minimart, write your articles or books in a simple and clear way

JINWEN XU
ProjLib@outlook.com
August 2021, Beijing

ABSTRACT

minimart is a series of styles and classes for you to typeset your articles or books in a simple and clear manner. The original intention in designing this series was to write drafts and notes that look simple yet not shabby. With the help of the ProjLib toolkit, also developed by the author, the classes provided here have multi-language support, preset theorem-like environments with clever reference support, and many other functionalities. Notably, using these classes, one can organize the author information in the AMS fashion, makes it easy to switch to journal classes later for publication.

Finally, this documentation is typeset using the minimart class (with the option classical). You can think of it as a short introduction and demonstration.

Contents

Before you start .......................................................... 1
1 Introduction .......................................................... 2
2 Usage and examples .................................................. 2
   2.1 How to load it ................................................... 2
   2.2 Example - minimart ............................................. 3
   2.3 Example - minimbook ......................................... 6
3 The options ............................................................ 7
4 Instructions by topic .................................................. 9
   4.1 Language configuration ........................................ 9
   4.2 Theorems and how to reference them ......................... 9
   4.3 Define a new theorem-like environment ..................... 10
   4.4 Draft mark ...................................................... 11
   4.5 Title, abstract and keywords .................................. 12
   4.6 Miscellaneous .................................................. 13
5 Known issues .......................................................... 13

Before you start

In order to use the package or classes described here, you need to:

1. install TeX Live or MikTeX of the latest possible version, and make sure that minimalist and projlib are correctly installed in your TeX system.
2. download and install the required fonts if needed.
3. be familiar with the basic usage of LaTeX, and knows how to compile your document with pdflatex, xelatex or luatex.

Corresponding to: minimalist 2021/08/11
minimalist is a series of styles and classes for you to typeset your articles or books in a simple and clear manner. The original intention in designing this series was to write drafts and notes that look simple yet not shabby.

The entire collection includes minimalist.sty, which is the main style shared by all of the following classes; minimart.cls for typesetting articles and minimbook.cls for typesetting books. They compile with any major \TeX engine, with native support to English, French, German, Italian, Portuguese (European and Brazilian) and Spanish typesetting via \useLanguage (see the instruction below for detail).

You can also found einfart and simplivre on CTAN. They are the enhanced version of minimart and minimbook with unicode support. With this, they can access to more beautiful fonts, and additionally have native support for Chinese, Japanese and Russian typesetting. On the other hand, they need to be compiled with Xe\TeX or Lua\TeX (not pdf\TeX).

With the help of the ProjLib toolkit, also developed by the author, the classes provided here have multi-language support, preset theorem-like environments with clever reference support, and many other functionalities such as draft marks, enhanced author information block, mathematical symbols and shortcuts, etc. Notably, using these classes, one can organize the author information in the AMS fashion, makes it easy to switch to journal classes later for publication. For more detailed information, you can refer to the documentation of ProjLib by running \texttt{texdoc projlib} in the command line.

2.1 How to load it

You can directly use minimart or minimbook as your document class. In this way, you can directly begin writing your document, without having to worry about the configurations.

\documentclass[minimart] or \documentclass[minimbook]

\textbf{Tip}

You may wish to use einfart or simplivre instead, which should produce better result. All the examples later using minimart or minimbook can be adopted to einfart and simplivre respectively, without further modification.

You can also use the default classes article or book, and load the minimalist package. This way, only the basic styles are set, and you can thus use your preferred fonts and page layout. All the features mentioned in this article are provided.

\documentclass[article] or \documentclass[book]
\usepackage[minimalist]
Let’s first look at a complete example of minimart (the same works for einfart).

\documentclass{minimart}
\usepackage{ProjLib}
\UseLanguage{French}
\begin{document}
\title{⟨title⟩}
\author{⟨author⟩}
\date{\PLdate{2022-04-01}}
\maketitle
\begin{abstract}
\dnf{⟨some hint⟩}
\end{abstract}
\begin{keyword}
AAA, BBB, CCC, DDD, EEE
\end{keyword}
\section{Un théorème}
\begin{theorem}\label{thm:abc}
\end{theorem}
Référence du théorème: \cref{thm:abc}
\end{document}

If you find this example a little complicated, don’t worry. Let’s now look at this example piece by piece.

2.2.1 Initialization

\documentclass{minimart}
\usepackage{ProjLib}

Initialization is straightforward. The first line loads the document class minimart, and the second line loads the ProjLib toolkit to obtain some additional functionalities.

2.2.2 Set the language

\UseLanguage{French}
This line indicates that French will be used in the document (by the way, if only English appears in your article, then there is no need to set the language). You can also switch the language in the same way later in the middle of the text. Supported languages include Simplified Chinese, Traditional Chinese, Japanese, English, French, German, Spanish, Portuguese, Brazilian Portuguese and Russian.

For detailed description of this command and more related commands, please refer to the section on the multi-language support.

2.2.3 Title, author information, abstract and keywords

\title{(title)}
\author{(author)}
\date{\PLdate{2022-04-01}}
\maketitle
\begin{abstract}
\langle abstract \rangle
\end{abstract}
\begin{keyword}
\langle keywords \rangle
\end{keyword}

This part begins with the title and author information block. The example shows the basic usage, but in fact, you can also write:

\begin{verbatim}
\author{(author 1)}
\email{\langle email 1 \rangle}
\author{(author 2)}
\email{\langle email 2 \rangle}
...
\end{verbatim}

In addition, you may also write in the \texttt{AMs} fashion, i.e.:

\begin{verbatim}
\title{(title)}
\author{(author 1)}
\email{\langle email 1 \rangle}
\author{(author 2)}
\email{\langle email 2 \rangle}
\date{\PLdate{2022-04-01}}
\subjclass{****}
\keywords{(keywords)}
\end{verbatim}

\footnote{The language Simplified Chinese, Traditional Chinese, Japanese and Russian requires Unicode support, thus the classes einfart or simplivre.}
\begin{abstract}
\abstract
\end{abstract}
\maketitle

2.2.4 Draft marks

\begin{verbatim}
\texttt{dnf}<\texttt{some hint}>
\end{verbatim}

When you have some places that have not yet been finished yet, you can mark them with this command, which is especially useful during the draft stage.

2.2.5 Theorem-like environments

\begin{verbatim}
\begin{theorem}\label{thm:abc}
Ceci est un théorème.
\end{theorem}
\end{verbatim}

Référence du théorème: \texttt{\cref{thm:abc}}

Commonly used theorem-like environments have been pre-defined. Also, when referencing a theorem-like environment, it is recommended to use \texttt{\cref{(label)}} — in this way, there is no need to explicitly write down the name of the corresponding environment every time.

\begin{tip}
If you wish to switch to the standard class later, just replace the first two lines with:

\begin{verbatim}
\documentclass{article}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino,amsfashion]{ProjLib}
\end{verbatim}

or to use the \texttt{\textsf{AMS}} class:

\begin{verbatim}
\documentclass[amsart]
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{ProjLib}
\end{verbatim}
\end{tip}
2.3 | Example - minimbook

Now let’s look at an example of minimbook (the same works for simplivre).

```latex
\documentclass{minimbook}
\usepackage{ProjLib}
\UseLanguage{French}
\begin{document}
\frontmatter
\begin{titlepage}
\langle code for titlepage \rangle
\end{titlepage}
\tableofcontents
\mainmatter
\part{\langle part title \rangle}
\parttext{\langle text after part title \rangle}
\chapter{\langle chapter title \rangle}
\section{\langle section title \rangle}
... 
\backmatter
... 
\end{document}
```

There is no much differences with minimart, only that the title and author information should be typeset within the titlepage environment. Currently no default titlepage style is given, since the design of the title page is a highly personalized thing, and it is difficult to achieve a result that satisfies everyone.

In the next section, we will go through the options available.
The minimalist offers the following options:

- The language options EN / english / English, FR / french / French, etc.
  - For the option names of a specific language, please refer to ⟨language name⟩ in the next section. The first specified language will be used as the default language.
  - The language options are optional, mainly for increasing the compilation speed. Without them the result would be the same, only slower.
- draft or fast
  - The option fast enables a faster but slightly rougher style, main differences are:
    * Use simpler math font configuration;
    * Do not use hyperref;
    * Enable the fast mode of ProjLib toolkit.

**Tip**

During the draft stage, it is recommended to use the fast option to speed up compilation. When in fast mode, there will be a watermark “DRAFT” to indicate that you are currently in the draft mode.

- allowbf
  - Allow boldface. When this option is enabled, the main title, the titles of all levels and the names of theorem-like environments will be bolded.
- classical
  - Classic mode. An antique-looking style will be used when this option is enabled, like in the current documentation.
- runin
  - Use the “runin” style for \subsubsection
- puretext or nothms
  - Pure text mode. Do not load theorem-like environments.
- delaythms
  - Defer the definition of theorem-like environments to the end of the preamble. Use this option if you want the theorem-like environments to be numbered within a custom counter.
- nothmmnum, thmmnum or thmmnum=⟨counter⟩
  - Theorem-like environments will not be numbered / numbered in order 1, 2, 3... / numbered within ⟨counter⟩. Here, ⟨counter⟩ should be a built-in counter (such as subsection) or a custom counter defined in the preamble (with the option delaythms enabled). If no option is used, they will be numbered within chapter (book) or section (article).
- regionalref, originalref
  - When referencing, whether the name of the theorem-like environment changes with the current language. The default is regionalref, i.e., the name corresponding to the current language is used; for example, when referencing a theorem-like environment in English context, the names "Theorem, Definition..." will be used no matter which language context the original environment is in. If originalref is enabled, then the
name will always remain the same as the original place; for example, when referencing a theorem written in the French context, even if one is currently in the English context, it will still be displayed as “Théorème”.

– In fast mode, the option originalref will have no effect.

Additionally, minimart and minimbook offers the following options:

• **a4paper** or **b5paper**
  – Optional paper size. The default paper size is 7in × 10in.

• **palatino, times, garamond, noto, biolinum | useosf**
  – Font options. As the name suggest, font with corresponding name will be loaded.
  – The useosf option is used to enable the old-style figures.

• **useindent**
  – Use paragraph indentation instead of inter-paragraph spacing.
4.1 Language configuration

minimart has multi-language support, including English, French, German, Italian, Portuguese (European and Brazilian) and Spanish. 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 (it is not case sensitive, for example, French and french have the same effect):

- Simplified Chinese: CN, Chinese, SChinese or SimplifiedChinese
- Traditional Chinese: TC, TChinese or TraditionalChinese
- English: EN or English
- French: FR or French
- German: DE, German or ngerman
- Italian: IT or Italian
- Portuguese: PT or Portuguese
- Portuguese (Brazilian): BR or Brazilian
- Spanish: ES or Spanish
- Japanese: JP or Japanese
- Russian: RU 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).

4.2 Theorems and how to reference them

Environments such as definition and theorem have been preset and can be used directly.

More specifically, preset environments include: assumption, axiom, conjecture, convention, corollary, definition, definition-proposition, definition-theorem, example, exercise, fact, hypothesis, lemma, notation, observation, problem, property, proposition, question, remark, theorem, and the corresponding unnumbered version with an asterisk * in the name.
The titles will change with the current language. For example, \texttt{theorem} will be displayed as “Theorem” in English mode and “Théorème” in French mode.

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

\begin{example}
\begin{verbatim}
\begin{definition}[Strange things] \label{def: strange} ...
\end{definition}
\end{verbatim}
will produce

\texttt{Definition 4.1 | (Strange things)} This is the definition of some strange objects. There is approximately a one-line spacing before and after the theorem environment, and there will be a symbol to mark the end of the environment.

$\texttt{\textbackslash cref\{def: strange\}}$ will be displayed as: \texttt{Definition 4.1}.

After using $\texttt{\textbackslash UseLanguage\{French\}}$, a theorem will be displayed as:

\texttt{Théorème 4.2 | (Inutile)} Un théorème en français.

By default, when referenced, the name of the theorem matches the current context. For example, the definition above will be displayed in French in the current French context : \texttt{Définition 4.1 et Théorème 4.2}. If you want the name of the theorem to always match the language of the context in which the theorem is located, you can add $\texttt{\textbackslash originalref}$ to the global options.

\section{Define a new theorem-like environment}

If you need to define a new theorem-like environment, you must first define the name of the environment in the language to use:

\begin{itemize}
  \item $\texttt{\textbackslash NameTheorem\{\langle language name\rangle\}\{\langle name of environment\rangle\}\{\langle name string\rangle\}}$
\end{itemize}

For $\langle language name\rangle$, please refer to the section on language configuration. When $\langle language name\rangle$ is not specified, the name will be set for all supported languages. In addition, environments with or without asterisk share the same name, therefore, $\texttt{\textbackslash NameTheorem\{envname\}\{\ldots\}}$ has the same effect as $\texttt{\textbackslash NameTheorem\{envname\}\{\ldots\}}$.

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

\begin{itemize}
  \item $\texttt{\textbackslash CreateTheorem\{*\{\langle name of environment\rangle\}}$
  \item $\texttt{\textbackslash CreateTheorem\{\langle name of environment\rangle\}}$
  \item $\texttt{\textbackslash CreateTheorem\{\langle name of environment\rangle\}\{\langle number like\rangle\}}$
  \item $\texttt{\textbackslash CreateTheorem\{\langle name of environment\rangle\}\{\langle number like\rangle\}$, which shares the counter $\langle number like\rangle$
\end{itemize}
• \CreateTheorem\{\textit{name of environment}\}\langle\textit{numbered within}\rangle

– Define a numbered environment \textit{name of environment}, numbered within the counter \textit{numbered within}

• \CreateTheorem\{\textit{name of environment}\}\langle\textit{existed environment}\rangle

\CreateTheorem*\{\textit{name of environment}\}\langle\textit{existed environment}\rangle

– Identify \textit{name of environment} with \textit{existed environment} or \textit{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 \texttt{thm} to write theorem.

2. To remove the numbering of some environments. For example, one can remove the numbering of the \texttt{remark} environment with \CreateTheorem\{remark\}\{remark*\}.

\textbf{TIP}

This macro utilizes the feature of \texttt{amsthm} internally, so the traditional \texttt{theoremstyle} is also applicable to it. One only needs declare the style before the relevant definitions.

Here is an example. The following code:

\NameTheorem[EN]{proofidea}{Idea}
\CreateTheorem*{proofidea*}
\CreateTheorem{proofidea}<subsection>

defines an unnumbered environment \texttt{proofidea*} and a numbered environment \texttt{proofidea} (numbered within \texttt{subsection}) respectively. They can be used in English context. The effect is as follows:

Idea | The \texttt{proofidea*} environment.

Idea 4.3.1 | The \texttt{proofidea} environment.

\textbf{4.4 Draft mark}

You can use \texttt{dnf} to mark the unfinished part. For example:

• \texttt{dnf} or \texttt{dnf...}. The effect is: \texttt{To be finished #1} or \texttt{To be finished #2: ...}.

The prompt text changes according to the current language. For example, it will be displayed as \texttt{Pas encore fini #3} in French mode.

Similarly, there is \texttt{needgraph}:

• \texttt{needgraph} or \texttt{needgraph...}. The effect is:

\texttt{A graph is needed here #1}

or

\texttt{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

\texttt{Il manque une image ici #3}
minimart has both the features of standard classes and that of the $\text{AMS}$ classes. Therefore, the title part can either be written in the usual way, in accordance with the standard class article:

\[
\text{maketitle} \begin{abstract} \langle \text{abstract} \rangle \end{abstract} \begin{keyword} \langle \text{keywords} \rangle \end{keyword}
\]

or written in the way of $\text{AMS}$ classes:

\[
\text{maketitle} \begin{abstract} \langle \text{abstract} \rangle \end{abstract} \begin{keyword} \langle \text{keywords} \rangle \end{keyword}
\]

The author information can contain multiple groups, written as:

\[
\text{maketitle} \begin{abstract} \langle \text{abstract} \rangle \end{abstract} \begin{keyword} \langle \text{keywords} \rangle \end{keyword}
\]

Among them, the mutual order of $\text{address}$, $\text{curraddr}$, $\text{email}$ is not important.
4.6 | Miscellaneous

4.6.1 On the line numbers
Line numbers can be turned on and off at any time. \texttt{linenumbers} is used to enable the line numbers, and \texttt{nolinenumbers} is used to disable them. For the sake of beauty, the title, table of contents, index and some other elements are not numbered.

4.6.2 On the footnotes in the title
In \texttt{section} or \texttt{subsection}, if you wish to add footnotes, you can only:
• first write \texttt{mbox{\protect\footnotemark}},
• then add \texttt{footnotetext{...}} afterwards.
This is a disadvantage brought about by the underline decoration of the title.

4.6.3 On the QED symbols
Since the font in the theorem-like environments is the same as that of the main text, in order to indicate where the environments end, a hollow QED symbol \texttt{□} is placed at the end of the theorem-like environments. However, if your theorem ends with an equation or list (itemize, enumerate, description, etc.), this symbol cannot be automatically placed in the correct position. In this case, you need to manually add a \texttt{\textbackslash qedhere} at the end of your equation or the last entry of your list to make the QED symbol appear at the end of the line.

Known issues
• The font settings are still not perfect.
• Since many features are based on the \texttt{Pro\textsc{f}L\textsc{i}b} toolkit, minimalist (and hence minimart, einfart and minimbook, simplivre) inherits all its problems. For details, please refer to the “Known Issues” section of the \texttt{Pro\textsc{f}L\textsc{i}b} documentation.
• The error handling mechanism is incomplete: there is no corresponding error prompt when some problems occur.
• There are still many things that can be optimized in the code.