Task Report v0.8

Geraldo Xexéo

Wednesday 9th June, 2021 - 19:21

Contents

1   Introduction

This package supports the maintenance and report of a task list..

2   How to use task-report

`task-report` is available as open source at `https://github.com/xexeo/task-report`. The stable distribution is in the folder `dist`, while the lastest, and unstable, will be in the root folder.

The only file you really need, besides this manual, is `task-report.sty`. This should be in your LATEX path, such as in the same folder that your main `.tex` file.

You can make suggestions, complain about bugs, and request features using GitHub's "Issues" feature, in `https://github.com/xexeo/task-report/issues` (you must be signed in).

2.1   Using from Overleaf

The best way to use `task-report` inside Overleaf is to link the distributed style file through its URL. To do that, inside your project, select first "Upload". An "Add files" window will appear, then select "From Externa URL" and enter `https://raw.githubusercontent.com/xexeo/task-report/main/dist/task-report.sty` as "URL to fetch the file from" and "task-report" as "File Name in This Project", as in Figure 1.

3   Package Options

showtasknumbers   show tasks numbers, default is false.

Users should never refer to a task by its number, since it can change from compilation to compilation if any task is inserted in among the existing ones. It is possible to use the `\label` macro to refer to tasks by number, page ou name.
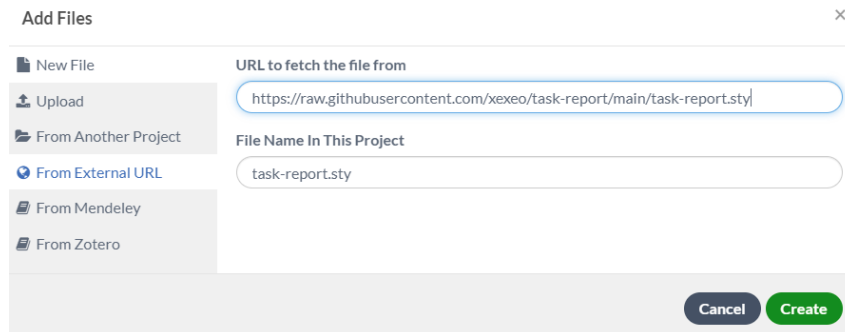
Figure 1: How to link a task-report.sty file in your project folder in Overleaf to the official distribution.

3.1 General behavior

4 Version

It is possible that the user wants to know the version being used. We provide two commands for it.

\taskreportversion   Provides the current version

\printtaskreportversion   Provides name and version

\taskreportversion was used in the title of this article. The current version is v0.8.

If needed, the second macro prints also the name of the package:

\printtaskreportversion

will result in:

task-report v0.8

Please state the version when reporting bugs.

5 Usage

\trtask   This is the main macro for the package. \trtask[⟨*status*⟩]{⟨*name*⟩}{⟨*due date*⟩}{⟨*when-assigned*⟩} {⟨*who-assigned*⟩}{⟨*to-who-assigned*⟩} {⟨*why*⟩} {⟨*when-done*⟩} {⟨*extra-text*⟩}

Altough it has 8 arguments parameters, all of them can be empty. It has 9 arguments because this is the standard LaTeX maximum.

There are 5 possible standard states that are associated with a color (*de-*

fault color): todo (red), doing(yellow), tocheck (cyan), checking (blue), and done(green). There is a pseudo-state other (black) that will be the color of any other word used of a state. Colors belong to the `xcolor` package and are configurable, as explained in section 6.1.

The example:

```
%\trtask[todo]{Make coffe}{Soon}{\today}{Geraldo Xexéo}{}{}{}{Please do the coffe strong
%\trtask[doing]{Drink a coffe}{In 5 days}{June, 1st 2021}{The Boss}{The Worker}{}{}{}
%\trtask[tocheck]{Roast the beef}{}{June, 1st 2021}{The Boss}{The Worker}{}{\today}{Rare
%\trtask[checking]{Drink another beer}{}{June, 1st 2021}{The Boss}{The Worker}{}{\today}
%\trtask[done]{Drink a beer}{}{\today}{The Boss}{The Worker}{Because I want}{\today}{}
%\trtask{}{}{}{}{}{}{}{}
%\trtask[done]{}{}{}{}{}{}{}{}
%\trtask[arg 1]{arg 2}{arg 3}{arg 4}{arg 5}{arg 6}{arg 7}{arg 8}{arg 9}
%
```

generates:

Make coffe

Status: todo
Due date: Soon
Assigned in: Wednesday 9th June, 2021
From: Geraldo Xexéo

Please do the coffe strong

Drink a coffe

Status: doing
Due date: In 5 days
Assigned in: June, 1st 2021
From: The Boss
To: The Worker

Roast the beef

Status: tocheck
Assigned in: June, 1st 2021
From: The Boss
To: The Worker
Done in: Wednesday 9$^{\text{th}}$ June, 2021

- -

Rare, plese

Drink another beer

Status: checking
Assigned in: June, 1st 2021
From: The Boss
To: The Worker
Done in: Wednesday 9$^{\text{th}}$ June, 2021

- -

Please check if you have age to drink in your country. If not, drink orange-juice.

Drink a beer

Status: done
Assigned in: Wednesday 9$^{\text{th}}$ June, 2021
From: The Boss
To: The Worker
Rationale: Because I want
Done in: Wednesday 9$^{\text{th}}$ June, 2021

Task without a name

Task without a name

Status: done

<div style="border:1px solid black; border-radius:10px;">

arg 2

Status: arg 1
Due date: arg 3
Assigned in: arg 4
From: arg 5
To: arg 6
Rationale: arg 7
Done in: arg 8

- -

arg 9

</div>

6 Configuration

6.1 Configuring Colors

`\trsetboxcolors`

Set the colors for the 6 types of standard states Black otherwise.

\trsetboxcolors{⟨*todo-color*⟩}{⟨*doing-color*⟩}{⟨*tocheck-color*⟩}{⟨*checking-color*⟩}{⟨*done-color*⟩}{⟨*other-color*⟩}

The example:

```
%\trsetboxcolors{magenta}{purple}{orange}{pink}{lime}{pink}
%
%\trtask[todo]{Drink a coffe}{next day}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask[doing]{Drink a coffe}{next day}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask[tocheck]{Drink a coffe}{next day}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask[checking]{Drink a coffe}{next day}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask[done]{Drink a coffe}{next day}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask{Drink a coffe}{\today}{next day}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%
```

generates:

Drink a coffe

Status: todo
Due date: next day
Assigned in: Wednesday 9th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: doing
Due date: next day
Assigned in: Wednesday 9th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: tocheck
Due date: next day
Assigned in: Wednesday 9th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

Coffe must be strong and add 1 spoon of sugar

<div style="border:1px solid #c08080;">

Drink a coffe

Status: checking
Due date: next day
Assigned in: Wednesday 9th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

- -

Coffe must be strong and add 1 spoon of sugar

</div>

<div style="border:1px solid #88aa00;">

Drink a coffe

Status: done
Due date: next day
Assigned in: Wednesday 9th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

- -

Coffe must be strong and add 1 spoon of sugar

</div>

<div style="border:1px solid #c08080;">

Drink a coffe

Due date: Wednesday 9th June, 2021
Assigned in: next day
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

- -

Coffe must be strong and add 1 spoon of sugar

</div>

6.2 Changing Field Names

\trsetfieldsnames   You can change all field names with command . This command can be used to internationalization or to make these box used by any other reason.

> \trsetfieldsnames{⟨*status*⟩}{⟨*duedate*⟩}{⟨*assigned*⟩} {⟨*from*⟩}{⟨*to*⟩} {⟨*done*⟩}{⟨*why*⟩}

For example:

```
%\trsetfieldsnames{Stage}{Must be ready in}{Demand date}{Who asked for}{To who}{Ready in
%\trtask[done]{Drink a coffe}{5 days}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%
```

generates:

> **Drink a coffe**
>
> **Stage**: done
> **Must be ready in**: 5 days
> **Demand date**: Wednesday 9$^{\text{th}}$ June, 2021
> **Who asked for**: The Boss
> **To who**: The Worker
> **Reasons**: To keep awake
> **Ready in**: Wednesday 9$^{\text{th}}$ June, 2021
>
> -
>
> Coffe must be strong and add 1 spoon of sugar

6.3   Modifying the Header

\trsettitlelabel   Sets the label for the header.

\trsettitlelabel{⟨*label-for-header*⟩}

```
% \trsettitlelabel{Task}
%
%\trtask[done]{Drink a coffe}{5 days}{\today}{The Boss}{The Worker}
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%
%
```

Task: Drink a coffe

Status: done
Due date: 5 days
Assigned in: Wednesday 9$^{\text{th}}$ June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9$^{\text{th}}$ June, 2021

- -

Coffe must be strong and add 1 spoon of sugar

Back to normal:

```
% \trsettitlelabel{}
%
%\trtask[done]{Drink a coffe}{5 days}{\today}{The Boss}{The Worker}
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%
```

Drink a coffe

Status: done
Due date: 5 days
Assigned in: Wednesday 9$^{\text{th}}$ June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9$^{\text{th}}$ June, 2021

- -

Coffe must be strong and add 1 spoon of sugar

6.4 Creating New States

`\traddstatecolor` Users can create their own state using

\traddstatecolor{⟨*state*⟩}{⟨*color*⟩}

For example:

```
%\traddstatecolor{waiting}{teal}
%\trtask[waiting]{Drink a coffe}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%
```

generates:

Drink a coffe

Status: waiting
Due date: ASAP
Assigned in: Wednesday 9th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Wednesday 9th June, 2021

- -

Coffe must be strong and add 1 spoon of sugar

7 Using Labels to Refer to Tasks

You can use labels, such as `\label{task:kanbam}` **in the last argument**[1] and refer to it with commands `\ref{task:kanbam}` (19), `\pageref{task:kanbam}` (17) ou `\nameref{task:kanbam}` (Make a LaTeX Kanbam document).

8 The List of Tasks

`\listoftasks`   A list of tasks can be generated with the command:

`\listoftasks`

Its title can be changed with the command:

`\trsettasklisttitle{⟨new-title⟩}`

The command:

`\setlength{\cfttasksnumwidth}{25pt}`

from package `tocloft` can be used to set the space used by the section indicator.

9 Warnings

This package uses another package that changes LaTeX's standard behavior for summary and lists. When you use it, you must explicitly change pages with `\newpage` before `\tableofcontents` or similar commands.

---

[1]Actually, they will probably break only if used in the second argument, that is used to name them

10 Implementation

10.1 Options

showtasknumbers

Since task numbers can change between compilations this default is set to false.

```
1 \newif\if@showtasknumbers\@showtasknumbersfalse%
2 \newif\if@cwtoclofttitles\@cwtoclofttitlesfalse
3 %
4 \DeclareOption{showtasknumbers}{\@showtasknumberstrue}%
5 \DeclareOption{toclofttitles}{\@cwtoclofttitlestrue}
6 %
7 \ProcessOptions\relax
```

10.2 Required Packages

- **mdframed** makes the frame aroung tasks

- **etoolbox** command ifstringempty

- **tocloft** create the list of tasks

Must defend coppe against tocloft

```
 8 \RequirePackage{tcolorbox}%
 9 \RequirePackage{etoolbox}%
10 \@ifclassloaded{coppe}%
11 {%
12     \RequirePackage[titles]{tocloft}%
13 }%
14 {%
15     \if@cwtoclofttitles%
16     \RequirePackage[titles]{tocloft}%
17     \else%
18     \RequirePackage{tocloft}%
19     \fi%
20 }%
21 \RequirePackage{hyperref}%
```

10.3 Solving problems with other packages

10.3.1 Problems with `abntex2` and `memoir`

`abntex2` causes error in `\newlistof`, you can't refer to `chapter`, `section` or other counter as optional argument

This error is actually `memoir` fault[1], since it emulates `tocloft` (and other packages). I can´t imagine why...

Therefore, we will avoid using the command with option when the class is present

```
22 \@ifclassloaded{memoir}%
23 {% TRUE
24     \newif\if@cwmemoirdefense\@cwmemoirdefensetrue%
25 }%
26 {% FALSE
27     \newif\if@cwmemoirdefense\@cwmemoirdefensefalse%
28 }%
29 %
```

10.4   Access to the current version

We provide some macros for the user to know the version being used.

\taskreportversion   Provides current version of coop-writing

```
30 \newcommand{\taskreportversion}{\tr@version}%
```

\printtaskreportversion   Provides package name and version

```
31 \newcommand{\printtaskreportversion}{task-report \taskreportversion}%
```

10.5   Color Management for States

10.6   Defining States Through a Simulated List

\traddstatecolor   Inserts states and their colors in a simulated list and uses them
\usestatecolor

```
32 \newcommand{\traddstatecolor}[2]{%
33     \@ifundefined{\string\color@#2}%
34     {%
35         \message{#2 not defined}%
36         \expandafter\ERROR%
37     }%
38     {%
39         \expandafter\gdef\csname tr@astate@\detokenize{#1}\endcsname{#2}%
40 }%
41 }%
42 %
43 \def\tr@usestatecolor#1{%
44     \ifcsname tr@astate@\detokenize{#1}\endcsname%
```

---

[1] CITE:Plese cite memoir package

```
45      \csname tr@astate@\detokenize{#1}\expandafter\endcsname%
46      \else%
47 \tr@astate@other%
48      \fi%
49 }%
```

\trsetboxcolors   Set the colors for the 6 types of standard states Black otherwise.

\trsetboxcolors{⟨*todo-color*⟩}{⟨*doing-color*⟩} {⟨*tocheck-color*⟩}{⟨*checking-color*⟩} {⟨*done-color*⟩}{⟨*other-color*⟩}

```
50 \newcommand{\trsetboxcolors}[6]{%
51 \traddstatecolor{todo}{#1}
52 \traddstatecolor{doing}{#2}
53 \traddstatecolor{tocheck}{#3}
54 \traddstatecolor{checking}{#4}
55 \traddstatecolor{done}{#5}
56 \traddstatecolor{other}{#6}
57 }%
```

10.7   Standard Colors

Sets the standard colors for the standard states

```
58 \trsetboxcolors{red}{yellow}{cyan}{blue}{green}{black}
```

10.8   Managing The Fields Names

\traddfieldname   Define field names - does not create a new field
\tr@usesfieldname

```
59 \newcommand{\traddfieldname}[2]{%
60      \expandafter\gdef\csname tr@fieldnames@\detokenize{#1}\endcsname{#2}%
61 }%
62 %
63 \def\tr@usesfieldname#1{%
64      \ifcsname tr@fieldnames@\detokenize{#1}\endcsname%
65      \csname tr@fieldnames@\detokenize{#1}\expandafter\endcsname%
66      \else%
67 Missing Field Name%
68      \fi%
69 }%
```

10.9   Standard Field Names

Sets the standard text for the fields

\trsetfieldsnames   Set the names for the field names and for the titles.
\trsettitlelabel

14

$\trsetfieldsnames\{\langle status \rangle\}\{\langle duedate \rangle\}\{\langle assigned \rangle\}\ \{\langle from \rangle\}\{\langle to \rangle\}\ \{\langle done \rangle\}\{\langle why \rangle\}$

$\trsettitlelabel\{\langle title\text{-}label \rangle\}$

```
70 \newcommand{\trsetfieldsnames}[7]{%
71     \traddfieldname{status}{#1}%
72     \traddfieldname{duedate}{#2}%
73     \traddfieldname{assigned}{#3}%
74     \traddfieldname{from}{#4}%
75     \traddfieldname{to}{#5}%
76     \traddfieldname{done}{#6}%
77     \traddfieldname{why}{#7}%
78 }%
79 %
80 \traddfieldname{defaulttitle}{Task without a name}%
81 %
82 \def\tr@defaulttitlelabel{}%
83 \newcommand{\trsettitlelabel}[1]{%
84     \def\tr@defaulttitlelabel{#1}%
85 }%
86 %
87 \trsetfieldsnames{Status}{Due date}{Assigned in}{From}{To}{Done in}{Rationale}
88 %
```

\trsetcolon

```
89 \newcommand{\trsetcolon}[1]{%
90     \traddfieldname{colonmark}{#1}
91 }
92 \trsetcolon{:}
```

\tr@currentboxcolor The dirt trick of using a global variable to store a value

```
93 \def\tr@currentboxcolor{\tr@usestatecolor{other}}%
```

\trstatuscolor Select the color based on the state

$\trstatuscolor\{\langle state \rangle\}$

```
94 \newcommand{\trstatuscolor}[1]%
95 {%
96 \ifblank{#1}%
97 {%
98 \def\tr@currentboxcolor{\tr@usestatecolor{other}}%
99 }%
100 {%
101 \def\tr@currentboxcolor{\tr@usestatecolor{#1}}%
102 }%
103 }%
```

10.10   List of Tasks

```
104 \newcommand{\trsettasklisttitle}[1]{%
105     \def\trlisttasks{#1}%
106 }%
107 \trsettasklisttitle{List of Tasks}%
108 %% cria a lista, depende do pacoto tcloft
109 \if@cwmemoirdefense%
110 \@ifundefined{chapter}%
111 {\newlistentry[section]{tasks}{trt}{0}}%
112 {\newlistentry[chapter]{tasks}{trt}{0}}%
113 \newlistof{listoftasks}{trt}{\trlisttasks}%
114 \else%
115 \@ifundefined{chapter}%
116 {\newlistof[section]{tasks}{trt}{\trlisttasks}}%
117 {\newlistof[chapter]{tasks}{trt}{\trlisttasks}}%
118 \fi%
```

10.11   Declaring Tasks

10.11.1   Helpers

\tr@field   Helpers to build the task \trtitle should be improved to chack if \tr@defaulttitlelabel
is not empty and use it

```
119 \newcommand{\tr@title}[2]{%
120 \ifx\tr@defaulttitlelabel\empty%
121     \if@showtasknumbers%
122         \textbf{#2\tr@usesfieldname{colonmark} #1}%
123     \else%
124         \textbf{#1}%
125     \fi%
126 \else%
127     \if@showtasknumbers%
128         \textbf{\tr@defaulttitlelabel\ #2\tr@usesfieldname{colonmark}\  #1}%
129     \else%
130         \textbf{\tr@defaulttitlelabel\tr@usesfieldname{colonmark}\  #1}%
131     \fi%
132 \fi%
133 }%
134 %
135 \newcommand{\tr@field}[2]{%
136 \textbf{#1}\tr@usesfieldname{colonmark} #2%
137 }%
138 %
```

10.11.2   Main task macro

\trtask   Describes a task

$\trtask[\langle status\rangle]\{\langle name\rangle\}\{\langle due\text{-}date\rangle\}$ $\{\langle when\text{-}assigned\rangle\}$ $\{\langle who\text{-}assigned\rangle\}$ $\{\langle to\text{-}who\text{-}assigned\rangle\}\{\langle why\rangle\}$ $\{\langle when\text{-}done\rangle\}\{\langle extra\text{-}text\rangle\}$

For example:

```
%\trtask[doing]{Make a \LaTeX\ Kanbam document}{soon}
% {2021-05-31}{Xexéo}{Geraldo Xexéo}{Help control tasks}
% {not yet}{Use colors\label{task:kanbam}}
%
```

generates:

Make a LᴬTEX Kanbam document

Status: doing
Due date: soon
Assigned in: 2021-05-31
From: Xexéo
To: Geraldo Xexéo
Rationale: Help control tasks
Done in: not yet

- -

Use colors

It is possible to reference the task (as Task 19 in page 17, named Make a LᴬTEX Kanbam document).

```
139 %  [status] , what, due-date when-assigned, from-who, to-who, why, when-done, extra
140 \newcounter{taskcounter}
141 %
142 \newcommand{\trtask}[9][]{%
143     \par%
144     \ifblank{#2}{
145     \def\trtask@temp{\tr@usesfieldname{defaulttitle}}%
146     }%
147     {%
148     \def\trtask@temp{#2}%
149     }%
150     \refstepcounter{tasks}%
151 % \refstepcounter updates \@currentlabel (\LaTeX magic!)
152 % therefore the next command must appear before #9, where label must be.
153     \refstepcounter{taskcounter}%
154     \protected@edef\@currentlabelname{\trtask@temp}% trying to make it work
155     \addcontentsline{trt}{tasks}{\protect\numberline{\thetasks}{\trtask@temp}}%
156     \trstatuscolor{#1}%
157     \begin{tcolorbox}[title=\tr@title{\trtask@temp}{\thetaskcounter},%
158 colback=\tr@currentboxcolor!5!white,%
159 colframe=\tr@currentboxcolor!75!black%
```

```
160 ]%
161         \ifblank{#1}{}%
162         {%
163             \tr@field{\tr@usesfieldname{status}}{#1}\newline%
164         }%
165         \ifblank{#3}{}%
166 {%
167     \tr@field{\tr@usesfieldname{duedate}}{#3}\newline%
168 }%
169 \ifblank{#4}{}%
170 {%
171     \tr@field{\tr@usesfieldname{assigned}}{#4}\newline%
172 }%
173 \ifblank{#5}{}%
174 {%
175     \tr@field{\tr@usesfieldname{from}}{#5}\newline%
176 }%
177 \ifblank{#6}{}%
178 {%
179     \tr@field{\tr@usesfieldname{to}}{#6}\newline%
180 }%
181 \ifblank{#7}{}%
182 {%
183     \tr@field{\tr@usesfieldname{why}}{#7}\newline%
184 }%
185 \ifblank{#8}{}%
186 {%
187     \tr@field{\tr@usesfieldname{done}}{#8}\newline%
188 }%
189         \ifblank{#9}{}%
190         {\tcblower%
191         #9%
192         }%
193     \end{tcolorbox}%
194 }%
```

End of Code

Change History

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

List of Tasks