ProjLib Toolkit

User Manual

Jinwen Xu

May 2021, Beijing

Abstract

The ProjLib toolkit is designed to simplify the preparation before writing LaTeX documents. With PLtoolkit loaded, you no longer need to set up the theorem-like environments nor configure the appropriate multilingual settings. Additionally, a series of auxiliary functionalities have been introduced.

1 PLtoolkit

1.1 How to load it

Just add the following line to your preamble:

```
\usepackage{PLtoolkit}
```

> **Attention**
>
> Since cleveref is used internally, PLtoolkit needs to be placed after varioref and hyperref.

1.2 Options

PLtoolkit offers the following options:

- `draft` or `fast`
  - Fast mode. The functionality is appropriately reduced to get faster compilation speed, recommended to use during the writing phase.
- `palatino`、`times`、`garamond`、`biolinum`
  - Font options. As the name suggest, font with corresponding name will be loaded.
- `author`
  - Load PLauthor. For more information about its functionality, see the section on this package.
- `amssim`
  - Load PLamssim. For more information about its functionality, see the section on this package.
- `nothms`
  - The theorem-like environments will not be defined. You should use this option if you wish to apply your own theorem styles.

In addition, there are also some options of the components that can (and should) be passed as global options of the document class, such as `paperstyle` and `preview` of PLpaper. For more information, please refer to the corresponding section.

Corresponding to: ProjLib 2021/05/12

2 THE COMPONENTS

2.1 PLAMSSIM

PLamssim is used to simulate some features of the amsart class in a standard class, including:

- `\address`, `\curraddr`, `\email` and `\dedicatory` macro(the first three are provided by PLauthor)
- `\keywords` macro
- `\subjclass` macro
- `\thanks` can be written outside `\author`
- The abstract environment can be placed before `\maketitle`

These modifications would only take place in standard classes. In the $\mathcal{AMS}$ classes, PLamssim does not have any effect.

2.2 PLAUTHOR

PLauthor offers `\address`, `\curraddr` and `\email`, and allows you to enter multiple groups of author information. The standard usage is like this:

```
\author{⟨author 1⟩}
\address{⟨address 1⟩}
\email{⟨email 1⟩}
\author{⟨author 2⟩}
\address{⟨address 2⟩}
\email{⟨email 2⟩}
...
```

The mutual order of `\address`, `\curraddr` and `\email` is not important.

2.3 PLDATE

PLdate offers the `\PLdate`⟨*yyyy-mm-dd*⟩ macro to convert ⟨*yyyy-mm-dd*⟩ into the date format of the currently selected language. For example, in current English context, `\PLdate`{2022-04-01} would become: April 1, 2022.

For details on how to select a language, please refer to the section on PLlang.

2.4 PLDRAFT

PLdraft offers the following macros:

- `\dnf` or `\dnf<...>`. The effect is: To be finished #1 or To be finished #2: ....
  The prompt text changes according to the current language. For example, it will be displayed as Pas encore fini #3 in French mode.
- `\needgraph` or `\needgraph<...>`. The effect is:

  A graph is needed here #1

  or

  A graph is needed here #2: ...

  The prompt text changes according to the current language. For example, in French mode, it will be displayed as

  Il manque une image ici #3

For details on how to select a language, please refer to the section on PLlang.

2.5 PLlang

PLlang offers multi-language support, including simplified Chinese, traditional Chinese, English, French, German, Japanese, and Russian (among them, Chinese, Japanese, and Russian require appropriate TEX engines and fonts to support). The language can be selected by the following macros:

- \UseLanguage{⟨*language name*⟩} is used to specify the language. The corresponding setting of the language will be applied after it. It can be used either in the preamble or in the main body. When no language is specified, "English" is selected by default.
- \UseOtherLanguage{⟨*language name*⟩}{⟨*content*⟩}, which uses the specified language settings to typeset ⟨*content*⟩. Compared with \UseLanguage, it will not modify the line spacing, so line spacing would remain stable when CJK and Western texts are mixed.

⟨*language name*⟩ can be:

- Simplified Chinese: Chinese, chinese, SChinese, schinese, SimplifiedChinese or simplifiedchinese
- Traditional Chinese: TChinese, tchinese, TraditionalChinese or traditionalchinese
- English: English or english
- French: French or french
- German: German, german or ngerman
- Japanese: Japanese or japanese
- Russian: Russian or russian

In addition, you can also add new settings to selected language:

- \AddLanguageSetting{⟨*settings*⟩}
    – Add ⟨*settings*⟩ to all supported languages.
- \AddLanguageSetting(⟨*language name*⟩){⟨*settings*⟩}
    – Add ⟨*settings*⟩ to the selected language ⟨*language name*⟩.

For example, \AddLanguageSetting(German){\color{orange}} can make all German text displayed in orange (of course, one then need to add \AddLanguageSetting{\color{black}} in order to correct the color of the text in other languages).

2.6 PLlogo

PLlogo offers the \ProjLib macro to draw the logo, which looks like ProjLib. It is similar to ordinary text macros and can be used with different font size macros:

\tiny:	ProjLib
\scriptsize:	ProjLib
\footnotesize:	ProjLib
\normalsize:	ProjLib
\large:	ProjLib
\Large:	ProjLib
\LARGE:	ProjLib
\huge:	ProjLib
\Huge:	ProjLib

2.7 PLPAPER

PLpaper is mainly used to adjust the paper color. It has the following options:

- `paperstyle = ⟨paper style name⟩`
  - Set the paper color style. The options available for ⟨paper style name⟩ are: `yellow`, `dark` and `nord`.
- `yellowpaper`、`darkpaper`、`nordpaper`
  - Set the paper color style. The effect is the same as `paperstyle` with the corresponding ⟨paper style name⟩ specified.
- `preview`
  - Preview mode. Crop the white edges of pdf file for the convenience of reading.

It is recommended to pass these options as global options of the document class. In this way, the paper settings would be clear at a glance.

2.8 PLTHM

PLthm offers the configuration of theorem-like environments. It has the following option:

- `nothms`
  - The theorem-like environments will not be defined. You should use this option if you wish to apply your own theorem styles.

Preset environments include: `assumption`, `axiom`, `conjecture`, `convention`, `corollary`, `definition`, `definition-proposition`, `definition-theorem`, `example`, `exercise`, `fact`, `hypothesis`, `lemma`, `notation`, `problem`, `property`, `proposition`, `question`, `remark`, `theorem`, and the corresponding unnumbered version with an asterisk * in the name. The display of these environments will change according to the current language. For example, `theorem` will be displayed as "Theorem" in English mode and "Théorème" in French mode. For details on how to select a language, please refer to the section on PLlang.

When referencing a theorem-like environment, it is recommended to use smart reference \cref{⟨label⟩}. In this way, there is no need to explicitly write down the name of the corresponding environment every time.

If you need to define a new theorem-like environment, you must first define the name of the environment in the language used: \⟨name of environment⟩⟨language abbr⟩, where ⟨language abbr⟩ is the abbreviation of language, which can be:

CN	Simplified Chinese	DE	German
TC	Traditional Chinese	JP	Japanese
EN	English	RU	Russian
FR	French		

REMIND

If you want to define an environment with an asterisk * after the name, you don't need to write an asterisk in the ⟨name of environment⟩ above. See the example below.

And then define this environment in one of following five ways:

- \CreateTheorem*{⟨name of environment⟩}
  - Define an unnumbered environment ⟨name of environment⟩
- \CreateTheorem{⟨name of environment⟩}
  - Define a numbered environment ⟨name of environment⟩, numbered in order 1,2,3,…

- \CreateTheorem{⟨*name of environment*⟩}[⟨*numbered like*⟩]
  - Define a numbered environment ⟨*name of environment*⟩, which shares the counter ⟨*numbered like*⟩
- \CreateTheorem{⟨*name of environment*⟩}<⟨*numbered within*⟩>
  - Define a numbered environment ⟨*name of environment*⟩, numbered within the counter ⟨*numbered within*⟩
- \CreateTheorem{⟨*name of environment*⟩}(⟨*existed environment*⟩)
  \CreateTheorem*{⟨*name of environment*⟩}(⟨*existed environment*⟩)
  - Identify ⟨*name of environment*⟩ with ⟨*existed environment*⟩ or ⟨*existed environment*⟩*.
  - This method is usually useful in the following two situations:
    1. To use a more concise name. For example, with \CreateTheorem{thm}(theorem), one can then use the name thm to write theorem.
    2. To remove the numbering of some environments. For example, one can remove the numbering of the remark environment with \CreateTheorem{remark}(remark*).

> **REMIND**
>
> It uses amsthm internally, so the traditional theoremstyle is also applicable to it. One only needs declare the style before the relevant definitions.

Here is an example. The following code:

```
\def\proofideanameEN{Idea}
\CreateTheorem*{proofidea*}
\CreateTheorem{proofidea}<subsection>
```

defines an unnumbered environment proofidea* and a numbered environment proofidea (numbered within subsection) respectively. They can be used in English context. The effect is as follows (the actual style is related to the document class):

Idea | The proofidea* environment.

Idea 2.8.1 | The proofidea environment.

3 KNOWN ISSUES

- PLauthor is still in its preliminary stage, and thus its effect is still not as good as the relatively mature authblk in some aspects.
- PLlang: It is still quite problematic with the configuration of polyglossia, so main features are implemented through babel for now.
- PLpaper: the preview option is mainly implemented with the help of package geometry, so it does not work quite as well in the KOMA document classes.
- PLthm: The numbering and theorem-style settings of the theorem-like environments cannot be accessed by the user at present.
- PLthm: The localization of cleveref is not yet complete for all supported languages of PLlang, especially for Chinese, Japanese and Russian.
- The error handling mechanism is incomplete: there is no corresponding error prompt when some problems occur.

4 Usage example

4.1 Standard classes

In standard classes, one usually only need to configure the page size, hyperlinks and load PLtoolkit before actually start writing the document. Below is a complete example.

```latex
\documentclass{article}
\usepackage[a4paper,margin=.75in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{PLtoolkit} % Load the toolkit and use font Palatino

\UseLanguage{French} % Use French from here

\begin{document}

\title{Le Titre}
\author{Auteur}
\date{\PLdate{2022-04-01}}

\maketitle

\begin{abstract}
    Ceci est un résumé. \dnf<Plus de contenu est nécessaire.>
\end{abstract}

\section{Un théorème}

%% Theorem-like environments can be used directly
\begin{theorem}\label{thm:abc}
    Ceci est un théorème.
\end{theorem}

Référence du théorème: \cref{thm:abc} % It is recommended to use clever reference

\end{document}
```

If PLamssim is loaded, then one can adopt the $\mathcal{AMS}$ writing style in the document (of course, the original way is also valid, so always adding the option `amssim` usually does not cause problems). This way, the line that introduces PLtoolkit should be written as:

```latex
\usepackage[amssim,palatino]{PLtoolkit}
```

4.2 THE $\mathcal{AMS}$ CLASSES

In $\mathcal{AMS}$ classes, one usually only need to configure the page size, hyperlinks and load PLtoolkit before actually start writing the document. Below is a complete example.

```
\documentclass{amsart}
\usepackage[a4paper,margin=.75in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{PLtoolkit} % Load the toolkit and use font Palatino

\UseLanguage{French} % Use French from here

\begin{document}

\title{Le Titre}
\author{Auteur 1}
\address{Adresse 1}
\email{\href{Courriel 1}{Courriel 1}}
\author{Auteur 1}
\address{Adresse 1}
\email{\href{Courriel 2}{Courriel 2}}
\date{\PLdate{2022-04-01}}
\subjclass{*****}
\keywords{...}

\begin{abstract}
    Ceci est un résumé. \dnf<Plus de contenu est nécessaire.>
\end{abstract}

\maketitle

\section{Première section}

%% Theorem-like environments can be used directly
\begin{theorem}\label{thm:abc}
    Ceci est un théorème.
\end{theorem}

Référence du théorème: \cref{thm:abc} % It is recommended to use clever reference

\end{document}
```