,nomargin]{fixme}
\usepackage{todonotes}
\newcommand{\FXUser}[2][{}]{\todo[inline,size=small]{{\bfseries #1:} #2}}
```

### 1.8.6 Altering the line spacing of todonotes

The `setspace` package lets you alter the line spacing of smaller sections of your document. The primary construct is the `spacing` environment, which is demonstrated below.

```latex
\begin{spacing}{0.5}
Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard latex distributions.
\end{spacing}
```
Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard LaTeX distributions.

Using the \texttt{spacing} environment we can define a new \texttt{todonote} command using the code below:

\begin{verbatim}
newcommand{\smalltodo}{
\begin{spacing}{0.5}
\end{spacing}}
\end{verbatim}

Todones with decreased line spacing can now be inserted as follows:

Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard LaTeX distributions.

\smalltodo{Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard LaTeX distributions.}

A different approach is given by Vitaly.

\begin{verbatim}
newcommand{\tinytodo}{
\begin{minipage}{\linewidth}
\renewcommand{\baselinestretch}{0.5}\selectfont#2\par
\end{minipage}}
\end{verbatim}

It looks like seen here.

\tinytodo{Some lines with a decreased line spacing. This is accomplished without using any special packages.}

1.8.7 Marking new / old sections

Sometimes a whole section has to be marked by some means. You might want to try the following.

\begin{verbatim}
\todo[inline, caption={Some text}]{
\begin{minipage}{\linewidth}
Some text that might differ from the text given to the caption option.
\end{minipage}
}
\end{verbatim}

It is important to add the \texttt{caption=\{text\}} option, otherwise \LaTeX{} will try to embed a \texttt{minipage} in the table of contents which triggers an error. Inside the \texttt{minipage} environment almost anything could be placed, except for other \texttt{todo} commands.

To streamline use the following command was suggested by Stefan Pinnow.

\begin{verbatim}
newcommand{\todoin}{
\begin{minipage}{\textwidth-4pt}
\textwidth-4pt#2\end{minipage}}
\end{verbatim}

\texttt{\todoin}
This example renders like
\begin{verbatim}
\texttt{Some text.}
\begin{align}
\sin(\theta)^2 + \cos(\theta)^2 = 1
\end{align}
\texttt{A formula and a list}
\begin{itemize}
\item An item
\end{itemize}
\end{verbatim}
Some text.

\[\sin(\theta)^2 + \cos(\theta)^2 = 1\] (1)

A formula and a list

\begin{itemize}
\item An item
\end{itemize}

1.8.8 Link to list of todos

Using the \texttt{hyperref} package it is possible to add a link from the inserted todonotes to the list of todos. The example were supplied by Andreas Plank.

\begin{verbatim}
% Define a counter for the inserted todonotes.
\newcounter{todoListItems}
\newcommand{\todoTrans}[2][\ ]{
  \addtocounter{todoListItems}{1}
  \todo[\caption={\protect\hyperlink{todo\thetodoListItems}{}Translation},
  #1]{#2 \hfill \hyperlink{todo\thetodoListItems}{$\uparrow$}}
}

% The idea behind the code is to embed a \texttt{hyperref} in each entry in the list of todos. In the todonotes a link to the entry in the list of todos is inserted as an arrow that points upwards. Using the \texttt{\todoTrans} command like below, the following two notes have been inserted.
\begin{verbatim}
\todoTrans{papirsflyver}
\todoTrans[inline]{damplokomotiv}
\end{verbatim}
\end{verbatim}

16
1.8.9 Numbered todonotes

The inserted todonotes can be argumented with the current subsubsection number. The code is shown below.

\newcommand{\ntodo}[2]{\todo[#1]{\thesubsubsection{}. #2}}

By changing \thesubsubsection to \thesection, the current section number can be inserted instead of the subsubsection number. The result looks like. Which were generated by the code
\ntodo{A subsection numbered todo.}.

1.8.10 Combining several modifications

Manduca have combined several of the modifications above into a highly specialized todo command. She uses the code:

\newcounter{todoListItems}
\newcommand{\sstodo}[2]{\addtocounter{todoListItems}{1}\todo[caption={\protect\hypertarget{todo\thetodoListItems}{}\thesection. #2}, #1]{\begin{spacing}{1} \hfill \hyperlink{todo\thetodoListItems}{#2} \end{spacing} }}

Using this approach it is possible to customize the behavior of the inserted notes to a very high degree.

1.8.11 Alter the appearance of the list of todos

Marco Daniel gives the following example of how to add section numbers to the elements in the list of todos. The code is slightly modified from http://tex.stackexchange.com/questions/18838/replacing-page-number-with-other-counter-in-listoftodos. An example of the modified list of todos is shown below, the complete code example is given in the example directory.

| This is a note . . . . . . . . . . . . . . . . . . . . . . see 1.1 at p. 2 |
| This is another note . . . . . . . . . . . . . . . . . . . . see 1.2 at p. 4 |

1.8.12 Tikz externalization issues

Using the tikz externalization framework together with todonotes can lead to some problems. One solution is to disable the externalization just before the todo command is issued and then reactivate externalization afterwards. The ruggedtodo handles this deactivation and reactivation.

\usetikzlibrary{external}
\tikzexternalize
\newcommand{\ruggedtodo}[2]{\tikzexternaldisable\todo[#1]{#2}\tikzexternalenable}
1.8.13 Highlight text to fix

Tobias Winchen provides the following example on how to highlight text related to the inserted todonote. Example wrong text continues here. Notice that the code relies on the soul package.

\makeatletter
\if@todonotes@disabled
\newcommand{\hlfix}{2}{#1}
\else
\newcommand{\hlfix}{2}{\texthl{#1}\todo{#2}}
\fi
\makeatother
Example \hlfix{wrong text}{fix text} continues here.
2 Implementation

Identifies the package and loads the packages dependences.

\ProvidesPackage{todonotes}[2018/11/22]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}
\RequirePackage{xcolor}
\RequirePackage{tikz}
\usetikzlibrary{positioning}
\RequirePackage{calc}

Implement a function for setting up the todonotes package.
\newcommand\setuptodonotes[1]{\presetkeys{todonotes}{#1}{}}

Some default values are set
\newcommand{\@todonotes@text}{}
\newcommand{\@todonotes@backgroundcolor}{orange}
\newcommand{\@todonotes@textcolor}{black}
\newcommand{\@todonotes@linecolor}{orange}
\newcommand{\@todonotes@bordercolor}{black}
\newcommand{\@todonotes@tickmarkheight}{0cm}
\newcommand{\@todonotes@textwidth}{\marginparwidth}
\newcommand{\@todonotes@textsize}{\normalsize}
\newcommand{\@todonotes@figwidth}{\linewidth}
\newcommand{\@todonotes@figheight}{4cm}
\newcommand{\@todonotes@figcolor}{black!40}
\AtBeginDocument{
\ifx\undefined\phantomsection
\newcommand{\phantomsection}{}
\fi
\ifdim\marginparwidth<2cm
\PackageWarning{todonotes}{The length marginparwidth is less than 2cm and will most likely cause issues with the appearance of inserted todonotes. The issue can be solved by adding a line like \setlength{\marginparwidth}{2cm} prior to loading the todonotes package.} \else\fi%
}

2.1 Declaration of options for the package

In this part the various options for the package are defined.

Define the default text strings and set localization options for the danish and german languages.
\newcommand{\@todonotes@todolistname}{Todo list}
\newcommand{\@todonotes@MissingFigureText}{Figure}
\newcommand{\@todonotes@MissingFigureUp}{Missing}
\newcommand{\@todonotes@MissingFigureDown}{figure}
\newcommand{\@todonotes@SetTodoListName}[1]
\renewcommand{\todonotes@todolistname}{\#1}
\newcommand{\todonotes@SetMissingFigureText}{\#1}
\newcommand{\todonotes@MissingFigureText}{\#1}
\newcommand{\todonotes@SetMissingFigureUp}{\#1}
\newcommand{\todonotes@SetMissingFigureDown}{\#1}
\newcommand{\@todonotes@SetTodoListName}{Llista de feines pendent}
\@todonotes@SetMissingFigureText{Figura}
\@todonotes@SetMissingFigureUp{Figura}
\@todonotes@SetMissingFigureDown{pendent}
\@todonotes@reverseMissingFigureTrianglefalse
\DeclareOptionX{catalan}{%}
\DeclareOptionX{croatian}{%}
\DeclareOptionX{danish}{%}
\DeclareOptionX{dutch}{%}
\DeclareOptionX{english}{%}
\DeclareOptionX{french}{%}
\DeclareOptionX{german}{%}
Create a counter, for storing the number of inserted todos.
\newcounter{@todonotes@numberoftodonotes}

Toggle whether the package should obey the global draft option.
\newif{\if@todonotes@obeyDraft}
\DeclareOptionX{obeyDraft}{\@todonotes@obeyDrafttrue}
\newif{\if@todonotes@isDraft}
\DeclareOptionX{draft}{\@todonotes@isDrafttrue}
\DeclareOptionX{draftcls}{\@todonotes@isDrafttrue}
\DeclareOptionX{draftclsnofoot}{\@todonotes@isDrafttrue}
\newif{\if@todonotes@obeyFinal}
\DeclareOptionX{obeyFinal}{\@todonotes@obeyFinaltrue}
\newif{\if@todonotes@isFinal}
\DeclareOptionX{final}{\@todonotes@isFinaltrue}
\DeclareOptionX{finalnofoot}{\@todonotes@isFinaltrue}

Make it possible to disable the functionality of the package. If this option is
given, the commands \texttt{\textbackslash todo} and \texttt{\textbackslash listoftodos} are defined as commands with
no effect. (But you can still compile you document with these commands).
\newif{\if@todonotes@disabled}
\DeclareOptionX{disable}{\@todonotes@disabledtrue}
Show small boxes in the list of todos with the color of the inserted todonotes.
\newif{\if@todonotes@colorinlistoftodos}
\DeclareOptionX{colorinlistoftodos}{\@todonotes@colorinlistoftodostrue}
The default style behaves bad when compiled with latex (some text placement
problems). The dvistyle option, changes the visual behavior to avoid this text
placement problem.
\newif{\if@todonotes@dviStyle}
\DeclareOptionX{dvistyle}{\@todonotes@dviStyletrue}
Create a color option.
\define@key{todonotes.sty}{color}{
\renewcommand{\@todonotes@backgroundcolor}{#1}
\renewcommand{\@todonotes@linecolor}{#1}}
Make the background color of the notes as an option.
\define@key{todonotes.sty}{backgroundcolor}{\renewcommand{\@todonotes@backgroundcolor}{#1}}
Make the text color of the notes as an option.
\define@key{todonotes.sty}{textcolor}{\renewcommand{\@todonotes@textcolor}{#1}}
Make the line color of the notes as an option.
\define@key{todonotes.sty}{linecolor}{\renewcommand{\@todonotes@linecolor}{#1}}
Make the color of the notes box color as an option.
\define@key{todonotes.sty}{bordercolor}{\renewcommand{\@todonotes@bordercolor}{#1}}
Make the height of the line start marking as an option.
\define@key{todonotes.sty}{tickmarkheight}{\renewcommand{\@todonotes@tickmarkheight}{#1}}
Set whether short captions given as arguments to the todo command should be
included in the inserted todonote.
\newif{\if@todonotes@prependcaptionglobal}
\@todonotes@prependcaptionglobalfalse
\DeclareOptionX{prependcaption}{\@todonotes@prependcaptionglobaltrue}
Make the text width as an option.
\define@key{todonotes.sty}{textwidth}{\renewcommand{\@todonotes@textwidth}{#1}}
Make the text size as an option, accept both size and textsize.
\define@key{todonotes.sty}{textsize}{\renewcommand{\@todonotes@textsize}{#1}}
\define@key{todonotes.sty}{size}{\renewcommand{\@todonotes@textsize}{#1}}

22
Add option for shadows behind the inserted notes
\newif\if@todonotes@shadowlibraryloaded
\@todonotes@shadowlibraryloadedfalse
\DeclareOptionX{loadshadowlibrary}{%
\usetikzlibrary{shadows}%
\@todonotes@shadowlibraryloadedtrue}
\newcommand{\@todonotes@shadowenabledbydefault}{noshadow}
\DeclareOptionX{shadow}{%
\renewcommand{\@todonotes@shadowenabledbydefault}{shadow}}

Add option for the default width of the figure inserted with \missingfigure.
\define@key{todonotes.sty}{figwidth}{\renewcommand{\@todonotes@figwidth}{#1}}
\define@key{todonotes.sty}{figheight}{\renewcommand{\@todonotes@figheight}{#1}}
\define@key{todonotes.sty}{figcolor}{\renewcommand{\@todonotes@figcolor}{#1}}

Make the text width as an option.
% Finally process the given options.
% \begin{macrocode}
\ProcessOptionsX*
% \end{macrocode}

If the obeyDraft is given, check whether one of the draft, draftcls or
draftclsnofoot options are given and enable or disable the functionality of this
package. If the obeyFinal option is given together with the final option the
todonotes are disabled. The disable option will overrule the effect of obeyDraft.
\if@todonotes@disabled
\else
\if@todonotes@obeyDraft
\@todonotes@disabledtrue
\if@todonotes@isDraft
\@todonotes@disabledfalse
\fi
\fi
\if@todonotes@obeyFinal
\@todonotes@disabledtrue
\if@todonotes@isFinal
\@todonotes@disabledfalse
\fi
\fi

2.2 Options for the todo command

In this part the various options for commands in the package are defined. Set an
arbitrarily fill color
\gdef\@todonotes@currentlinecolor{\@todonotes@linecolor}%
\gdef\@todonotes@currentbackgroundcolor{\@todonotes@backgroundcolor}%
\gdef\@todonotes@currenttextcolor{\@todonotes@textcolor}%
Toggle whether there is a shadow behind the inserted notes.
\newif\if@todonotes@useshadow
\define@key{todonotes}{shadow}{\@todonotes@useshadowtrue}
\define@key{todonotes}{noshadow}{\@todonotes@useshadowfalse}

Define height of the inserted tickmark.
\define@key{todonotes}{tickmarkheight}{\renewcommand{\@todonotes@tickmarkheight}{#1}}
\renewcommand{\@todonotes@tickmarkheight}{#1}

Set a relative font size
\newcommand{\@todonotes@sizecommand}{}
\define@key{todonotes}{size}{\renewcommand{\@todonotes@sizecommand}{#1}}
\renewcommand{\@todonotes@sizecommand}{#1}

Should the todo item be disabled?
\newif\if@todonotes@localdisable
\define@key{todonotes}{disable}{\@todonotes@localdisabletrue}
\define@key{todonotes}{nodisable}{\@todonotes@localdisablefalse}

Should the todo item be included in the list of todos?
\newif\if@todonotes@appendtolistoftodos
\define@key{todonotes}{list}{\@todonotes@appendtolistoftodostrue}
\define@key{todonotes}{nolist}{\@todonotes@appendtolistoftodosfalse}

Should the todo item be displayed inline?
\newif\if@todonotes@inlinenote
\define@key{todonotes}{inline}{\@todonotes@inlinenotetrue}
\define@key{todonotes}{notineline}{\@todonotes@inlinenotefalse}
\newif\if@todonotes@prependcaption
\define@key{todonotes}{prepend}{\@todonotes@prependcaptiontrue}
\define@key{todonotes}{noprepend}{\@todonotes@prependcaptionfalse}

Should the note in the margin be connected to the insertion point in the text?
\newif\if@todonotes@line
\define@key{todonotes}{line}{\@todonotes@linetrue}
\define@key{todonotes}{notineline}{\@todonotes@linefalse}

Should the connection between note and insertion point be drawn in a fancy way? (does only work if line=true)
Author option.

Should the text in the list of todos be different from the text in the todonote?

Change the current figure width and height.

Change the width of an inline note.

Change if inline note is written to a separate line or not.

Preset values of the options
2.3 The main code part

Here is the actual macros defined. If the option "disable" was passed to the package define empty commands.

\if@todonotes@disabled%
\newcommand{\listoftodos}[1][]{}
\newcommand{\@todo}[2][]{}
\newcommand{\missingfigure}[2][]{}
\else % \if@todonotes@disabled
Define the \listoftodos command and define the appearance of the list of todos.
\newcommand{\listoftodos}[1][\@todonotes@todolistname]{\@ifundefined{chapter}{\section*{#1}}{\chapter*{#1}} \@starttoc{tdo}}
\newcommand{\l@todo}{\@dottedtocline{1}{0em}{2.3em}}
Define styles used by the todo command
\tikzstyle{notestyleraw} = [
\draw=\@todonotes@currentbordercolor,  
\fill=\@todonotes@currentbackgroundcolor,  
\text=\@todonotes@currenttextcolor,  
\line width=0.5pt,  
\text width = \@todonotes@textwidth - 1.6 ex - 1pt,  
\inner sep = 0.8 ex,  
\rounded corners=4pt]
\@todo Define the \@todo command.
\newcommand{\@todo}[2][]{%
Use the global value for determining the default prepend behavior.
\if@todonotes@prependcaptionglobal%
\@todonotes@prependcaptiontrue%
\else%
\@todonotes@prependcaptionfalse%
\fi%
Store the original text for later usage and parse the given options.
\renewcommand{\@todonotes@text}{#2}%
\renewcommand{\@todonotes@caption}{#2}%
\setkeys{todonotes}{#1}%

Add shadows to the inserted todonotes.
\if@todonotes@useshadow%
  \if@todonotes@shadowlibraryloaded%
    \tikzstyle{notestyle} = \[notestyleraw,\]
    general shadow={shadow xshift=0.5ex, shadow yshift=-0.5ex,\}
    opacity=1,fill=black!50}\]%
  \else%
    \PackageWarning{todonotes}{Trying to put a shadow below a todonote,}
    but the loadshadowlibrary option was not given when loading
    the todonotes package}%
  \tikzstyle{notestyle} = \[notestyleraw\]%
\else%
  \tikzstyle{notestyle} = \[notestyleraw\]%
\fi%

Update notestyles
\tikzstyle{notestyleleft} = \[%
  notestyle,\%
  left\]%
\tikzstyle{connectstyle} = \[%
  thick,\%
  draw=\@todonotes@currentlinecolor\]%
\tikzstyle{inlinenotestyle} = \[%
  notestyle,\%
  text width=\@todonotes@inlinewidth - 1.6 ex - 1 pt\]%

If the option disable is given to the command, no output is generated.
\if@todonotes@localdisable%
\else%
Add the item to the list of todos. When the option colorinlistoftodos is given
to the package a small colored square is added in front of the text.
\addtocounter{todonotes@numbertodonotes}{1}%
\if@todonotes@appendtolistoftodos%
  \phantomsection%
\fi%
\if@todonotes@captiongiven%
  \@todonotes@addElementToListOfTodos%
\fi%
\if@todonotes@captiongiven%
  \if@todonotes@prependcaption%
  \renewcommand{\@todonotes@caption}{#2}%
  \fi%
\fi%
\if@todonotes@captiongiven%
  \@todonotes@addElementToListOfTodos%
\fi%
\fi%

Prepend the short caption given if it is requested
\if@todonotes@captiongiven%
  \if@todonotes@prependcaption%
  \fi%
\fi%

27
Place the todonote as indicated by the options (inline or in a marginpar), below is the code for the inline placement.

\if@todonotes@inlinenote
\@todonotes@drawInlineNote\else\@todonotes@drawMarginNoteWithLine\fi\if@todonotes@inlinenote\fi\if@todonotes@localdisable\}

\newcommand\addElementToListOfTodos{\if@todonotes@colorinlistoftodos\addcontentsline{tdo}{todo}{\fcolorbox{\@todonotes@currentbordercolor}{\@todonotes@currentbackgroundcolor}{\textcolor{\@todonotes@currentbackgroundcolor}{o}}}\else\fi\}
drawMarginNoteWithLine Define helper function drawMarginNoteWithLine.

When the todonote should be placed inside a marginpar, the code below is applied. First is the current location in the document stored; this enables us later to connect this point with the inserted todonote.

\ifvmode\vspace*{-\parskip} % backup if we are already in vertical mode
\vskip-\baselineskip % (and don’t loose that space after a pagebreak ... %
\noindent
\fi
\begin{tikzpicture}[remember picture, overlay, baseline=-0.75ex]
\node [coordinate] (inText) {};\end{tikzpicture}
\marginpar[\{\@todonotes@drawMarginNote\@todonotes@drawLineToLeftMargin\}]{\{\@todonotes@drawMarginNote\@todonotes@drawLineToRightMargin\}}

In the book documentclass (which is a twoside layout), the \marginpar marginpar macro takes two arguments \marginpar[left]{right}. If both arguments are given, latex will decide in which side the margin note has to be inserted, and then use the corresponding commands.

\addElementToListOfTodos Define helper function addElementToListOfTodos.

\newcommand\addElementToListOfTodos{\if@todonotes@colorinlistoftodos\addcontentsline{tdo}{todo}{\fcolorbox{\@todonotes@currentbordercolor}{\@todonotes@currentbackgroundcolor}{\textcolor{\@todonotes@currentbackgroundcolor}{o}}}\else\fi\}
\def\@todonotes@caption{}%
\else
\addcontentsline{tdo}{todo}{\@todonotes@caption}%
\fi}

drawInlineNote Define helper function useSizeCommand.
\newcommand{\@todonotes@useSizeCommand}{%\ifcsname \expandafter\string\@todonotes@sizecommand\endcsname\csname \expandafter\string\@todonotes@sizecommand\endcsname\else\@todonotes@sizecommand\fi%}
drawInlineNote Define helper function drawInlineNote.
\newcommand{\@todonotes@drawInlineNote}{%\if@todonotes@dviStyle{%\if@todonotes@inlinepar\par\noindent\fi%\begin{tikzpicture}[remember picture]%;\draw node[linenotestyle]{\if@todonotes@authorgiven%\noindent \@todonotes@author: \@todonotes@text\else\@todonotes@text\fi};%\end{tikzpicture}%;\if@todonotes@inlinepar\par\fi}%;\else%\begin{tikzpicture}[remember picture]%;\draw node[linenotestyle,font=\@todonotes@useSizeCommand]{\if@todonotes@authorgiven%\noindent \@todonotes@author: \@todonotes@text\else\@todonotes@text\fi}%;\end{tikzpicture}%;\if@todonotes@inlinepar\par\fi}%;\else%\begin{tikzpicture}[remember picture]%;\draw node[linenotestyle]{\if@todonotes@authorgiven%\noindent \@todonotes@author: \@todonotes@text\else\@todonotes@text\fi}%
\end{tikzpicture}%;\if@todonotes@inlinepar\par\fi}%;\fi}%
drawMarginNote Define helper function drawMarginNote.
\newcommand{\@todonotes@drawMarginNote}{%\if@todonotes@dviStyle{%\begin{tikzpicture}[remember picture]%;\end{tikzpicture}%}
\begin{tikzpicture}[remember picture]%;\end{tikzpicture}%
\begin{tikzpicture}[remember picture]%;\end{tikzpicture}%
\begin{tikzpicture}[remember picture]%;\end{tikzpicture}%
drawLineToRightMargin Define helper function drawLineToRightMargin.
\newcommand{\@todonotes@drawLineToRightMargin}{% 
\if@todonotes@line
\if@todonotes@fancyline
\tikz[remember picture, overlay]{% 
\tikzstyle{both}=[line width=3pt, draw, opacity=0.15] 
\tikzstyle{line}=[shorten >=5pt, line cap=round] 
\tikzstyle{head}=[shorten >=-1pt, dash pattern=on 0pt off 1pt, ->] 
\foreach \s in {line,head} {% 
\draw[\s,\s] (inNote.north west).. controls +(0:0) and +(90:1.5)..([yshift=1ex] inText);% 
}"% 
\} 
\else 
\begin{tikzpicture}[remember picture, overlay] 
\draw[connectstyle] (\[yshift=-0.2cm + \@todonotes@tickmarkheight\] inText)% 
-|- (\[xshift=-0.2cm\] inNote.west)% 
\end{tikzpicture}% 
\fi} 
\} 
\fi 
\end{tikzpicture}% 
\hbadness \originalHbadness%
\fi} 
\end{tikzpicture}% 
\hbadness \originalHbadness%
\fi}%
drawLineToLeftMargin Define helper function `drawLineToLeftMargin`.
\newcommand{\@todonotes@drawLineToLeftMargin}{%
  \if@todonotes@line%
    \if@todonotes@fancyline%
      \tikz[remember picture, overlay]{%
        \tikzstyle{both}=[line width=3pt, draw, opacity=0.15]%
        \tikzstyle{line}=[shorten >=5pt, line cap=round]%
        \tikzstyle{head}=[shorten >=-1pt, dash pattern=on 0pt off 1pt,->]
        \foreach \s in {line, head} {%
          \draw[both, \s] (inNote.north east).. controls +(0:0) and +(90:1.5)..(\[yshift=1ex\] inText);%
        \}
      \}
    \else%
      \begin{tikzpicture}[remember picture, overlay]%
        \draw[connectstyle] (\[yshift=-0.2cm + \@todonotes@tickmarkheight\] inText) -| (\[yshift=-0.2cm\] inText) -| (\[xshift=0.2cm\] inNote.east) -| (inNote.east);%
      \end{tikzpicture}%
    \fi%
  \fi}%
\fi}%
\missingfigure Defines the `\missingfigure` macro.
\newcommand{\missingfigure}[2][2]{%
  \setkeys{todonotes}{#1}%
  \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
  \par
  \noindent
  \hfill
  \begin{tikzpicture}%
    \draw[fill=\@todonotes@currentfigcolor, draw = black!40, line width=2pt]
    (-2, -0.5*\@todonotes@currentfigheight-0.5cm)
    rectangle + (\@todonotes@currentfigwidth, \@todonotes@currentfigheight);%
  \draw (2, -0.5) node[right, text]
    \width=\@todonotes@currentfigwidth-4.5cm, font=\@todonotes@useSizeCommand} {#2};%
  \draw[red, fill=white, rounded corners = 5pt, line width=10pt]
    (30:2cm) -- (150:2cm) -- (270:2cm) -- cycle;
  \draw (0, 0.3) node {\@todonotes@MissingFigureUp};%
  \draw (0, -0.3) node {\@todonotes@MissingFigureDown};%
  \end{tikzpicture}\hfill
  \null\par
\} % Ending `\missingfigure` command
\fi % Ending `@todonotes@ifdisabled
\dototoc \quad \text{Inserts a reference to the list of todos in the table of contents. If } chapter \text{ is defined, } chapter \text{ is used as level otherwise will } section \text{ be used. The } \dotototoc \text{ command respects the disable option.}

514 \newcommand{\dototoc}{\%}
515 \if@todonotes@disabled
516 \else
517 \addcontentsline{toc}{\@ifundefined{chapter}{section}{chapter}}{\@todonotes@todolistname}\
519 \fi
520 \}

\todo \quad \text{Define the } \todo \text{ command as a redirection to } @\todo.

521 \newcommand{\todo}[2][\%]{\%}
522 \if@inlabel \leavevmode \fi
523 \if@noskipsec \leavevmode \fi
524 \if@todonotes@inlinepar
525 \ifhmode
526 \@bsphack
527 \if@todonotes@vmodefalse
528 \else
529 \@savsf\@m
530 \@savsk\z@
531 \@todonotes@vmodetrue
532 \fi
533 \fi
534 \{%@todo[#1]{#2}\%
535 \@esphack\%
536 \if@todonotes@vmode \par \fi
537 \else\%
538 \%@todo[#1]{#2}\%
539 \fi\%
540 \newif\if@todonotes@vmode
Change History

0.1
General: The first version of the package .......................... 1

0.2
General: Updated the option handling of the package ........ 1

0.2.1
General: Slightly modified by Kjell Magne Fauske to support notes in the left margin (for documentstyle book). ........ 1

0.2.2
General: Added a missingfigure command ....................... 1

0.2.3
General: Made a dependency on the calc package ............... 1

0.3
General: Delayed the requirements for the hyperref package until begin document and added an optional argument to the todo command for adding inline todonotes (Idea from Patrick Toche) .......................... 1

0.3.1
General: Added some options to the todo macro (Idea: Patrick Toche) and made the listoftodos point at the inserted todos and not only the current / previous section, subsection or figure using the phantomsection macro. ........ 1

0.4
General: Modified the behaviour of the inline todonotes, to avoid empty lines around the inline todonotes. ....................... 1

0.4.1
General: Added the option colorinlistoftodos which inserts a small box with the used fillcolor of the todonotes in the list of todos. ....................... 1

0.4.2
General: Fixed a bug with the disable option to the package. . 1

0.5
General: Created a dtx file containing both source code and documentation of the package .......................... 1

0.5.1
General: Updated the documentation .............................. 1

0.5.2
General: Fixed a bug that prevented the usage of the option french for babel. Bug report by Thomas Braun. ........ 1

0.6
General: Added the caption option to the todo command. ........ 1

0.6.1
General: Added a new usecase with decreased line spacing. .... 1

0.6.2
General: Added a usecase by Fabrice Niessen. .................. 1

0.7
General: Added language options on request from Peter Zimmermann. ....................... 1

0.7.1
General: Reworked the color options for both the whole package and the todo command. General code clean up. Added the prependcaption package option. ....................... 1

0.7.2
General: Avoid to change the fontsize inside the list of todos, fixing a bug revealed by Vladimir Zhuravlev. ....................... 1

0.7.3
General: The localization options (danish and german) and the disable options, were all flawed by naming inconsistencies that
made then break the package.

This have been fixed. . . . . . . . 1

0.7.4
General: Fixed a bug related to
the caption option for the todo
command. Introduced a
counter of the number of
inserted todos. . . . . . . . . . . . . 1

0.7.5
General: Fixed a typo in a
macroname. . . . . . . . . . . . . . 1

0.7.6
General: Added a textsize option
for the package and the
prepend / noprepend option
for the todo command. . . . . . . 1

0.8
General: Added three new
translations french, spanish
and catalan thanks to Richard
Dominique and Joan Queralt.
Improved the visual
appearance of the inserted
notes (rounded corners and
optional shadows) with code
from Joan Queralt. Found an
untranslated texstring
"Figure" in the source. Added
a figwidth option to the
missingfigure command, patch
by Paul Ivanov. . . . . . . . . . . . . 1

0.8.1
General: Added a space between
the colored square and the text
in the list of todos. Added a
new usecase for marking old /
new sections. Made the name
of listoftodos changeable. . . . . . 1

0.8.2
General: Italian translation by
Gustavo Cevolani. Removed
the dependence on the hyperref
package. . . . . . . . . . . . . . . . 1

0.8.3
General: Added a use case for
linking to the list of todos, idea
from Andreas Plank.
Introduced a package option
for listening to the draft option
given to the document class. . . 1

0.8.4
General: Fixed a bug related to
the obeyDraft option. . . . . . . 1

0.8.5
General: Added two new usecases
(enumeration of inserted
todonotes and how to set
custom default values).
Changed the order of the use
case examples. . . . . . . . . . . . 1

0.8.6
General: Added a portuguese
translation by Og DeSouza. . . 1

0.8.7
General: Updated portuguese
translation. Added a ngerman
alias for the german translation
suggested by Michael
Niedermair. . . . . . . . . . . . . . 1

0.8.8
General: Added a new usecase
from Vitaly. Fixed a bug
reported by Oscar Gustafsson.
Explained why the placement
of todonotes in the margin fails
in certain custom document
classes. . . . . . . . . . . . . . . . . . . 1

0.8.9
General: Added a dutch
translation by Ruben Ruben
Vermeersch. . . . . . . . . . . . . . 1

0.9.0
General: Added a english option as
suggested by Marco Berghoff. . 1

0.9.1
General: Added the todototoc
command by idea from Sven
Augustin. . . . . . . . . . . . . . . 1

0.9.2
General: Use chapter (if available)
for the list of todos heading. . . 1

0.9.3
General: Make an internal
definition of the todo
command, for easing
redefinition of the command
behaviour. . . . . . . . . . . . . . . . . . . 1

0.9.4
General: Make the disable option
work on a local scale. . . . . . . 1

34
0.9.5
General: Code simplification by extracting functionality to smaller macros. ............... 1

0.9.6
General: Give fontsize to TikZ. Align notes with line where note is set. Added new option fancyline. Patches by Benjamin Kellermann. ........ 1

0.9.7
General: Updated documentation. 1

0.9.8
General: Suppress warnings about underfull / overful boxes generated by the inserted todonotes. Patch by Peter M Schuler. ............... 1

0.9.9
General: Added author option, implementation provided by Xavier Alameda-Pineda. Example of modifying the listoftodos removing some protect commands with no effect. ..................... 1

1.0.0
General: Mention trouble with the classicthesis style. Refer to some alternatives to the package. Added todoin command as suggested by Stefan Pinnow. Described how to use tikz externalize with todonotes. Added obeyDraft and obeyFinal options. ........ 1

1.0.1
General: Fix spacing issues reported by Jonathan Zachhuber and Brent Longborough. Added figheight option to the missingfigure command as suggested by Kim Albertsson. ..................... 1

1.0.2
General: Added Swedish translation by Emil Lundberg. Added usecase by Tobias Winchen. Mentioned that default arguments can be set using the presetkeys command. Updated list of alternatives to the todonotes package. Draw borders around coloured boxes in the list of todos, patch by Ze Loff. ..................... 1

1.0.3
General: Added the option figcolor to missing figure, patch by Pascal Hebbeker. Added Croatian translation by Ican Kokan. Changed default with of missingfigure. Removed some underfull box warnings, solution by Ernst Blecha. ........ 1

1.0.4
General: Restructured documentation and placed some examples in the doc/examples subdirectory. .... 1

1.0.5
General: Example of how to change the space eating behaviour added by Anselm Wagner. Reduced the width of missingfigure to avoid bad box warnings. ..................... 1

1.0.6
General: Changed how whitespaces are handled around inserted todos to better mimick macros like index (Suggestion by Frank Mittelbach). Fix missing character warnings using patch by Niels Anders Danielsson. Use the specified textsize for missingfigure, patch by Johannes Twittmann. Added new options for inline todonotes based on code from Daniel Krenn. Added warning message when marginparwidth is less than 2cm, as suggested by Erik Boesen. ..................... 1

1.0.7
General: Changed the warning about marginparwidth to a proper warning message. ........ 1

1.0.8
General: Improved spacing

35
behaviour so it mimics the
index macro, patch by Frank
Mittelbach. Center graphics
vertically in missingfigure,
patch by Philipp Allgeuer.
Added new option
tickmarkheight as suggested by
Richard Niland. 1.1.0
General: Many improvements to
the user interface by Frank
Mittelbach. Added the
command setuptodonotes.
Added textcolor as an option.
Marked the insertion point of
notes with tickmarks with a
customizable height. Steamline
which options that are
accepted to size commands
(both commands (backslash
tiny) and text (tiny) are now
supported). 1.1.1
1.1.2
General: Implemented shadow and
noshadow options for the todo
command. 1.1.3
General: Fix issue 36 and 37. 1.1.4
General: Fix issue 48 and make
colors used in the last todo
globally available. 1.1.5
General: Fix issue 51 by avoid
loading the tikz shadows
library by default. 36