Task Report v0.1

Geraldo Xexéo

Monday 31st May, 2021 - 21:32

Contents

1 Introduction

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

1

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
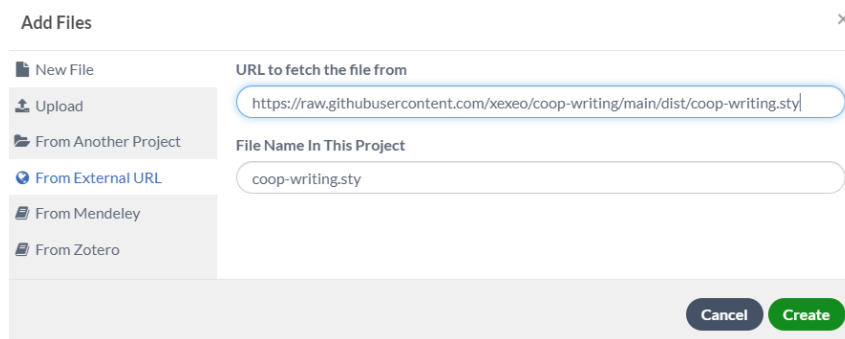Project", as in Figure 1.

Figure 1: How to link a task-report.sty file in your project folder in Overleaf to
the official distribution.

3 Package Options

There are no package options up to now

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 v0.1.

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

\printtaskreportversion

will result in:

task-report v. v0.1

Please state the version when reporting bugs.

5 Usage

\trtask

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

Altough it has 7 mandatory parameters, all of them can be empty.

There are 5 possible states that are associated with a color (*default 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 5.1.

The example:

```
%\trtask[todo]{Mage coffe}{\today}{Geraldo Xexéo}{}{}{}{Please do the coffe strong}
%\trtask[doing]{Drink a coffe}{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}{}{F
%\trtask[done]{Drink a beer}{\today}{The Boss}{The Worker}{\today}{}{}
%\trtask{}{}{}{}{}{}{}
%\trtask[done]{}{}{}{}{}{}{}{}
%\trtask{Grab a hot-dog}{3}{4}{5}{6}{7}{A hot dog must be done with bread and sausage}
%
```

generates:

Mage coffe

Status: todo
Assigned in: Monday 31st May, 2021
From: Geraldo Xexéo

Please do the coffe strong

Drink a coffe

Status: doing
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
Rationale: Monday 31st May, 2021

Rare, plese

Drink another beer

Status: checking
Assigned in: June, 1st 2021
From: The Boss
To: The Worker
Rationale: Monday 31st May, 2021

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

> **Drink a beer**
>
> **Status**: done
> **Assigned in**: Monday 31st May, 2021
> **From**: The Boss
> **To**: The Worker
> **Rationale**: Monday 31st May, 2021

> **Task not defined**

> **Task not defined**
>
> **Status**: done

> **Grab a hot-dog**
>
> **Assigned in**: 3
> **From**: 4
> **To**: 5
> **Rationale**: 6
> **Done in**: 7
>
> -
>
> A hot dog must be done with bread and sausage

5.1 Configuration Options

`\trsetboxcolors` Set the colors for the 4 types of standard states Black otherwise.

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

The example:

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

```
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask[done]{Drink a coffe}{\today}{The Boss}{The Worker}%
%{To keep awake}{\today}{Coffe must be strong and add 1 spoon of sugar}
%\trtask{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: todo
Assigned in: Monday 31st May, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Monday 31st May, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: doing
Assigned in: Monday 31st May, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Monday 31st May, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: tocheck
Assigned in: Monday 31st May, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Monday 31st May, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: checking
Assigned in: Monday 31st May, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Monday 31st May, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: done
Assigned in: Monday 31st May, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Monday 31st May, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Assigned in: Monday 31st May, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Monday 31st May, 2021

Coffe must be strong and add 1 spoon of sugar

5.2 Creating New States

Users can create their own state using

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

> **Drink a coffe**
>
> **Status**: waiting
> **Assigned in**: Monday 31ˢᵗ May, 2021
> **From**: The Boss
> **To**: The Worker
> **Rationale**: To keep awake
> **Done in**: Monday 31ˢᵗ May, 2021
>
> -
>
> Coffe must be strong and add 1 spoon of sugar

6 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.

7 Implementation

This package was initially written in Portuguese. As it become more useful, it was translated to English. There are still some signs of this in the code.

7.1 Required Packages

- `mdframed` makes the frame aroung tasks

- `etoolbox` command ifstringempty

```
1 \RequirePackage{tcolorbox}%
2 \RequirePackage{etoolbox}%
3 \RequirePackage{xstring}%
```

7.2 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

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

`\printtaskreportversion` Provides package name and version

```
5 \newcommand{\printtaskreportversion}{task-report v. \taskreportversion}%
```

7.3 Color Management for Standard States

`\tr@todoboxcolor`
`\tr@doingboxcolor`
`\tr@doneboxcolor`
`\tr@tocheckboxcolor`
`\tr@checkingboxcolor`
`\tr@otherwisecolor`

Global variables for the box colors.

```
 6 \def\tr@todoboxcolor{red}%
 7 \def\tr@doingboxcolor{yellow}%
 8 \def\tr@tocheckboxcolor{cyan}%
 9 \def\tr@checkingboxcolor{blue}%
10 \def\tr@doneboxcolor{green}%
11 \def\tr@otherwisecolor{black}%
```

`\trsetboxcolors`

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

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

```
12 \newcommand{\trsetboxcolors}[6]{%
13 \def\tr@todoboxcolor{#1}%
14 \def\tr@doingboxcolor{#2}%
15 \def\tr@tocheckboxcolor{#3}%
16 \def\tr@checkingcolor{#4}%
17 \def\tr@doneboxcolor{#5}%
18 \def\tr@otherwisecolor{#6}%
19 }%
```

`\tr@currentboxcolor`

The dirt trick of using a global variable to store a value

```
20 \def\tr@currentboxcolor{\tr@otherwisecolor}%
```

7.4 User-defined states

`\traddstatecolor`
`\usestatecolor`

```
21 \def\traddstatecolor#1#2{%
22 \@ifundefined{\string\color@#2}{%
23 \message{#2 not defined}%
24 \ERROR%
25 }%
26 {%
27 \expandafter\gdef\csname%
28 tr@astate@\detokenize{#1}\endcsname{#2}%
29 }}%
30 %
31 \def\tr@usestatecolor#1{%
32 \ifcsname tr@astate@\detokenize{#1}\endcsname%
33 \csname tr@astate@\detokenize{#1}\expandafter\endcsname%
34 \else%
35 \tr@otherwisecolor%
36 \fi%
37 }%
```

\trstatuscolor Select the color based on the state

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

```
38 \newcommand{\trstatuscolor}[1]%
39 {%
40     \def\tr@currentboxcolor{\tr@otherwisecolor}%
41     \ifblank{#1}{}{%
42     \IfStrEqCase{#1}{%
43     {todo}{\def\tr@currentboxcolor{\tr@todoboxcolor}}%
44     {doing}{\def\tr@currentboxcolor{\tr@doingboxcolor}}%
45     {done}{\def\tr@currentboxcolor{\tr@doneboxcolor}}%
46     {tocheck}{\def\tr@currentboxcolor{\tr@tocheckboxcolor}}%
47     {checking}{\def\tr@currentboxcolor{\tr@checkingboxcolor}}%
48     } [%
49     \def\tr@currentboxcolor{\tr@usestatecolor{#1}}
50 ]%
51     }%
52 }%
```

7.5 Declaring Tasks

7.5.1 Helpers

\trstatus
\trwhen
\trwhy
\trwhat
\trwho

Helpers to build the task

```
53 \newcommand{\trstatus}[2][Status]{%
54     \textbf{#1}: #2%
55 }%
56 \newcommand{\trwhen}[2][Assigned in]{%
57     \textbf{#1}: #2%
58 }%
59 \newcommand{\trwho}[2][To]{%
60     \textbf{#1}: #2%
61 }%
62 \newcommand{\trwhat}[2][What]{%
63     \textbf{#1}: #2%
64 }%
65 \newcommand{\trwhy}[2][Rationale]{%
66     \textbf{#1}: #2%
67 }%
68 %
```

7.5.2 Main task macro

\trtask Describes a task

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

For example:

```
%\trtask[Help students organize]{doing}
% {Prepare a \LaTeX\ style to support task report}
% {2021-05-31}{Xexéo}{Geraldo Xexéo}{}
%
```

generates:

> **Prepare a LATEX style to support task report**
>
> **Status**: doing
> **Assigned in**: 2021-05-31
> **From**: Xexéo
> **To**: Geraldo Xexéo
> **Rationale**: Help students organize
> **Done in**: not yet
>
> -
>
> This is an idea to help understanding the work being done

```
69 %  [status] , what, when-assigned, from-who, to-who, why, when-done, extra
70 \newcommand{\trtask}[8][]{%
71     \par%
72     \ifblank{#2}{
73     \def\trtask@temp{Task not defined}%
74     }%
75     {%
76     \def\trtask@temp{#2}%
77     }%
78     \trstatuscolor{#1}%
79     \begin{tcolorbox}[title=\textbf{\trtask@temp},%
80          colback=\tr@currentboxcolor!5!white,%
81          colframe=\tr@currentboxcolor!75!black%
82         ]%
83         \ifblank{#1}{}%
84         {%
85             \trstatus{#1}\newline%
86         }%
87         \ifblank{#3}{}%
88         {%
89             \trwhen{#3}\newline%
90         }%
91         \ifblank{#4}{}%
92         {%
93             \trwho[From]{#4}\newline%
94         }%
95         \ifblank{#5}{}%
```

```
 96          {%
 97              \trwho{#5}\newline%
 98          }%
 99          \ifblank{#6}{}%
100          {%
101          \trwhy{#6}\newline%
102          }%
103          \ifblank{#7}{}%
104          {%
105              \trwhen[Done in]{#7}\newline%
106          }%
107          \ifblank{#8}{}%
108          {\tcblower%
109          #8%
110          }%
111      \end{tcolorbox}%
112 }%
```

End of Code

Change History

v0.1

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.