

The `todonotes` package*

Henrik Skov Midtiby
`henrikmidtiby@gmail.com`

October 24, 2008

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 <code>todo</code> command	3
1.4	Options for the <code>missingfigure</code> command	4
1.5	Options for the <code>listoftodos</code> command	4
1.6	Known issues	5
1.6.1	Package loading order	5
1.6.2	Wrapping of long lines in list of todos	5
1.7	Usage methods	5
1.7.1	Define new commands with fixed options	5
1.7.2	Combination with the <code>fixme</code> package	6
2	Implementation	6
2.1	Declaration of options for the package	6
2.2	Options for the <code>todo</code> command	7
2.3	The main code part	8

*This document corresponds to `todonotes.dtx`, dated 2008/10/24.

1 Introduction

The `todonotes` package makes three commands available to the user: `\todo[]{}{}`, `\missingfigure{}{}` and `\listoftodos`. `\todo[]{}{}` and `\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.

1.1 Usage

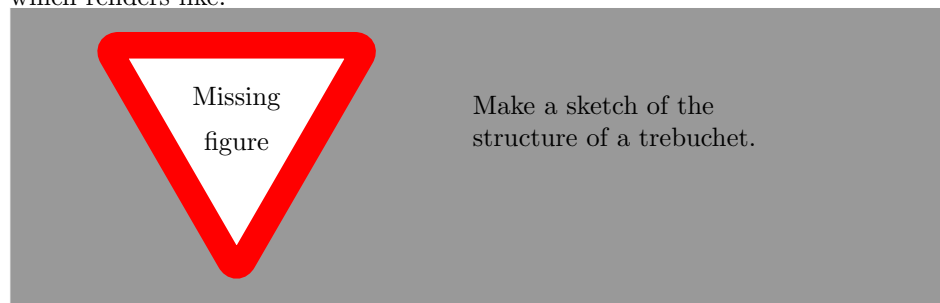
`\todo` My most common usage of the `todonotes` package, is to insert an uncusomized `todonotes` somewhere in a latex document. An example of this usage is the command

Make a cake ...

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

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

`\missingfigure{Make a sketch of the structure of a trebuchet.}` which renders like.



`\listoftodos` 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
A red note.	3
And a green note	3

■ A note with no line connecting it to the placement in the original text. .	4
■ A todonote placed in the text	4
■ Fill those circles	4
■ A note with a large fontsize.	4
■ Note with very small fontsize.	4
■ Short note	4

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.
colorinlistoftodos	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 usefull 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	color=val sets the default background color of the inserted todonotes to val . The default value is color=orange . val must be a color understood by the xcolor package.
bordercolor	bordercolor=val sets the default border color of the inserted todonotes to val . The default value is color=black . val must be a color understood by the xcolor package.
textwidth	textwidth=length sets the width of a todo item in the margin to length . The width of inline todonotes will allways be the same as the current line width.
dvistyle	When a document with todonotes is compiled with plain latex (to a dvi-file), there is an issue with the visual appearance ¹ . The option dvistyle changes the appearance of the inserted todonotes to avoid this problem.

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.

color **color=val** sets the background color of the inserted todonotes to **val**.

The default value can be set by the color option when the todonotes package is loaded. The todo notes inserted in this paragraph is created with the commands **\todo[color=red!40]{A red note.}** and **\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}}
```

line / noline If you want to get rid of the line connecting the inserted note with the place in

¹The problem is placement of text inside the colored boxes.

A note with no line connecting it to the placement in the original text.

the text where the note occurs in the latex code, the option `noline` can be used.

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

`inline / noline`

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.

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

A todonote placed in the text

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

```
\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}
```

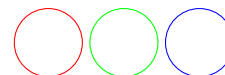


Figure 1: A text explaining the image.

Fill those circles ...

`size`

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

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

Note with very small fontsize.

A note with a large fontsize.

`list / nolist`

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

`caption`

The `caption` option enables the user to specify a short description of the `todonote` that are inserted in the list of todos instead of the full `todonote` text. .

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

A very long and tedious note that cannot be on one line in the list of todos.

1.4 Options for the `missingfigure` command

Currently the `missingfigure` command takes no optional arguments.

1.5 Options for the `listoftodos` command

Currently the `listoftodos` command takes no optional arguments.

1.6 Known issues

1.6.1 Package loading order

The `todonotes` package requires the following packages.

- `ifthen`
- `xkeyval`
- `hyperref`
- `xcolor`
- `tikz`
- `calc`
- `graphicx` (is loaded via the `tikz` package)

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.

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.

1.7 Usage methods

In this section I have collected some different methods to use the `todonotes` package.

1.7.1 Define new commands with fixed options

Often there is a need for marking different classes of things to do (add reference, rewrite, ...). One way to do this, is to define some new commands as shown below (idea from Florent B.).

```
\newcommand{\addref}{\todo[color=red!40]{Add reference.}}  
\newcommand{\rewrite}[1]{\todo[color=green!40]{#1}}
```

To distinguish between the different types of todos, the `todonotes` package can be loaded with the `colorinlistoftodos` option, which adds small colored squares to the list of todos.

```
\usepackage[colorinlistoftodos]{todonotes}
```

1.7.2 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).

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

2 Implementation

Identifies the package and loads the packages dependences.

```
1 \ProvidesPackage{todonotes}[2008/10/24]
2 \RequirePackage{ifthen}
3 \RequirePackage{xkeyval}
4 \RequirePackage{hyperref}
5 \RequirePackage{xcolor}
6 \RequirePackage{tikz}
7 \RequirePackage{calc}
```

Some default values are set

```
8 \newcommand{\@todonotes@backgroundcolor}{orange}
9 \newcommand{\@todonotes@bordercolor}{black}
10 \newcommand{\@todonotes@textwidth}{\marginparwidth}
```

2.1 Declaration of options for the package

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

Make it possible to disable the functionality of the package. If this option is given, the commands `\todo{}` and `\listoftodos` are defined as commands with no effect. (But you can still compile you document with these commands).

```
11 \newif{\if@todonotes@disabled}
12 \DeclareOptionX{disable}{\@todonotes@disabledtrue}
```

Show small boxes in the list of todos with the color of the inserted todonotes.

```
13 \newif{\if@todonotes@colorinlistoftodos}
14 \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.

```
15 \newif{\if@todonotes@dviStyle}
16 \DeclareOptionX{dvistyle}{\@todonotes@dviStyletrue}
```

Make the background color of the notes as an option.

```
17 \define@key{todonotes}.sty}%
18 {color}{\renewcommand{\@todonotes@backgroundcolor}{#1}}
```

Make the color of the notes box color as an option.

```
19 \define@key{todonotes.sty}%  
20   {bordercolor}{\renewcommand{\@todonotes@bordercolor}{#1}}
```

Make the text width as an option.

```
21 \define@key{todonotes.sty}%  
22   {textwidth}{\renewcommand{\@todonotes@textwidth}{#1}}
```

Finally process the given options.

```
23 \ProcessOptionsX
```

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

```
24 \newcommand{\@todonotes@fillcolor}{}%  
25 \define@key{todonotes}{color}{\renewcommand{\@todonotes@fillcolor}{#1}}%
```

Set a relative font size

```
26 \newcommand{\@todonotes@sizecommand}{}%  
27 \define@key{todonotes}{size}{\renewcommand{\@todonotes@sizecommand}{#1}}%
```

Should the todo item be included in the list of todos?

```
28 \newif\if@todonotes@appendtolistoftodos%  
29 \define@key{todonotes}{list}[]{\@todonotes@appendtolistoftodostrue}%  
30 \define@key{todonotes}{nolist}[]{\@todonotes@appendtolistoftodosfalse}%
```

Should the todo item be displayed inline?

```
31 \newif\if@todonotes@inlinenote%  
32 \define@key{todonotes}{inline}[]{\@todonotes@inlinenotetrue}%  
33 \define@key{todonotes}{noinline}[]{\@todonotes@inlinenotefalse}%
```

Should the note in the margin be connected to the insertion point in the text?

```
34 \newif\if@todonotes@line%  
35 \define@key{todonotes}{line}[]{\@todonotes@linetrue}%  
36 \define@key{todonotes}{noline}[]{\@todonotes@linefalse}%
```

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

```
37 \newcommand{\@todonotes@caption}{}%  
38 \newif\if@todonotes@captiongiven%  
39 \define@key{todonotes}{caption}%  
40 {\renewcommand{\@todonotes@caption}{#1}}%  
41 \@todonotes@captiongiventrue}%
```

Preset values of the options

```
42 \presetkeys  
43   {todonotes}  
44   {color=\@todonotes@backgroundcolor, noinline,  
45   line, list, size=\normalsize}{}%
```

2.3 The main code part

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

```
46 \if@todonotes@disabled
47   \newcommand{\listoftodos}{}
48   \newcommand{\todo}[2] [] {}
49   \newcommand{\missingfigure}[1] {}
50 \else % \if@todonotes@disabled
```

Define the \listoftodos command and define the appearance of the list of todos.

```
51 \newcommand{\listoftodos}
52   {\section*{Todo list} \@starttoc{tdo}}
53 \newcommand{\l@todo}
54   {\@dottedtocline{1}{0em}{2.3em}}
```

Define styles used by the todo command

```
55 \tikzstyle{notestyle} = [draw=\@todonotes@bordercolor, line width=0.5pt,
56   text width = \@todonotes@textwidth - 1.6 ex - 1pt, inner sep = 0.8 ex]
57 \tikzstyle{notestyleleft} = [notestyle, left]
58 \tikzstyle{connectstyle} = [thick]
59 \tikzstyle{inlinenotestyle} = [notestyle, text width=\linewidth - 1.6 ex - 1 pt]
```

\todo Define the todo command

```
60 \newcommand{\todo}[2] [] {%
61   \renewcommand{\@todonotes@caption}{#2}%
62   \setkeys{todonotes}{#1}%
```

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.

```
63 \if@todonotes@appendtolistoftodos%
64   \phantomsection%
65 \if@todonotes@captiongiven%
66 \else%
67   \renewcommand{\@todonotes@caption}{#2}
68 \fi%
69   \if@todonotes@colorinlistoftodos%
70     \addcontentsline{tdo}{todo}{\protect{%
71       \colorbox{\@todonotes@fillcolor}{\textcolor{\@todonotes@fillcolor}{\tiny i}} %
72       \@todonotes@caption}}%
73   \else%
74     \addcontentsline{tdo}{todo}{\protect{\@todonotes@caption}}%
75   \fi%
76 \fi%
```

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

```
77 \if@todonotes@inlinenote%
78   \if@todonotes@dviStyle%
79     {\par\noindent\begin{tikzpicture}[remember picture] \draw node[inlinenotestyle, %
80       fill=\@todonotes@fillcolor] {};\end{tikzpicture}\par}%
```



```

81      {\noindent \@todonotes@sizecommand #2}%
82      {\par\noindent\begin{tikzpicture}[remember picture] \draw node[inline notestyle, %
83      fill=\@todonotes@fillcolor] {}; \end{tikzpicture}\par}%
84    \else%
85      {\par\noindent\begin{tikzpicture}[remember picture] \draw node[inline notestyle, %
86      fill=\@todonotes@fillcolor] {\@todonotes@sizecommand #2}; \end{tikzpicture}\par}%
87    \fi%
88  \else%

```

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.

```

89 \begin{tikzpicture}[remember picture, baseline=-0.75ex]%
90   \node [coordinate] (inText) {};%
91 \end{tikzpicture}%
92 \marginpar[{\% Draw note in left margin
93 \if@todonotes@dviStyle%
94   \begin{tikzpicture}[remember picture] %
95   \draw node[notestyle, fill=\@todonotes@fillcolor] {}; %
96 \end{tikzpicture}}\%
97   \begin{minipage}{\@todonotes@textwidth}%
98   \@todonotes@sizecommand #2%
99   \end{minipage} \,%
100  \begin{tikzpicture}[remember picture] %
101  \draw node[notestyle, fill=\@todonotes@fillcolor] (inNote) {};%
102 \end{tikzpicture}%
103   \if@todonotes@line%
104     \begin{tikzpicture}[remember picture, overlay]%
105       \draw[connectstyle, draw=\@todonotes@fillcolor]%
106       ([yshift=-0.2cm] inText)%
107       -| ([xshift=0.2cm] inNote.east)%
108       -| (inNote.east);%
109     \end{tikzpicture}%
110   \fi%
111 \else%
112   \begin{tikzpicture}[remember picture] %
113   \draw node[notestyle, fill=\@todonotes@fillcolor] (inNote) %
114   {\@todonotes@sizecommand #2}; %
115 \end{tikzpicture}%
116   \if@todonotes@line%
117     \begin{tikzpicture}[remember picture, overlay]%
118       \draw[connectstyle, draw=\@todonotes@fillcolor]%
119       ([yshift=-0.2cm] inText)%
120       -| ([xshift=0.2cm] inNote.east)%
121       -| (inNote.east);%
122     \end{tikzpicture}%
123   \fi%
124 \fi%

```

In the book documentclass (which is a twoside layout), the `\marginpar` marginpar

macro takes two arguments `\marginpar []`, it is possible to let latex decide in which

```

125 }]{% Draw note in right margin
126 \if@todonotes@dviStyle%
127   \begin{tikzpicture}[remember picture] %
128 \draw node[notestyle, fill=\@todonotes@fillcolor] {}; %
129 \end{tikzpicture} \\\ %
130   \begin{minipage}{\@todonotes@textwidth}%
131   \@todonotes@sizecommand #2%
132   \end{minipage} \\\ %
133   \begin{tikzpicture}[remember picture] %
134 \draw node[notestyle, fill=\@todonotes@fillcolor] (inNote) {};%
135 \end{tikzpicture} %
136   \if@todonotes@line%
137     \begin{tikzpicture}[remember picture, overlay]%
138       \draw[connectstyle, draw=\@todonotes@fillcolor]%
139         ([yshift=-0.2cm] inText)%
140         -| ([xshift=-0.2cm] inNote.west)%
141         -| (inNote.west);%
142     \end{tikzpicture}%
143   \fi%
144 \else%
145   \begin{tikzpicture}[remember picture] \draw node[notestyle, fill=\@todonotes@fillcolor]
146 (inNote) {\@todonotes@sizecommand #2}; \end{tikzpicture}%
147   \if@todonotes@line%
148     \begin{tikzpicture}[remember picture, overlay]%
149       \draw[connectstyle, draw=\@todonotes@fillcolor]%
150         ([yshift=-0.2cm] inText)%
151         -| ([xshift=-0.2cm] inNote.west)%
152         -| (inNote.west);%
153     \end{tikzpicture}%
154   \fi%
155 \fi%
156 }%
157 \fi%
158 }%

```

`\missingfigure` Defines the `\missingfigure` macro.

```

159 \newcommand{\missingfigure}[1]{
160 \addcontentsline{tdo}{todo}{Figure: \protect{#1}}%
161 \par
162 \noindent
163 \begin{tikzpicture}
164 \draw[fill=black!40, draw = white, line width=0pt]
165   (-3, -2.5) rectangle +(\textwidth, 4cm);
166 \draw (3, 0) node[right, text width=4cm] {#1};
167 \draw[red, fill=white, rounded corners = 5pt, line width=10pt]
168   (30:2cm) -- (150:2cm) -- (270:2cm) -- cycle;
169 \draw (0, 0.3) node {Missing};
170 \draw (0, -0.3) node {figure};
171 \end{tikzpicture}

```

```

172 }% Ending \missingfigure command
173 \fi % Ending \@todonotes@ifdisabled

```

Change History

0.1	General: The first version of the package	1		the inline todonotes, to avoid empty lines around the inline todonotes.	1
0.2	General: Updated the option handling of the package	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
0.2.1	General: Slightly modified by Kjell Magne Fauske to support notes in the left margin (for documentstyle book).	1	0.4.2	General: Fixed a bug with the disable option to the package. . . .	1
0.2.2	General: Added a missingfigure command	1	0.5	General: Created a dtx file containing both source code and documentation of the package	1
0.2.3	General: Made a dependency on the calc package	1	0.5.1	General: Updated the documentation	1
0.3	General: Delayed the requirements for the hyperref package untill begin document and added an optional argument to the todo command for adding inline todonotes (Idea from Patrick Toche)	1	0.5.2	General: Fixed a bug that prevented the usage of the option french for babel. Bug report by Thomas Braun.	1
0.4	General: Modified the behaviour of		0.6	General: Added the caption option to the todo command.	1