

Task Report v0.6

Geraldo Xexéo

Sunday 6th June, 2021 - 13:31

Contents

1	Introduction	2
2	How to use task-report	2
2.1	Using from Overleaf	2
3	Package Options	2
3.1	General behavior	3
4	Version	3
5	Usage	3
6	Configuration	6
6.1	Configuring Colors	6
6.2	Changing Field Names	8
6.3	Modifying the Header	9
6.4	Creating New States	10
7	Using Labels to Refer to Tasks	11
8	The List of Tasks	11
9	Warnings	11
10	Implementation	12
10.1	Options	12
10.2	Required Packages	12
10.3	Solving problems with other packages	12
10.3.1	Problems with <code>abntex2</code> and <code>memoir</code>	12
10.4	Access to the current version	13
10.5	Color Management for States	13
10.6	Defining States Through a Simulated List	13

10.7 Standard Colors	14
10.8 Managing The Fields Names	14
10.9 Standard Field Names	14
10.10List of Tasks	15
10.11Declaring Tasks	16
10.11.1 Helpers	16
10.11.2 Main task macro	16

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 latest, 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 External 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.

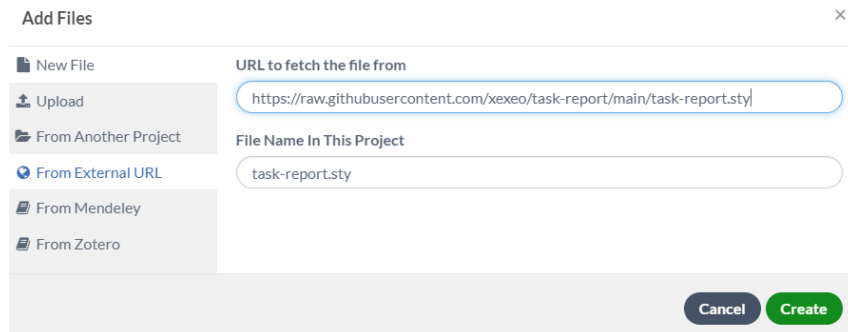


Figure 1: How to link a task-report.sty file in your project folder in Overleaf to the official distribution.

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 v0.6.

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

`\printtaskreportversion`

will result in:

task-report v0.6

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>}`

Although it has 8 arguments parameters, all of them can be empty. It has 9 arguments because this is the standard L^AT_EX 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: Sunday 6th 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: Sunday 6th June, 2021

Rare, please

Drink another beer

Status: checking
Assigned in: June, 1st 2021
From: The Boss
To: The Worker
Done in: Sunday 6th 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: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: Because I want
Done in: Sunday 6th June, 2021

Task without a name

Task without a name

Status: done

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

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: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: doing
Due date: next day
Assigned in: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: tocheck
Due date: next day
Assigned in: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: checking
Due date: next day
Assigned in: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Status: done
Due date: next day
Assigned in: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th June, 2021

Coffe must be strong and add 1 spoon of sugar

Drink a coffe

Due date: Sunday 6th June, 2021
Assigned in: next day
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th June, 2021

Coffe must be strong and add 1 spoon of sugar

6.2 Changing Field Names

`\trsetfieldsnames` You can change all field names with command `\trsetfieldsnames`. 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: Sunday 6th June, 2021
Who asked for: The Boss
To who: The Worker
Reasons: To keep awake
Ready in: Sunday 6th 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: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th 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: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th 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: Sunday 6th June, 2021
From: The Boss
To: The Worker
Rationale: To keep awake
Done in: Sunday 6th 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 argument¹ and refer to it with commands `\ref{task:kanbam}` (19), `\pageref{task:kanbam}` (17) ou `\nameref{task:kanbam}` (Make a L^AT_EX 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 L^AT_EX's standard behavior for summary and lists. When you use it, you must explicitly change pages with `\newpage` before `\tableofcontents` or similar commands.

¹Actually, 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 %
3 \DeclareOption{showtasknumbers}{\@showtasknumberstrue}%
4 %
5 \ProcessOptions\relax
```

10.2 Required Packages

- `mdframed` makes the frame around tasks
- `etoolbox` command `ifstringempty`
- `tocloft` create the list of tasks

```
6 \RequirePackage{tcolorbox}%
7 \RequirePackage{etoolbox}%
8 \RequirePackage{tocloft}%
9 \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` fault¹, 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

```
10 \@ifclassloaded{memoir}%
11 {% TRUE
12   \newif\if@cwmemoirdefense\@cwmemoirdefensetrue%
13 }%
14 {% FALSE
15   \newif\if@cwmemoirdefense\@cwmemoirdefensefalse%
```

¹CITE:Plese cite memoir package

```

16 }%
17 %

```

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
18 \newcommand{\taskreportversion}{\tr@version}%

\printtaskreportversion Provides package name and version
19 \newcommand{\printtaskreportversion}{task-report \taskreportversion}%

```

10.5 Color Management for States

10.6 Defining States Through a Simulated List

```

\ttraddstatecolor Inserts states and their colors in a simulated list and uses them
\usestatecolor
20 \newcommand{\traddstatecolor}[2]{%
21   \@ifundefined{string\color@#2}%
22   {%
23     \message{#2 not defined}%
24     \expandafter\ERROR%
25   }%
26   {%
27     \expandafter\gdef\csname tr@astate@\detokenize{#1}\endcsname{#2}%
28   }%
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@astate@other%
36   \fi%
37 }%

\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}

38 \newcommand{\trsetboxcolors}[6]{%
39 \traddstatecolor{todo}{#1}
40 \traddstatecolor{doing}{#2}

```

```

41 \traddstatecolor{tocheck}{#3}
42 \traddstatecolor{checking}{#4}
43 \traddstatecolor{done}{#5}
44 \traddstatecolor{other}{#6}
45 }%

```

10.7 Standard Colors

Sets the standard colors for the standard states

```

46 \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
47 \newcommand{\traddfieldname}[2]{%
48   \expandafter\gdef\csname tr@fieldnames@\detokenize{#1}\endcsname{#2}%
49 }%
50 %
51 \def\tr@usesfieldname#1{%
52   \ifcsname tr@fieldnames@\detokenize{#1}\endcsname%
53   \csname tr@fieldnames@\detokenize{#1}\expandafter\endcsname%
54   \else%
55 Missing Field Name%
56   \fi%
57 }%

```

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
\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-label \rangle}

58 \newcommand{\trsetfieldsnames}[7]{%
59   \traddfieldname{status}{#1}%
60   \traddfieldname{duedate}{#2}%
61   \traddfieldname{assigned}{#3}%
62   \traddfieldname{from}{#4}%
63   \traddfieldname{to}{#5}%
64   \traddfieldname{done}{#6}%
65   \traddfieldname{why}{#7}%
66 }%

```

```

67 %
68 \traddfieldname{defaulttitle}{Task without a name}%
69 %
70 \def\tr@defaulttitlelabel{}%
71 \newcommand{\trsettitlelabel}[1]{%
72     \def\tr@defaulttitlelabel{#1}%
73 }%
74 %
75 \trsetfieldsnames{Status}{Due date}{Assigned in}{From}{To}{Done in}{Rationale}
76 %

\trsetcolon

77 \newcommand{\trsetcolon}[1]{%
78     \traddfieldname{colonmark}{#1}%
79 }
80 \trsetcolon{:}

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

81 \def\tr@currentboxcolor{\tr@usestatecolor{other}}%

\trstatuscolor Select the color based on the state

    \trstatuscolor{\langle state \rangle}

82 \newcommand{\trstatuscolor}[1]{%
83 {%
84     \ifblank{#1}%
85 {%
86     \def\tr@currentboxcolor{\tr@usestatecolor{other}}%
87 }%
88 {%
89     \def\tr@currentboxcolor{\tr@usestatecolor{#1}}%
90 }%
91 }%

```

10.10 List of Tasks

```

92 \newcommand{\trsettasklisttitle}[1]{%
93     \def\trlisttasks{#1}%
94 }%
95 \trsettasklisttitle{List of Tasks}%
96 %% cria a lista, depende do pacote tcloft
97 \if@cwmemoirdefense%
98 \@ifundefined{chapter}%
99 {\newlistentry[section]{tasks}{trt}{0}}%
100 {\newlistentry[chapter]{tasks}{trt}{0}}%
101 \newlistof{listoftasks}{trt}{\trlisttasks}%

```

```

102 \else%
103 \@ifundefined{chapter}%
104 {\newlistof[section]{tasks}{trt}{\trlisttasks}}%
105 {\newlistof[chapter]{tasks}{trt}{\trlisttasks}}%
106 \fi%

```

10.11 Declaring Tasks

10.11.1 Helpers

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

```

107 \newcommand{\tr@title}[2]{%
108 \ifx\tr@defaulttitlelabel\empty%
109     \if@showtasknumbers%
110         \textbf{#2\tr@usesfieldname{colonmark} #1}%
111     \else%
112         \textbf{#1}%
113     \fi%
114 \else%
115     \if@showtasknumbers%
116         \textbf{\tr@defaulttitlelabel\ #2\tr@usesfieldname{colonmark}\ #1}%
117     \else%
118         \textbf{\tr@defaulttitlelabel\tr@usesfieldname{colonmark}\ #1}%
119     \fi%
120 \fi%
121 }%
122 %
123 \newcommand{\tr@field}[2]{%
124 \textbf{#1}\tr@usesfieldname{colonmark} #2%
125 }%
126 %

```

10.11.2 Main task macro

`\trtask` Describes a task

```

\trtask[<status>]{<name>}{<due-date>}{<when-assigned>}{<who-assigned>}
{<to-who-assigned>}{<why>}{<when-done>}{<extra-text>}

```

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 L^AT_EX 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 L^AT_EX Kanbam document).

```
127 % [status] , what, due-date when-assigned, from-who, to-who, why, when-done, extra
128 \newcounter{taskcounter}
129 %
130 \newcommand{\trtask}[9] [] {%
131     \par%
132     \ifblank{#2}{%
133         \def\trtask@temp{\tr@usesfieldname{defaulttitle}}%
134     }%
135     {%
136         \def\trtask@temp{#2}%
137     }%
138     \refstepcounter{tasks}%
139 % \refstepcounter updates \@currentlabel (\LaTeX magic!)
140 % therefore the next command must appear before #9, where label must be.
141     \refstepcounter{taskcounter}%
142     \protected@edef\@currentlabelname{\trtask@temp}% trying to make it work
143     \addcontentsline{trt}{tasks}{\protect\numberline{\thetasks}{\trtask@temp}}%
144     \trstatuscolor{#1}%
145     \begin{tcolorbox}[title=\tr@title{\trtask@temp}{\thetaskcounter},%
146 colback=\tr@currentboxcolor!5!white,%
147 colframe=\tr@currentboxcolor!75!black%
148 ]%
149         \ifblank{#1}{}%
150         {%
151             \tr@field{\tr@usesfieldname{status}}{#1}\newline%
152         }%
153         \ifblank{#3}{}%
154     {%
155         \tr@field{\tr@usesfieldname{duedate}}{#3}\newline%
156     }%
157     \ifblank{#4}{}%
158     {%
159         \tr@field{\tr@usesfieldname{assigned}}{#4}\newline%
```

```

160 }%
161 \ifblank{#5}{}%
162 {%
163     \tr@field{\tr@usesfieldname{from}}{#5}\newline%
164 }%
165 \ifblank{#6}{}%
166 {%
167     \tr@field{\tr@usesfieldname{to}}{#6}\newline%
168 }%
169 \ifblank{#7}{}%
170 {%
171     \tr@field{\tr@usesfieldname{why}}{#7}\newline%
172 }%
173 \ifblank{#8}{}%
174 {%
175     \tr@field{\tr@usesfieldname{done}}{#8}\newline%
176 }%
177     \ifblank{#9}{}%
178     {\tcblower%
179     #9%
180     }%
181 \end{tcolorbox}%
182 }%

```

End of Code

Change History

v0.1		v0.5	
General: First version	1	General: Allows redefining header and colon	1
v0.2		v0.6	
General: User can create new states	1	General: Numbers in tasks and references	1
v0.3		v0.7	
General: List of Tasks	1	General: Memoir class (used abntex2) badly emulates tocloft	1
v0.4			
General: Allows redefining field names	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.

Symbols	I	<code>\tcblower</code> 178
<code>\@currentlabel</code> 139	<code>\if@cwmemoirdefense</code>	<code>\textbf</code> 110,
<code>\@currentlabelname</code> . 142 12, 15, 97	112, 116, 118, 124
<code>\@cwmemoirdefensefalse</code>	<code>\if@showtasknumbers</code>	<code>\thetaskcounter</code> ... 145
..... 15 1, 109, 115	<code>\thetasks</code> 143
<code>\@cwmemoirdefensetrue</code>	<code>\ifblank</code> 84, 132, 149,	<code>\tr@astate@other</code> .. 35
..... 12	153, 157, 161,	<code>\tr@currentboxcolor</code>
<code>\@ifclassloaded</code> ... 10	165, 169, 173, 177	81, 86, 89, 146, 147
<code>\@ifundefined</code> 21, 98, 103	<code>\ifcsname</code> 32, 52	<code>\tr@defaulttitlelabel</code>
<code>\@showtasknumbersfalse</code>	<code>\ifx</code> 108 70,
..... 1		72, 108, 116, 118
<code>\@showtasknumberstrue</code>	L	<code>\tr@field</code> <u>107</u> ,
..... 3	<code>\LaTeX</code> 139	151, 155, 159,
		163, 167, 171, 175
	N	<code>\tr@title</code> 107, 145
<code>_</code> 116, 118	<code>\newif</code> 1, 12, 15	<code>\tr@usesfieldname</code> .
A	<code>\newline</code> 151, 155, 159,	.. <u>47</u> , 110, 116,
<code>\addcontentsline</code> .. 143	163, 167, 171, 175	118, 124, 133,
	<code>\newlistentry</code> .. 99, 100	151, 155, 159,
C	<code>\newlistof</code> 101, 104, 105	163, 167, 171, 175
<code>\color@</code> 21	<code>\numberline</code> 143	<code>\tr@usestatecolor</code> .
<code>\csname</code> .. 27, 33, 48, 53	 31, 81, 86, 89
	P	<code>\tr@version</code> 18
D	<code>\par</code> 131	<code>\traddfieldname</code> ...
<code>\DeclareOption</code> 3	<code>\printtaskreportversion</code>	.. <u>47</u> , 59–65, 68, 78
<code>\detokenize</code> 27, <u>19</u>	<code>\traddstatecolor</code> ..
32, 33, 48, 52, 53	<code>\ProcessOptions</code> 5 <u>20</u> , 39–44
	<code>\protect</code> 143	<code>\trlisttasks</code>
E	<code>\protected@edef</code> ... 142	.. 93, 101, 104, 105
<code>\else</code> 34, 54,		<code>\trsetboxcolors</code> . <u>38</u> , 46
102, 111, 114, 117	R	<code>\trsetcolon</code> <u>77</u>
<code>\empty</code> 108	<code>\refstepcounter</code> ...	<code>\trsetfieldsnames</code> . <u>58</u>
<code>\end</code> 181 138, 139, 141	<code>\trsettasklisttitle</code>
<code>\endcsname</code> 27,	<code>\relax</code> 5 92, 95
32, 33, 48, 52, 53	<code>\RequirePackage</code> ... 6–9	<code>\trsettitlelabel</code> .. <u>58</u>
<code>\ERROR</code> 24		<code>\trstatuscolor</code> . <u>82</u> , 144
<code>\expandafter</code>	S	<code>\trtask</code> <u>127</u>
.. 24, 27, 33, 48, 53	<code>\string</code> 21	<code>\trtask@temp</code> .. 133,
		136, 142, 143, 145
F	T	
<code>\fi</code> 36, 56,	<code>\taskreportversion</code> .	U
106, 113, 119, 120 <u>18</u> , 19	<code>\usestatecolor</code> <u>20</u>

List of Tasks

5.1	Make coffe	4
5.2	Drink a coffe	4
5.3	Roast the beef	4
5.4	Drink another beer	5
5.5	Drink a beer	5
5.6	Task without a name	5
5.7	Task without a name	5
5.8	arg 2	5
6.1	Drink a coffe	6
6.2	Drink a coffe	7
6.3	Drink a coffe	7
6.4	Drink a coffe	7
6.5	Drink a coffe	8
6.6	Drink a coffe	8
6.7	Drink a coffe	9
6.8	Drink a coffe	9
6.9	Drink a coffe	10
6.10	Drink a coffe	11
10.1	Make a L ^A T _E X Kanbam document	16