

Proof-at-the-end, or how to move proofs in appendix in LaTeX

Léo Colisson

Contents

1	Introduction	1
2	Demo	2
3	Quickstart	2
3.1	Super Quick QuickStart	2
3.2	Install	4
3.3	Use in your project	5
3.4	A note on <code>hyperref</code>	6
4	Use cases	7
4.1	Configuration and how to use and create styles	7
4.2	Usual styles	9
4.3	Categories, or how to move proofs in different sections	10
4.4	Comments	10
4.5	Restate a theorem	11
4.6	Translate the links	11
4.7	Write a sketch of proof in the main text	12
4.8	Split the article in two documents, one for the main body and one for the appendix	12
5	List of options	13
6	Package options	15
7	Troubleshooting	16
7.1	The link after the proof does not mention the section, but something else (an equation, a figure, etc.)	16
7.2	I get an error <code>ERROR: File ended while scanning use of \@xverbatim.</code> when using <code>verbatim</code> inside a theorem	16
7.3	The proof writes <code>Proof (Proof of Lemma X)</code> (or something else) instead of <code>Proof of Lemma X.</code>	17

8 Changelog **17**

9 Contributions **18**

WARNING: this package is definitely usable, but may not be as tested and stable as standard packages. Note also that it may change a bit in the future, but we will of course try to avoid as much as possible backward incompatibilities. To ensure your document won't change when the package get updates, it is always safer to copy the `proof-at-the-end.sty` file at the root of your project.

1 Introduction

This package aims to provide a way to easily move proofs in the appendix. You can:

- Move proofs in different places/sections by giving different “categories” to the theorems
- Create links from theorem to proof, and from proof to theorem (and configure the text)
- Restate the theorem in appendix (or before)
- Choose what you want to keep in the main text or in appendix (you can put in appendix either everything, just the proof, the theorem and the proof by keeping (or not) a copy in the body of the article. . .)
- Duplicate the proof in appendix and in the current section, practical to use syntex during the proof writing
- Add text that would appear only in the appendix (or in both body and appendix)
- Easily change the defaults, and create your own styles/environments
- Include sketch of proof in the main text, and full proof in appendix
- Change the text of the link, for example to translate into another language
- Have a nice environment-based commands in order to mimic the usual theorem/proof structure.
- Separate the main article and the body in two documents (some journals may ask for a separate file for the appendix)

NB: This project is hosted on github at <https://github.com/leo-colisson/proof-at-the-end> . Feel free to contribute, report bugs, or read/copy-paste the documentation/examples from there.

Disclaimer: This package is still in beta and not considered as stable.

This package is licensed under LPPL v1.3, and the last version of this package on CTAN is 2019/05/21.

2 Demo

If you just want to see an example of what you can do, you can directly open the file `demo.pdf` (also available online at <https://github.com/leo-colisson/proof-at-the-end/raw/master/demo.pdf>) to see what is possible, or generate it with

```
git clone https://github.com/leo-colisson/proof-at-the-end.git
pdflatex demo.tex && pdflatex demo.tex
```

3 Quickstart

3.1 Super Quick QuickStart

If `proof-at-the-end` it's not installed in your CTAN distribution, copy the `proof-at-the-end.sty` file in your project. Then, in your project, create new theorem/lemma environments (using any tool you like, like `asmthm`, `ntheorem` and `thmtools`), load the library using `\usepackage[createShortEnv]{proof-at-the-end}` (note that you need the version 2022/01/28 to have `createShortEnv` defined, otherwise you need to use directly `theoremEnd`) and write your theorem using:

```
\begin{thmE}[My title][end, restate]
  I am a restated theorem whose proof goes in appendix (compile me twice).
\end{thmE}
\begin{proofE}
  And I am a proof.
\end{proofE}
```

The options (here `restate` and `end`) tells what should go in appendix, how to configure the links... Here the proof should go in appendix, and the `restate` options states that the theorem should be restated before the proof. Note that in case you don't want any title but do want to add options, make sure to insert an empty pair of brackets like in `\begin{thmE}[] [end, text link=]` (the option `text link` is used to choose the text of the link to the proof, writing `text link=` is used to display no link at all).

You can find below a full example to compile which should produce this output (sorry, this is a screenshot, that way I can export it easily in the github page). Just make sure to compile twice to get references.

You can directly copy them (with a more complete demo) in the github page here <https://github.com/leo-colisson/proof-at-the-end/>.

```
\documentclass{article}

\usepackage{xparse} % Not needed with recent LaTeX
\usepackage{amsthm}
```

1 Theorems

Theorem 1.1 (My title). *I am a theorem*

See `proof` on page `1`

2 Proofs

Theorem 1.1 (My title). *I am a theorem*

Proof of `Theorem 1.1` And I am a proof. □

Figure 1: Screenshot of the quickstart

```
% Create new theorems, or use ntheorem/thmtools/...
\newtheorem{thm}{Theorem}[section]
% Default uses autoref, but you can also use cleveref, see the documentation.
% Say to autoref that "thm" are Theorems.
\newcommand{\thmautorefname}{Theorem}

% Load the library. createShortEnv automatically creates the shortcuts
% thmE, theoremE, lemmaE, propositionE, corollaryE, proofE. See \newEndThm for more details.
\usepackage[createShortEnv]{proof-at-the-end}

\begin{document}

\section{Theorems}

% If you don't have any title, leave an empty bracket like \begin{thmE}[][end]
\begin{thmE}[My title][end, restate]
  I am a theorem
\end{thmE}
\begin{proofE}
  And I am a proof.
\end{proofE}

\section{Proofs}
\printProofs

\end{document}
```

3.2 Install

If your CTAN distribution is recent enough, you have nothing to do (if you are using overleaf, note that you can configure your project to use a more recent TeXLive distribution: this packages is included starting from TeXLive 2019). Otherwise if it's not yet in your CTAN distribution, first download the `proof-at-the-end.sty` file and insert it in the root of your project with the following commands on unix (you can also clone this repository if you prefer, or just manually download or copy/paste the files on Windows).

```
cd <your project>
repopratend="https://raw.githubusercontent.com/leo-colisson/proof-at-the-end"
wget ${repopratend}/master/proof-at-the-end.sty
```

It also requires a recent version of xparse (not present in TeXLive 2018 and before basically), otherwise you will get an error like:

```
Unknown argument type 'b' replaced by 'm'
```

While we used to provide for simplicity a `xparse.sty` file (version 2018-04-12) in older versions of this repository, we removed it to avoid license issues and conflicts with recent LaTeX distributions that already include xparse. Nevertheless, if you are using an older distribution, you can also download xparse manually here, unzip it, and copy the `.sty` files contained in `tex/latex/l3packages/xparse/` at the root of your project. On linux, you can do that via:

```
cd <your project>
wget http://mirrors.ctan.org/install/macros/latex/contrib/l3packages.tds.zip
unzip -d . -j l3packages.tds.zip 'tex/latex/l3packages/xparse/*'
rm -f l3packages.tds.zip xparse.ltx
```

If you have an old distribution of LaTeX (2017 or before basically), you may also need a more recent expl3 or you will get an error like:

```
Package xparse Error: Support package l3kernel too old.
```

To install it, download the zip files <http://mirrors.ctan.org/install/macros/latex/contrib/l3kernel.tds.zip> and <http://mirrors.ctan.org/install/macros/latex/contrib/l3backend.tds.zip> (l3backend is since July 1st, 2019 a dependency of l3kernel), unzip, and copy the content of the directories `tex/latex/l3kernel/` (first archive) and `tex/latex/l3backend/` (second archive) into your project. On linux it's a matter of two commands in your project:

```
cd <your project>
wget http://mirrors.ctan.org/install/macros/latex/contrib/l3kernel.tds.zip
unzip -d . -j l3kernel.tds.zip 'tex/latex/l3kernel/*'
rm l3kernel.tds.zip
wget http://mirrors.ctan.org/install/macros/latex/contrib/l3backend.tds.zip
unzip -d . -j l3backend.tds.zip 'tex/latex/l3backend/*'
rm l3backend.tds.zip
```

If you don't want to pollute your main project with all these files, you may be interested to put them in a subfolder and update the environment variable `TEXINPUTS` or, if you use `latexmk` or `overleaf`, you can write instead a `latexmkrc` file as explained here.

If you still have errors, it's likely that your LaTeX distribution is just too old (LaTeX 2016 seems to be too old to install a recent `l3exp`). So just upgrade to a decent version, and you should not have any troubles anymore.

You can test your install by compiling the `demo.tex` file present in the repository. When it compiles, you are ready to start!

3.3 Use in your project

Now, you can load the library in your project by simply using:

```
\usepackage{proof-at-the-end}
```

Then, you can configure your theorem/lemma/... environments as usual, by using any counter you like...:

```
\usepackage{amssymb, amsthm, amsmath}
% Theorems
\newtheorem{thm}{Theorem}[section]
\newtheorem*{thm*}{Theorem}
\providecommand*\thmautorefname{Theorem}
% Lemmata
\newtheorem{lemma}[thm]{Lemma}
\newtheorem*{lemma*}{Lemma}
\providecommand*\lemmaautorefname{Lemma}
```

And inside your document, you can use the following syntax to create a new theorem:

```
\begin{theoremEnd}[OPTIONS]{THEOREM ENVIRONMENT}[OPTIONAL TITLE]
  YOUR THEOREM, with eventually labels like \label{thm:OPTIONAL LABEL}
\end{theoremEnd}
\begin{proofEnd} %% Optional environment
  YOUR (OPTIONAL) PROOF
\end{proofEnd}
```

For example:

```
\begin{theoremEnd}{thm}[Yes I can have a title]
  \label{thm:ilikelabels}
  Creating a new theorem is easy
\end{theoremEnd}
\begin{proofEnd}
  You want a proof? Here is it!
\end{proofEnd}
```

Since `theoremEnd` may be a bit verbose to use, we provide since the version 2022/01/28, an option: if you load the program using `\usepackage[createShortEnv]{proof-at-the-end}`, you will automatically have the shortcut environments `proofE` (to replace `proof`), `thmE` (to replace `thm`), `theoremE` (to replace... ok you got the pattern), `lemmaE`, `propositionE` and `corollaryE`. Note that it is your role to define the initial commands `thm...`, but you don't need to define them all, only those that you use. Then, you can use them like:

```
\begin{thmE}[My title][end, restate]
  I am a restated theorem whose proof goes in appendix (compile me twice).
\end{thmE}
\begin{proofE}
  And I am a proof.
\end{proofE}
```

Finally, to display the proofs, just use `\printProofs` where you like:

```
\printProofs
```

If you would like to display a lemma instead, just change `{thm}` into `{lemma}`, or into any other theorem environment you defined! You can now compile safely your document ;)

NB: if you want to make sure all the references are linked correctly, make sure to compile twice the document!

Isn't it simple ?

3.4 A note on hyperref

An older version of the package explicitly loaded `hyperref` with no option, and an option clash could occur if the user wanted to add options to `hyperref`. We now push the loading to the very end to avoid that issue, but if you want to load packages *after* `hyperref` (like `cleveref`), it is your job to include `hyperref`, for example by putting at the end of the preamble something like:

```
\usepackage[colorlinks]{hyperref}
\usepackage{cleveref}
```

4 Use cases

4.1 Configuration and how to use and create styles

You can very easily configure this package, and choose how each theorem/proof must be displayed by providing a value in `OPTIONS`. For example, if you would like to keep the proof of a theorem in the main text like any normal theorem, use the `normal` option.

```

\begin{theoremEnd}[normal]{thm}[A title]
  You can easily turn a theorem back into a normal theorem!
\end{theoremEnd}
\begin{proofEnd}
  And keep the proof with you!
\end{proofEnd}

```

If you loaded the package using `\usepackage[createShortEnv]{proof-at-the-end}` (starting from 2022/01/28), you also have preconfigured shortcuts for multiple environments, namely `proofE` (to replace `proof`), `thmE` (to replace `thm`), `theoremE` (to replace... ok you got the pattern), `lemmaE`, `propositionE`, `propertyE`, `factE` and `corollaryE`. This way, you can simplify the above as:

```

\begin{thmE}[A title][normal]
  You can easily turn a theorem back into a normal theorem!
\end{thmE}
\begin{proofE}
  And keep the proof with you!
\end{proofE}

```

(You can of course remove the [...] containing the options and the title, however if you want to add options without any title, **make sure to use empty brackets for the title**, like `\begin{thmE}[] [normal]`).

Since 2022/01/28, you can also create new theorem environment that directly come with a preconfigured style (you may also find `\prateendSetLocal` useful if you want to change the style in a whole section). For instance to create a `thmNormal` environment having the `normal` option and a `thmSpecial` environment having the all `end`, `category=mySpecial` options, just do:

```

\newEndThm[normal]{thmNormal}{thm} % 'thm' is the name of the existing thm environment
\newEndThm[all end, category=mySpecial]{thmSpecial}{thm}

```

This way, you can simply use `\begin{thmSpecial}[Special title] ... \end{thmSpecial}` to create a theorem going in the special category (see below). (you also have `\newEndProof[options]{newProofEnv}` to define new proofs, but it is less useful as options are typically given in the theorem).

The options are in fact a set of keys/values, thanks to `pgfkeys`. So you can combine them with comma separated list like that (order matters, as the right-most values may overwrite configuration set by left-most values):

```

\begin{theoremEnd}[proof at the end,
  no link to proof,
  text proof={Difficult proof}
]{thm}[A title]
  Each theorem can have a custom configuration!
\end{theoremEnd}
\begin{proofEnd}

```


Quite practical, isn't it?

```
\end{proofEnd}
```

You can easily create your own styles like that:

```
\pgfkeys{/prAtEnd/my great style/.style={
  proot at the end,
  no link to proof,
  text proof={Difficult proof},
}
}
```

You can also change the default configuration when you load the package by nesting the configuration into a `conf` key:

```
\usepackage[conf={normal, one big link}]{proof-at-the-end}
```

Note however that for now it is *not* possible to use macros directly inside the options when you load the package, so if you need to use more complicated configuration, you can overwrite the `global custom defaults` style for global configuration, and the `local custom defaults` style for local configuration (useful for example if you want to define a category for a single section):

```
\pgfkeys{/prAtEnd/global custom defaults/.style={
  one big link={Go to proof on page-\pageref*{proof:prAtEnd\pratendcountercurrent}}
}
}
```

or, for short (needs version > 2022/02/04):

```
\pratendSetGlobal{one big link="blabla"}
```

and for local configuration:

```
\pgfkeys{/prAtEnd/local custom defaults/.style={
  category=greattheorem
}
}
```

for short (needs version > 2022/02/04):

```
\pratendSetLocal{category=greattheorem}
```

Then you can use like that:

```
\begin{thmE}[Title]
  Here is a normal theorem with the proof in the main text.
\end{thmE}
\begin{proofE}
  The (optional) proof
\end{proofE}

\begin{thmE}[Title][end]
```

```

    Here is a theorem whose proof goes to the end.
\end{thmE}
\begin{proofE}
    The proof
\end{proofE}

\begin{thmE}[Title][all end]
    Here is a theorem that goes with the proof at the end.
\end{thmE}
\begin{proofE}
    The proof
\end{proofE}

```

Note also that it is possible to give options to the `proofEnd` environment, but it is usually useless, as it will automatically pick the parameters from the last `theoremEnd` environment. However, if for some reasons you want to change the options of the proof only, you can do it, but do it as your own risks ;)

4.2 Usual styles

We predefined some pretty common styles/options. The full list is at the end of this document, but here is a list of the most practical ones:

- `normal`: turn the theorem into a “normal” theorem, with the proof in the main text and nothing in appendix
- `category=yourcategory`: change the category of the theorem (see next sub-section)
- `end`: put the proof in appendix
- `all end`: put both the theorem and the proof in appendix
- `debug`: make sure the proof is written in the main text as well. Practical when you write the proof to be able to use `synctex` (if you use `synctex` with the proof in appendix, you will be unfortunately moved to a temporary file that this library is using. . . so **make sure you don't modify the files named like `*pratttheenddefaultcategory.tex` or all your changes will be lost at the next compilation!**).
- `one big link`: if you prefer to have a single big link instead of two links (one for the proof, one for the page)
- `one big link translated=Your translation`: to change/translate the text of the link easily
- `text link section`: put a link looking like “See proof in section XX.”. You can use `text link=` to remove this link.
- `text link section full proof`: put a link looking like “See full proof in section XX.”
- `text proof translated=Your translation`: to change/translate the text of the proof at the end easily
- `global custom defaults`: empty style that you can modify to change the configuration (globally)

- `local custom defaults`: empty style that you can modify to change the configuration (locally). Practical to set a category for a single section.

4.3 Categories, or how to move proofs in different sections

Let's imagine that you have some proofs that are easy to do, and some proofs that are long but interesting. You may want to put the easy proofs in a different place than the long proofs. It is super easy to do, you just need to give a category name to the option `category` like here:

```
\begin{theoremEnd}[category=mylongproofs]{thm}[A title]
  You can easily change the place of the proofs
\end{theoremEnd}
\begin{proofEnd}
  Just use a different category name!
\end{proofEnd}
```

and give this category name to `\printProofs` in the section where you would like to display the proofs:

```
\printProofs[mylongproofs]
```

4.4 Comments

You can also move some text in the appendix by using:

```
\textEnd{Your text that should go in appendix}
```

You can also give it a category as explained above, or configure it to be displayed in both the main text and at the end of the file with:

```
\textEnd[both]{I am a comment that is written in both the main text
and the appendix}
```

You can also use the environment notation like that:

```
\begin{textAtEnd}[options]
  You can also use the environment syntax.
\end{textAtEnd}
```

4.5 Restate a theorem

It is easy to restate a theorem in the appendix, to have both the theorem in the main text and in the appendix: just use the option `restate`:

```
\begin{theoremEnd}[end, restate]{thm}[A title]
  This theorem will be displayed both in main text and appendix.
\end{theoremEnd}
\begin{proofEnd}
  Just use restate option.
\end{proofEnd}
```

You can also use the option `restate command=yourcustomcommand` in order to create a macro `\yourcustomcommand` that will restate the theorem wherever you want (but after the definition).

If you want to (re)state a theorem *before* its definition (say in the introduction), there is also a special environment `theoremEndRestateBefore` that requires a (unique) custom name that you need to provide also later on in place of the real theorem with the option `restated before`:

```
\section{Introduction}
\begin{theoremEndRestateBefore}{thm}[Title]{anamethatisusedtorestate}
  It is possible to state the theorem before
  in the introduction, and restate it later
\end{theoremEndRestateBefore}

\section{Real definition}
\begin{theoremEnd}[restated before]{thm}
  anamethatisusedtorestate
\end{theoremEnd}
\begin{proofEnd}
  Proof of the theorem, put in place of the theorem the unique name
\end{proofEnd}
```

4.6 Translate the links

The more powerful way to change the text of the links is to redefine `text link` and `text proof` (see section List of options for more details). However we defined also some easy way to redefine the text using `one big link translated` and `text proof translated`. For example, to create your french style you can do:

```
\pgfkeys{/prAtEnd/french/.style={
  one big link translated={Voir preuve page},
  text proof translated={Preuve du}
}
}
```

4.7 Write a sketch of proof in the main text

You can include a sketch of proof in the main text by simply adding a proof in between `theoremEnd` and `proofEnd`. An alias option `see full proof` can also be used to change the link into “See full proof on page X.”:

```
\begin{theoremEnd}[see full proof]{thm}
  I can also write a sketch of proof, and put the full proof in appendix.
\end{theoremEnd}
\begin{proof}
  Hint: look at the alias options.
```

```

\end{proof}
\begin{proofEnd}
  You just use ``see full proof'' as an option
\end{proofEnd}

```

4.8 Split the article in two documents, one for the main body and one for the appendix

Since 2022/02/04, it is possible to include the appendix in a separate file. Internally, this library creates a new file having the form `NAME_OF_FILE-pratendNAME_CATEGORY.tex` containing the proofs to include (then, `\printProofs` only input that file). You can input that file from other files if you want to separate the body from the proof, but it won't work if you have restate theorems. You should use the option `external appendix` (starting from version 2022/02/04) to restate appropriately the theorem:

```
\usepackage[createShortEnv,conf={external appendix}]{proof-at-the-end}
```

Then, compile your main file and create a new file for the appendix, load the theorems environments as for the main file, in the preamble load the `xr` package to properly have references (otherwise you won't have the appropriate number for the theorems) using something like:

```

\usepackage{xr}
\externaldocument{name_of_main_file_without_extension}

```

and in this new document just add the appendix wherever you want:

```
\includeExternalAppendix{name_of_main_file_without_extension}
```

You can see an exemple in the github repository, with the two files `demo_external_appendix.tex` and `demo_external_appendix_part2.tex`. Whenever your proofs change in the main paper, make sure to compile first the main document (so that the auxiliary file gets updated), and then compile the appendix document. Note that you can still use `\printProofs` in the main document if you like (while writting your paper, it is certainly easier to keep the proofs and theorems in the same document).

5 List of options

Here is the list of fundamental options supported. Most options have a no version, with `no` written before. Note that you may prefer to use directly the alias/styles (see next paragraph) as the options listed here are quite fundamental and atomic.

- `category`: category of the proof (if you want to put proofs at several places), can be anything
- `proof here/no proof here`: put (or not) the proof in the main text

- `proof end/no proof end`: display the proof in appendix
- `restate/no restate`: restate the theorem in appendix
- `link to proof/no link to proof`: Display a link to the proof in the main text
- `opt all end/no opt all end`: put the theorem and proof only in appendix. You may prefer the alias `all end`, that also makes sure that the proof is indeed displayed in appendix.
- `text link`: text of the link to the proof, defaults to


```
{See \hyperref[proof:prAtEnd\pratendcountercurrent]{proof}
on page-\pageref{proof:prAtEnd\pratendcountercurrent}.}
```
- `text proof`: text displayed in place of “Proof” in the appendix. Defaults to


```
{Proof of \string\pratendRef{thm:prAtEnd\pratendcountercurrent}}
```
- `restate command`: name of a unique macro (without backslash) that will be defined as an alias to restate the theorem wherever you want
- `restated before`: if the theorems has been stated before (with `\theoremProofEndRestateBefore`), then we just need to put the restate command in place of the theorem, and enable this option
- `both/no both`: only for `\textInAppendix`, specifies that the text must be present in both the main text and the appendix.
- `external appendix`: to ensure the appendix can be included in another file (see details above), starting from version 2022/02/04.

Here are all the alias/styles (you can create you own as well), they are practical to quickly define a behaviours, but are made of the basic options listed above:

- `normal`: like a ‘normal’ theorem, without any proof in the appendix, and with a proof displayed in the main text. Shortcut for `proof here, no all end, no proof end, no link to proof, no restate, no both`.
- `end`: theorems whose proof need to go in the appendix. Shorcut for `proof at the end, link to proof`.
- `all end`: makes sure both the theorem and the proof are in appendix. Alias of `end`, `opt all end`.
- `proof at the end`: theorems whose proof need to go in the appendix contrary to `end` it does not make sure that there is a link to the proof. Shorcut for `no proof here, no all end, proof end, no both`.
- `debug`: make sure the proof is written in the main text as well (alias of `proof here, no opt all end`), it is quite practical to use when you write a proof to be able to use syntex features to move between the pdf and the file.
- `no link to theorem`: Remove the link from the proof to the theorem, alias of `text proof={\proofname}`

- **stared** (or **no number**): when you use the stared version of a theorem you don't have any number, so `cref` fails to write a nice link to the theorem. This option changes the text of "Proof", by keeping the link but writing only Proof. Equivalent to `text proof={\string\mbox{\string\hyperref[thm:prAtEnd\pratendcountercurrent]{\proofname}}}`
- **see full proof**: useful when you want to write in the main text only a sketch of proof, this alias writes a link `See full proof on page X..` Equivalent to `text link={See \hyperref[proof:prAtEnd\pratendcountercurrent]{full proof} on page~\pageref{proof:prAtEnd\pratendcountercurrent}.}`
- **one big link**: instead of two links, one for page, one for proof, put just one link around everything. It can also accept an optional argument which will be the text of the link, like `one big link=Go to the proof.` The default value is `See proof on page~\pageref*{proof:prAtEnd\pratendcountercurrent}.}`
- **one big link translated**: This is like `one big link`, but automatically add the page at the end (and a big link around). Practical to quickly define a translation like `one big link translated=Voir preuve page.` See also `text proof translated.`
- **text link section**: Put a link to the proof looking like "See proof in section X". Defaults to `text link={See \hyperref[proof:prAtEnd\pratendcountercurrent]{proof} in \pratendSectionlikeCref.}`
- **text link section full proof**: Put a link to the proof looking like "See full proof in section X". Defaults to `text link={See \hyperref[proof:prAtEnd\pratendcountercurrent]{full proof} in \pratendSectionlikeCref.}`
- **default text link**: default text for the link to the proof, equivalent of `text link={See \hyperref[proof:prAtEnd\pratendcountercurrent]{proof} on page~\pageref{proof:prAtEnd\pratendcountercurrent}.}`
- **default text proof**: default text for the proof in appendix, equivalent of `text proof={Proof of \string\pratendRef{thm:prAtEnd\pratendcountercurrent}}`
- **text proof only theorem**: equivalent of `text proof={\string\pratendRef{thm:prAtEnd\pratendcountercurrent}}` helpful if the class uses a custom proof environment that would write `Proof (Proof of Lemma X)` instead of `Proof of Lemma X.`: adding this style turns it into `Proof (Lemma X)`. See more details in the troubleshooting section.
- **text proof translated**: like `default text proof`, but takes one argument and use it instead of `Proof of`. Example: `text proof translated={Preuve du}`
- **text link external appendix**: style used to configure the text to display when the appendix is in an external file. Defaults to `text link={The proof is in the appendix.}`
- **external appendix**: useful when the the proofs are moved in an external file (read more above). In particular, it will ensure that normal theorems are used instead of `restate` since `restate` does not work across files. This will also apply `text link external appendix` to change the text accordingly.
- **only external appendix**: like `external appendix`, but without chang-

ing the text.

- **bare defaults:** default style that is loaded before anything else that configure by default a link to the proof, put the proof in appendix, use the category `defaultcategory`. It is an alias of `end`, `link to proof`, `no restate`, `category=defaultcategory`, `default text link`, `default text proof`, `restate command=pratenddummymacro`.
- **configuration options:** style that contains the options used to load the package. It is called right after **bare defaults**. Note that you cannot insert macro in the options, overwrite **global custom defaults** instead
- **global custom defaults:** empty style that you can overwrite to change the global defaults
- **local custom defaults:** empty style that you can overwrite to change the “local” defaults, like `category`
- **all defaults:** all the defaults, equivalent of **bare defaults**, **configuration options**, **global custom defaults**, **local custom defaults**

6 Package options

The package comes with multiple options: - The `conf={CONFIGURATION}` option is used to configure the default style. Usage: `\usepackage[conf={normal, text link section}]{proof-at-the-end}` - The `disablePatchSection` is useful to stop the library from patching the section-like commands and chapters (by default, we automatically add a `\label` at the end of the section which is used by `text link section` to find the Appendix). If you enable this option, you can manually add the label by using `\pratendAddLabel` after the appendix section (without any argument). - `commandRef=NAMECOMMAND`: By default, theorem are referenced to using `\autoref{}`. You can change the value of `NAMECOMMAND` to use another command, like `commandRef=Cref` to use `\Cref{}`. In that case, make sure to load both `hyperref` and `cleveref` (typically at the very end of your preamble) as `cleveref` needs to be loaded after `hyperref`. Internally, the library creates and uses `\pratendRef` instead of `\autoref`. - `createShortEnv`: creates (using `newEndThm` and `newEndProof`) the shortcut environments `proofE` (to replace `proof`), `thmE` (to replace `thm`), `theoremE` (to replace... ok you got the pattern), `lemmaE` and `corollaryE`. Note that it is still your job to define the `thm`, `lemma` environments.

7 Troubleshooting

Here are some common issues you may have, with explanations to solve them.

7.1 The link after the proof does not mention the section, but something else (an equation, a figure, etc.)

Sometimes, you may see something like `See proof in Equation A` instead of `See proof in Appendix A`. This issue was reported before and should be solved on newer versions (starting from 2022/01/27) by automatically patching sections. If you have this issue, just upgrade (for instance by copying the `proof-at-the-end.sty` file at the root of your project). We proceed by patching the `\chapter`, `\section`, `\subsection`, `subsubsection` and `\paragraph` commands to add `\pratendAddLabel` which will help the package to find the label of the current section. If you prefer the old behavior, you can disable it using the `disablePatchSection` option (more details in the package options).

7.2 I get an error `ERROR: File ended while scanning use of \@xverbatim`. when using `verbatim` inside a `theorem`

It turns out that this library needs to use the `+b` option of environments in order to manipulate appropriately the `theorem/proof`. Unfortunately, it means that it is not possible to use `verbatim` environments inside. While this answer suggests that it is impossible to avoid this issue, other people reported that it may be possible to use catcodes to capture the environment body `verbatim`, before using `scantokens` to reparse it, or to use `filecontentsdef` (see for instance this comment),). However, it may be possible that this make `syntex unusable`, so it may not worth the effort.

In anycase, there exists some workarounds, some of the are for instance give in the TeX FAQ. The simplest solution, if your code is simple/short enough, is to use `\texttt` (you can replace `backslash` with `\textbackslash` inside etc, and include it in a `verse` if you have multiple lines). If you have a longer text, you may also like `lrbox` to put your content inside a box, for instance like that:

```
%% Create the box
\newsavebox{\myEndBox}

\begin{lrbox}{\myEndBox}
  \begin{minipage}{1.0\linewidth}
    \begin{verbatim}
\newEndThm[normal]{thmE}{thm}
\newEndProof{proofE}{proof}
\end{verbatim}
    \end{minipage}
\end{lrbox}

%% Create a theorem:
\begin{thmE}
  Insert the box:
```

```

\begin{center}
  \usebox{\myEndBox}
\end{center}
\end{thmE}

```

7.3 The proof writes Proof (Proof of Lemma X) (or something else) instead of Proof of Lemma X.

If the proof starts with something different from Proof of Lemma X, you are likely to use a class/package that defines a `proof` environment different from the one defined in `amsthm`, like `svjour3` by Springer. To fix this, you have two options: if you want to keep using the original `proof` environment, and get something like Proof (Lemma X), you can either configure:

```

\prateGlobal{
  text proof={\string\prateRef{thm:prAtEnd\pratecountercurrent}},
}

```

or, if you use a version greater than 2025/06/11 (included), you can simply call the `text proof only theorem`, either like in `\prateGlobal{text proof only theorem}` or directly when loading the package:

```

\usepackage[createShortEnv, conf={text proof only theorem}]{proof-at-the-end}

```

Another option is to replace your class `proof` environment with, for instance, the one provided by `amsthm` like in:

```

% Removes the environment 'proof' defined by the class and replaces it with the one of amsthm
\let\proof\undefined
\let\endproof\undefined
\usepackage{amsthm}

```

but ensure that `amsthm` does not produce additional conflicts. This method will write Proof of Lemma X. before starting the proof.

8 Changelog

- 2025/06/11: Add `text proof only theorem`
- 2022/02/07:
 1. issue 2 was not really solved in fact... Now, I use a more robust method to detect the current section: I write in an AUX file the label of the proof section. This way, I don't need anymore to use dirty tricks to recover the section label.
 2. Change the default text when the appendix is in an external file, and add options like `text link external appendix` to customize the text appearing when the appendix is in an external file. Of course, you can still change it using `text link` directly.

- 2022/02/04:
 1. Add a way to put theorems in different files. (see `external appendix`)
 2. Change the path for auxiliary files (should be transparent for the user)
 3. Add commands to change local/global configuration easily (see `\pratendSetLocal`, `\pratendSetGlobal`)
- 2022/02/01:
 1. Fix a typo when defining the shortcut for lemma
 2. Add a shortcut for proposition
 3. Fix a small spacing issue in the text before the proof in Appendix
- 2022/01/28:
 1. Fix the issue when using sharps in a proof <https://github.com/leo-colisson/proof-at-the-end/issues/7>.
 2. Provide `\newEndThm`, `\newEndProof` and the option `createShortEnv` to quickly create environments.
- 2022/01/27:
 1. Patch `chapters/sections/...` to better detect the current section and fix issue 2. This can be disabled using the package option `disablePatchSection`.
 2. Add an option `commandRef` to use other ref libraries, like `cleveref` instead of `autoref`.
 3. Normal restatable theorem do not need to be restated. Fix issue 8.

9 Contributions

Feel free to contribute, report bugs, and send pull requests on the github repository <https://github.com/leo-colisson/proof-at-the-end> !

NB: the documentation is generated from the Markdown file `README.md` thanks to pandoc. These commands may help you:

```

%% Compile the demo
make demo
%% Clean the project
make clean
%% Generate the documentation
make doc
%% Generate a package for CTAN
make package

```