

ProjLib 工具集

使用指南

许锦文

2021 年 5 月，北京

摘要

ProjLib 工具集之设计目的为简化 \LaTeX 文档撰写前的准备工作。只需要加载 PJLtoolkit，多语言设置就已准备就绪，定理类环境已被设置好可供直接使用，并且引入了一系列辅助功能。

1 PJLTOOLKIT

1.1 如何加载

只需要在导言部分加入这一行即可：

```
\usepackage{PJLtoolkit}
```

注意事项

由于其内部使用了 cleveref，PJLtoolkit 需要放在 varioref、hyperref 的后面。

1.2 选项

PJLtoolkit 提供下列选项以供选择：

- draft 或 fast
 - 快速模式。功能会适当缩减，但能够提高编译速度，建议在撰写阶段使用。
- palatino、times、garamond、biolinum
 - 字体选项。顾名思义，会加载相应名称的字体。
- author
 - 加载 PJLauthor。关于其详细功能，请参阅有关这一宏包的小节。
- amssim
 - 加载 PJLamssim。关于其详细功能，请参阅有关这一宏包的小节。
- nothms
 - 不设定定理类环境。如果你希望使用自己的定理样式，可以使用这一选项。

另外，还有一些组件的参数可以作为文档类的全局参数传递，例如 PJLpaper 的 paperstyle 和 preview 等。详细信息可以参阅对应的小节。

对应版本. ProjLib 2021/05/19

2 具体组件

2.1 PJLAMSSIM

PJLamssim 用于模拟 amsart 文档类的部分功能，包括：

- `\address`、`\curraddr`、`\email`、`\dedicatory` 命令 (前三者由 PJLauthor 提供)
- `\keywords` 命令
- `\subjclass` 命令
- `\thanks` 可以写在 `\author` 之外
- `abstract` 环境可以放在 `\maketitle` 的前面

这些功能只在标准文档类中启用。在 \LaTeX 文档类中，PJLamssim 不会起到任何效果。

2.2 PJLAUTHOR

PJLauthor 提供了 `\address`、`\curraddr`、`\email` 命令，并且允许输入多组用户信息。标准的输入方式是这样的：

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

其中 `\address`、`\curraddr`、`\email` 的相互顺序是不重要的。

2.3 PJLDATE

PJLdate 提供了 `\PJLdate{yyyy-mm-dd}` 命令，以将 `{yyyy-mm-dd}` 转换为对应语言的日期格式显示。例如，在当前的中文语境下，`\PJLdate{2022-04-01}` 会被显示为：2022 年 4 月 1 日。

关于如何选定语言，请参阅关于 PJLlang 的小节。

2.4 PJLDRAFT

PJLdraft 提供了下列命令：

- `\dnf` 或 `\dnf<...>`。效果为：`这里尚未完成 #1` 或 `这里尚未完成 #2：...`。
其提示文字与当前语言相对应，例如，在法语模式下将会显示为 `Pas encore fini #3`。
- `\needgraph` 或 `\needgraph<...>`。效果为：

`这里需要一张图片 #1`

或

`这里需要一张图片 #2：...`

其提示文字与当前语言相对应，例如，在法语模式下将会显示为

`Il manque une image ici #3`

关于如何选定语言，请参阅关于 PJLlang 的小节。

2.5 PJLLANG

PJLlang 提供了多语言支持, 包括简体中文、繁体中文、英文、法文、德文、日文、俄文 (其中中文、日文、俄文需要相应的 \TeX 引擎与字体支持)。可以通过下列命令来选定语言:

- `\UseLanguage{⟨language name⟩}`, 用于指定语言, 在其后将使用对应的语言设定。
 - 既可以用于导言部分, 也可以用于正文部分。在不指定语言时, 默认选定 “English”。
- `\UseOtherLanguage{⟨language name⟩}{⟨content⟩}`, 用指定的语言的设定排版 `⟨content⟩`。
 - 相比较 `\UseLanguage`, 它不会对行距进行修改, 因此中西文字混排时能够保持行距稳定。

`⟨language name⟩` 有下列选择:

- 简体中文: Chinese、chinese、SChinese、schinese、SimplifiedChinese 或 simplifiedchinese
- 繁体中文: TChinese、tchinese、TraditionalChinese 或 traditionalchinese
- 英文: English 或 english
- 法文: French 或 french
- 德文: German、german 或 ngerman
- 日文: Japanese 或 japanese
- 俄文: Russian 或 russian

另外, 还可以通过下面的方式来填加相应语言的设置:

- `\AddLanguageSetting{⟨settings⟩}`
 - 向所有支持的语言增加设置 `⟨settings⟩`。
- `\AddLanguageSetting(⟨language name⟩){⟨settings⟩}`
 - 向指定的语言 `⟨language name⟩` 增加设置 `⟨settings⟩`。

例如, `\AddLanguageSetting(German){\color{orange}}` 可以让所有德语以橙色显示 (当然, 还需要再加上 `\AddLanguageSetting{\color{black}}` 来修正其他语言的颜色)。

2.6 PJLLOGO

PJLlogo 提供了 `\ProjLib` 命令用于绘制 Logo, 效果为: ProjLib。它与普通的文字指令效果类似, 可以用于不同的字号:

<code>\tiny:</code>	ProjLib
<code>\scriptsize:</code>	ProjLib
<code>\footnotesize:</code>	ProjLib
<code>\normalsize:</code>	ProjLib
<code>\large:</code>	ProjLib
<code>\Large:</code>	ProjLib
<code>\LARGE:</code>	ProjLib
<code>\huge:</code>	ProjLib
<code>\Huge:</code>	ProjLib

2.7 PJLPAPER

PJLpaper 主要用于调节纸张颜色。它支持下列选项：

- `paperstyle = <paper style name>`
 - 设定纸张色彩样式。<paper style name> 可供选择的选项有：yellow、dark 与 nord。
- `yellowpaper`、`darkpaper`、`nordpaper`
 - 设定纸张色彩样式。效果与相应名称的 `paperstyle` 相同。
- `preview`
 - 预览模式，将会把 pdf 文件的白边去掉以方便阅读。

为了使用的方便，建议把这些选项作为文档类的全局参数，这样对于文档的纸张设定一目了然。

2.8 PJLTHM

PJLthm 提供定理类环境的设置。它支持下列选项：

- `nothms`
 - 不设定定理类环境。如果你希望使用自己的定理样式，可以使用这一选项。

预设的定理类环境包括：assumption、axiom、conjecture、convention、corollary、definition、definition-proposition、definition-theorem、example、exercise、fact、hypothesis、lemma、notation、problem、property、proposition、question、remark、theorem，以及相应的带有星号 * 的无编号版本。这些定理类环境在显示时会依据当前语言而相应变化，例如在中文模式下 theorem 会显示为“定理”，而在英文模式下则会显示为“Theorem”。关于如何选定语言，请参阅关于 PJLlang 的小节。

在引用定理类环境时，建议使用智能引用 `\cref{<label>}`。这样就不必每次都写上相应环境的名称了。

若需要定义新的定理类环境，首先要定义这个环境在所用语言下的名称 `\<name of environment>{<language abbr>}`，其中 `<language abbr>` 是语言的简写，分别为：

CN	简体中文	DE	德文
TC	繁体中文	JP	日文
EN	英文	RU	俄文
FR	法文		

提示

如果要定义名称后带有星号 * 的环境，那么在上面的 `<name of environment>` 中不用写星号。

然后用下面五种方式之一定义这一环境：

- `\CreateTheorem*{<name of environment>}`
 - 定义不编号的环境 `<name of environment>`
- `\CreateTheorem{<name of environment>}`
 - 定义编号环境 `<name of environment>`，按顺序编号
- `\CreateTheorem{<name of environment>}[<numbered like>]`
 - 定义编号环境 `<name of environment>`，与 `<numbered like>` 计数器共用编号

- `\CreateTheorem{<name of environment>}<numbered within>`
 - 定义编号环境 `<name of environment>`，在 `<numbered within>` 计数器内编号
- `\CreateTheorem{<name of environment>}<existed environment>`
`\CreateTheorem*{<name of environment>}<existed environment>`
 - 将 `<name of environment>` 与 `<existed environment>` 或 `<existed environment>*` 等同。
 - 这种方式通常在两种情况下比较有用：
 1. 希望定义更简洁的名称。例如，使用 `\CreateTheorem{thm}(theorem)`，便可以直接用名称 `thm` 来撰写定理。
 2. 希望去除某些环境的编号。例如，使用 `\CreateTheorem{remark}(remark*)`，便可以去除 `remark` 环境的编号。

提示

其内部使用了 `amsthm`，因此传统的 `theoremstyle` 对其也是适用的，只需在相关定义前标明即可。

下面提供一个例子。这三行代码：

```
\def\proofideanameCN{思路}
\CreateTheorem*{proofidea*}
\CreateTheorem{proofidea}<subsection>
```

可以分别定义不编号的环境 `proofidea*` 和编号的环境 `proofidea` (在 `subsection` 内编号)，它们支持在简体中文语境中使用，效果如下所示 (具体样式与所在的文档类有关)：

思路 | `proofidea*` 环境。

思路 2.8.1 | `proofidea` 环境。

3 目前存在的问题

- PJLauthor 仍然处于初步阶段，在很多方面还远远比不上相对成熟的 `authblk`。
- PJLlang：针对 `polyglossia` 的设置仍然存在许多问题，因此现在主要功能都是通过 `babel` 实现的。
- PJLpaper 的 `preview` 功能主要是通过 `geometry` 宏包实现的，因此在 KOMA 文档类中效果不好。
- PJLthm 对于定理类环境的编号与样式设定目前还无法由用户更改。
- PJLthm：智能引用针对所有 PJLlang 已支持语言的本地化尚不完整，主要是中文、日文与俄文。
- 错误处理功能不完善，在出现一些问题时没有相应的错误提示。
- 代码中有许多可优化之处，有些部分耗时过长，特别是 PJLthm 对定理类环境的定义。

4 使用示例

4.1 标准文档类

在标准文档类中，通常只需简要设置页面尺寸、超链接，再载入 PjLtoolkit，即可直接开始写作。下面是一段完整的示例。

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

\UseLanguage{French} % Use French from here

\begin{document}

\title{Le Titre}
\author{Auteur}
\date{\PjLdate{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}
```

如果使用了 PjLamssim，那么文章中就可以采用 $\mathcal{A}\mathcal{M}\mathcal{S}$ 文档类的写法 (当然，此时原始的写法也是成立的，因此始终添加 amssim 这一参数通常是没有问题的)。此时上文引入 PjLtoolkit 的那一行应该写为：

```
\usepackage[amssim,palatino]{PjLtoolkit}
```

4.2 $\mathcal{A}\mathcal{M}\mathcal{S}$ 文档类

在 $\mathcal{A}\mathcal{M}\mathcal{S}$ 文档类中，通常只需简要设置页面尺寸、超链接，再载入 PjLtoolkit，即可直接开始写作。下面是一段完整的示例。

```
\documentclass{amsart}
\usepackage[a4paper,margin=.75in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{PjLtoolkit} % 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{\PjLdate{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}
```
