The tutodoc class Tutorial-style documentation

Christophe, BAL

Dec 4, 2024 - Version 1.7.0

The tutodoc class ¹ is used by its author to semantically produce documentation of LAT_EX packages and classes in a tutorial style ² using a sober rendering for reading on screen.

Remark: this documentation is also available in French.

Last changes

ያ Break.

- Format: the scrartcl class replaces the venerable article. This implies better placement of the margin notes with the options retained for loading scrartcl.
- LATEX code: the macro \tdocinlatex has been renamed \tdoclatexin.
- Color key names will be hyphenated where necessary: this implies the following changes.
 - 1. Indicate the latest changes: the colchges option of the environments has been renamed col-chges.
 - 2. Showcases: for the environment tdocshowcase and the macro \tdocshowcaseinput, the colstripe and coltext options have been renamed col-stripe and col-text.

Fix.

• Admonitions: for the \newkeytheorem used with the draft theme, postheadhook = \leavevmode has been added (this is necessary because the content can naturally be of the list type).

New.

- Documentation: addition of a section listing dependencies.
- Class options.
 - 1. Options not specific to tutodoc are passed on to the class in charge of general formatting.
 - 2. The scrartcl options fontsize and DIV can't be used because their values are fixed by tutodoc.
- The macro \tdocinEN respects the English linguistic rules.
- $\bullet\,$ Indicate the latest changes.
 - 1. Add the environment $\begin{tdoctodo} \dots \end{tdoctodo}$.
 - 2. Each environment has a new option col for the colour of the content indicating changes.

C Update.

- draft theme and changes: the environments for the latest changes stop to use icons.
- Documentation: the theme gallery uses a better fake example.

Technical information.

- Simplified organisation of configuration files in the final project.
 - 1. CSS-like: use of one file per theme with a name like tutodoc-bw.css.cls.
 - 2. Locale: use of names like tutodoc-en.loc.cls.

¹The name comes from "tuto·rial-type doc·umentation".

²The idea is to produce an efficient PDF file that can be browsed for one-off needs. This is generally what is expected of coding documentation.

Contents

I Dependencies

tutodoc admits the following dependencies (the dates in brackets are those of the versions used during the latest tests).

•	scrartcl.cls	(2024/10/24)	•	clrstrip.sty	(2021/08/28)
•	csquotes.sty	(2024/04/04)	•	fontawesome5.sty	(2022/05/02)
•	geometry.sty	(2020/01/02)	•	hyperref.sty	(2024/11/05)
•	inputenc.sty	(2024/02/08)	•	keytheorems.sty	(2024/11/11)
•	marginnote.sty	(2018/08/09)	•	minted.sty	(2024/11/17)
•	tcolorbox.sty	(2024/10/22)			

II General formatting imposed

1 Font size and page geometry

The scrartcl class is loaded via the fontsize = 10pt option, and the geometry package manages the page dimensions.

& Warning.

The macros for dating and versioning presented in the XI section require fixed settings for page geometry and font size.

2 Titles and table of contents

The selected settings are directly visible in this documentation.

3 Dynamic links

The hyperref package is imported, if it hasn't already been, and the settings chosen are just for the colors of links relating to citations, files, internal links, and finally url (this color will depend on the theme chosen).

III What language is used by the tutodoc class?

This documentation loads the babel package via \usepackage[english]{babel} a package that tutodoc does not load. On the other hand, the tutodoc class identifies en as the main language used by babel.³ As this language is included in the list of languages taken into account, see below, the tutodoc class will produce the expected effects.

• en : English. • es : Spanish. • fr : French.

1 Note.

Packages babel and polyglossia are taken into account.

& Caution.

If the choice of main language is not made in the preamble, the mechanism used will fail with unintended side effects (see warning that follows).

🙎 Warning.

When a language is not supported by tutodoc, a warning message is issued, and English is selected as the language for tutodoc.

 $^{^3 \}text{Technically},$ we use \BCPdata{language} which returns a language in short format.

IV What does that mean in "English"?

The macro \tdocinEN and its starred version are useless for English speakers because they have the following effects.

Cool and top stand for \tdocinEN*{cool} and \tdocinEN{top}.

Cool and top stand for "cool" and "top" in English.

The macro \tdocinEN and its starred version are based on \tdocquote: for example, "semantic" is obtained via tdocquote{semantic}.

i Note.

As the text "in English" is translated into the language detected by tutodoc, the macro \tdocinEN and its starred version become useful for non-English speakers.

V Choose your theme

To modify the general layout, there is the tutodoc class option theme = <choice> where <choice> can take the following values.

- bw: a black-and-white theme with some shades of grey.
- color: a coloured theme: this is the default value. w
- dark: a dark theme ideal for resting the eyes.
- draft: a theme for a printout such as to look for content errors that aren't necessarily easy to spot in front of a screen.

i Note.

At the end of this document, after the change history, you'll find a gallery of use cases for these different themes: go to appendix page 23.

VI Highlighting content

1 Note.

The environments presented in this section ^a add a short title indicating the type of information provided. This short text will always be translated into the language detected by the tutodoc class.

^aThe formatting comes from the keytheorems package.

1 Content in the reading flow

💉 Important.

All the environments presented in this section share the same counter, which will be reset to zero as soon as a section with a level at least equal to a \section is opened.

i Examples

Numbered examples, if required, are indicated via \begin{tdocexa} ... \end{tdocexa}, which offers an optional argument for adding a mini-title. Here are two possible uses.

\begin{tdocexa}
 An example...
\end{tdocexa}

\begin{tdocexa} Example VI.1. An example...

\begin{tdocexa} [Mini title]
 Useful?
\end{tdocexa}

\end{tdocexa}

• Tip.

It can sometimes be useful to return to the line at the start of the content. The code below shows how to proceed (this trick also applies to the tdocrem environment presented next). Note in passing that the numbering follows that of the previous example as desired.

ii Some remarks

Everything happens via \begin{tdocrem} ... \end{tdocrem}, which works identically to the tdocexa environment, as shown in the following example.

2 Flashy content

i Note.

The formatting proposed here is the default one, but others are possible by changing the theme: see the gallery of use cases in the appendix page 23. As for the icons, they are obtained via the fontawesome5 package, and the \t docicon macro manages the spacing in relation to the text. a

^aFor example, \fbox{tdocicon{faBed}{Fatigued}} produces ☐ Fatigued ...

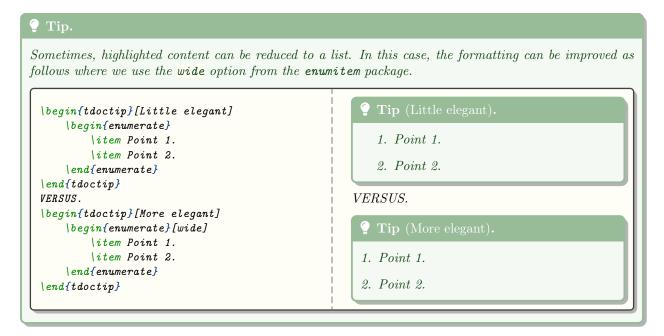
i A tip

The tdoctip environment is used to give tips. Here's how to use it.

```
\begin{tdoctip}
A tip.
\end{tdoctip}

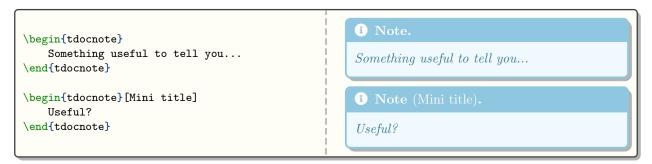
\begin{tdoctip} [Mini title]
Useful?
\end{tdoctip}

\text{Useful?}
```



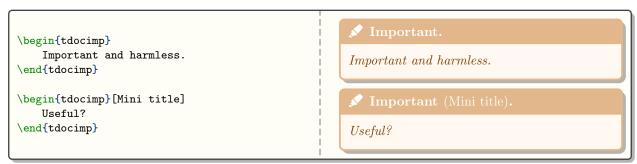
ii Informative note

The tdocnote environment is used to highlight useful information. Here's how to use it.



iii Something important

The tdocimp environment is used to indicate something important but harmless.



iv Caution about a delicate point

The tdoccaut environment is used to indicate a delicate point to the user. Here's how to use it.

```
\begin{tdoccaut}
    Caution, caution...
\end{tdoccaut}

\begin{tdoccaut} (Mini title)
    Useful?

\end{tdoccaut}

Useful?
```

v Warning of danger

The tdocwarn environment is used to warn the user of a trap to avoid. Here's how to use it.



VII Specify packages, classes, macros or environments

Here's what you can type semantically.

```
\tdoccls{myclass} is for...
                                                    myclass is for...
\tdocpack{mypackage} is for...
                                          11
                                                    mypackage is for...
\tdocmacro{onemacro} is for...
                                          //
                                                    \onemacro is for...
\tdocenv{env} produces...
                                          //
                                                    \begin{env} ...\end{env} produces...
\tdocenv[{[opt1]<opt2>}]{env}
                                          //
                                                    \begin{env}[opt1]<opt2>...\end{env}
Just \tdocenv*{env}...
                                          11
                                                    Just env...
Finally \t ([opt1] < pt2>)] {env}...
                                                    Finally env...
        % For copy and paste.
```

Remark VII.1. Unlike \tdoclatexin, \tdocenv and \tdocenv* macros don't color the text they produce. In addition, \tdocenv{monenv} produces \begin{monenv}...\end{monenv} with spaces to allow line breaks if required.

🙎 Warning.

The optional argument of the \t docenv macro is copied and pasted ^a when rendering. This may sometimes require the use of protective braces, as in the example above.

 a Remember that almost anything is possible from now on.

VIII Origin of a prefix or suffix

To explain the names chosen, there is nothing like indicating and explaining the short prefixes and suffixes used. This is easily done as follows.

```
      \tdocpre{sup} relates to...
      \sup relates to...

      \tdocprewhy{sup.erbe} means...
      \sup erbe means...

      \emph{\tdocprewhy{sup.er} for...}
      \sup er for...
```

Remark VIII.1. The choice of a full stop to split a word allows words with a hyphen to be used, as in \tdocprewhy{bric.k-breaker} which gives bric·k-breaker.

IX A real-life rendering

It is sometimes useful to render code directly in the documentation. This type of rendering must be dissociable from the explanatory text.

1 With a colored stripe

Example IX.1 (With default text). It can be useful to show a real rendering directly in a document. ⁴ This is done via \begin{tdocshowcase} ... \end{tdocshowcase} as follows.

```
| begin{tdocshowcase}
| bfseries A bit of code \LaTeX.
| bigskip
| emph{\large End of the awful demo.}
| end{tdocshowcase}
```

The result is the following rendering. ⁵

Start of the real output

A bit of code LATEX.

End of the awful demo.

End of the real output

Remark IX.2. See the section 4 on page 12 to easily obtain code followed by its actual rendering as in the previous example.

1 Note.

The explanatory texts adapt to the language detected by tutodoc.

Example IX.3 (Change the colors and/or the texts).

This will produce the following.

i Note.

In the previous example, the text uses the proposed darkened orange. On the other hand, red is used as a base to obtain the colors used for the strip: the transformations used depend on the theme chosen.^a You should also be aware that behind the scenes, the macro \tdocruler is used.

```
\tdocruler[red]{A decorated pseudo-title}

A decorated pseudo-title
```

⁴Typically when making a demo.

⁵Behind the scenes, the strip is created effortlessly using the clrstrip package.

^aFor example, the themes bw and draft ignore the key col-stripe!

& Warning.

With the default settings, if the code to be formatted begins with an opening bracket, use \string as in the following example.

```
\begin{tdocshowcase}
\string[This works...]
\end{tdocshowcase}
```

This will produce the following.

[This works...]

End of the real output

2 Without a coloured strip

The rendering of \begin{tdocshowcase} ...\end{tdocshowcase} with a coloured strip may not be suitable, or sometimes may not be acceptable despite the work done by clrstrip. It is possible not to use a coloured strip, as we will see straight away.

Example IX.4. The use of \begin{tdocshowcase} [nostripe] ... \end{tdocshowcase} indicate to not use clrstrip. Here is an example.

This will produce the following.

Start of the real output

End of the real output

Example IX.5 (Change the colors and/or the texts).

This will produce the following horror.

3 By importing the LATEX code

To obtain renderings by importing the code from an external file, simply use the \tdocshowcaseinput macro whose option uses the syntax of that of \begin{tdocshowcase} ...\end{tdocshowcase} and the mandatory argument corresponds to the path of the file.

Example IX.6. The following was obtained via \tdocshowcaseinput{external.tex}.

Blablobli, blablobli, blablobli, blablobli, blablobli...

 \blacksquare End of the real output \blacksquare

As for \tdocshowcaseinput[col-stripe = red, col-text = orange!75!black]{external.tex}, this will produce the color change shown below.

Blablobli, blablobli, blablobli, blablobli, blablobli...

End of the real output

X Use cases in LATEX

Documenting a package or class is best done through use cases showing both the code and the corresponding result. 6

1 "Inline" codes

The \tdoclatexin macro ⁷ can be used to type inline code in a similar way to \verb or like a standard macro (see brace management in the last case below). Here are some examples.

```
1: \tdoclatexin|\$a^b = c\$|
2: \tdoclatexin+\tdoclatexin|\$a^b = c\$|+ \\
3: \tdoclatexin\{\tdoclatexin\{\$a^b = c\$}\}

1: \$a^b = c\$
2: \tdoclatexin|\$a^b = c\$|
3: \tdoclatexin\{\$a^b = c\$}
```

i Note.

The \tdoclatexin macro can be used in a footnote: see below. a In addition, a background color is deliberately used to subtly highlight the codes \LaTeX.

^a\$minted = TOP\$ has been typed \tdoclatexin+\$minted = TOP\$+ in this footnote...

2 Directly typed codes

Example X.1 (Side by side). Using \begin{tdoclatex}[sbs]...\end{tdoclatex}, we can display a code and its rendering side by side. Consider the following code.

```
\begin{tdoclatex}[sbs]

$A = B + C$
\end{tdoclatex}
```

This will produce the following.

Example X.2 (Following). \begin{tdoclatex} ... \end{tdoclatex} produces the following result, which corresponds to the default option std. 8

```
A = B + C
A = B + C
```

Example X.3 (Just the code). Via \begin{tdoclatex}[code] ... \end{tdoclatex}, we'll just get the code as shown below.

 $^{^6\}mathrm{Code}$ is formatted using the minted package.

⁷The name of the macro \tdoclatexin comes from "in·line LATEX".

⁸std refers to the "standard" behaviour of tcolorbox in relation to the minted library.

```
$A = B + C$
```

& Warning.

With default formatting, if the code begins with an opening bracket, the default option must be explicitly indicated. Consider the following code.

```
\begin{tdoclatex}[std]
    [Strange... Or not!]
\end{tdoclatex}
```

This will produce the following.

```
[Strange... Or not!]
[Strange... Or not!]
```

Another method is to use the \string primitive. Consider the following code.

```
\begin{tdoclatex}
\string[Strange... Or not!]
\end{tdoclatex}
```

This will produce the following.

```
[Strange... Or not!]
```

3 Imported codes

For the following codes, consider a file with the relative path examples-listing-xyz.tex, and with the following contents.

```
% Just one demo.
$x y z = 1$
```

The \tdoclatexinput macro, shown below, expects the path of a file and offers the same options as the tdoclatex environment.

Example X.4 (Side by side).

```
\ttdoclatexinput[sbs]{examples-listing-xyz.tex}
```

This produces the following layout.

```
\begin{cases} % Just one demo. \\ \$x \ y \ z = 1 \end{cases} xyz = 1
```

Example X.5 (Following).

This produces the following formatting where the default option is std.

```
% Just one demo.
$x y z = 1$
```

```
xyz = 1
Example X.6 (Just the code).
 \tdoclatexinput[code] {examples-listing-xyz.tex}
This produces the following layout.
 % Just one demo.
 $x y z = 1$
4 Imported codes put into practice
Example X.7 (Showcase). The following comes from \tdoclatexshow{examples-listing-xyz.tex}.
                                  ■ Start of the rendering in this doc. ■
 % Just one demo.
 $x y z = 1$
This gives:
                                         Start of the real output
xyz = 1
                                         ■ End of the real output ■
                                    ■ End of rendering in this doc. ■

   Note.

 The default texts take into account the language detected by tutodoc.
Example X.8 (Changing the explanatory text). Using the key explain, you can use custom text. Thus,
tdoclatexshow[explain = Here is the rendering.] {examples-listing-xyz.tex} will give the follow-
                             Start of the rendering in this doc.
 % Just one demo.
 $x y z = 1$
Here is the rendering.
                                         ■ Start of the real output ■
xyz = 1
                                         ■ End of the real output ■
                                   ■ End of rendering in this doc. ■
Example X.9 (The options available). In addition to the explanatory text, it is also possible to use all the
options of tdocshowcase environment, see IX page 8. Here is an example to illustrate this.
 \ttdoclatexshow[explain
                           = What comes next is colorful...,
                 before
                           = Rendering below.,
```

```
\tdoclatexshow[explain = What comes next is colorful...,
before = Rendering below.,
after = Finished rendering.,
col-stripe = orange,
col-text = blue!70!black]
{examples-listing-xyz.tex}
```

This will produce the following.

Start of the rendering in this doc.

```
% Just one demo.
$x y z = 1$
```

What comes next is colorful...

xyz=1Finished rendering.

End of rendering in this doc.

XI Indicate changes

To make it easier to monitor a project, it is essential to provide a history indicating the changes made when a new version is published.

1 When?

You can either date something, or version it, in which case the version number can be dated.

Example XI.1 (Dating new products). The \tdocdate macro is used to indicate a date in the margin, as in the following example.

This gives:

2023-09-24

2020-05-08

Example XI.2 (Versioning new features, possibly with a date). Associating a version number with a new feature is done using the \tag{tdocversion} macro, with the colour and date being optional arguments.

■ End of the real output

This gives:

■ Start of the real output ■

10.2.0-beta 2023-12-01 10.2.0-alpha

 \blacksquare End of the real output

Example XI.3 (Caution with paragraph titles). The following example shows that a date and/or version must be placed just after a paragraph title, and not before it.

This gives:

Start of the real output

1.2.3 2024-11-23 A well-versioned title. Blah, blah,

2024-11-23

End of the real output

Example XI.4 (Adjust vertical positioning). If required, you can modify the vertical offset used to place dates and versions in the margin, the default value being (-8 pt).

 $This\ gives:$

■ Start of the real output ■

This is what it looks like without vertical movement.

1.2.3 2024-10-29 End of the real output

Important

- 1. The \tdocdate and \tdocversion macros require two compilations.
- 2. The final rendering of the dates takes into account the language detected by tutodoc: for example, if French is selected, the dates will be displayed in the format DD/MM/YYYY.

☆ Caution.

Only the use of the digital format YYYY-MM-DD is verified, a and this is a choice! Why? Quite simply because dating and versioning explanations should be done semi-automatically to avoid any human bugs.

^aTechnically, checking the validity of a date using LATEX3 presents no difficulty.

2 What's new?

tutodoc offers the macro \tdocstartproj and different environments to indicate quickly and clearly what has been done during the changes made, or to come.⁹

i Note.

For icons, see the note at the beginning of the section 2 page 5.

i Sobriety first

Example XI.5 (Just for the very first version).

\tdocstartproj{1st version of the project.}	\$ 1st version of the project.
---	--------------------------------

Example XI.6 (For new features).

\begin{tdocnew} \item Info 1 \item Info 2 \end{tdocnew}	♦ New.• Info 1• Info 2	
---	---	--

Example XI.7 (For updates).

\begin{tdocupdate} \item Info 1 \item Info 2 \end{tdocupdate}	C Update. • Info 1 • Info 2	

Example XI.8 (For breaks).

\begin{tdocbreak} \item Info 1 \item Info 2 \end{tdocbreak}	P Break.Info 1Info 2	
---	--	--

Example XI.9 (For problems).

\begin{tdocprob} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
--

Example XI.10 (For fixes).

\begin{tdocfix} \item Info 1 \item Info 2 \end{tdocfix}	Fix.Info 1Info 2	
---	--	--

⁹The user doesn't need all the technical details.

Example XI.11 (Roadmap).

\begin{tdoctodo} \item Info 1 \item Info 2 \end{tdoctodo}	⊘ Todo.• Info 1• Info 2	
---	--	--

Example XI.12 (Technical information).

\begin{tdoctech} \item Info 1 \item Info 2 \end{tdoctech}	■ Technical information.• Info 1• Info 2
---	--

Example XI.13 (Selectable themes with an icon).

```
\begin{tdoctopic}{To hide}<\faEyeSlash>
% An icon from fontawesome5.
\item Info 1...
\item Info 2...
\end{tdoctopic}

\text{To hide.}

Info 1...

Info 2...
```

Example XI.14 (Selectable themes without icons).

\begin{tdoctopic}{End of icons} \item Info 1 \item Info 2 \end{tdoctopic}	End of icons. ● Info 1 ● Info 2
---	-----------------------------------

ii De la couleur si besoin

iii Color if necessary

It may be useful to highlight certain changes: this can only be done by modifying the content color.

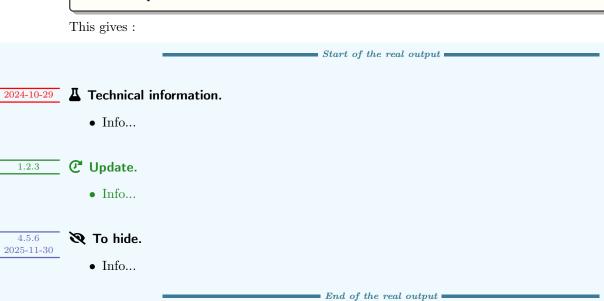
Example XI.15 (A flashy first version).

```
\tdocstartproj[DarkOrchid]%
{Brightly colored version 1.}
```

Example XI.16 (Outstanding fixes).

3 The what and the when

The optional keys **col**, **date** and **version** allow to date and version a change of a particular type. Here are some examples of use.



XII Ornament

Let's finish this documentation with a small formatting tool that is very useful.

Bla, bla, bla	
\tdocsep % Practical for demarcation.	Bla, bla, bla
This works with enumerations.	This works with enumerations.
<pre>\begin{itemize} \item Underline. \end{itemize}</pre>	• Underline.
\tdocsep % Uniform behaviour.	Ble, ble, ble
Ble, ble, ble	

XIII Contribute

i Note.

You don't need to be a coder to take part in translations, including those that are useful for the running of tutodoc.

1 Complete the translations

1 Note.

The author of tutodoc manages the French and English versions of the translations.

😾 Caution.

Although we're going to explain how to translate the documentation, it doesn't seem relevant to do so, as English should suffice these days.^a

^aThe existence of a French version is simply a consequence of the native language of the author of tutodoc.

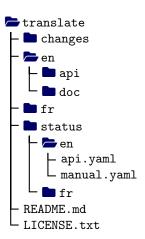


Figure 1: Simplified view of the translation folder

The translations are roughly organized as in figure 1 where only the folders important for the translations have been "opened". A little further down, the section v explains how to add new translations.

i The fr and en folders

These two folders, managed by the author of tutodoc, have the same organization; they contain files that are easy to translate even if you're not a coder.

ii The changes folder

This folder is a communication tool where important changes are indicated without dwelling on minor modifications specific to one or more translations.

iii The status folder

This folder is used to keep track of translations from the project's point of view. Everything is done via well-commented YAML files, readable by a non-coder.

iv The README.md and LICENSE.txt files

The LICENSE.txt file is aptly named, while the README.md file takes up in English the important points of what is said in this section about new translations.

v New translations

Important.

The api folder contains translations relating to the functionalities of tutodoc. Here you'll find TXT files for editing with a text or code editor, but not with a word processor. The content of these files uses commented lines in English to explain what tutodoc will do; these lines begin with //. Here's an extract from such a file, where translations are made after each = sign, without touching the preceding, as this initial piece is used internally by the tutodoccode.

```
// #1: year in format YYYY like 2023.
// #2: month in format MM like 04.
// #3: day in format DD like 29.
date = #1-#2-#3
```

 $^{^{10}}$ This was the organization on October 5, 2024, but it's still relevant today.

```
// #1: the idea is to produce one text like
       "this word means #1 in English".
in_EN = #1 in english
```

i Note.

The doc folder is reserved for documentation. It contains files of type TEX that can be compiled directly for real-time validation of translations.

& Warning.

Only start from one of the fr and en folders, as these are the responsibility of the tutodoc author.

Let's say you want to add support for Italian from files written in English. 11

Method 1: git.

- 1. Obtain the entire project folder via https://github.com/bc-tools/for-latex/tree/tutodoc. Do not use the main branch, which is used to freeze the latest stable versions of projects in the single https://github.com/bc-tools/for-latex repository,.
- 2. In the tutodoc/contrib/translate folder, create a it copy of the en folder, with the short name of the language documented in the page "IIETF language tag" from Wikipedia.
- 3. Once the translation is complete in the it folder, you'll need to communicate it via https://github. com/bc-tools/for-latex/tree/tutodoc using a classic git push.

Method 2: communicate by e-mail.

- 1. By e-mail with the subject "tutodoc CONTRIB en FOR italian", request a version of the English translations (note the use of the English name for the new language). Be sure to respect the subject of the e-mail, as the author of tutodoc automates the pre-processing of this type of e-mail.
- 2. You will receive a folder named italian containing the English version of the latest translations. This folder will be the place for your contribution.
- 3. Once the translation is complete, you will need to compress your italian file in zip or rar format before sending it by e-mail with the subject "tutodoc - CONTRIB - italian".

2 Improving the source code



Important.

If you want to participate in tutodoc you'll need to use the LATEX3 programming paradigm.

Participation as a coder is made via the https://github.com/bc-tools/for-latex/tree/tutodoc repository corresponding to the tutodoc development branch.

☆ Caution.

Do not use the main branch, which is used to freeze the latest stable versions of projects in the single https://github.com/bc-tools/for-latex repository.

XIV History

2024-12-04

ያ Break.

• Format: the scrartcl class replaces the venerable article. This implies better placement of the margin notes with the options retained for loading scrartcl.

 $^{^{11}\}mathrm{As}$ mentioned above, there is no real need for the \mathtt{doc} documentation folder.

- LATEX code: the macro \tdocinlatex has been renamed \tdoclatexin.
- Color key names will be hyphenated where necessary: this implies the following changes.
 - 1. Indicate the latest changes: the colchges option of the environments has been renamed col-chges.
 - 2. Showcases: for the environment tdocshowcase and the macro \tdocshowcaseinput, the colstripe and coltext options have been renamed col-stripe and col-text.

Fix.

• Admonitions: for the \newkeytheorem used with the draft theme, postheadhook = \leavevmode has been added (this is necessary because the content can naturally be of the list type).

New.

- Documentation: addition of a section listing dependencies.
- Class options.
 - 1. Options not specific to tutodoc are passed on to the class in charge of general formatting.
 - 2. The scrartcl options fontsize and DIV can't be used because their values are fixed by tutodoc.
- The macro \tdocinEN respects the English linguistic rules.
- Indicate the latest changes.
 - 1. Add the environment $\begin{tdoctodo} \dots \end{tdoctodo}$.
 - 2. Each environment has a new option col for the colour of the content indicating changes.

C Update.

- draft theme and changes: the environments for the latest changes stop to use icons.
- Documentation: the theme gallery uses a better fake example.

⚠ Technical information.

- Simplified organisation of configuration files in the final project.
 - 1. CSS-like: use of one file per theme with a name like tutodoc-bw.css.cls.
 - 2. Locale: use of names like tutodoc-en.loc.cls.

1.6.2 2024-10-30

New.

- The macros \tdocdate and \tdocversion has a new final optional argument <voffset> to choose a specific vertical offset.
- Better environments to indicate the changes made.
 - 1. The new optional keys col, date and version allow to date and version a change of a specific topic.
 - 2. Use of \paragraph for the title.

C Update.

- Version and changes: the font of the margin notes will always have a normal shape.
- Ornament: use of a \cleaders to avoid orphean rules at the bottom of a page.

1.6.1 2024-10-28

Technical information.

• The naming rules of CTAN need the use of CSS files named tutodoc-*.css.cls.sty.

1.6.0 2024-10-27

P Break.

- The showcase environment and its descendants: the color key has been renamed col-stripe.
- The macro \tdoclinkcolor has been renamed tutodoc@link@color for internal use.

New.

- The theme class option allows you to choose different formatting themes.
- Change log: addition of the tdoctech environment for technical information.
- The showcase environment and its descendants: the col-text key can also be used to change the text color.
- The new functionalities have been documented.

C Update.

• Change log: the tdocupdate environment uses the icon | C | instead of | % |

ℱ Fix.

• The Spanish translations were not included in the previous version! Don't laugh too hard...

1.5.0 2024-10-19

Technical information.

• Version 3 of minted is taken into account.

ያ Break.

- The tutodoc class replaces the now-defunct tutodoc package (for the moment, the young class offers no specific options).
- The \tdocruler macro is now used via \tdocruler[<color>]{<text>} (remember that the old syntax was \tdocruler{<text>}{<color>}).

New.

- The class is usable in Spanish.
- The documentation contains a new section explaining how to contribute.

Fix.

- The \tdocdate macro did not handle date format and formatting.
- Colored frames did not color text after a page break.

 $1.4.0 \\ 2024-09-28$

ያ Break.

- The tdoccaution environment has been renamed tdoccaut for simplified input.
- Content highlighting: examples and remarks, indicated via the tdocexa and tdocrem environments, are always numbered using a common counter.
- The unused macro \tdocxspace has been deleted.

New.

- Change log: the \tdocstartproj macro is used to manage the case of the first public version.
- Code factorization: the \tdocicon macro is responsible for adding icons in front of text.

C Update.

- Colors: the \tdocdarkcolor and \tdoclightcolor macros offer an optional argument.
 - 1. \tdocdarkcolor: the amount of color in relation to black can be optionally defined.
 - 2. \tdoclightcolor: the transparency rate can be optionally defined.
- Content highlighting: reduced space around content in colored frames.
- Versioning: better vertical spacing thanks to \vphantom.

 $1.3.1 \\ 2024-09-26$

New.

• Star version of \tdocenv to display only the environment name.

1.3.0 2024-09-25

• Version 3 of minted cannot be used for the moment as it contains bugs: see https://github.com/gpoore/minted/issues/401. We therefore force temporarily the use of version 2 of minted.

P Break.

• The tdocimportant environment has been renamed tdocimp for simplified input.

New.

- Change log: proposed environments use icons.
- Content highlighting: colored frames with icons are proposed for the following environments.
 - 1. tdoccaution
- 2. tdocimp

3. tdocnote

4. tdoctip

5. tdocwarn

1.2.0-a 2024-08-23

C Update.

- \tdocversion
 - 1. The version number is above the date.
 - 2. The spacing is better managed when the date is absent.

Fix.

• Content highlighting: the French translations of "caution" and "danger" were incorrect.

1.1.0 2024-01-06

New.

- Change log: two new environments.
 - 1. \begin{tdocbreak} ...\end{tdocbreak} for breaking changes which are not backward compatible.
 - 2. \begin{tdocprob} ...\end{tdocprob} for identified problems.
- \tdoclatexin: a light yellow is used as the background color.

1.0.1 2023-12-08



- \tdocenv: spacing is now correct, even if the babel package is not loaded with the French language.
- \begin{tdocshowcase}[nostripe]...\end{tdocshowcase}: page breaks around "framing" lines should be rare from now on.

1.0.0 2023-11-29

First public version of the project.

Appendix – Theme gallery



Each example is exactly a two pages long PDF and has been inserted directly into this document (so don't be surprised by the page numbers).

The "bw" theme

I. Liens

A very big link, but at least we can see it.

II. LATEX listings

Typing in on-line code such as $E = m c^2 \neq \frac{3}{14}$ is useful, as is showing use cases such as the following one.

```
Formatted \LaTeX\ code is great: $E = m c^2$ or $pi \neq \frac{3}{14}$. Formatted LATeX code is great: E = mc^2 or pi \neq \frac{3}{14}.
```

There's also a less invasive side-by-side mode. Nice! No?

```
Formatted \LaTeX\ code is great: \\ $E = m c^2$ or $\pi \neq \frac{3}{14}$. Formatted LATeX code is great: E = mc^2 \text{ or } \pi \neq \frac{3}{14}.
```

III. Highlighting, versioning and dating

1. tdocexa, tdocrem

1.7.0 2024-12-04 In the flow of the text, it is always useful to be able to provide examples and comments to supplement the main content.

Example III.1. What to say 1? I don't know, but it's nice. No?

Remark III.2. What to say ²? I don't know, but it's nice. No?

2. tdocnote, tdoctip...

Depending on the context of use, it is sometimes necessary to be able to highlight content by indicating its degree of importance.

```
What to say a? I don't know, but it's nice. No?

aLet's not forget the footnotes...
```

• Tip.

What to say? I don't know, but it's nice. No?

🖍 Important.

What to say? I don't know, but it's nice. No?

Caution

What to say? I don't know, but it's nice. No?

¹Let's not forget the footnotes...

 $^{^2}$ Let's not forget the footnotes...

& Warning.

What to say? I don't know, but it's nice. No?

3. tdocbreak, tdocfix...

& A new demonstration section...

Ø Todo.

 $\bullet\,$ A gallery would be welcome...

In a change log, it is important to visualise the types of changes clearly. This makes it easier for the user to read!

ያ Break.	⊁ Fix.	♦ New.	Ö Problem.
• Infos	• Infos	• Infos	• Infos
■ Technical information.	C Update.	Ø Todo.	
• Infos	• Infos	• Infos	

The "color" theme

I. Liens

A very big link, but at least we can see it.

II. LATEX listings

Typing in on-line code such as $E = m c^2 \neq \frac{14}$ is useful, as is showing use cases such as the following one.

```
Formatted \LaTeX\ code is great: $E = m c^2$ or $pi \neq \frac{3}{14}$. Formatted IATeX code is great: E = mc^2 or pi \neq \frac{3}{14}.
```

There's also a less invasive side-by-side mode. Nice! No?

```
Formatted \LaTeX\ code is great: \\ $E = m c^2$ or $\pi \neq \frac{3}{14}$. Formatted LATeX code is great: E = mc^2 \text{ or } \pi \neq \frac{3}{14}.
```

III. Highlighting, versioning and dating

1. tdocexa, tdocrem

1.7.0 2024-12-04 In the flow of the text, it is always useful to be able to provide examples and comments to supplement the main content.

Example III.1. What to say ¹? I don't know, but it's nice. No ?

Remark III.2. What to say ²? I don't know, but it's nice. No?

2. tdocnote, tdoctip...

Depending on the context of use, it is sometimes necessary to be able to highlight content by indicating its degree of importance.

```
What to say a? I don't know, but it's nice. No?

aLet's not forget the footnotes...
```

Tip.

What to say? I don't know, but it's nice. No ?

🖍 Important.

What to say? I don't know, but it's nice. No?

☆ Caution.

What to say? I don't know, but it's nice. No?

¹Let's not forget the footnotes...

 $^{^2}$ Let's not forget the footnotes...

& Warning.

What to say? I don't know, but it's nice. No?

3. tdocbreak, tdocfix...

 $\mbox{\mbox{\mbox{$\rlap 4$}}} \ A$ new demonstration section...

Ø Todo.

• A gallery would be welcome...

In a change log, it is important to visualise the types of changes clearly. This makes it easier for the user to read!

Problem.

● Infos...

Problem.

• Infos...

• Infos...

• Infos...

• Infos...

• Infos...

The "dark" theme

I. Liens

A very big link, but at least we can see it.

II. LATEX listings

Typing in on-line code such as $E = m c^2 \neq \frac{14}{14}$ is useful, as is showing use cases such as the following one.

```
Formatted \LaTeX\ code is great: $E = m c^2$ or $pi \neq \frac{3}{14}$. 

Formatted IATeX code is great: E = mc^2 or pi \neq \frac{3}{14}.
```

There's also a less invasive side-by-side mode. Nice! No?

```
Formatted \LaTeX\ code is great: \\ $E = m c^2$ or $\pi \neq \frac{3}{14}$. Formatted LATeX code is great: E = mc^2 \text{ or } \pi \neq \frac{3}{14}.
```

III. Highlighting, versioning and dating

1. tdocexa, tdocrem

1.7.0 2024-12-04 In the flow of the text, it is always useful to be able to provide examples and comments to supplement the main content.

Example III.1. What to say ¹? I don't know, but it's nice. No ?

Remark III.2. What to say ²? I don't know, but it's nice. No?

2. tdocnote, tdoctip...

Depending on the context of use, it is sometimes necessary to be able to highlight content by indicating its degree of importance.

What to say ? I don't know, but it's nice. No?

aLet's not forget the footnotes...

• Tip.

What to say? I don't know, but it's nice. No ?

Important.

What to say? I don't know, but it's nice. No ?

☆ Caution.

What to say? I don't know, but it's nice. No ?

 $^{^1\}mathrm{Let's}$ not forget the footnotes...

²Let's not forget the footnotes...

& Warning.

What to say? I don't know, but it's nice. No ?

3. tdocbreak, tdocfix...

& A new demonstration section...

Ø Todo.

• A gallery would be welcome...

In a change log, it is important to visualise the types of changes clearly. This makes it easier for the user to read!

Problem.

Infos...

Infos..

The "draft" theme

I. Liens

A very big link, but at least we can see it.

II. LATEX listings

Typing in on-line code such as $E = m c^2 \neq \frac{3}{14}$ is useful, as is showing use cases such as the following one.

```
Formatted \LaTeX\ code is great: $E = m c^2$ or $pi \neq \frac{3}{14}$. 
 Formatted IATEX code is great: E=mc^2 or pi\neq \frac{3}{14}.
```

There's also a less invasive side-by-side mode. Nice! No?

```
Formatted LATeX\ code is great: \\ $E = m c^2$ or $\pi \neq \frac{3}{14}$. 
 E = mc^2 \text{ or } \pi \neq \frac{3}{14}.
```

III. Highlighting, versioning and dating

1. tdocexa, tdocrem

1.7.0 2024-12-04 In the flow of the text, it is always useful to be able to provide examples and comments to supplement the main content.

Example III.1. What to say ¹? I don't know, but it's nice. No ?

Remark III.2. What to say ²? I don't know, but it's nice. No?

2. tdocnote, tdoctip...

Depending on the context of use, it is sometimes necessary to be able to highlight content by indicating its degree of importance.

Note III.3. What to say ³? I don't know, but it's nice. No?

Tip III.4. What to say? I don't know, but it's nice. No?

Important III.5. What to say? I don't know, but it's nice. No?

Caution III.6. What to say? I don't know, but it's nice. No?

Warning III.7. What to say? I don't know, but it's nice. No ?

3. tdocbreak, tdocfix...

[Init] A new demonstration section...

Todo.

• A gallery would be welcome...

In a change log, it is important to visualise the types of changes clearly. This makes it easier for the user to read!

Break. Fix. New. Problem.

• Infos... • Infos... • Infos... • Infos...

Technical information. Update. Todo.

• Infos... • Infos...

¹Let's not forget the footnotes...

²Let's not forget the footnotes...

³Let's not forget the footnotes...