Projlib

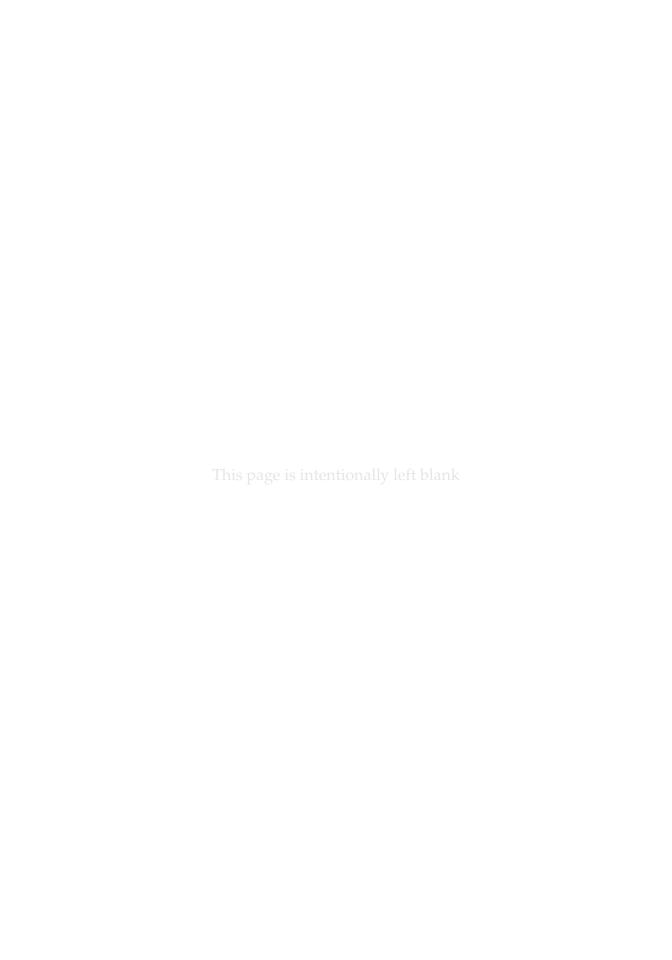
SIMPLIVRE

WRITE YOUR BOOKS IN
A SIMPLE AND CLEAR WAY

Corresponding to: simplivre 2022/07/31

JINWEN XU

July 2022, in Paris



Preface

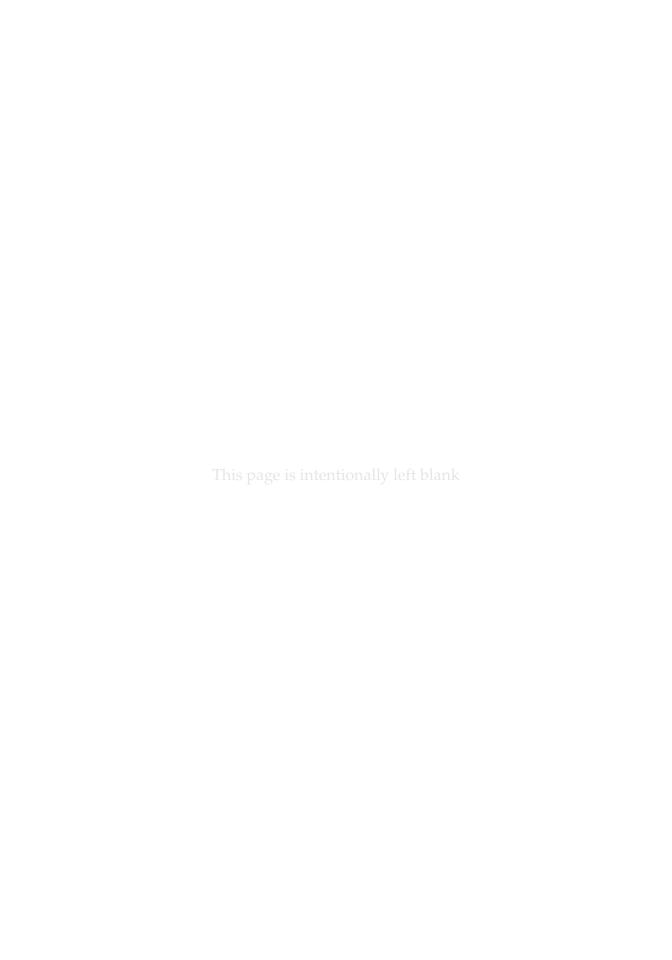
simplivre is part of the minimalist class series. Its name is taken from French words "simple" and "livre" (for "book"). The entire collection includes minimart and einfart for type-setting articles, and minimbook and simplivre for typesetting books. My original intention in designing them was to write drafts and notes that look simple yet not shabby.

simplivre has multi-language support, including Chinese (simplified and traditional), English, French, German, Italian, Japanese, Portuguese (European and Brazilian), Russian and Spanish. These languages can be switched seamlessly in a single document. Due to the usage of custom fonts, einfart requires XHATEX or LualATEX to compile.

This documentation is typeset using simplivre (with the option use boldface). You can think of it as a short introduction and demonstration.

TIP

Multi-language support, theorem-like environments, draft marks and some other features are provided by the <code>Projlib</code> toolkit. Here we only briefly discuss how to use it with this document class. For more detailed information, you can refer to the documentation of <code>Projlib</code>.



Contents

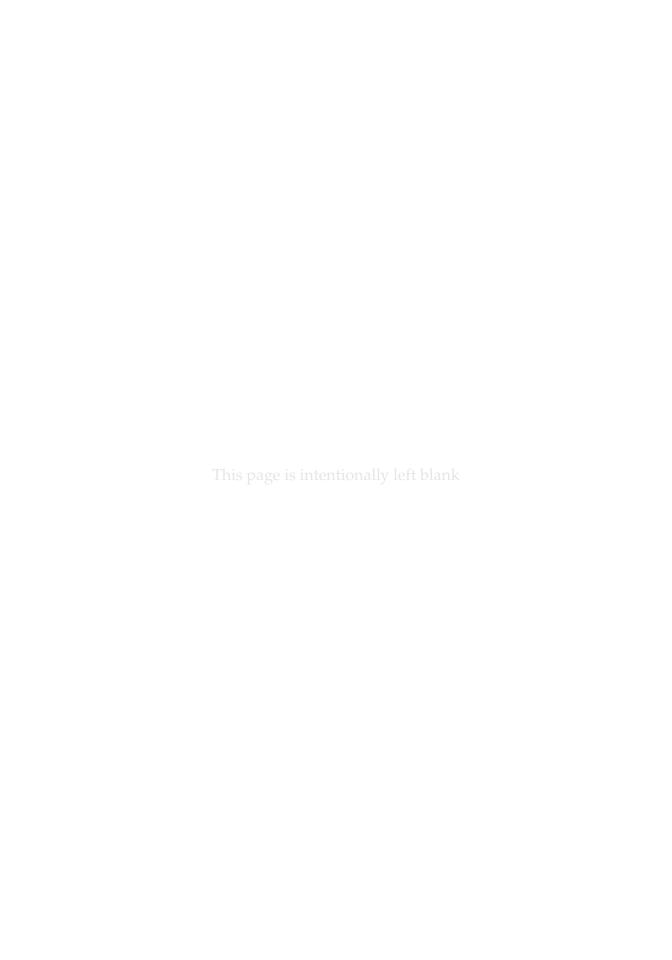
Pre	tace .														1
				I INSTRUCT	'ION										
				INSTRUCT	ION										
Bef	ore you	ı start													3
1	Usage and examples										5				
	1.1	How to	load it												5
	1.2	Example - A complete document								5					
		1.2.1	Initialization												6
		1.2.2	Set the langua	age											6
		1.2.3	Draft marks												6
		1.2.4	Theorem-like	environmen	ts										7
2	On th	e default	fonts												9
3	The o	ptions													11
4	Instructions by topic									13					
	4.1		ge configuratio												13
	4.2		ns and how to												14
	4.3		Oraft mark					14							
	4.4	Miscella	Miscellaneous			14									
		4.4.1	On the line n	umbers											15
		4.4.2	On the footno	otes in the titl	е										15
5	Know	n issues													17
			1	II DEMONSTR.	ATIOI	V									
			j	221110110110	11101	•									
6	Head 6.1		evel 0 (chapter) g on Level 1 (se Heading on I	ection)											21

6.2	Lists	
	6.2.1	Example for list (itemize)
	6.2.2	Example for list (enumerate)
	6.2.3	Example for list (description)

PΔ	R	ГΠ	

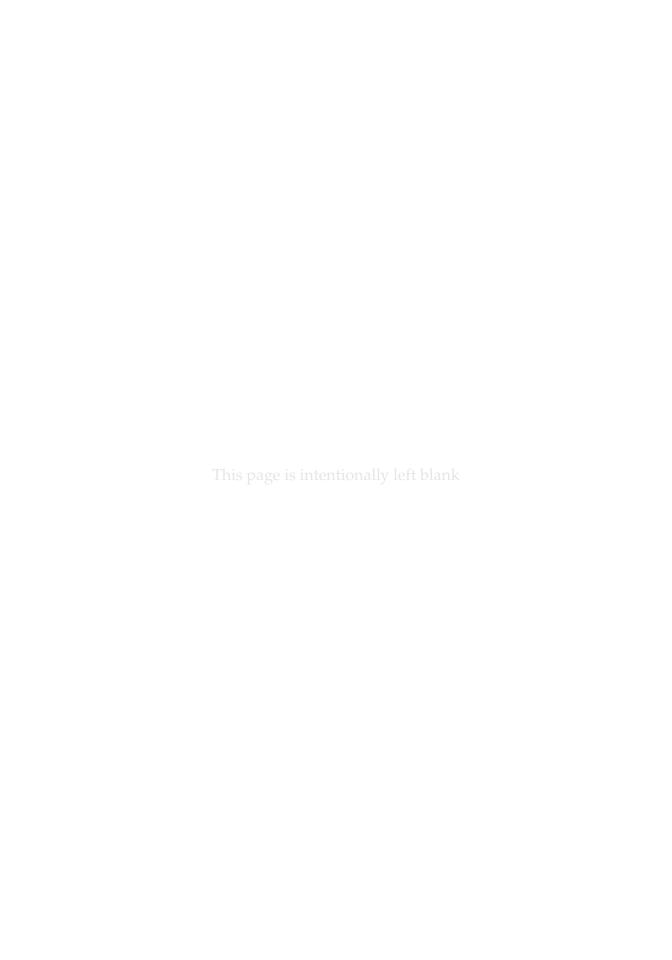
INSTRUCTION

You can add some introductory text here via \parttext\(\lambda text\).



Before you start

- In order to use the package or classes described here, you need to:
- install TeX Live or MikTeX of the latest possible version, and make sure that minimalist and projlib are correctly installed in your TeX system.
- download and install the required fonts, see the section "On the default fonts".
- be familiar with the basic usage of LATEX, and know how to compile your document with pdfLATEX, XALATEX or LuaLATEX.



Usage and examples

1.1 How to load it

One only needs to put

```
\documentclass{simplivre}
```

as the first line to use the simplivre class.

ATTENTION

You need to use either XHETEX or LualATEX engine to compile.

1.2 Example - A complete document

Let's first look at a complete document.

```
\documentclass{simplivre}
2 \usepackage{ProjLib}
  \UseLanguage{French}
6 \begin{document}
  \frontmatter
  \begin{titlepage}
      ⟨code for titlepage⟩
11
  \end{titlepage}
  \tableofcontents
14
15
  \mainmatter
16
17
  \part{\(part \title\)}
  \parttext{\langle text after part title\}}
19
21 \chapter{\chapter title\}
```

```
22
  \section{\langle section title \rangle}
24
   \DNF < (some hint)>
25
26
   \begin{theorem}\label{thm:abc}
27
       Ceci est un théorème.
28
   \end{theorem}
  Référence du théorème: \cref{thm:abc}
31
  \backmatter
33
34
35
  \end{document}
```

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

1.2.1 Initialization

```
\documentclass{simplivre} \usepackage{ProjLib}
```

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

1.2.2 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.

1.2.3 Draft marks

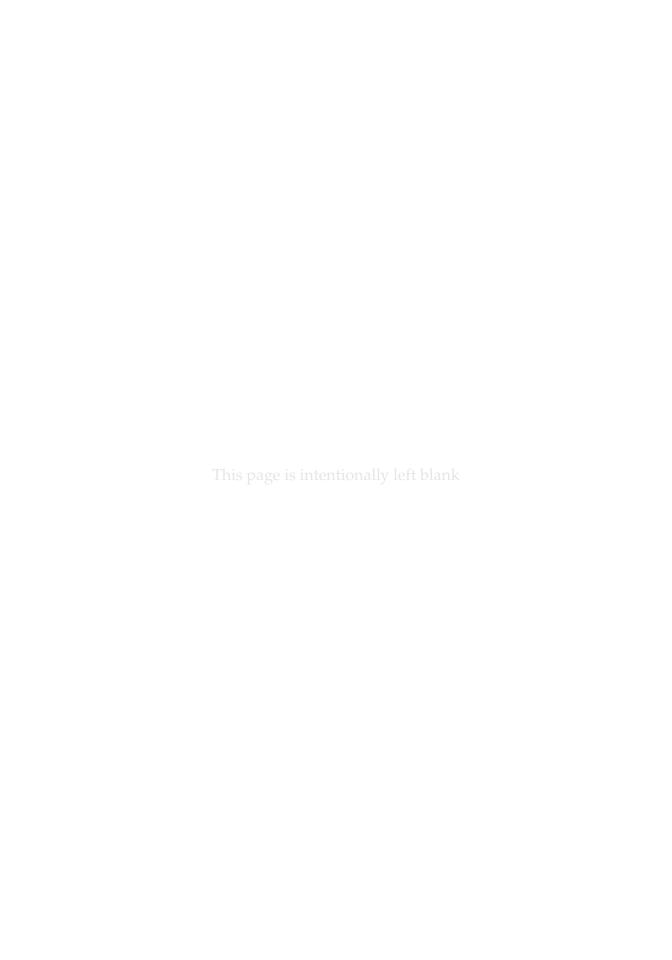
$\DNF < (some hint)>$

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.

1.2.4 Theorem-like environments

```
\begin{theorem}\label{thm:abc}
   Ceci est un théorème.
\end{theorem}
Référence du théorème: \cref{thm:abc}
```

Commonly used theorem-like environments have been pre-defined. Also, when refer-there is no need to explicitly write down the name of the corresponding environment every time.



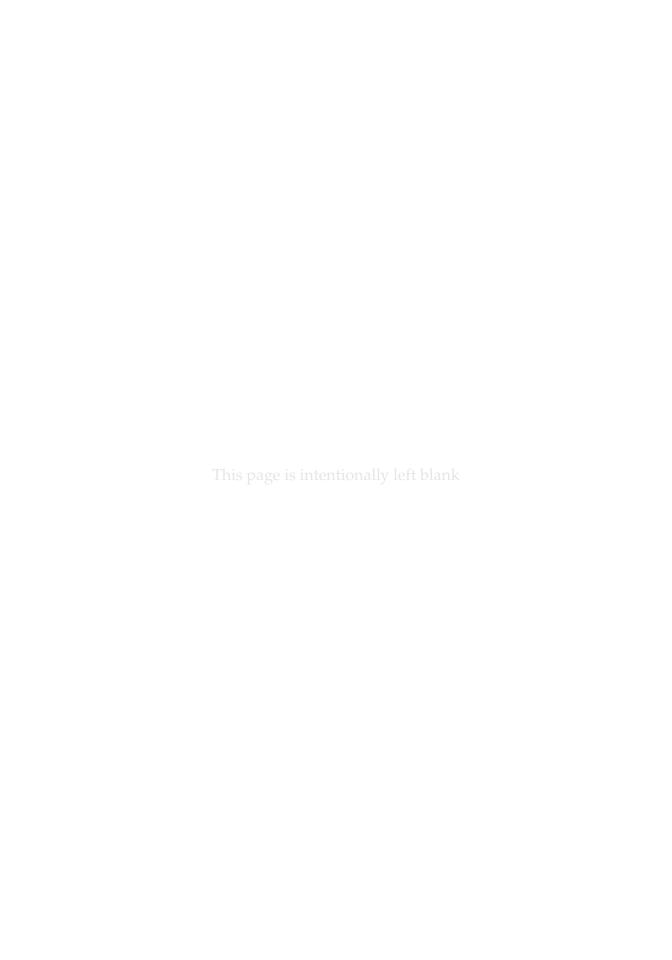
On the default fonts

By default, this document class uses Palatino Linotype as the English main font; Source Han Serif, Source Han Sans and Source Han Mono as the Chinese main font, sans serif font and typewriter font; and partially uses Neo Euler as the math font. You need to download and install these fonts by yourself. The Source Han font series can be downloaded at https://github.com/adobe-fonts (It is recommended to download the Super-OTC version, so that the download size is smaller). Neo Euler can be downloaded at https://github.com/khaledhosny/euler-otf. When the corresponding font is not installed, the font that comes with TeX Live will be used instead, and the effect may be discounted.

In addition, Source Code Pro is used as the English sans serif font, New Computer Modern Mono as the English monospace font, as well as some symbols in the mathematical fonts of Asana Math, Tex Gyre Pagella Math, and Latin Modern Math. These fonts are already available in TeX Live or MikTeX, which means you don't need to install them yourself.

- English main font. English sans serif font. English typewriter font.
- 简体中文主要字体,简体中文无衬线字体,简体中文等宽字体
- 繁體中文主要字體,繁體中文無襯線字體,繁體中文等寬字體
- 日本語のメインフォント、日本語のサンセリフフォント、日本語の等幅フォント
- Math demonstration: α , β , γ , δ , 1, 2, 3, 4, a, b, c, d,

$$\operatorname{li}(x) := \int_{2}^{\infty} \frac{1}{\log t} \, \mathrm{d}t$$



The options

simplivre 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.

a4paper or b5paper

- Paper size options. The default paper size is $7 \text{in} \times 10 \text{in}$.

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.

use boldface

 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.

useindent

Use paragraph indentation instead of inter-paragraph spacing.

25 runin

Use the "runin" style for \subsubsection

nothms

- Pure text mode. Do not load theorem-like environments.
- nothmnum, thmnum or thmnum=\(\langle counter \rangle \)
 - Theorem-like environments will not be numbered / numbered in order 1, 2, 3... / numbered within (counter). 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.

In addition, the commonly used oneside and twoside options are also available. Twopage layout is used by default.

Instructions by topic

4.1 Language configuration

simplivre has multi-language support, including Chinese (simplified and traditional), English, French, German, Italian, Japanese, Portuguese (European and Brazilian), Russian and Spanish. The language can be selected by the following macros:

- \UseLanguage {\language name \rangle} 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\} {\language name\}, which uses the specified language settings to typeset \(\language\). Compared with \UseLanguage, it will not modify the line spacing, so line spacing would remain stable when CJK and Western texts are mixed.

 $\langle language \ name \rangle$ 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{\(\settings\\)\}
 - Add ⟨settings⟩ to all supported languages.
- \AddLanguageSetting(\language name\) \{\language 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, 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 \cref{abel} . In this way, there is no need to explicitly write down the name of the corresponding environment every time.

EXAMPLE

```
\begin{definition}[Strange things] \label{def: strange} ...
```

will produce

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.

\cref{def: strange} will be displayed as: Definition 4.1. After using \UseLanguage{French}, a theorem will be displayed as:

Тне́опѐме 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: la Définition 4.1 et le 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 originalref to the global options.

TIP

Please refer to the documentation of the package create-theorem to see how to declare new theorem-like environments.

4.3 Draft mark

You can use \DNF to mark the unfinished part. For example:

• \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.

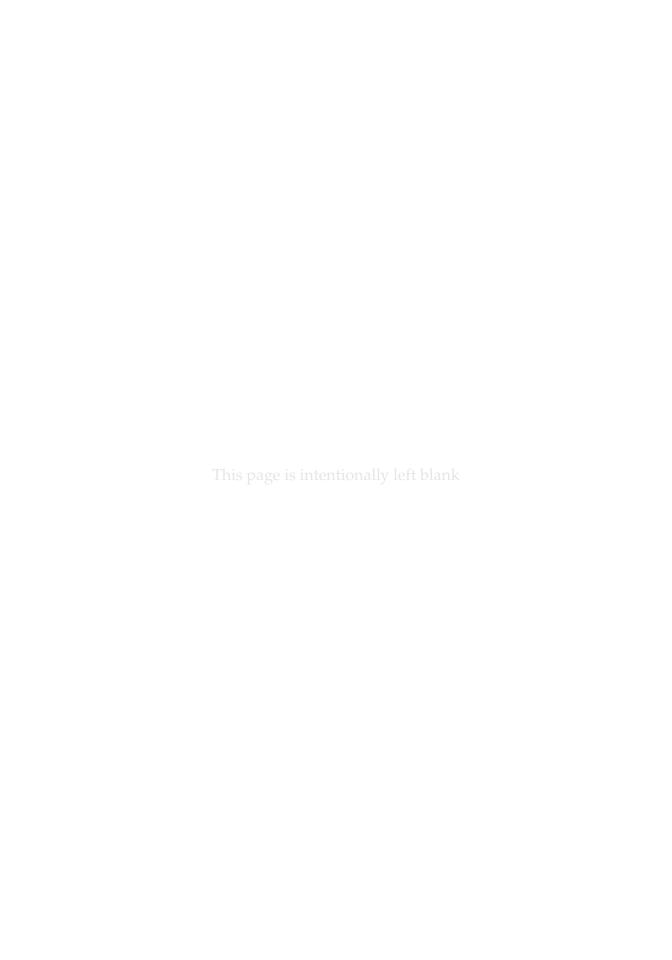
4.4 Miscellaneous

4.4.1 On the line numbers

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

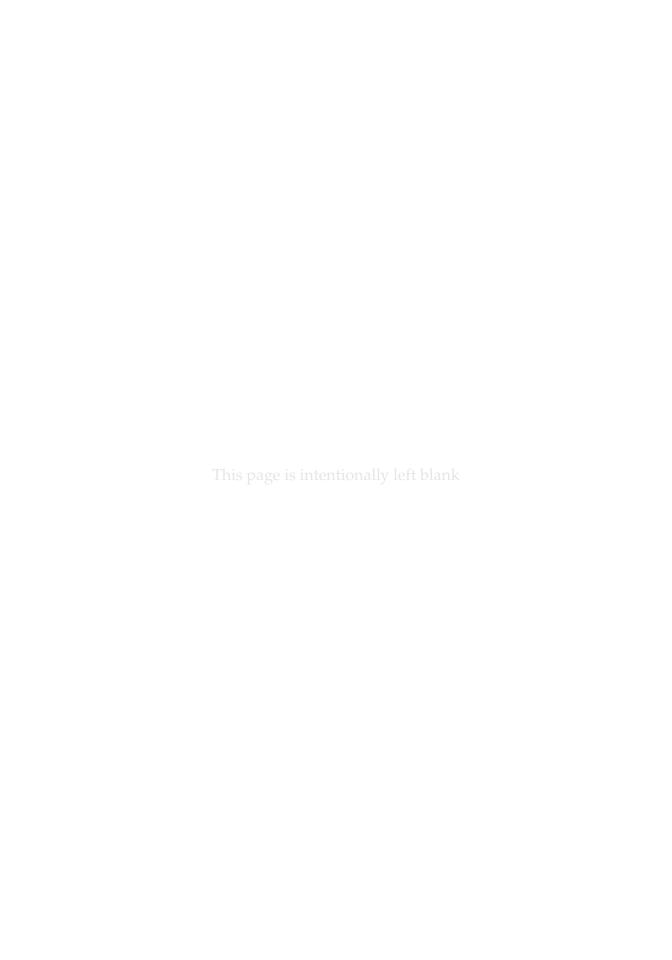
4.4.2 On the footnotes in the title

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



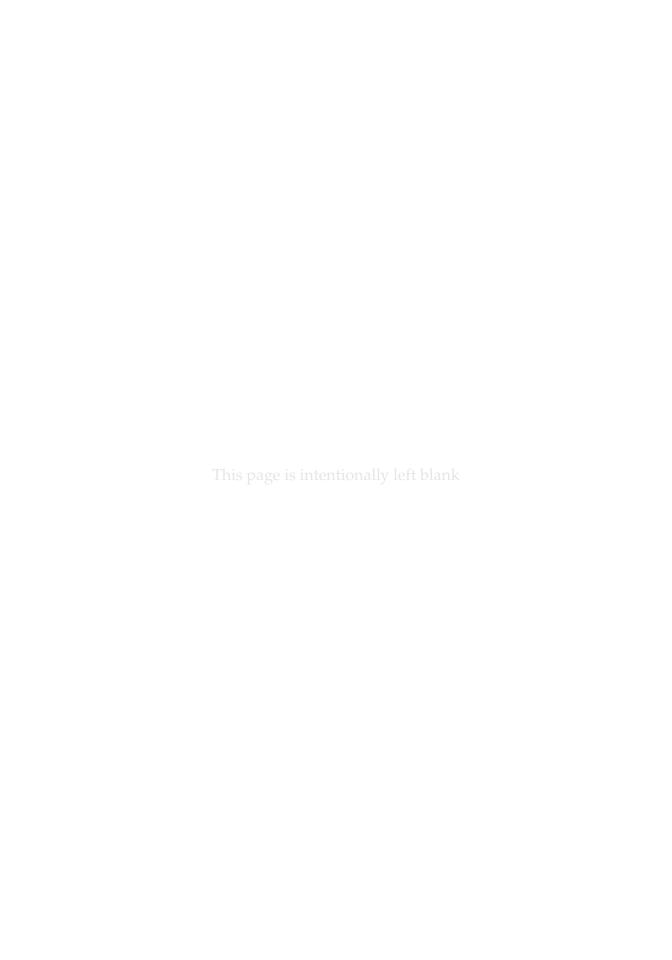
Known issues

- The font settings are still not perfect.
- Since many features are based on the <code>Projlib</code> toolkit, minimalist (and hence minimart, einfart and minimbook, simplivre) inherits all its problems. For details, please refer to the "Known Issues" section of the <code>Projlib</code> 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.



PART II

DEMONSTRATION



Heading on Level 0 (chapter)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.1 Heading on Level 1 (section)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.1.1 Heading on Level 2 (subsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.1.1.1 Heading on Level 3 (subsubsection)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Heading on Level 4 (paragraph) Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6.2 Lists

6.2.1 Example for list (itemize)

- First item in a list
- Second item in a list
 - Third item in a list
- Fourth item in a list
- Fifth item in a list

6.2.1.1 Example for list (4*itemize)

- First item in a list
 - First item in a list
 - First item in a list
 - * First item in a list
 - * Second item in a list
 - Second item in a list
 - Second item in a list
 - Second item in a list

6.2.2 Example for list (enumerate)

- 1) First item in a list
- 2) Second item in a list
- 24 3) Third item in a list
 - 4) Fourth item in a list
 - 5) Fifth item in a list

6.2.2.1 Example for list (4*enumerate)

- 1) First item in a list
 - i) First item in a list
 - a) First item in a list
 - A. First item in a list
 - B. Second item in a list
 - b) Second item in a list
 - ii) Second item in a list
- 2) Second item in a list

6.2.3 Example for list (description)

- First item in a list

 Second item in a list
- Third item in a list
- 4 Fourth item in a list
 - Fifth item in a list

6.2.3.1 Example for list (4*description)

First item in a list

Second item in a list

- Below is a list of outstanding items, followed by the corresponding page numbers.
- **#1 (14)**
- **#2 (14):...**
- #3 $\langle 14 \rangle$