novathesis

The Manual

A LaTeX template for theses and dissertations

João M. Lourenço
NOVA UNIVERSITY LISBON · LPPL 1.3C

AUGUST 2026

v8.5.0

novathesis



The Manual

A LaTeX template for theses and dissertations

João M. Lourenço
NOVA UNIVERSITY LISBON · LPPL 1.3C

AUGUST 2026

v8.5.0

A Very Long and Impressive Thesis Title with a Forced Line Break

Copyright © John Very Longname Doe, *School*, *University*.
The *School* and the *University* have the right, perpetual and without geographical limitations, to archive and publish this dissertation in print or digital form, or by any other existing or future means, and to disseminate it through scientific repositories and to allow its copying and distribution for non-commercial educational or research purposes, provided that proper credit is given to the author and the publisher.

Dedicatory lorem ipsum.

Acknowledgements

Acknowledgments are personal text and should be a free expression of the author.

However, without any intention of conditioning the form or content of this text, I would like to add that it usually starts with academic thanks (instructors, etc.); then institutional thanks (Research Center, Department, Faculty, University, FCT / MEC scholarships, etc.) and, finally, the personal ones (friends, family, etc.).

But I insist that there are no fixed rules for this text, and it must, above all, express what the author feels.

”

“You cannot teach a man anything; you can only help him discover it in himself.”

— Galileo, Somewhere in a book or speach
(Astronomer, physicist and engineer)

Abstract

Regardless of the language in which the dissertation is written, usually there are at least two abstracts: one abstract in the same language as the main text, and another abstract in some other language.

The abstracts' order varies with the school. If your school has specific regulations concerning the abstracts' order, the The **nova**thesis LaTeXtemplate (**nova**thesis) (LaTeX) template will respect them. Otherwise, the default rule in the **nova**thesis template is to have in first place the abstract in *the same language as main text*, and then the abstract in *the other language*. For example, if the dissertation is written in Portuguese, the abstracts' order will be first Portuguese and then English, followed by the main text in Portuguese. If the dissertation is written in English, the abstracts' order will be first English and then Portuguese, followed by the main text in English. However, this order can be customized by adding one of the following to the file `5_packages.tex`.

```
\ntsetup{abstractorder={<LANG_1>,...,<LANG_N>}}
\ntsetup{abstractorder={<MAIN_LANG>={<LANG_1>,...,<LANG_N>}}}
```

For example, for a main document written in German with abstracts written in German, English and Italian (by this order) use:

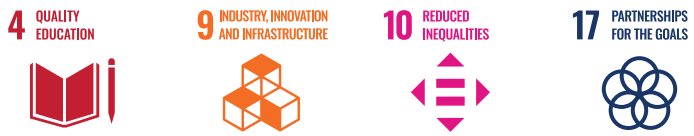
```
\ntsetup{abstractorder={de={de,en,it}}}
```

Concerning its contents, the abstracts should not exceed one page and may answer the following questions (it is essential to adapt to the usual practices of your scientific area):

1. What is the problem?
2. Why is this problem interesting/challenging?
3. What is the proposed approach/solution/contribution?
4. What results (implications/consequences) from the solution?

Keywords: One keyword · Another keyword · Yet another keyword · One keyword more · The last keyword

Sustainable Development Goals (SDG):



This work contributes to SDG 9 (Industry, Innovation, and Infrastructure) by advancing resilient and secure cyber-infrastructure through lightweight, decentralized edge-inference techniques. Furthermore, it supports SDG 16 (Peace, Justice, and Strong Institutions) by promoting privacy-preserving endpoint security, and SDG 12 (Responsible Consumption and Production) through energy- and resource-efficient software design.

Resumo

Independentemente da língua em que a dissertação está escrita, geralmente esta contém pelo menos dois resumos: um resumo na mesma língua do texto principal e outro resumo numa outra língua.

A ordem dos resumos varia de acordo com a escola. Se a sua escola tiver regulamentos específicos sobre a ordem dos resumos, o template (LaTeX) **nova**thesis irá respeitá-los. Caso contrário, a regra padrão no template **nova**thesis é ter em primeiro lugar o resumo *no mesmo idioma do texto principal* e depois o resumo *no outro idioma*. Por exemplo, se a dissertação for escrita em português, a ordem dos resumos será primeiro o português e depois o inglês, seguido do texto principal em português. Se a dissertação for escrita em inglês, a ordem dos resumos será primeiro em inglês e depois em português, seguida do texto principal em inglês. No entanto, esse pedido pode ser personalizado adicionando um dos seguintes ao arquivo `5_packages.tex`.

```
\abstractorder(<MAIN_LANG>):={<LANG_1>,...,<LANG_N>}
```

Por exemplo, para um documento escrito em Alemão com resumos em Alemão, Inglês e Italiano (por esta ordem), pode usar-se:

```
\ntsetup{abstractorder={de={de,en,it}}}
```

Relativamente ao seu conteúdo, os resumos não devem ultrapassar uma página e frequentemente tentam responder às seguintes questões (é imprescindível a adaptação às práticas habituais da sua área científica):

1. Qual é o problema?
2. Porque é que é um problema interessante/desafiante?
3. Qual é a proposta de abordagem/solução?
4. Quais são as consequências/resultados da solução proposta?

Palavras-chave: Primeira palavra-chave · Outra palavra-chave · Mais uma palavra-chave · A última palavra-chave

Objetivos de Desenvolvimento Sustentável (ODS):



Este trabalho contribui para o ODS 9 (Indústria, Inovação e Infraestrutura) ao promover uma ciberinfraestrutura resiliente e segura por meio de técnicas leves e descentralizadas de inferência na borda (edge-inference). Além disso, apoia o ODS 16 (Paz, Justiça e Instituições Eficazes) ao incentivar a segurança de endpoints com preservação da privacidade, e o ODS 12 (Consumo e Produção Responsáveis) através de um design de software eficiente em termos de energia e recursos.

Disclosure of Usage of Generative Artificial Intelligence

The author, John Very Longname Doe, acknowledges the use of the following generative artificial intelligence tools in the development of this thesis. These technologies were utilized under strict supervision, and all AI-generated outputs were critically reviewed and verified for accuracy. The author accepts full responsibility for the validity and originality of the content, confirming that the use of these tools aligns with all institutional policies and standards of academic integrity.[1]

Generative AI tools used: ChatGPT-5.2, Gemini 3 Pro, Claude 4.5 Opus, and Perplexity 12.12.25.

Activities supported by Generative Artificial Intelligence Tools:[2]

Conceptualization

- ☐ Idea generation
- ☑ **Defining the research objective**
- ☐ Formulating research questions and hypotheses
- ☑ **Feasibility assessment and risk evaluation**
- ☐ Hypothesis viability assessment
- ☐ Simulated debate and argument testing

Literature Review

- ☐ Search and Discovery
- ☐ Literature summarization and synthesis
- ☑ **Concept Mapping and Systematization**
- ☐ Market/patent landscape analysis
- ☐ Gap Identification and Novelty Evaluation
- ☐ Cross-lingual literature comprehension

Methodology

- ☐ Experimental Design Optimization
- ☐ Development of experimental or research protocols
- ☑ **Methodological Instrument Selection**

[1] This declaration was generated using XeLaTeX [9], the LaTeX package `aidisclose` and its companion website `aidisclose.org` [7].

[2] This list extends the GAIDeT taxonomy [8].

Software Development and Automation

- ☐ Code Generation
- ☑ **Debugging and Repair**
- ☐ Algorithm design
- ☐ Refactoring and Optimization
- ☑ **Process automation**
- ☐ Code documentation and comment generation

Data Management

- ☐ Data collection
- ☐ Data analysis
- ☐ Synthetic data generation
- ☐ Validation
- ☑ **Quantitative Plotting and Charting**
- ☐ De-identification and anonymization support
- ☐ Data cleaning
- ☐ Reproducibility and rerun checks
- ☑ **Audio-to-text transcription and diarization**
- ☑ **Data curation and organization**
- ☐ Data labeling and annotation assistance

Visuals and Multimedia

- ☐ Synthetic Asset Generation
- ☑ **Image Enhancement and Editing**
- ☐ Diagrammatic and Schematic Design

Writing and Editing

- ☐ Drafting Text
- ☐ Formulation of conclusions
- ☐ Citation formatting and bibliography management
- ☐ Polishing and Editing
- ☐ Tone Adjustment
- ☑ **Press releases and outreach materials**
- ☐ Abstract and Executive Summary Generation
- ☐ Translation
- ☐ Title Generation

Ethics Review

- ☑ **Bias analysis and discrimination assessment**
- ☐ Ethical risk analysis
- ☐ Data confidentiality monitoring
- ☐ Monitoring compliance with ethical standards

Quality Assurance

- ☑ **Simulated Peer Review**
- ☐ Identification of limitations
- ☐ Publication support
- ☐ Consistency checking against field trends
- ☑ **Publication strategy and journal selection**

Additional comment #1: The AI was used primarily for refining the code in Section 3, but the algorithm logic was derived manually.

Additional comment #2: This is a second additional comment!

Additional comment: This is a third additional comment! It is starred and, thus, does not get a number. Also, this comment is longer and, hopefully, will span to the next line!

Additional comment #3: This is the third non-starred additional comment!

Contents

Abstract vi

Resumo viii

Disclosure of Usage of Generative Artificial Intelligence x

Contents xii

List of Figures xvii

List of Tables xviii

Glossary xxi

Acronyms xxii

Symbols xxiii

1 Introduction 1

1.1 The *novathesis* Template . . . 1

1.1.1 Your Time is Precious . . . 1

1.1.2 Recognition . . . 2

1.2 Getting Started . . . 2

1.3 Your Thesis Roadmap . . . 2

1.4 Getting Help . . . 3

1.4.1 Suggestions, Bugs and Feature Requests . . . 4

1.5 Donors . . . 4

1.6 Disclaimer . . . 4

2 novathesis Template *User's Manual* 5

2.1 Introduction . . . 5

2.2 Getting Started . . . 5

2.2.1 Local LaTeX Installation . . . 5

2.2.2 Downloading the Template . . . 6

2.2.3 Overleaf . . . 6

2.3 Project Structure . . . 7

2.3.1 The `0-Config/` Files . . . 7

2.4 Configuring the Document (`1_novathesis.tex`) . . . 7

2.4.1 Essential Options . . . 7
2.4.2 Appearance Options . . . 9
2.4.3 Print Control Options . . . 10
2.4.4 Language Options . . . 12
2.4.5 Sustainable Development Goals (SDGs) . . . 12
2.4.6 Indexing . . . 16
2.4.7 Glossary and Miscellaneous Options . . . 16
2.5 Cover Metadata (`3_cover.tex`) . . . 17
2.5.1 Title and Subtitle . . . 17
2.5.2 Department, Degree, and Specialization . . . 18
2.5.3 Author . . . 18
2.5.4 Advisers . . . 18
2.5.5 Evaluation Committee . . . 19
2.5.6 Dates and Sponsors . . . 19
2.6 Book Spine . . . 19
2.6.1 Enabling and Sizing . . . 19
2.6.2 Spine Title . . . 20
2.6.3 Customising the Spine with `\SpineSetup` . . . 20
2.6.4 Example . . . 22
2.7 Bibliography (`2_biblatex.tex`) . . . 22
2.7.1 General Behavioral Options . . . 23
2.7.2 Choosing a Citation Style . . . 23
2.7.3 Webography (Separate Online Bibliography) . . . 23
2.8 Adding Content (`4_files.tex`) . . . 23
2.8.1 Bibliography Files . . . 23
2.8.2 Front Matter . . . 24
2.8.3 Abstracts . . . 24
2.8.4 Glossaries, Acronyms, and Symbols . . . 24
2.8.5 Main Chapters . . . 24
2.8.6 Appendices and Annexes . . . 24
2.8.7 Custom Covers . . . 25
2.8.8 Bundled Institutional Images . . . 25
2.9 Custom Packages and Commands (`5_packages.tex`) . . . 25
2.10 Document Status Workflow . . . 26
2.11 Building the Document . . . 26
2.11.1 Using `make` . . . 26
2.11.2 Using `latexmk` Directly . . . 28
2.11.3 Building for Another School . . . 28
2.11.4 Overleaf . . . 28
2.11.5 Portable Document Format/Archive (PDF/A) Archival Output 28

3 Artificial Intelligence Disclosure Statement 30
3.1 Why Disclose AI Use? . . . 30

3.2 Configuration . . . 30
3.3 The Standard aidisclose Checklist . . . 31
3.3.1 Using the Companion Website (Recommended) . . . 31
3.3.2 Manual Configuration . . . 31
3.4 What if No AI was Used? . . . 31
3.5 Placement in the Document . . . 31
3.6 Detailed Taxonomy Descriptions . . . 32

4 A Short LaTeX Tutorial with Examples 33
4.1 Document Structure . . . 33
4.1.1 Adding a New Chapter . . . 33
4.1.2 Cross-references . . . 34
4.2 The `cleveref` Package . . . 34
4.2.1 Basic Usage . . . 35
4.2.2 Teaching `cleveref` about a New Object Type . . . 35
4.3 Text Formatting . . . 36
4.3.1 Basic Emphasis . . . 36
4.3.2 Lists . . . 36
4.3.3 Footnotes . . . 37
4.4 Equations . . . 37
4.5 Tables . . . 38
4.6 Figures . . . 39
4.6.1 Sub-figures . . . 40
4.7 Bibliography and Citations . . . 41
4.7.1 Citation Commands . . . 41
4.7.2 Compilation with Biber . . . 41
4.8 Acronyms and Glossaries . . . 41
4.9 Indexing . . . 43
4.10 Source Code Listings . . . 43
4.11 To-do Notes . . . 45
4.12 Generating the Final PDF . . . 45
4.12.1 Algorithms . . . 45

Bibliography 46

Appendix A: novathesis Macro Reference 48
A.1 Template Bootstrap . . . 48
A.2 Document Identity . . . 49
A.3 Institutional Metadata . . . 50
A.4 People — Advisers and Committee . . . 51
A.5 File Registration — `\ntaddfile` . . . 53
A.6 Content Rendering . . . 54
A.7 Cover Customisation . . . 55

A.8 Document Conditionals . . . 55
A.9 Hook System . . . 56
A.10 Miscellaneous Utilities . . . 57
A.11 Summary Table . . . 58

Appendix B: The novathesis Cover Page System **60**
B.1 Architecture Overview . . . 60
B.2 Cover Identifiers . . . 61
B.3 Setting Up the Cover Canvas . . . 61
B.3.1 Cover-Specific Margins . . . 61
B.3.2 Background Colour and Image . . . 61
B.4 The `\ntaddtocover` Command . . . 62
B.4.1 Attribute Reference . . . 63
B.4.2 Stacking Elements Vertically . . . 65
B.4.3 Embedding TikZ Code via `tikz=true` . . . 66
B.5 The `\ntAddToCoverTikz` Command . . . 67
B.6 Cover Hooks . . . 68
B.7 Debugging a Cover Layout . . . 69
B.8 Cover Rendering: Execution Order Summary . . . 70
B.9 Step-by-Step: Adding Cover Support for a New School . . . 71
B.10 Real-World Examples . . . 74
B.10.1 NOVA/FCT: Mixed LaTeX and TikZ . . . 74
B.10.2 ULISBOA/IST: Pure LaTeX Box Stacking . . . 74
B.11 Quick Reference . . . 75

Appendix C: The Build System **76**
C.1 Everyday Building . . . 76
C.2 Command-Line Overrides . . . 77
C.3 Faster pdfLaTeX and XeLaTeX Builds . . . 78
C.4 Building Without `make` . . . 78
C.5 Maintainer Tools . . . 78
C.5.1 Building One School Variant . . . 79
C.5.2 Building the Full Matrix . . . 79
C.5.3 Other Maintainer Targets . . . 81

Appendix D: Migrating from 7.10.x **82**
D.1 What Changed, and Why . . . 82
D.2 The Migration, Step by Step . . . 82
D.3 If Something Looks Wrong . . . 84

Appendix E: Detailed AI Taxonomy Descriptions **85**

Annex I: Lorem Ipsum Annex **90**
I.1 A section A . . . 90

I.2 A section B 90
I.2.1 A section B.1 91
I.2.2 A section B.2 91

List of Figures

1.1 The novathesis project web page in GitHub. 3

3.1 The aidisclose.org web site 32

4.1 Bitmap image (PNG/JPG): quality degrades at larger sizes. 39
4.2 Vector image (PDF): scales perfectly at any size. 40
4.3 Two sub-figures in one float. 40

List of Tables

2.1 Project structure and access guidelines. 7
2.2 Configuration files in `0-Config/`. 7
2.3 The 17 UN Sustainable Development Goals (SDGs), in the `normal`, `inverted` and `mono` icon variants. 14
2.4 Institutional logo images bundled in `novathesisFiles/Images/`. 25
2.5 Effect of `docstatus` on the generated document. 26

4.1 Core `cleveref` commands. 35
4.2 Common text formatting commands. 36
4.3 A simple table. 38
4.4 A multi-page table using `xltabular` with row colors. 38
4.5 Common `biblatex` citation commands. 41
4.6 Pre-configured `minted` environments. 43

List of Algorithms

1 Example algorithm using `algorithm2e`. 45

List of Listings

4.1 *Hello, World!* in Python. 44
4.2 *Hello, World!* in C. 44
4.3 *Hello, World!* in LaTeX. 44

Glossary

bibliography A systematic list of the books of a specific author or publisher, or on a specific subject, cited in the document. (pp. *1*, *42*)

framework A basic structure underlying a system, concept, or text, providing a foundation for development. (pp. *1*, *42*)

LaTeX A document preparation system based on TeX, widely used for the communication and publication of scientific documents in many fields, including mathematics, physics, and computer science. (pp. *1*, *42*)

Acronyms

AI Artificial Intelligence (pp. *30*, *31*, *42*, *85–87*, *89*)

API Application Programming Interface (p. *60*)

CI Continuous Integration (p. *77*)

DI Department of Computer Science (p. *1*)

DOI Digital Object Identifier (p. *42*)

FCT NOVA School of Science and Technology (pp. *1*, *6*, *74*)

JRE Java Runtime Environment (p. *82*)

MSc Master of Science (pp. *26*, *64*, *66*, *74*)

NOVA NOVA University Lisbon (p. *1*)

novathesis The **novathesis** LaTeXtemplate (pp. *vi*, *viii*, *xvii*, *1–4*)

PDF Portable Document Format (pp. *2*, *6*, *26*, *27*, *39*, *42*, *45*, *49*, *57*, *59*, *70*, *76*, *77*, *79*, *80*, *83*)

PDF/A Portable Document Format/Archive (pp. *xiii*, *8*, *26–29*)

PhD Doctor of Philosophy (pp. *6*, *64*, *66*, *74*)

STEM Science, Technology, Engineering, and Mathematics (pp. *2*, *42*)

URL Uniform Resource Locator (p. *42*)

Symbols

α the significance level (pp. 1, 43)

$\hbar$ the reduced Planck constant (pp. 1, 43)

ω the angular frequency (pp. 1, 43)

$\mathbb{R}$ the set of real numbers (pp. 1, 43)

$\sum$ the sum of a sequence (pp. 1, 43)

1 Introduction



This is the novathesis LaTeX template Version 8.5.0 from 2026-10-10.

This work is licensed under the LaTeX Project Public License v1.3c. To view a copy of this license, visit the LaTeX project public license.

This first Chapter introduces the novathesis template and how it is organized. In Chapter 2 you can find some specific instructions on how to use this template. Chapter 4 shows some examples and give some hints on how to write your text. Please read these next Chapters carefully.

1.1 The *novathesis* Template

The novathesis template was born at the Department of Computer Science (DI) of NOVA School of Science and Technology (FCT) of NOVA University Lisbon (NOVA), Portugal. But the user base grew… initially grew to other Departments of NOVA FCT, then to other Schools of NOVA, and later to other Schools of other Universities. Currently more than 25 Schools are natively supported by the novathesis template. The full list of supported Schools can be found at the Project's GitHub page.

1.1.1 Your Time is Precious

Did you learn how to drive by sitting by the wheel and throwing your car into the road? Most probably you did take your time *learning the rules* and *practicing* first! Likewise, it is not wise to throw yourself at the task of writing a thesis/dissertation in LaTeX without seriously considering the following recommendation!

> If you are going to spend zillions of hours writing your thesis/dissertation using the novathesis LaTeX template (or some other LaTeX template), be wise and spend a couple of hours learning how to use it properly by reading this manual. And then, be even wiser, and spend a few more hours learning some LaTeX. I am sure that the time you are investing now will pay itself countless times before you submit your thesis/dissertation.
>
> — *João Lourenço*

1.1.2 Recognition

The novathesis template was born in 1996, and what you see now accumulates to many many hundreds (thousands?!) of working hours, unpaid and stolen from family and friends. This work is available to the community under the LaTeX Project Public License v1.3c, which means you are entitled to use it for free and change it at your will. However, if you decide to use this template to write your thesis/dissertation, **be fair to the developers** and:

1. Cite the novathesis manual [6] in a place of your choice (e.g., in the *Acknowledgments*) of your thesis/dissertation with "`\cite{novathesis-manual}`" . If you cite it this way, the correct entry will be added automatically to your bibliography (no need to worry with the necessary BibTeX entry, as it will be added automatically);

2. Go to the project web page in GitHub and give the project a star (marked with a red ellipse at the top-right in Figure 1.1); and

3. Make a donation by visiting the novathesis project page and clicking in the button marked with a green ellipse at the top-center in Figure 1.1). Alternatively, just click HERE and your browser will be directed to the right page.

1.2 Getting Started

The novathesis template provides an *easy to use* environment for writing your thesis. Detailed instructions on how to set up the template using Overleaf or a local LaTeX installation are provided in the *User's Manual* (see section 2.2). This is particularly useful for students in Science, Technology, Engineering, and Mathematics (STEM) fields who need to produce high-quality Portable Document Format (PDF) documents.

1.3 Your Thesis Roadmap

Writing a thesis is a journey. Here is the recommended workflow for using this template:

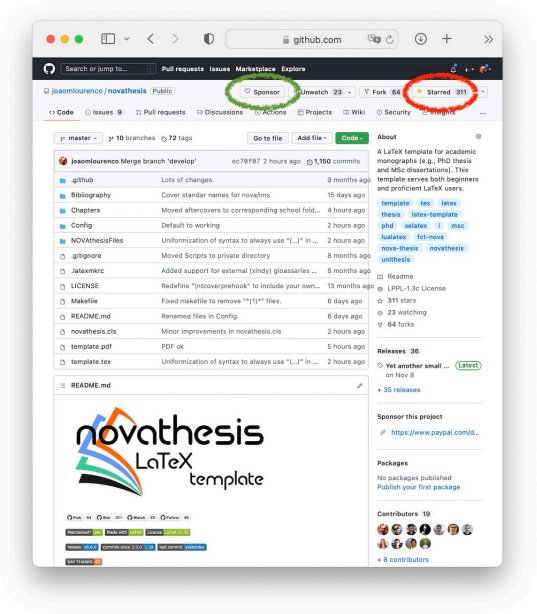


Figure 1.1: The novathesis project web page in GitHub.

1. **Configuration:** Start by editing the files in `0-Config/`. Select your school, set your document type (PhD, MSc, etc.), and fill in your metadata (title, author, supervisors) in `3_cover.tex`.

2. **Front Matter:** Write your acknowledgments, dedication, and abstracts in the `1-FrontMatter/` folder.

3. **Main Content:** Create separate `.tex` files for each chapter in `2-MainMatter/`. Register them in `0-Config/4_files.tex` to include them in the build.

4. **Bibliography:** Keep your references in `4-Bibliography/bibliography.bib`. The template uses `biblatex` to handle formatting automatically.

5. **Build & Iterate:** Use `make` or your editor's build command frequently to check the results.

1.4 Getting Help

No! You don't have to use this template to write your thesis. You don't even have to use LaTeX. However, writing a thesis is serious stuff, and which tool you shall use to write it is not a decision to make lighthearted.

LaTeX is hard enough by itself. This template aims at making your life easier, but not easy. If you choose to use this template to write your thesis, you are very welcome. However, don't expect me to provide you help with LaTeX. Look for help with your friends (you have some friends, don't you?), or search the web, or try even to read some book(s) on LaTeX. In the end you will certainly find the experience rewarding.

When you come to the point of "*How do I do this with the* novathesis *template?*", remember…

1. To check the novathesis wiki and have some hope! :D
2. Google is your best friend.
3. Search the GitHub Discussions page for a question related to yours. *If and only if* you don't find one, then post your own question in English please!
4. Search the NOVAtheis Facebook group for a question related to yours. *If and only if* you don't find one, then post your own question in either Portuguese or English, at your preference.

When you post your own question, remember to **always** state the novathesis version number you are using and referring to.

Please do not attempt to contact me directly (email, Messenger, etc)…
I WILL NOT REPLY!

1.4.1 Suggestions, Bugs and Feature Requests

Help: If you just need some help, see above Section 1.4.

Suggestion: Suggestions Do you have a suggestion/recommendation? Please add it to the wiki and help other users!

Bug: Bugs Did you find a bug? Please open an issue. Thanks!

New Feature: Feature Requests Would you like to request a new feature (or support of a new School)? Please open an issue. Thanks!

1.5 Donors

The list of *Donors* is available in the novathesis Project page.

1.6 Disclaimer

Although the novathesis template is endorsed by some Schools (e.g., linked from NOVA FCT web site), the novathesis template **this not an official template** for any School.

The novathesis template exists to make your life easier and we do our best to make it compliant to the supported (+25) Schools' regulations but, in the end of the line, you and only you are accountable for both the look and the contents of the document you submit as your thesis/dissertation.

2

NOVATHESIS TEMPLATE *USER'S MANUAL*

2.1 Introduction

This chapter is a complete guide to the novathesis template. It covers Everything from the first-time setup to the fine-tuning of covers, fonts, citation styles, and the document workflow. All user customization flows through the files inside the `0-Config/` folder — the rest of the project tree is never edited directly.

For a reference to the build system (`make` and the command-line overrides), see Chapter C. For a guide to the cover page system, see Chapter B.

2.2 Getting Started

2.2.1 Local LaTeX Installation

novathesis requires a *full* LaTeX installation. The two major distributions are:

TeX Live The reference distribution maintained by LaTeX developers. The full scheme occupies around 5 GB but includes every package you will ever need. Strongly recommended unless disk space is very tight. Download from `https://www.tug.org/texlive/`.

MiKTeX Installs a minimal base and downloads missing packages on the fly. A good choice when disk space is limited. Packages installed on the fly may occasionally fail, so keep the installation up to date. Download from `https://miktex.org/`.

After installation, update all packages via the *TeX Live Manager* or the *MiKTeX Console* before compiling the template for the first time.

Recommended engine. LuaLaTeX is the recommended engine for most schools because it handles Unicode fonts natively. XeLaTeX is a close alternative. pdfLaTeX works but cannot load OpenType/TrueType fonts.

School-specific fonts. Some Schools or font-styles require fonts not available in LaTeX. In normal conditions, they will be downloaded automatically. However, if they are not,

you may download the necessary fonts from https://github.com/joaomlourenco/novathesis-extras/tree/main/Fonts, unzip (if necessary), and copy the `*.ttf` files to `novathesisFiles/FontStyles/Fonts/`.

Security: compiling runs external programs (`--shell-escape`)

novathesis compiles with `--shell-escape` *enabled*, because several features run external programs during the build: `minted` calls Pygments to highlight source code, selecting a bundled proprietary font (Calibri, Arial, …) downloads it over the network, and printing the SDG icons (Section 2.4.5) downloads them the same way. (Glossaries no longer need shell-escape: since 8.0 they are built by `bib2gls`, which `latexmk` runs directly.) With shell-escape, **compiling a document can run commands on your computer with your own user privileges** — there is no sandbox.

Building the *official* template with *your own* content is safe. But you should only compile `.tex` files, school configurations, and font styles that you *trust* — treat a thesis project received from someone else the same way you would review any script before running it.

2.2.2 Downloading the Template

Clone the GitHub repository (recommended — you get all update history):

```
git clone --depth=1 https://github.com/joaomlourenco/novathesis.git
```

Or download the latest ZIP directly:

```
# Click "Code > Download ZIP" on https://github.com/joaomlourenco/novathesis
```

The top-level folder is your working directory. Rename it to anything you like. Open `template.tex` in your editor and compile it — you should get a Portable Document Format (PDF) immediately with the default settings (NOVA School of Science and Technology (FCT) Doctor of Philosophy (PhD) in English).

2.2.3 Overleaf

Overleaf users can import the template directly from the GitHub link. Once imported:

1. Set the *Compiler* to **LuaLaTeX** (Menu → Compiler).
2. Set the *Main document* to `template.tex`.
3. Set the *Bibliography tool* to **Biber** (Menu → Bibliography).
4. Press *Recompile*.

On Overleaf you cannot run `make`; all customization must be done by editing the `0-Config/` files directly (see Section 2.3).

2.3 Project Structure

Table 2.1 summarises the folders and top-level files in the project. Items marked **(edit)** are intended for user customization; items marked **(do not edit)** are part of the template engine.

Table 2.1: Project structure and access guidelines.

Path	Purpose	Access
`template.tex`	Main document entry point	do not edit
`novathesis.cls`	Document class	do not edit
`0-Config/`	All user configuration	**edit**
`1-FrontMatter/`	Abstracts, glossaries, front pages	**edit**
`2-MainMatter/`	Thesis chapters	**edit**
`3-BackMatter/`	Appendices and annexes	**edit**
`4-Bibliography/`	`.bib` database files	**edit**
`5-Figures/`	Images	**edit**
`novathesisFiles/`	Template internals (styles, schools, strings)	do not edit
`Scripts/`	Utility scripts	do not edit
`.Build/`	Maintainer build tools (`nt-variant.sh`)	do not edit

2.3.1 The `0-Config/` Files

All customization lives in `0-Config/`. Table 2.2 describes each file's purpose.

Table 2.2: Configuration files in `0-Config/`.

File	Purpose
`1_novathesis.tex`	Document options (`\ntsetup{...}`)
`2_biblatex.tex`	Bibliography/biblatex configuration
`3_cover.tex`	Cover metadata (title, author, advisers, committee)
`4_files.tex`	List of content files to include
`5_packages.tex`	User-added packages and custom macros
`9_*.tex`	School-specific overrides (department, degree names)

The *active* setting for any option is the *uncommented* line; commented-out lines show available values as documentation:

```
% \ntsetup{doctype=msc}    % commented = not active
\ntsetup{school=nova/fct}  % uncommented = active
```

2.4 Configuring the Document (`1_novathesis.tex`)

2.4.1 Essential Options

`doctype` The type of document being written. Default: `phd`.

```
\ntsetup{doctype=phd}      % PhD thesis
\ntsetup{doctype=phdplan}  % PhD thesis plan
\ntsetup{doctype=phdprop}  % PhD thesis proposal
\ntsetup{doctype=msc}      % MSc dissertation
\ntsetup{doctype=mscplan}  % MSc dissertation plan
\ntsetup{doctype=bsc}      % BSc report
\ntsetup{doctype=plain}    % Other report
```

The `*plan` and `*prop` document types produce a *simplified* front matter: only the abstracts and the AI disclosure statement are included. Pages such as the dedication, acknowledgements, quote, and evaluation committee are suppressed regardless of the `docstatus` setting.

`school` The school identifier. This selects the institutionally-compliant cover layout. Default: `nova/fct`.

```
\ntsetup{school=nova/fct}      % NOVA FCT (default)
\ntsetup{school=nova/fcsh}     % NOVA FCSH
\ntsetup{school=nova/itqb/green}
\ntsetup{school=ulisboa/ist}
\ntsetup{school=uminho/eeng}
% ... see 1_novathesis.tex for the full list
```

`docstatus` The workflow stage. Controls which front-matter pages are included. Default: `working`. See Section 2.10 for the full implications of each value.

```
\ntsetup{docstatus=working}      % drafting (default)
\ntsetup{docstatus=provisional}  % submitted; no committee yet
\ntsetup{docstatus=final}        % post-examination; committee, book
↪  spine, PDF/A-2b
```

`pdfa` Portable Document Format/Archive (PDF/A) archival output toggle. Default: `true`. In normal usage, PDF/A-2b output is produced automatically in `docstatus=final`; set this option to `false` to opt out. See Section 2.11.5 for details.

```
\ntsetup{pdfa=true}   % default behavior
\ntsetup{pdfa=false}  % disable PDF/A output
```

`lang` Main language of the document body (two- or three-letter language code). Default: `en`.

```
\ntsetup{lang=en}   % English (default)
\ntsetup{lang=pt}   % Portuguese
\ntsetup{lang=de}   % German
% cat, cz, dk, es, fr, gr, it, nl, pl, sk, uk also available
```

`media` Target output medium. Default: `screen`.

```
\ntsetup{media=screen}  % equal margins, colored links (default)
\ntsetup{media=paper}   % binding margins, black links
```

2.4.2 Appearance Options

style/font Body font family. Default: newpx (Palatino-like). Options marked with (*) require LuaLaTeX or XeLaTeX.

```
\ntsetup{style/font=newpx}        % Palatino-like (default)
\ntsetup{style/font=libertine}    % Linux Libertine
\ntsetup{style/font=opensans}     %
\ntsetup{style/font=arial}        % (*) Arial
\ntsetup{style/font=calibri}      % (*) Calibri
\ntsetup{style/font=newsgott}     % (*) NewsGotT (UMINHO)
% Any standard LaTeX font package name is also accepted.
```

For system fonts (LuaLaTeX/XeLaTeX only), provide the font name and attributes explicitly:

```
\ntsetup{style/font=Times New Roman}
\ntsetup{style/font/attributes={
  Extension       = .ttf,
  Ligatures       = TeX,
  UprightFont     = *,
  ItalicFont      = * Italic,
  BoldFont        = * Bold,
  BoldItalicFont  = * Bold Italic,
}}
```

style/font/fallback What to do when a font style marked (*) cannot be used: its files are not available and could not be downloaded (see Section 2.2.1), or the document is compiled with pdfLaTeX, which cannot use them. Default: false, which prints an error that explains how to get the font. With true only a warning is printed, and a font that ships with TeX Live and works with pdfLaTeX, XeLaTeX and LuaLaTeX is used instead: Calibri → Carlito, Verdana → DejaVu Sans, Arial and News Gothic T → TeX Gyre Heros, Futura and Jost → TeX Gyre Adventor.

```
\ntsetup{style/font/fallback=true}
```

style/chapter Chapter heading style. Default: bar.

```
\ntsetup{style/chapter=bar}          % (default)
\ntsetup{style/chapter=bar-compact}  % less vertical space
\ntsetup{style/chapter=elegant}      % classical
\ntsetup{style/chapter=bluebox}
\ntsetup{style/chapter=compact}      % minimal vertical space
\ntsetup{style/chapter=ist}          % ULISBOA-IST
\ntsetup{style/chapter=fmv}          % ULISBOA-FMV
```

color/links Color of hyperlinks in screen media. Default: a dark blue. Any color name from the xcolor package works.

```
\ntsetup{color/links=SteelBlue}
\ntsetup{color/links=black}   % effectively disables colored links
```

`style/url` URL font style.

```
\ntsetup{style/url=same}  % use main body font for URLs
```

`numbering/floats` Float (figure/table) numbering scheme. Default: `chapter` (*chapter.number*).

```
\ntsetup{numbering/floats=chapter}     % 1.1, 1.2, 2.1 … (default)
\ntsetup{numbering/floats=sequential}  % 1, 2, 3 …
```

2.4.3 Print Control Options

These options control which supplementary pages are included in the generated PDF. Many have defaults that vary with `school` and `docstatus`:

`print/frontpage` Print a front page (second cover). Default: `false`.

```
\ntsetup{print/frontpage=true}
```

`print/committee` Print the evaluation committee. Default: `false` for `working`/`provisional`; `true` for `final`.

```
\ntsetup{print/committee=true}   % force on
\ntsetup{print/committee=false}  % force off
```

`print/statement` Print the declaration/integrity statement. Default: `false`.

```
\ntsetup{print/statement=true}
```

`print/copyright` Print the copyright notice. Default: `true`.

```
\ntsetup{print/copyright=false}
```

`print/appendixname` Prefix "Appendix"/"Annex" to the chapter number in appendix/annex headings too (the PDF bookmarks already show it – see §2.8.6). Default: `false`.

```
\ntsetup{print/appendixname=true}
```

`spine/layout` Print the book spine. Default: `false` for `working` and `provisional`; `trim` (printed) for `final`.

```
\ntsetup{spine/layout=false}  % suppress spine even in final
\ntsetup{spine/layout=trim}   % force spine even in provisional
```

`print/index` Print a word index at the end. Default: `false`.

```
\ntsetup{print/index=true}
```

`print/timestamp` Print a generation timestamp on the cover. Default: `true` for `working`; `false` otherwise.

```
\ntsetup{print/timestamp=false}
```

print/webography Title for a separate bibliography section that collects only @online entries. When empty (default), all references are printed together.

```
\ntsetup{print/webography=Webography}
```

cite/template Add the reference to the novathesis template to your document's bibliography. Default: true.

novathesis is developed and maintained for free, and citations are the only currency in which that work is repaid. If you would rather not include the reference, simply turn the option off — no hard feelings.

```
\ntsetup{cite/template=false}
```

print/aidisclosure Print an AI use disclosure statement. Default: aidisclose (uses 1-FrontMatter/7-aidisclose.tex with the aidisclose package).

```
\ntsetup{print/aidisclosure=false}             % suppress
\ntsetup{print/aidisclosure=aidisclose}        % default package
\ntsetup{print/aidisclosure=aidisclosure.tex} % custom file
```

print/otherlists Where the "other lists" (list of figures, tables, listings, …) are printed. Default: frontmatter.

```
\ntsetup{print/otherlists=frontmatter}  % after the ToC (default)
\ntsetup{print/otherlists=backmatter}   % before the appendices
```

print/glossaries Where the glossaries (glossary, acronyms, symbols, chemical symbols) are printed, or false to suppress them. Default: frontmatter.

```
\ntsetup{print/glossaries=frontmatter}  % default
\ntsetup{print/glossaries=backmatter}
\ntsetup{print/glossaries=false}        % do not print
```

listof/skip Skip individual "list of …" entries (listoffigures, listoftables, listofalgorithms, listoflistings, or any custom name \ntaddlistof registers in 0-Config/6_list_of.tex) without commenting out their \ntaddlistof line. glossaries is also accepted here – it is just a friendlier spelling of print/glossaries=false, since glossaries aren't registered via \ntaddlistof at all. Default: empty (nothing skipped).

```
\ntsetup{listof/skip={listoftables,listofalgorithms,glossaries}}
```

abstract/title Print the work title with each abstract, before or after the abstract text. Default: none. The alignment of the title block is set with abstract/title/align (left, center, or right; default: center).

```
\ntsetup{abstract/title=none}    % default
\ntsetup{abstract/title=before}  % title above the abstract
\ntsetup{abstract/title=after}   % title below the abstract
\ntsetup{abstract/title/align=left}
```

2.4.4 Language Options

`abstractorder` Ordered list of languages for the abstract sequence. Default: `{lang, en}` when `lang≠en`; otherwise `{en, pt}`.

```
\ntsetup{abstractorder={en,pt}}
\ntsetup{abstractorder={en,pt,uk,gr}}   % add Ukrainian & Greek
```

`lang/cover` Language for the cover and front page. Default: same as `lang`.

```
\ntsetup{lang/cover=pt}   % Portuguese cover even if body is English
```

`lang/copyright` Language for the copyright message. Default: same as `lang`.

`lang/extra` Additional languages loaded in the document (besides the main language and abstract languages).

```
\ntsetup{lang/extra={de,es}}
```

2.4.5 Sustainable Development Goals (SDGs)

novathesis can print the United Nation's SDG icons at the bottom of the abstract page(s). All three options below should be set together:

```
\ntsetup{print/sdgs/list={3,7,13}}    % SDG numbers to print
\ntsetup{print/sdgs/type=inverted}    % inverted | normal | mono
\ntsetup{print/sdgs/size=2cm}         % icon size
\ntsetup{print/sdgs/hspace=5mm}       % horizontal gap between icons
\ntsetup{print/sdgs/vspace=4mm}       % vertical gap below icons
```

The `type` option selects the icon variant: `inverted` (colour on white background), `normal` (colour on coloured background), or `mono` (monochrome on white background). Table 2.3[1] lists all 17 SDGs with their icon in the three variants, title, and goal description.

Automatic download of the SDG icons. The SDG icons are *not* shipped with the template. Just like the school-specific fonts (Section 2.2.1), they are downloaded automatically when needed from `https://github.com/joaomlourenco/SDGs` into `novathesisFiles/Images/sdg-<lang>/`, and reused from there in later compilations, so the download happens only once. Only what is really necessary is fetched, namely the icons listed in `print/sdgs/list`, in the variant chosen with `print/sdgs/type`, and in the language of each abstract.

At the time of writing the icons exist upstream in English, Portuguese, Spanish, French and German. For an abstract written in any other language, the English icons are used instead, so the English set is always downloaded as a fallback. No list of languages is hardwired in the template: the upstream folder is simply the language code in upper case (`pt` → `PT`, …), so icons for a new language added upstream are

[1] Table 2.3 is intentionally in a page with differente dimensions (corresponding to a rotated page) to illustrate the funcitonality of the `stocksize` package.

used automatically, with no change to novathesis, as long as the new folder is named after the corresponding language code.

As for the fonts, this needs `--shell-escape` (see the warning box in page 6) and a network connection. A language with no icons upstream simply produces one warning per compilation and falls back to English. If the download fails, you may get the files by hand from the address above and copy them into that folder.

Table 2.3: The 17 UN Sustainable Development Goals (SDGs), in the `normal`, `inverted` and `mono` icon variants.

Normal	Inverted	Mono	Title	Goal
			**1 — No Poverty**	End poverty in all its forms everywhere.
			**2 — Zero Hunger**	End hunger, achieve food security, improve nutrition, and promote sustainable agriculture.
			**3 — Good Health and Well-being**	Ensure healthy lives and promote well-being for all at all ages.
			**4 — Quality Education**	Ensure inclusive and equitable quality education and promote lifelong learning opportunities.
			**5 — Gender Equality**	Achieve gender equality and empower all women and girls.
			**6 — Clean Water and Sanitation**	Ensure availability and sustainable management of water and sanitation for all.
			**7 — Affordable and Clean Energy**	Ensure access to affordable, reliable, sustainable, and modern energy for all.
			**8 — Decent Work and Economic Growth**	Promote sustained, inclusive, and sustainable economic growth, full employment, and decent work.

Normal	Inverted	Mono	Title	Goal
			**9 — Industry, Innovation and Infrastructure**	Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.
			**10 — Reduced Inequalities**	Reduce inequality within and among countries.
			**11 — Sustainable Cities and Communities**	Make cities and human settlements inclusive, safe, resilient, and sustainable.
			**12 — Responsible Consumption and Production**	Ensure sustainable consumption and production patterns.
			**13 — Climate Action**	Take urgent action to combat climate change and its impacts.
			**14 — Life Below Water**	Conserve and sustainably use the oceans, seas, and marine resources.
			**15 — Life on Land**	Protect, restore, and promote sustainable use of terrestrial ecosystems and halt biodiversity loss.
			**16 — Peace, Justice and Strong Institutions**	Promote peaceful and inclusive societies, provide access to justice, and build accountable institutions.
			**17 — Partnerships for the Goals**	Strengthen the means of implementation and revitalize the global partnership for sustainable development.

2.4.6 Indexing

novathesis provides built-in support for generating a word index at the end of the document.

Enabling the index. To generate and print the index, enable the `print/index` toggle in `0-Config/1_novathesis.tex`:

```
\ntsetup{print/index=true}
```

When active, the template automatically initializes the indexing engine and adds an "Index" (or "Índice Remissivo") section to the back matter.

Indexing words with `\ntindex`. While the standard LaTeX `\index{word}` command only adds an entry to the index file (printing nothing in the text), the novathesis `\ntindex` command is more convenient:

```
This is a \ntindex{recommendation} for users.
```

The above code prints "recommendation" in the document **and** adds it to the index.

Custom keys and sub-entries. You can use an optional argument to specify a different index key or create sub-entries:

```
\ntindex[template!]{Version} % Prints "Version", indexes under Template >
↪  Version
\ntindex[project web page]{GitHub} % Prints "GitHub", indexes under Project
↪  web page
```

Debugging the index. If you want to see exactly which words have been indexed while proofreading, enable the index debug mode:

```
\ntsetup{debug={index}}
```

This will highlight every word indexed with `\ntindex` with a small label in the margin.

2.4.7 Glossary and Miscellaneous Options

```
% Hyperlink color for glossary entries (default: black)
\ntsetup{color/glossaries/gls=SteelBlue}

% Show back-references in glossary entries (default: true)
\ntsetup{glossaries/list/reverse=false}

% Glossary column widths
\ntsetup{glossaries/layout={>{\raggedright}p{0.2\textwidth}X}}

% Show letter-group headings before each alphabetic group (default: none)
\ntsetup{glossaries/groupheadings={acronym,symbols}}
```

```
% Rename a glossary's printed/TOC title (default: unchanged)
\ntsetup{glossaries/titles={symbols=List of Symbols}}

% Include the Table of Contents as an entry in itself (default: true)
\ntsetup{tocintoc=false}

% Book spine width (default: 1cm)
\ntsetup{spine/width=2cm}

% Debug helpers (print grid and bounding boxes on cover/spine)
\ntsetup{debug={cover,spine}}
```

Letter-group headings. Glossary, acronym, and symbols lists are always alphabetically grouped by first letter, but by default no heading is printed before each group. To show a bold heading ("A", "B", …) before each letter group, opt in per glossary type with `glossaries/groupheadings`:

```
% Show group headings only for the acronyms and symbols lists
\ntsetup{glossaries/groupheadings={acronym,symbols}}
```

Valid type names are `glossary`, `acronym`, `symbols`, or any custom type registered with `\ntaddfile{glossaries}[<type>]{...}`. More than one type needs the extra pair of braces shown above, to protect the comma from `\ntsetup`'s own top-level comma splitting; a single type does not, e.g. `\ntsetup{glossaries/groupheadings=acronym}`.

Renaming a glossary's title. To rename the printed (and table-of-contents) title of a glossary type – for example "Symbols" to "List of Symbols" – use `glossaries/titles` with a comma-separated list of `type=Title` pairs:

```
\ntsetup{glossaries/titles={symbols=List of Symbols,
                            acronym=List of Abbreviations}}
```

As with `glossaries/groupheadings`, wrap multiple pairs in one extra pair of braces (a single pair does not need it). Neither the type name nor the title text may contain a literal comma or equals sign – the parser is a simple split, not a full tokenizer.

The un-typed "main" glossary. Internally, `bib2gls` calls the un-typed glossary you feed via `\ntaddfile{glossaries}[glossary]{...}` `main`, not `glossary`. Both options above accept `glossary` as an alias for it, so you never need to write `main` yourself.

2.5 Cover Metadata (`3_cover.tex`)

2.5.1 Title and Subtitle

`\nttitle` accepts an optional *type* and a *language*:

```
\nttitle(main,en){Full Thesis Title}
\nttitle(main,pt){Título Completo da Tese}
\nttitle(sub,en){Optional Subtitle}
\nttitle(sub,pt){Subtítulo Opcional}
% Variants for specific contexts (if needed):
\nttitle(cover,en){Shorter Title for the Cover}
\nttitle(abstract,en){Title Version for the Abstract}
\nttitle(spine,en){Very Short Title}
```

Use \\ to force line breaks anywhere in a title.

2.5.2 Department, Degree, and Specialization

```
\ntdepartment*(en){Name of the Department}
\ntdepartment*(pt){Nome do Departamento}

\ntdegreename*(en){Computer Science}
\ntdegreename*(pt){Engenharia Informática}

\ntspecialization*(en){Distributed Systems}    % omit if not applicable
\ntspecialization*(pt){Sistemas Distribuídos}
```

NOVA FCT users: open `0-Config/9_nova_fct.tex` instead and uncomment the correct department/degree entries.

2.5.3 Author

```
\ntauthorname(m){John Very Longname Doe}{John Doe}
% (m) = male, (f) = female
% First argument: full name for the cover
% Second argument: short name for other uses
```

For the author's previous degree (printed on some covers):

```
\ntauthordegree(en){Master of Science in Computer Science}
\ntauthordegree(pt){Mestrado em Engenharia Informática}
```

For the author's date/place of birth (`school=other/huberlin` only – its cover prints *"geb. am … in …"*; every other school's cover simply ignores these):

```
\SetAuthorBirthDate*()={1998-03-03}   % ISO format; no language key
\SetAuthorBirthPlace*(de)={Berlin}    % one line per cover language you use
\SetAuthorBirthPlace*(en)={Berlin}
```

2.5.4 Advisers

```
% Principal adviser
\ntaddperson{adviser}(a,f){Mary Adviser}
      [position=Full Professor, institution=NOVA University Lisbon]
% Co-advisers
```

```
\ntaddperson{adviser}(c,m){John Co-Adviser}
      [position=Associate Professor, institution=IST]
```

Valid role codes: `a` (adviser), `c/co` (co-adviser), `t` (tutor). Gender codes: `m` (male), `f` (female). The `title`, `position`, and `institution` fields are all optional; a legacy `{name, position, institution}` comma string is still accepted too, but is deprecated (see Appendix A).

2.5.5 Evaluation Committee

Committee members are only printed when `docstatus=final` (or when forced with `\ntsetup{print/committee=true}`):

```
\ntaddperson{committee}(c,m){Chair Name}
      [position=Full Professor, institution=NOVA University Lisbon]
\ntaddperson{committee}(r,m){Rapporteur}
      [position=Associate Professor, institution=University of Porto]
\ntaddperson{committee}(m,f){Member Name}
      [position=Assistant Professor, institution=University X]
\ntaddperson{committee}(g,f){Guest Member}
      [position=Position, institution=Institution]
```

Valid roles: `c` (chair), `r` (rapporteur/arguente), `a` (adviser, if in committee), `co` (co-adviser, if in committee), `m` (member), `g` (guest).

2.5.6 Dates and Sponsors

```
\ntdate(submission){2025-09-30}   % YYYY-MM-DD preferred
\ntdate(exam){2026-01-15}

% Sponsors (logos or text, per language)
\ntsponsors(1,logo,en){\includegraphics[height=1.5cm]{republica_portuguesa_CTES}}
\ntsponsors(2,logo,en){\includegraphics[height=1.5cm]{FCT_MEC_logo_positive}}
```

2.6 Book Spine

The *book spine* is the narrow strip printed on the bound edge of the thesis, showing the year, title, author name, and school logo from top to bottom. It is generated automatically in `final` mode and typeset as a separate page (rotated 90°) appended after the back cover.

2.6.1 Enabling and Sizing

`spine/layout` Controls when the spine is printed. Default: `false` for `working` and `provisional`; `trim` for `final`.

```
\ntsetup{spine/layout=false}  % suppress spine even in final
\ntsetup{spine/layout=trim}   % force spine even in provisional/working
```

spine/width Thickness of the spine strip (= book spine width). Default: `1cm`. Set this to match the actual thickness of the printed and bound book.

```
\ntsetup{spine/width=2cm}
```

2.6.2 Spine Title

When the full thesis title is too long for the spine, supply a shorter variant with `\nttitle`:

```
\nttitle(spine,en){A Very Short Title}
\nttitle(spine,pt){Um Título Muito Curto}
```

If no `spine` variant is defined, the `cover` variant is used; if that is also absent, the `main` title is used.

2.6.3 Customising the Spine with `\SpineSetup`

`\SpineSetup` is the main command for customising the spine appearance. It is typically called in the school `.clo` file, but users may call it in `0-Config/3_cover.tex` or `0-Config/5_packages.tex` to override or extend the school defaults.

```
\SpineSetup[<preamble code>]{<key=value list>}
```

The optional *preamble code* is executed first (useful for conditionals based on `\ifmscdoc` / `\ifphddoc`), followed by setting the key–value pairs listed in the mandatory argument. Multiple calls accumulate; later calls override earlier ones for the same key.

Global Appearance Keys

bg/image File name of a full-spine background image (without extension).

bg/image/angle Rotation angle of the background image. Allowed values: `0`, `90`, `-90/270`, `180`. Default: `0`.

bg/color Background colour of the whole spine strip. Use `none` for transparent. Default: `none`.

bg/opacity Opacity of the background image (`0`–`1`). Default: `1`.

fg/color Default text colour for all boxes. Default: `black`.

frame/color Colour of the outer frame drawn around the spine strip. Use `none` to suppress. Default: `none`.

frame/width Width of the outer frame. Default: `0pt`.

boxsep Gap inserted between adjacent content boxes. Default: `5mm`.

margin/top, margin/bottom Margins between the spine edge and the top/bottom of the content area. Defaults: `1mm`.

margin/left, margin/right Margins at the left and right (= top and bottom when the spine is rotated) of the content area. Defaults: `5mm` / `0mm`.

Box Content and Order

The spine is divided into a sequence of named *boxes*. Four built-in boxes are available:

Box name	Content	Default length
`date`	Submission year	1.5 cm
`title`	Thesis title (spine variant)	16.6 cm
`author`	Author name (short form)	5.0 cm
`logo`	School logo image	4.1 cm

`order` Comma-separated list specifying which boxes to print and in what order (left-to-right along the rotated spine, i.e. top-to-bottom when reading the bound book). Default: `date,title,author,logo`.

```
\SpineSetup{order={date,title,author,logo}}
```

`author` Text placed in the `author` box. Default: `\GetDocAuthor(name,short)`.

`title` Text placed in the `title` box. Default: `\GetDocTitle(\@LANG@COVER,spine)`.

`date` Text placed in the `date` box. Default: `\GetDocDate(submission,year)`.

Per-Box Keys

Each built-in box (`date`, `title`, `author`, `logo`) accepts the following keys, prefixed with `box/<name>/`:

`box/<name>/len` Length (height when the spine is laid horizontally) of the box.

```
\SpineSetup{box/logo/len=4.5cm}
```

`box/<name>/extrasep` Additional gap added *after* this box. Default: `0mm`.

`box/<name>/fontstyle` Font-switching command applied to the box content. Default: `\sffamily`.

`box/<name>/bg/color` Background colour of the box. Default: `cyan` (override per school as needed).

`box/<name>/bg/opacity` Opacity of the box background. Default: `1`.

`box/<name>/bg/image` Background image file for the box.

```
\SpineSetup{box/logo/bg/image={my-school-logo}}
```

`box/<name>/bg/image/scale` Scale factor (0–1) applied to the maximum box dimensions before fitting the image. The default `1` fills the whole box; smaller values add proportional padding around the image.

```
\SpineSetup{box/logo/bg/image/scale=0.75}  % logo at 75% of box size
```

box/<name>/bg/image/angle Rotation angle for the per-box background image. Default: `0`. Doctype-specific overrides are also available: `box/<name>/bg/image/angle/msc` and `box/<name>/bg/image/angle/phd`.

box/<name>/fg/color Foreground (text) colour of the box. Default: `black`.

```
\SpineSetup{box/title/fg/color=white}
```

box/<name>/frame/color Frame colour drawn around the box. Use `none` to suppress. Default: `none`.

box/<name>/frame/width Width of the box frame. Default: `0pt`.

box/<name>/halign Horizontal alignment of box content: `left`, `center`, `right`. Default: `center`.

box/<name>/valign Vertical alignment of box content: `top`, `center`, `bottom`. Default: `center`.

2.6.4 Example

The following example (similar to the `nova/fct` school) sets a full-spine background image, makes all text white, and places the school logo as a box background image — choosing different images for MSc and PhD:

```
\SpineSetup[%
  \def\spnimg{my-spine-blue}%         % default image
  \ifphddoc{\def\spnimg{my-spine-green}}{} % PhD override
]{
  bg/image                    = {\spnimg},
  bg/image/angle              = {-90},
  box/logo/bg/image           = {my-school-logo},
  box/date/fg/color           = {white},
  box/title/fg/color          = {white},
  box/author/fg/color         = {white},
}
```

A minimal setup that only changes the logo box background:

```
\SpineSetup{
  box/logo/bg/image = {my-school-logo},
}
```

2.7 Bibliography (`2_biblatex.tex`)

novathesis uses `biblatex` with the `Biber` backend.

2.7.1 General Behavioral Options

```
\ntbibsetup{
  backend      = biber,        % recommended; 'bibtex' also works
  giveninits   = true,         % print initials only
  maxbibnames  = 1000,         % list all authors (no 'et al')
  backref      = true,         % show page numbers for each citation
  backrefstyle = three,        % compact back-reference style
  urldate      = iso,          % ISO dates for URL access
  date         = iso,          % ISO dates for publications
}
```

2.7.2 Choosing a Citation Style

Uncomment exactly one style block in `0-Config/2_biblatex.tex`:

Numeric (sorted by first author) The default. Citations like [1]:

```
\ntbibsetup{style=numeric-comp, sortcites=true, sorting=nyt}
```

Numeric (citation order) Sequential citations [1], [2], …:

```
\ntbibsetup{style=numeric-comp, sortcites=true, sorting=none}
```

Alpha Citations like [Lou24]:

```
\ntbibsetup{style=alphabetic, sortcites=true, sorting=nyt}
```

Author-year Citations like (Lourenço, 2024):

```
\ntbibsetup{style=authoryear-comp, sortcites=true, sorting=nyt}
```

Discipline-specific IEEE, APA, Vancouver, Chicago are also available — see the style blocks in `2_biblatex.tex`.

2.7.3 Webography (Separate Online Bibliography)

To collect all `@online` entries in a separate section, set a title:

```
\ntsetup{print/webography=Webography}
```

When this option is active, `@online` references are excluded from the main bibliography list and printed in their own section at the same location.

2.8 Adding Content (`4_files.tex`)

All content files are registered in `0-Config/4_files.tex` using `\ntaddfile`.

2.8.1 Bibliography Files

```
\ntaddfile{bib}{bibliography.bib}
\ntaddfile{bib}{extra.bib}   % multiple .bib files are allowed
```

Place `.bib` files in `4-Bibliography/`.

2.8.2 Front Matter

```
\ntaddfile{dedication}{dedication}          % 1-FrontMatter/dedication.tex
\ntaddfile{acknowledgements}{acknowledgements}
\ntaddfile{quote}{quote}
% Dedication, quote, and acknowledgements are skipped in 'working' mode.

% Integrity statement (school-specific files available in novathesisFiles)
\ntaddfile{statement}[en]{univ-school/statement-en}
```

2.8.3 Abstracts

```
\ntaddfile{abstract}[pt]{abstract-pt}   % 1-FrontMatter/abstract-pt.tex
\ntaddfile{abstract}[en]{abstract-en}
% Other supported language codes: de, es, fr, gr, it, nl, uk, zhs, zht
```

The order in which abstracts appear is controlled by `\ntsetup{abstractorder=...}` (Section 2.4.4).

2.8.4 Glossaries, Acronyms, and Symbols

```
\ntaddfile{glossaries}[glossary]{glossary}   % 1-FrontMatter/glossary.tex
\ntaddfile{glossaries}[acronym]{acronyms}
\ntaddfile{glossaries}[symbols]{symbols}
```

Each file uses standard `glossaries` package commands such as `\newglossaryentry`, `\newacronym`, etc. To include them in your text, use the `\gls{label}` command.

2.8.5 Main Chapters

```
\ntaddfile{chapter}{chapter-introduction}   %
↪  2-MainMatter/chapter-introduction.tex
\ntaddfile{chapter}{chapter-methods}
\ntaddfile{chapter}{chapter-results}
\ntaddfile{chapter}{chapter-conclusion}
```

Chapters are included in the order they appear in `4_files.tex`. To restrict a chapter to specific document types, pass an optional list:

```
\ntaddfile{chapter}[mscplan,msc]{chapter-proposal}  % only in \gls{msc} docs
```

2.8.6 Appendices and Annexes

```
\ntaddfile{appendix}{appendix-a}   % 2-MainMatter/appendix-a.tex
\ntaddfile{annex}{annex-b}         % 2-MainMatter/annex-b.tex
```

The same optional doctype filter applies to appendices and annexes.

By default, the appendix/annex chapter heading shows only the number (e.g. a big "A"), while the PDF bookmark already reads "Appendix A". Set `print/appendixname=true` to also prefix "Appendix"/"Annex" to the printed heading – see the `print/appendixname` option.

2.8.7 Custom Covers

To override an automatically-generated cover with a custom file:

```
\ntaddfile{cover}[1]{cover-page}   % 1-FrontMatter/cover-page.tex
% cover ids: 1 (main cover), 2 (front page), N (back cover), spine
```

2.8.8 Bundled Institutional Images

novathesis bundles a set of institutional logo images in `novathesisFiles/Images/`. These files are *not* in the default graphics search path, so reference them with the `\DIRIMAGES` macro (or copy them to `5-Figures/` to use just the file name):

```
\includegraphics[height=1.5cm]{\DIRIMAGES/FCT_logo_positive}
```

Table 2.4 shows the available images.

Table 2.4: Institutional logo images bundled in `novathesisFiles/Images/`.

Preview	File name	Description
	`FCT_logo_positive`	FCT logo (colour)
	`FCT_logo_negative`	FCT logo (negative / white)
	`republica_portuguesa`	República Portuguesa emblem
	`republica_portuguesa_CTES`	República Portuguesa — CTES variant
	`Logo_Governo_Portugal`	Governo de Portugal logo
	`Logo_UE_FSE`	EU / Fundo Social Europeu logo

2.9 Custom Packages and Commands (`5_packages.tex`)

Add any extra `\usepackage` commands and custom macros in `0-Config/5_packages.tex`. This file is loaded after the template core so any redefinitions here take precedence.

This file also contains the pre-defined `minted` code environments used throughout the demo chapters:

`latexcode` Syntax-highlighted LaTeX listing with a light beige background.

`bashcode` Shell/Bash listing with the same style.

`pythoncode` Python listing with the same style.

`\latexinline{...}` Inline highlighted code fragments.

Use these environments in your chapters for consistent formatting:

```
\ntsetup{school=nova/fct}
```

```
make lua
```

To add a language-string rename (e.g. change the TOC heading):

```
\ntlangsetup{en/contents=Table of Contents}
\ntlangsetup{pt/contents=Índice}
```

2.10 Document Status Workflow

The `docstatus` option controls which pages are included and how the cover is rendered. Table 2.5 summarises the differences.

Table 2.5: Effect of `docstatus` on the generated document.

Feature	`working`	`provisional`	`final`
Timestamp on cover	yes	no	no
Dedication / quote	no	yes	yes
Acknowledgements	no	yes	yes
Integrity statement	no	yes	yes
Evaluation committee	no	no	yes
Book spine	no	no	yes
PDF/A-2b output	no	no	yes

See Section 2.11.5 for details on the PDF/A archival output.

A typical workflow:

1. Write with `docstatus=working` (fast, no distractions).
2. Switch to Master of Science (MSc) when submitting to the examination jury — a clean PDF without the committee members.
3. Switch to `final` after the examination — includes the committee, the book spine, and produces a PDF/A-2b archival document (see Section 2.11.5).

2.11 Building the Document

2.11.1 Using `make`

The `Makefile` at the project root wraps `latexmk` and provides a set of commands to manage the entire document lifecycle. You can run `make help` at any time to see the full list of available targets.

Basic Compilation. These targets handle the multi-pass build (including bibliography) using specific engines:

```
make        # build with LuaLaTeX (recommended default)
make lua    # build using LuaLaTeX
make xe     # build using XeLaTeX
make pdf    # build using pdfLaTeX
```

Viewing and Debugging.

```
make view   # build (if needed) and open the PDF in the default viewer
make watch  # rebuild automatically every time a file is saved
make log    # show the LaTeX log file
```

Cleaning Files.

```
make clean      # remove auxiliary files (keep the PDF)
make distclean  # same as clean, but also remove the PDF and SyncTeX files
```

Variables. The behavior is adjusted with variables appended to the command:

```
make V=1                        # verbose (raw LaTeX output)
make BATCH=1                    # batch mode, never stops at errors
make TL=2024                    # use the TeX Live 2024 release
make NT="doctype=msc,lang=pt"   # override \ntsetup options (see below)
make FILE=mythesis              # explicit root file
make SIZE=10                    # cap the final PDF at ~10MB (see below)
```

Capping the File Size. Some submission systems (repositories, conference portals, …) enforce a maximum PDF size. `SIZE=x` (megabytes) shrinks the build to fit: after the normal LaTeX run, novathesis binary-searches, via Ghostscript, the lowest raster-image resolution that keeps the PDF under `x` MB, and re-embeds every image at that resolution. Vector graphics, fonts and text are untouched — only raster images (photos, screenshots, rasterised plots) are resampled, so a document with little or no raster content may never reach a very small target; when that happens the build still succeeds, with a warning, producing the smallest PDF it could reach.

```
make SIZE=10           # build, then shrink to about 10MB
make lua SIZE=5 NT="lang=pt"
```

Requires Ghostscript (`gs`) on `PATH`. The pre-shrink build is kept as `<FILE>.pdf.orig` next to the shrunk `<FILE>.pdf`, so re-running `make` without `SIZE` restores full quality. Because downsampling rewrites every image stream, re-validate PDF/A compliance (Section 2.11.5) if you combine `SIZE` with `docstatus=final`.

Overrides. The `NT` variable accepts any comma-separated list of `\ntsetup` options; they take precedence over the values in `0-Config/1_novathesis.tex`, without editing any file. This is the easiest way to build for another school, language, or document type. See Chapter C for the full reference.

2.11.2 Using `latexmk` Directly

```
latexmk -pdflua -shell-escape -file-line-error template
```

The `latexmkrc` file at the project root is loaded automatically and already configures the auxiliary directory, Biber, and the glossaries.

Remember to select **biber** (not bibtex) as the bibliography processor — either in your editor preferences or via `latexmk`'s default settings.

2.11.3 Building for Another School

The `NT` overrides build for a specific school without modifying the `0-Config/` files. This is also what the CI pipeline and the maintainers use to generate sample PDFs. For full documentation see Chapter C.

Quick example:

```
make NT="school=nova/fct,doctype=phd,lang=en,docstatus=final"
```

2.11.4 Overleaf

On Overleaf the build is triggered by pressing *Recompile*. Ensure the settings described in Section 2.2.3 are applied (LuaLaTeX compiler, Biber bibliography, `template.tex` as main file). Overleaf compiles on their servers so `make` is not available — all configuration is done by editing the `0-Config/` files.

2.11.5 PDF/A Archival Output

Archival compliance When `\ntsetup{docstatus=final}` is set, novathesis automatically produces a **PDF/A-2b** conformant document — the archival-quality format required by most institutional repositories and digital libraries. No configuration beyond setting `docstatus=final` is needed.

How it works (single-pass). novathesis loads the `pdfx` package during the normal LaTeX preamble, just before `hyperref`. This single-pass approach requires no extra compilation run beyond the usual cross-reference and bibliography passes performed by every LaTeX document.

The XMP metadata packet (title, author, language) is written from the cover-page data defined in `0-Config/3_cover.tex` and embedded into the PDF automatically.

PDF/A standard choice. novathesis targets **PDF/A-2b** (not 1b). PDF/A-1b prohibits transparency, which is incompatible with the TikZ cover-page graphics used by virtually all supported schools. PDF/A-2b lifts this restriction while remaining accepted by all major institutional repositories.

XMP metadata. The thesis title, author name, and language are taken directly from the cover-page data in `0-Config/3_cover.tex` and passed to `pdfx`, which embeds them in the required XMP packet. No additional configuration is needed.

Images and colour spaces. The most common cause of PDF/A validation failures is an embedded image with a device-dependent colour space (plain RGB or CMYK) and no ICC profile. PDF/A-2b requires every included image to carry an embedded ICC profile or use a calibrated colour space. Practical guidance by format:

JPEG / JPG Most cameras and photo-editors embed an sRGB ICC profile automatically. Screenshots and diagrams exported from presentation or diagramming software often do not. Check the source file and add a profile before including it (*e.g.* via GIMP's *Image* → *Mode* → *Assign Colour Profile*, or with ImageMagick: `magick -profile sRGB.icc input.jpg output.jpg`).

PNG Inkscape embeds an sRGB profile when exporting via *File* → *Export PNG Image*. Images from other sources should be checked; if the profile is missing, add it with the same tools as for JPEG above.

TIFF LaTeX engines cannot include TIFF directly. Convert to PDF or PNG with an embedded ICC profile using an image-editing tool (GIMP, ImageMagick, or Photoshop) before including the result.

PDF (figures) An included PDF figure must itself use device-independent colour spaces throughout. Figures produced by Matplotlib, Inkscape, and TikZ are almost always clean. Scanned or converted PDFs may not be — if VeraPDF reports colour-space errors linked to a specific figure, pre-process it with Ghostscript before re-including:

```
gs -dBATCH -dNOPAUSE -sDEVICE=pdfwrite \
   -sColorConversionStrategy=sRGB \
   -o figure-fixed.pdf figure-original.pdf
```

The safest choice is **vector PDF** (TikZ, Inkscape, Matplotlib): it contains no raster data and is intrinsically colour-space clean.

Opting out. To disable PDF/A output even in `final` mode, uncomment the escape hatch in `0-Config/1_novathesis.tex`:

```
\ntsetup{pdfa=false}
```

Verifying compliance. After the final build, validate with:

- **VeraPDF** (open source, https://verapdf.org) — the reference PDF/A validator.
- **Adobe Acrobat Pro** — *Tools* → *Print Production* → *Preflight* → *PDF/A compliance*.
- **PDF/A Live!** (online service) — quick check without local installation.

3

Artificial Intelligence Disclosure Statement

As Generative Artificial Intelligence (Artificial Intelligence (AI)) becomes increasingly integrated into academic research, transparency regarding its use is essential. The novathesis template provides built-in support for rendering a formal *Artificial Intelligence Disclosure Statement*.

3.1 Why Disclose AI Use?

Academic integrity requires that authors are transparent about the tools and methods used to produce their work. Disclosing AI use:

- **Promotes Transparency:** It allows readers and examiners to understand which parts of the research process were assisted by AI.
- **Ensures Accountability:** Regardless of AI assistance, the **author remains solely responsible** for the accuracy, originality, and ethical integrity of the thesis. AI-generated "hallucinations" or errors are the author's responsibility to correct.
- **Complies with Regulations:** Many universities and publishers now require a formal declaration of AI use. *Please check your school's specific regulations regarding Generative AI.*

3.2 Configuration

The rendering of the statement is controlled by the `\ntsetup{print/aidisclosure=OPTION}` option in the file `0-Config/1_novathesis.tex`.

Possible values for `OPTION` are:

`false`	Omit the AI Disclosure Statement entirely.
`aidisclose`	(Default) Render the *standard* checklist using the `aidisclose` package.
`filename.tex`	Render a custom file (e.g., `aidisclosure.tex`) from the `1-FrontMatter/` folder.

3.3 The Standard aidisclose Checklist

The novathesis template uses the `aidisclose` package [7] (also on GitHub), which follows the *GAIDeT (Generative AI Delegation Taxonomy)* [8]. This provides a structured checklist of tasks (e.g., brainstorming, proofreading, code generation) where AI might have been employed.

3.3.1 Using the Companion Website (Recommended)

The most efficient way to generate your disclosure is to use the companion web site at `aidisclose.org` (see Figure 3.1).

1. Visit aidisclose.org.
2. Fill in your name and the AI tools used (e.g., ChatGPT, Claude, Gemini).
3. Select the appropriate checkboxes in the *Taxonomy of Delegated Tasks*.
4. Click the green button to generate the LaTeX snippet.
5. Copy the snippet and replace the contents of `0-Config/7-aidisclose.tex` in your project.

3.3.2 Manual Configuration

Alternatively, you can manually edit `0-Config/7-aidisclose.tex` by activating specific tags:

```
\AIDactivate{w:poly}  % For Polishing and Editing
\AIDactivate{s:gen}   % For Code Generation
```

3.4 What if No AI was Used?

If you did **not** use any Generative AI tools, you have two choices:

1. **Deactivate it:** Set `\ntsetup{print/aidisclosure=false}` in `0-Config/1_novathesis.tex`.
2. **Negative Declaration:** Some schools prefer an explicit statement that no AI was used. In this case, create a custom file in `1-FrontMatter/aidisclosure.tex` with your declaration and set `\ntsetup{print/aidisclosure=aidisclosure.tex}`.

3.5 Placement in the Document

By default, the AI Disclosure Statement is printed in the **Front Matter**, typically appearing after the Integrity Statement and before the Abstracts. It is automatically included in the PDF when `docstatus` is set to `provisional` or `final`.

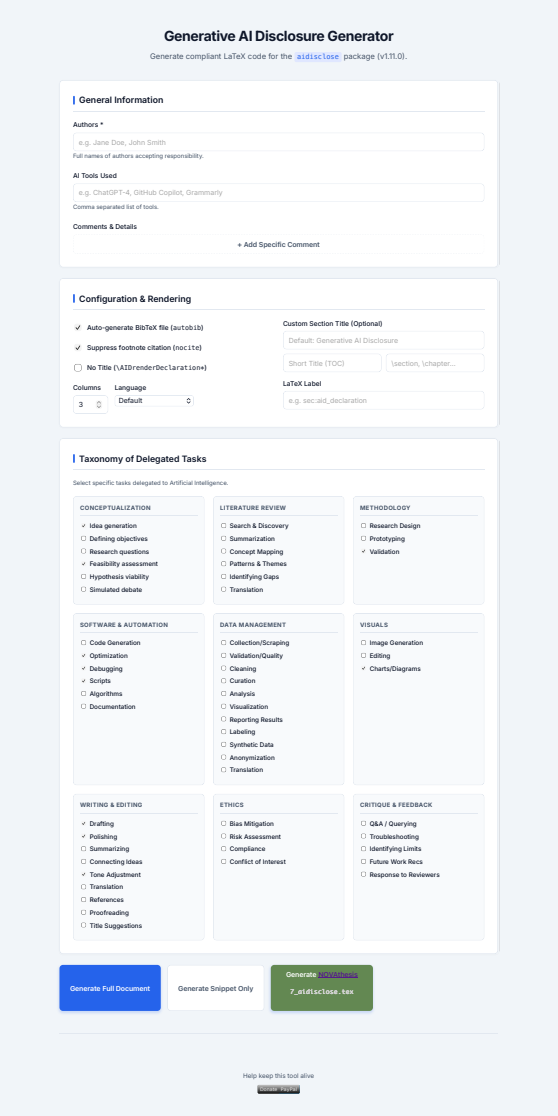


Figure 3.1: The aidisclose.org web site

3.6 Detailed Taxonomy Descriptions

For a detailed breakdown of what is included in each category (Objective, Scope, and Keys), please refer to Appendix E.

4

A Short LaTeX Tutorial with Examples

This chapter is a practical guide to writing your thesis with LaTeX and the novathesis template. It covers the most common tasks: structuring your document, formatting text, inserting figures and tables, typesetting equations, managing citations, and producing the final PDF.

Use these examples as a starting point. The Overleaf documentation and TeX Stack Exchange are excellent resources for additional help.

Do not edit "`template.tex`" or "`novathesis.cls`"

All your customizations go into the files inside `0-Config/`. Your chapter content lives in `2-MainMatter/` and is listed in `0-Config/4_files.tex`. The root file `template.tex` and the class file `novathesis.cls` are **never** edited directly.

4.1 Document Structure

The novathesis template follows a strict separation of concerns, with configuration separated from content. A detailed overview of the folder hierarchy and the purpose of each file is provided in section 2.3.

4.1.1 Adding a New Chapter

Create a new `chapter-mywork.tex` file in `2-MainMatter/`, then register the new file in configuration file `0-Config/4_files.tex`:

```
\ntaddfile{chapter}{chapter-mywork}
```

Inside the file, start with a `\chapter` command and give it a label:

```
\chapter{My Work}
\label{cha:my_work}

\section{Overview}
\label{sec:overview}
```

The sectioning hierarchy is: `\chapter` → `\section` → `\subsection` → `\subsubsection` → `\paragraph` → `\subparagraph`. Avoid going deeper than `\subsection` unless absolutely necessary.

4.1.2 Cross-references

Place a `\label` immediately after the element you want to reference — a chapter, section, figure, table, or equation — and then use `\autoref` to cite it:

```
See \autoref{cha:my_work} for details.
The results are in \autoref{fig:results} and \autoref{tab:results}.
```

`\autoref` automatically prepends the correct word (*Chapter*, *Figure*, *Table*, …) in the current language. Use `\ref` if you only want the number.

Tip: non-breaking space before \ref

Always write `Figure~\ref{fig:x}` with a non-breaking space (~) so the word and number never split across lines.

4.1.2.1 Teaching `\autoref` about a new object type

`\autoref` looks up the label prefix to determine the word to prepend. When you define a new numbered environment — say, a `listing` or `algorithm` — you register its name by defining the macro `\<counter>autorefname`:

```
% In 0-Config/5_packages.tex, after \usepackage{algorithm2e}:
\newcommand{\algorithmautorefname}{Algorithm}

% For a custom theorem environment with counter "theorem":
\newcommand{\theoremautorefname}{Theorem}
```

If your document is multilingual you can conditionally define the name based on the active language:

```
\addto\extrasenglish{\renewcommand{\algorithmautorefname}{Algorithm}}
\addto\extrasportuges{\renewcommand{\algorithmautorefname}{Algoritmo}}
```

4.2 The `cleveref` Package

The `cleveref` package is a more powerful alternative to `\autoref`. Load it *last* in your preamble (in `0-Config/5_packages.tex`) with:

```
\AtEndPreamble{
  \usepackage{cleveref}
}
```

cleveref must be loaded last

cleveref must be loaded *after* hyperref and after any package that defines new float or theorem environments. In the novathesis template, add it as above and at the end of 0-Config/5_packages.tex.

4.2.1 Basic Usage

Table 4.1: Core cleveref commands.

Command	Result
\cref{key}	Lowercase: *figure 1*, *chapter 2*, …
\Cref{key}	Capitalised: *Figure 1*, *Chapter 2*, …
\cref{k1,k2,k3}	Multiple refs in one call: *figures 1, 2 and 3*
\crefrange{a}{b}	Range: *figures 1 to 3*
\labelcref{key}	Number only (no type word)

Use \Cref at the start of a sentence and \cref elsewhere:

```
\Cref{cha:my_work} presents the proposed approach.
The results are summarised in \cref{tab:results,fig:results}.
```

A key advantage over \autoref is that \cref can handle a comma-separated list of labels and automatically formats them — even across different object types — as a single, grammatically correct phrase.

4.2.2 Teaching cleveref about a New Object Type

For every new numbered environment you create, register its singular and plural names with \crefname (lowercase) and \Crefname (capitalised):

```
% In 0-Config/5_packages.tex, after defining the environment:
\crefname{algorithm}{algorithm}{algorithms}   % \cref{}
\Crefname{algorithm}{Algorithm}{Algorithms}  % \Cref{}

% Custom theorem environment:
\crefname{theorem}{theorem}{theorems}
\Crefname{theorem}{Theorem}{Theorems}
```

The third argument is the *plural* form used when multiple labels are listed in a single \cref{k1,k2} call.

For multilingual documents, wrap the definitions in language hooks:

```
\addto\extrasenglish{%
  \crefname{algorithm}{algorithm}{algorithms}
  \Crefname{algorithm}{Algorithm}{Algorithms}
}
```

```
\addto\extrasportuges{%
  \crefname{algorithm}{algoritmo}{algoritmos}
  \Crefname{algorithm}{Algoritmo}{Algoritmos}
}
```

4.3 Text Formatting

4.3.1 Basic Emphasis

Table 4.2: Common text formatting commands.

Command	Output	Use for
`\emph{text}`	*text*	semantic emphasis (adapts to context)
`\textbf{text}`	**text**	bold
`\textit{text}`	*text*	italic
`\texttt{text}`	`text`	monospace / code
`\textsc{text}`	TEXT	small capitals
`\underline{text}`	text	underline (use sparingly)

To emphasize some words, prefer `\emph` over `\textit`. Inside an italic context `\emph` automatically switches to upright, while `\textit` does not.

4.3.2 Lists

Unordered list (`itemize`):

```
\begin{itemize}
  \item First item
  \item Second item
  \item[$\star$] Custom bullet
\end{itemize}
```

- First item
- Second item
- ⋆ Custom bullet

Ordered list (`enumerate`):

```
\begin{enumerate}[a)]
  \item Alpha-labelled item
  \item Another item
\end{enumerate}
```

a) Alpha-labelled item

b) Another item

Description list (`description`):

```
\begin{description}
  \item[PDF] Portable Document Format
  \item[URL] Uniform Resource Locator
\end{description}
```

PDF Portable Document Format

URL Uniform Resource Locator

4.3.3 Footnotes

Footnotes[1] are created with `\footnote{text}` placed immediately after the word they annotate — no space before the command. They are numbered automatically and appear at the bottom of the page.

4.4 Equations

LaTeX excels at mathematical typesetting. Inline formulas are delimited by `\(...\)`, for example the discriminant is $)\Delta = b^2 - 4ac$.

Display-style (unnumbered) equations use `\[...\]`:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Numbered equations use the `equation` environment and are referenced with `\autoref`:

```
\begin{equation}
  E = mc^2
  \label{eq:einstein}
\end{equation}
As shown in \autoref{eq:einstein}, energy and mass are equivalent.
```

$$E = mc^2 \tag{4.1}$$

As shown in Equation 4.1, energy and mass are equivalent.

For multi-line aligned equations, use the `align` environment:

```
\begin{align}
  f(x) &= x^2 + 2x + 1 \label{eq:fx}\\
       &= (x+1)^2      \label{eq:fx_factored}
\end{align}
```

$$f(x) = x^2 + 2x + 1 \tag{4.2}$$
$$= (x + 1)^2 \tag{4.3}$$

[1] This is a simple footnote.

4.5 Tables

Professional tables follow these rules:

1. Never use vertical lines.
2. Place the caption *above* the table.
3. Use `\toprule`, `\midrule`, and `\bottomrule` from `booktabs`.
4. Use `xltabular` for tables that may span multiple pages.

Table 4.3 shows a minimal example.

```
\begin{table}[htbp]
  \caption{A simple table.}
  \label{tab:simple_table}
  \centering
  \begin{tabular}{lcc}
    \toprule
    Method   & Precision & Recall \\
    \midrule
    Baseline & 0.72      & 0.68   \\
    Proposed & 0.89      & 0.91   \\
    \bottomrule
  \end{tabular}
\end{table}
```

Table 4.3: A simple table.

Method	Precision	Recall
Baseline	0.72	0.68
Proposed	0.89	0.91

The `ShadedTabular` environment (defined in `0-Config/5_packages.tex`) provides automatic zebra striping using `colortbl`, which is already loaded by novathesis. For long tables, use `xltabular` as illustrated in Table 4.4.

Table 4.4: A multi-page table using `xltabular` with row colors.

Test	Anomalies	Warnings	Correct	Categories	Missed
Connection [3]	2	2	1	*C*	1
Coordinates'03 [1]	1	4	1	*2B*, *1C*	0
Local Variable [1]	1	2	1	*A*	0
NASA [1]	1	1	1	—	0
Coordinates'04 [2]	1	4	1	*3C*	0
Buffer [2]	0	7	0	*2A*, *1B*, *2C*, *2D*	0

Double-Check [2]	0	2	0	*1A, 1B*	0
StringBuffer [4]	1	0	0	—	1
Account [10]	1	1	1	—	0
Jigsaw [10]	1	2	1	*C*	0
Over-reporting [10]	0	2	0	*1A, 1C*	0
Under-reporting [10]	1	1	1	—	0
Allocate Vector [5]	1	2	1	*C*	0
Knight Moves [3]	1	3	1	*2B*	0
Total	12	33	10	5A, 6B, 10C, 2D	2

4.6 Figures

Store image files in `5-Figures/`. Always prefer *vector* formats (PDF) over *bitmap* formats (PNG, JPG): vector images scale without quality loss, while bitmap images become pixelated when enlarged. Compare Figure 4.1 and Figure 4.2 by *zooming in* on both.



Figure 4.1: Bitmap image (PNG/JPG): quality degrades at larger sizes.

A minimal figure:

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.8\linewidth]{my-diagram}
  \caption{A clear, descriptive caption.}
  \label{fig:my_diagram}
\end{figure}
```

The placement specifier `[htbp]` means: try **h**ere, then **t**op, then **b**ottom, then a float **p**age. Avoid `[H]` (from the `float` package) — it defeats LaTeX's float algorithm

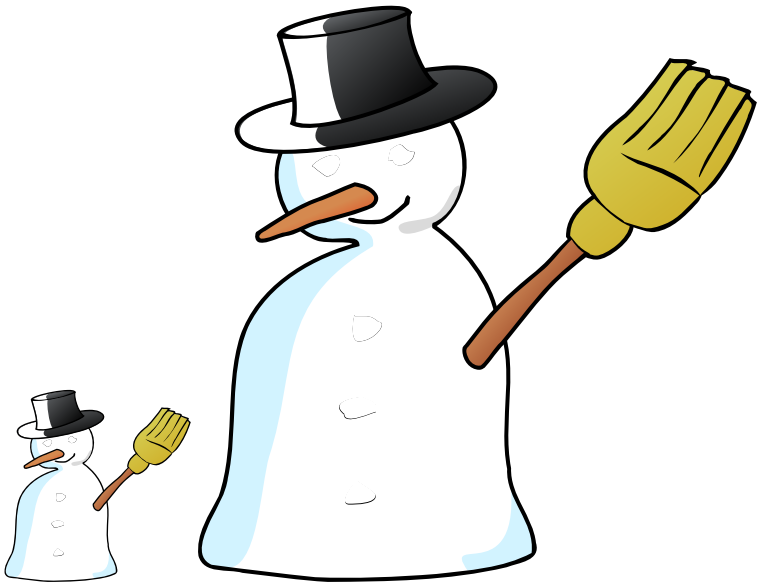


Figure 4.2: Vector image (PDF): scales perfectly at any size.

and causes large white gaps.

4.6.1 Sub-figures

Use memoir's `\subbottom` to combine multiple images in one float (Figure 4.3).

```
\begin{figure}[htbp]
  \centering
  \subbottom[Wool ball\label{fig:sub_wool}]{%
    \includegraphics[height=1in]{knitting-vectorial}}%
  \qquad
  \subbottom[Snowstorm\label{fig:sub_snow}]{%
    \includegraphics[height=1in]{snowstorm-vectorial}}%
  \caption{Two sub-figures in one float.}
  \label{fig:subfig_example}
\end{figure}
```



(a) Wool ball (b) Snowstorm

Figure 4.3: Two sub-figures in one float.

Reference the whole figure as Figure 4.3 or the sub-figures separately: Subfigure 4.3(a) and Subfigure 4.3(b).

4.7 Bibliography and Citations

The template uses `biblatex` with the `Biber` backend. Add your references to the `.bib` file(s) configured in `0-Config/2_biblatex.tex`.

4.7.1 Citation Commands

Table 4.5: Common `biblatex` citation commands.

Command	Result
`\cite{key}`	Standard citation: [1] or (Author, Year)
`\cite{k1,k2,k3}`	Multiple citations in one bracket
`\textcite{key}`	Inline: Author (Year)
`\parencite{key}`	Parenthetical: (Author, Year)
`\footcite{key}`	Citation in a footnote
`\citeauthor{key}`	Author name only
`\citeyear{key}`	Year only

Always attach the citation to the preceding word with a non-breaking space:

```
...the shunting-yard algorithm~\cite{wiki:shuntingyard} was first...
...as shown by several authors~\cite{Artho03,Artho04,Beckman08}.
```

Only references actually cited in the text appear in the bibliography list.

4.7.2 Compilation with Biber

After changing citations or the `.bib` file, a full build requires four passes:

```
lualatex template
biber   template
lualatex template
lualatex template
```

The provided `make` targets handle all of this automatically:

```
make lua    # full build: LuaLaTeX + biber
make clean  # remove auxiliary files
```

4.8 Acronyms and Glossaries

The novathesis template uses the `glossaries-extra` package with the `bib2gls` backend to manage acronyms and technical terms. Definitions are kept in `.bib` files: `1-FrontMatter/acronyms.bib`, `glossary.bib` and `symbols.bib`.

Changed in 8.0: glossary entries are now `.bib` files

Entries used to be written as LaTeX commands (`\newacronym`, `\newglossaryentry`) in `.tex` files. They are now `.bib` entries read by `bib2gls`. Upgrading a document from 7.10.x or earlier? Run `make glsbib` and see Appendix D.

Defining an Acronym:

```
@acronym{gcd,
  short = {GCD},
  long  = {Greatest Common Divisor}
}
```

Using it in text:

```
The \gls{gcd} is...  % First use: Greatest Common Divisor (GCD)
The \gls{gcd} is...  % Subsequent use: GCD
The \glspl{gcd} are... % Plural: GCDs
```

Using `\gls` ensures that acronyms are expanded on their first use and that a consistent "List of Acronyms" is generated automatically in the front matter. For instance, we can talk about AI and how it relates to Science, Technology, Engineering, and Mathematics (STEM) fields. Documents are often shared as PDF files, which may include a Uniform Resource Locator (URL) or a Digital Object Identifier (DOI) for reference.

The glossary is used for technical terms like LaTeX or the framework used in your research. You should also properly manage your bibliography.

Defining a glossary term or a symbol:

```
@entry{framework,
  name        = {framework},
  description = {A basic structure underlying a system}
}

@symbol{omega,
  name        = {\ensuremath{\omega}},
  sort        = {omega},     % ALWAYS sort symbols explicitly
  description = {the angular frequency},
  type        = {symbols}
}
```

The `sort` key deserves a word. `bib2gls` sorts on the `name` unless you say otherwise, which is fine for ordinary words but meaningless for a symbol, whose name is math markup: sorting on `\ensuremath{\omega}` puts entries in an order no reader can predict. Give every symbol — and every entry whose name is a command — an explicit `sort` key.

Using Symbols: Symbols are called using the `\gls` command. For example, the angular frequency is ω, the significance level is α, and the reduced Planck constant is $\hbar$. We often work with the set of real numbers $\mathbb{R}$ and calculate the $\sum$ of a sequence.

4.9 Indexing

novathesis provides the `\ntindex` command to simultaneously typeset and index a word:

```
This is a \ntindex{recommendation} for all users.
See the \ntindex[template!]{Version} number.
```

The first example prints "recommendation" and adds it to the index. The second example prints "Version" and adds it to the index under the parent entry "Template".

To print the index at the end of the document, ensure the following is set in `0-Config/1_novathesis.tex`:

```
\ntsetup{print/index=true}
```

For more details on index customization and debugging, see subsection 2.4.6.

4.10 Source Code Listings

The template loads the `minted` package. Three environments are pre-configured in `0-Config/5_packages.tex`:

Table 4.6: Pre-configured `minted` environments.

Environment / Command	Language	Use
`latexcode`	LaTeX	block listing
`\latexinline{}`	LaTeX	inline snippet
`bashcode`	Bash	block listing
`\bashinline{}`	Bash	inline snippet
`pythoncode`	Python	block listing
`\pythoninline{}`	Python	inline snippet

Block listing:

```
\begin{pythoncode}
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n-1) + fibonacci(n-2)
\end{pythoncode}
```

```python
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n-1) + fibonacci(n-2)
```

Inline code:

```
The function \pythoninline{fibonacci(n)} returns the $n$-th term.
```

The function `fibonacci(n)` returns the n-th term.

Captioned listings (appear in the List of Listings): Wrap any `minted` environment inside a `listing` float and add `\caption` and `\label` to register it in the list of listings.

```
\begin{listing}[htbp]
  \caption{Hello, World! in Python.}
  \label{lst:hello-python}
  \begin{pythoncode}
    print("Hello, World!")
  \end{pythoncode}
\end{listing}
```

```python
print("Hello, World!")
```

Listing 4.1: *Hello, World!* in Python.

```c
#include <stdio.h>

int main(void) {
    printf("Hello, World!\n");
    return 0;
}
```

Listing 4.2: *Hello, World!* in C.

```
\documentclass{article}
\begin{document}
  Hello, World!
\end{document}
```

Listing 4.3: *Hello, World!* in LaTeX.

Requirement: `--shell-escape`

`minted` calls the Pygments library at compile time and requires the `--shell-escape` flag. This flag is already included in the provided `Makefile`. Because this flag lets the build run external programs, only compile content you trust — see the security note in Section 2.2.

4.11 To-do Notes

During writing, annotate your draft with `\todo` notes:

```
Some text that needs more work.\todo{Add reference here}
\todo[inline]{This entire section needs revision.}
```

Noice that `\ntsetup{docstatus=final}` will suppresses them automatically all `\todo` calls.

4.12 Generating the Final PDF

The **nova**thesis template supports several LaTeX engines. For detailed instructions on how to build your document, use the `Makefile`, or generate archival-quality PDF/A output, please refer to section 2.11.

4.12.1 Algorithms

To typeset algorithms, add to `0-Config/5_packages.tex`:

```
\usepackage{algorithm2e}
\RestyleAlgo{ruled}
```

And optionally list algorithms in the front matter via `0-Config/4_files.tex`:

```
\ntaddlistof{listofalgorithms}
```

Algorithm 1: Example algorithm using `algorithm2e`.

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
    read current;
    if *understand* **then**
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;

Reference it as algorithm 1.

Bibliography

[1] C. Artho, K. Havelund, and A. Biere. "High-Level Data Races". In: *The First International Workshop on Verification and Validation of Enterprise Information Systems (VVEIS'03)*. 2003. URL: citeseer.ist.psu.edu/artho03highlevel.html (cit. on p. 38).

[2] C. Artho, K. Havelund, and A. Biere. "Using Block-Local Atomicity to Detect Stale-Value Concurrency Errors". In: *ATVA*. Ed. by F. Wang. Vol. 3299. Lecture Notes in Computer Science. Springer, 2004, pp. 150–164. ISBN: 3-540-23610-4. DOI: 10.1007/978-3-540-30473-1_11 (cit. on pp. 38, 39).

[3] N. E. Beckman, K. Bierhoff, and J. Aldrich. "Verifying Correct Usage of Atomic Blocks and Typestate". In: *SIGPLAN Not.* 43.10 (2008), pp. 227–244. ISSN: 0362-1340. DOI: 10.1145/1449955.1449783 (cit. on pp. 38, 39).

[4] C. Flanagan and S. N. Freund. "Atomizer: a dynamic atomicity checker for multithreaded programs". In: *POPL '04: Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. Venice, Italy: ACM, 2004, pp. 256–267. ISBN: 1-58113-729-X. DOI: 10.1145/964001.964023 (cit. on p. 39).

[5] *IBM's Concurrency Testing Repository*. http://www.haifa.il.ibm.com/projects/verification/contest/index.html. Link currently inactive/removed. 2007 (cit. on p. 39).

[6] J. M. Lourenço. ***nova**thesis — The Manual: A LaTeX template for theses and dissertations*. Version 8.5.0. 2026. DOI: 10.5281/zenodo.21938603 (cit. on pp. 2, 92).

[7] J. M. Lourenço. *The `aidisclose` package: Generative AI disclosure checklist and statements*. Version 1.13.0. 2025. URL: https://ctan.org/pkg/aidisclose/aidisclose-doc.pdf (cit. on pp. x, 31).

[8] Y. Suchikova, N. Tsybuliak, J. A. T. da Silva, and S. Nazarovets. "GAIDeT (Generative AI Delegation Taxonomy): A Taxonomy for Humans to Delegate Tasks to Generative Artificial Intelligence in Scientific Research and Publishing". In: *Accountability in Research* 33.3 (2026). DOI: 10.1080/08989621.2025.2544331 (cit. on pp. x, 31).

[9] The LaTeX Project team. *LaTeX – A document preparation system*. 1985. URL: https://www.latex-project.org (cit. on p. x).

[10] C. von Praun and T. R. Gross. "Static Detection of Atomicity Violations in Object-Oriented Programs". In: *Journal of Object Technology* 3.6 (2004), pp. 103–122. DOI: 10.5381/jot.2004.3.6.a5 (cit. on p. 39).

A

novathesis Macro Reference

This appendix documents the most important macros provided by the novathesis template, grouped by purpose. All signatures follow the notation used in the IntelliSense completion file `novathesis.cwl`:

- `{`⟨*arg*⟩`}` — mandatory argument
- `[`⟨*arg*⟩`]` — optional argument
- `(`⟨*arg*⟩`)` — novathesis parenthesised argument (part of the command name, not a standard LaTeX delimiter)
- `*` — starred variant with modified behaviour

A.1 Template Bootstrap

`\documentclass{novathesis}`

`\documentclass[`⟨*options*⟩`]{novathesis}`
The only document class provided by novathesis. Pass engine-level or user options as a comma-separated list in the optional argument; all template-specific options are better placed in `\ntsetup` (see below).

`\ntsetup`

`\ntsetup{`⟨*key=value,...*⟩`}`

`\ntsetup[`⟨*scope*⟩`]{`⟨*key=value,...*⟩`}`
The *master configuration command*. Every template option is set here; it may be called multiple times and the last assignment wins. Place all calls in `0-Config/1_novathesis.tex`.
The most commonly used keys are listed below.

Key	Typical values	Effect
`school`	`nova/fct`, `ulisboa/ist`, …	Selects the institutional cover & strings
`doctype`	`phd`, `msc`, `bsc`, `plain`	Document type
`lang`	`en`, `pt`, `de`, …	Main document language
`media`	`screen`, `paper`	Optimises layout for on-screen or print
`docstatus`	`working`, `provisional`, `final`	Controls watermarks and certain front-matter sections

```
\ntsetup{school=nova/fct, doctype=phd, lang=en, media=screen}
\ntsetup{docstatus=working}
```

A.2 Document Identity

`\nttitle`

`\nttitle(`⟨*type*⟩`,`⟨*lang*⟩`){`⟨*title*⟩`}`
Sets one of the document's titles. Different *type* variants are used in different locations (cover page, spine, abstract header, etc.) and can be provided in multiple languages.

type	Where used
`main`	Cover page, headers, PDF metadata
`sub`	Subtitle
`cover`	Overrides the cover title only
`abstract`	Abstract page header
`spine`	Spine label (usually a short form)

lang is any supported language code: `en`, `pt`, `de`, `es`, `fr`, `it`, `nl`, `pl`, `cz`, `dk`, `sk`, `cat`, `gr`, `uk`, `zhs`, `zht`.

```
\nttitle(main,en){Thesis Title in English}
\nttitle(main,pt){Título da Tese em Português}
\nttitle(spine,en){Short Title}
```

\ntauthorname \ntauthordegree

\ntauthorname(⟨*gender*⟩){⟨*full name*⟩}{⟨*short name*⟩}

\ntauthordegree(⟨*lang*⟩){⟨*degree*⟩}

\ntauthorname registers the author's full and short names. The *gender* argument (`m` or `f`) controls grammatical agreement in Portuguese strings.

\ntauthordegree sets the author's pre-existing academic degree for display on the cover.

```
\ntauthorname(m){João Maria Lourenço}{João Lourenço}
\ntauthordegree(en){Bachelor in Computer Science}
\ntauthordegree(pt){Licenciado em Engenharia Informática}
```

\ntdate

\ntdate(⟨*type*⟩){⟨*YYYY-MM-DD*⟩}

\SetDateISO(⟨*type*⟩){⟨*YYYY-MM-DD*⟩}

\PrintDateISO{⟨*YYYY-MM-DD*⟩}

\todayiso

\ntdate registers a document date. *type* is either `submission` (date sent to the jury) or `exam` (public defence date).

\SetDateISO / **\PrintDateISO** are lower-level helpers for ISO 8601 date handling elsewhere in the document. **\todayiso** expands to today's date in ISO format. The date string may be `YYYY-MM-DD`, `YYYY-MM`, or just `YYYY`.

```
\ntdate(submission){2025-09-30}
\ntdate(exam){2025-11-14}
```

A.3 Institutional Metadata

\ntdepartment \ntdegreename \ntspecialization

\ntdepartment(⟨*lang*⟩){⟨*name*⟩}

\ntdepartment*(⟨*lang*⟩){⟨*name*⟩}

\ntdegreename(⟨*lang*⟩){⟨*name*⟩}

\ntdegreename*(⟨*lang*⟩){⟨*name*⟩}

\ntspecialization(⟨*lang*⟩){⟨*name*⟩}

\ntspecialization*(⟨*lang*⟩){⟨*name*⟩}

Override the school-default department name, degree name, or specialisation string. Provide values for every language used in the document (at minimum the document language and Portuguese for national universities). The starred form sets an abbreviated variant used on the spine and narrow covers.

```
\ntdepartment(en){Department of Computer Science}
\ntdepartment(pt){Departamento de Informática}
\ntdegreename(en){Doctor of Philosophy in Computer Science}
\ntdegreename(pt){Doutor em Engenharia Informática}
```

\ntsponsors

\ntsponsors(⟨*order*⟩,⟨*type*⟩,⟨*lang*⟩){⟨*content*⟩}
Registers a funding sponsor for display on the cover. *order* is a positive integer that controls display order. *type* is either `text` (a string) or `logo` (an `\includegraphics` call or similar).

```
\ntsponsors(1,logo,en){\includegraphics[height=1cm]{fct-logo}}
\ntsponsors(2,text,en){Supported by FCT grant XXXXXX/XX/XXXXX}
```

A.4 People — Advisers and Committee

\ntaddperson

\ntaddperson{⟨*class*⟩}(⟨*role*⟩,⟨*gender*⟩){⟨*name*⟩}[⟨*title=..., position=..., institution=...*⟩]
Registers a person (adviser or jury member). Only ⟨*name*⟩ is required; `title`, `position`, and `institution` are independent optional fields, given as `key=value` pairs in the trailing bracket. All fields are typeset verbatim on the cover and in the jury table.

class	role	Meaning
`adviser`	`a`	Principal adviser
	`c` / `co`	Co-adviser
	`t`	Tutor
`committee`	`c`	Chair
	`r`	Rapporteur
	`a`	Adviser
	`co`	Co-adviser
	`m`	Member
	`g`	Guest

The *gender* argument (`m` / `f`) governs grammatical agreement in Portuguese texts.

```
\ntaddperson{adviser}(a,m){João Maria Lourenço}%
  [position=Full Professor,
```

```
   institution=NOVA School of Science and Technology, UNL]
\ntaddperson{committee}(c,m){Maria Fernanda Távora}%
  [title=Prof.,
   position=Full Professor,
   institution=Instituto Superior Técnico, ULisboa]
```

Upgrading from an older document

Earlier novathesis versions took a single `{name, position, institution}` comma string instead of the bracket above – that form still works (only the first two commas are significant; a comma-containing institution like "*NOVA School of Science and Technology, UNL*" is kept whole), but now emits a deprecation warning (once per `\ntaddperson` call) suggesting the `title/position/institution` form instead. There is no rush to convert existing documents.

`\ntprintpersons`

`\ntprintpersons{`⟨*class*⟩`}{`⟨*roles*⟩`}`

`\GetPersonsAsList{`⟨*class*⟩`}{`⟨*roles*⟩`}[`⟨*separator*⟩`]`

`\ntprintpersons` typesets a formatted table of persons filtered by class and role(s). Use it inside cover or front-matter files to display the jury or adviser list.

`\GetPersonsAsList` returns the same persons as a flat list with an optional custom separator.

```
\ntprintpersons{adviser}{a,co}
\ntprintpersons{committee}{c,r,m}
```

`\ntforeachperson`

`\ntforeachperson{`⟨*class*⟩`}{`⟨*role*⟩`}{`⟨*macro*⟩`}`

`\ntpersoncount{`⟨*class*⟩`}{`⟨*role*⟩`}`

`\ntpersongender{`⟨*class*⟩`}{`⟨*role*⟩`}{`⟨*result-macro*⟩`}`

Lower-level accessors for custom cover pages or school `.clo` files that need person data directly, without going through `\ntprintpersons`.

`\ntforeachperson` calls ⟨*macro*⟩`{title}{name}{position}{institution}{gender}` once for every person registered under ⟨*class*⟩/⟨*role*⟩, in the order they were added with `\ntaddperson`. Any field a given person did not set comes through as an empty group.

`\ntpersoncount` expands to the number of people registered (`0` if none) – useful for choosing between a singular and plural label.

\ntpersongender sets ⟨*result-macro*⟩ to `m` if at least one registered person is male, otherwise `f` – the same "any male makes the group masculine" rule novathesis uses for Portuguese grammatical agreement on group labels like "Supervisor(s)".

```
\newcommand{\printshortname}#1#2#3#4#5{#2\\}
\ntforeachperson{committee}{m}{\printshortname}
```

A.5 File Registration — \ntaddfile

\ntaddfile

\ntaddfile{⟨*class*⟩}{⟨*filename*⟩}

\ntaddfile{⟨*class*⟩}[⟨*key*⟩]{⟨*filename*⟩}

The single command used to register *every* user-provided file with the template. Files are rendered later by the corresponding \ntprint* call in `template.tex`. The *filename* must be given without extension; the `.tex` suffix is appended automatically.

Multiple calls accumulate; files are printed in registration order.

class	key	Notes
`bib`	—	BibTeX/Biber `.bib` database
`abstract`	language code (`en`, `pt`, …)	One per language; key selects the language
`glossaries`	`glossary` | `acronym` | `symbols`	Glossary-type files loaded by `glossaries-extra`
`chapter`	optional doctype list	Main-matter chapters; key restricts to listed doctypes
`appendix`	optional doctype list	Back-matter appendices
`annex`	optional doctype list	Back-matter annexes
`cover`	`1` | `2` | *N* | `spine`	Overrides the built-in cover layout for slot *N*
`dedication`	—	
`acknowledgements`	—	
`quote`	—	
`statement`	language code	Signed statement; key selects the language

```
\ntaddfile{bib}{bibliography}
\ntaddfile{abstract}[en]{abstract-en}
\ntaddfile{abstract}[pt]{abstract-pt}
```

```
\ntaddfile{glossaries}[acronym]{acronyms}
\ntaddfile{chapter}{chapter-introduction}
\ntaddfile{chapter}[phd,msc]{chapter-advanced}% only in PhD and MSc
\ntaddfile{appendix}{app-build}
\ntaddfile{annex}{annex-data}
```

A.6 Content Rendering

\ntprint \ntprintlist \ntprintindex

\ntprint{⟨*key*⟩}

\ntprintlist{⟨*key*⟩}

\ntprintindex

High-level rendering commands used exclusively in `template.tex`. **\ntprint** outputs a single named section; **\ntprintlist** outputs all items registered under the given key (e.g., all chapters). **\ntprintindex** prints the word index if \ntsetup{print/index=true} is set.

These abstract away the complex logic of conditionally printing front-matter elements, list-of-figures, etc. depending on the current `docstatus` and `doctype`.

Common `key` values include: `frontcover`, `frontmatter`, `mainmatter`, `backmatter`, `backcover`.

\ntaddlistof

\ntaddlistof{⟨*listof*⟩}

\ntaddlistof[⟨*matter*⟩]{⟨*listof*⟩}

Registers a "List of …" to be printed by \ntprint{frontmatter} or \ntprint{backmatter}. The optional ⟨*matter*⟩ argument is either `frontmatter` (default) or `backmatter`, and controls where the list is typeset. ⟨*listof*⟩ is the internal identifier of the list (e.g. `figures`, `tables`, `listings`).

```
% Print list of figures in front matter (default)
\ntaddlistof{figures}
% Print list of listings in back matter
\ntaddlistof[backmatter]{listings}
```

A.7 Cover Customisation

\ntaddtocover

\ntaddtocover{⟨*cover-ids*⟩}{⟨*content*⟩}

\ntaddtocover[⟨*scope*⟩]{⟨*cover-ids*⟩}{⟨*content*⟩}

Appends *content* to one or more cover slots identified by *cover-ids* (a comma-separated list of slot identifiers such as `1`, `2`, `spine`, `N`). Use the optional *scope* to restrict the addition to a particular school or document type context.

```
\ntaddtocover{1,2}{\textbf{CONFIDENTIAL} - not for distribution}
```

\ntAddToCoverTikz

\ntAddToCoverTikz(⟨*layer*⟩){⟨*cover-ids*⟩}{⟨*tikz-code*⟩}

\ntAddToCoverTikz<⟨*label*⟩>[⟨*tikz-options*⟩](⟨*layer*⟩)[⟨*more-opts*⟩]{⟨*cover-ids*⟩}{⟨*tikz-code*⟩}

Injects raw TikZ drawing code onto the cover canvas at the specified *layer* (e.g. `bg`, `fg`, `text`). Coordinates use the full cover dimensions. Use this for advanced cover layouts in school `.clo` files.

```
\ntAddToCoverTikz(bg){1}{%
  \fill[blue!20] (0,0) rectangle (\paperwidth,3cm);
}
```

A.8 Document Conditionals

\ifphddoc, \ifmscdoc, …

\ifdocclass{⟨*class*⟩}{⟨*true*⟩}{⟨*false*⟩}

\ifphddoc{⟨*true*⟩}{⟨*false*⟩}

\ifmscdoc{⟨*true*⟩}{⟨*false*⟩}

\ifbscdoc{⟨*true*⟩}{⟨*false*⟩}

\ifplaindoc{⟨*true*⟩}{⟨*false*⟩}

\ifdocstatus{⟨*status*⟩}{⟨*true*⟩}{⟨*false*⟩}

\ifdocfinal{⟨*true*⟩}{⟨*false*⟩}

\ifdocprovisional{⟨*true*⟩}{⟨*false*⟩}

\ifdocworking{⟨*true*⟩}{⟨*false*⟩}

\ifcoverdefined{⟨*cover-id*⟩}{⟨*true*⟩}{⟨*false*⟩}

\ifxeorlua{⟨*xe/lua code*⟩}{⟨*pdflatex code*⟩}

A family of Boolean predicates for conditional content. All follow the same two-argument form: *true code* and *false code*. The *false* argument may be omitted (empty

by default).

```
% Acknowledgements only in the final version
\ifdocfinal{%
  \ntaddfile{acknowledgements}{acknowledgements}%
}

% PhD-specific chapter
\ifphddoc{%
  \ntaddfile{chapter}[phd]{chapter-related-work}%
}

% Engine-dependent font choice
\ifxeorlua{%
  \setmainfont{TeX Gyre Pagella}%
}{%
  \usepackage{newpxtext}%
}
```

A.9 Hook System

\NTAddToHook \NTDeclareHook \NTRunHook

\NTDeclareHook{⟨*hook-name*⟩}

\NTAddToHook{⟨*hook-name*⟩}{⟨*code*⟩}

\NTRunHook{⟨*hook-name*⟩}

novathesis exposes a lightweight hook system used primarily in school `.clo` files to inject code at well-defined points (cover rendering, font loading, language setup, etc.) without modifying the class source.

Hooks are declared once with **\NTDeclareHook**, populated with **\NTAddToHook** (can be called multiple times — code accumulates), and executed at the right moment with **\NTRunHook**.

```
% In a school .clo file:
\NTAddToHook{cover/before}{\MySchoolSetupCode}
\NTAddToHook{fonts/load}{\setmainfont{TeX Gyre Pagella}}
```

A.10 Miscellaneous Utilities

\shiftbookmarks

\shiftbookmarks{⟨*content*⟩}

\shiftbookmarks[⟨*levels*⟩]{⟨*content*⟩}

Shifts all PDF bookmark levels inside *content* by the given number of *levels* (default: 1). Used at the start of back matter to nest appendix and annex bookmarks one level below the "Back Matter" root bookmark.

```
\shiftbookmarks[1]{\ntprintbibsingle \ntprintappendices \ntprintannexes}
```

\ntselectlang

\ntselectlang{⟨*lang-code*⟩}

Switches the active Babel / Polyglossia language mid-document and updates all novathesis localised strings accordingly. Used internally when rendering multi-language abstracts; available for advanced users who need language changes inside chapters.

```
\ntselectlang{pt}
\noindent\textbf{Resumo} - \lipsum[1]
\ntselectlang{en}
```

\debuggrid

\debuggrid

\debuggrid[⟨*color*⟩]

Overlays a grid on the current page to aid layout debugging. Default colour is `red!30`. Wrap in a local group or remove before final submission.

```
\debuggrid[blue!20]   % draw a blue grid on the current page
```

\ntindex

\ntindex{⟨*word*⟩}

\ntindex[⟨*key*⟩]{⟨*word*⟩}

Typesets ⟨*word*⟩ in the document and adds it to the index. If ⟨*key*⟩ is provided, it is used as the index entry while ⟨*word*⟩ remains the visible text. If ⟨*key*⟩ ends with `!`, it is treated as a parent entry and ⟨*word*⟩ is added as a sub-entry.

```
This is a \ntindex{recommendation}.
See the \ntindex[template!]{Version} number.
```

\novathesis

\novathesis
Typesets the template name **nova**thesis in the canonical stylised form. Use it whenever referring to the template inside your document.

```
This document was prepared using the \novathesis{} template.
```

A.11 Summary Table

The table below lists all documented macros with a short description and the configuration file where they are typically used.

Macro	Purpose	Typical location
\documentclass {novathesis}	Load the template class	template.tex
\ntsetup	Master document configuration	0-Config/1_novathesis.tex
\nttitle	Set title (type + language)	0-Config/3_cover.tex
\ntauthorname	Author full and short name	0-Config/3_cover.tex
\ntauthordegree	Author's pre-existing degree	0-Config/3_cover.tex
\ntdate	Submission / exam date	0-Config/3_cover.tex
\ntdepartment	Override department name	0-Config/3_cover.tex
\ntdegreename	Override degree name	0-Config/3_cover.tex
\ntspecialization	Override specialisation name	0-Config/3_cover.tex
\ntsponsors	Funding sponsor (text or logo)	0-Config/3_cover.tex
\ntaddperson	Register adviser or jury member	0-Config/3_cover.tex
\ntprintpersons	Typeset adviser/jury table	Cover .tex files
\ntaddfile	Register any user file with the template	0-Config/4_files.tex
\ntprint	Print one named document section	*Internal*
\ntprintlist	Print all files in a registered list	Internal
\ntprintindex	Print the word index	print/index
\ntaddlistof	Register a "List of …"; optional arg frontmatter (default) or backmatter sets where it prints	0-Config/6_list_of.tex

(continued on next page)

Macro	Purpose	Typical location
`\ntaddtocover`	Append content to cover slot(s)	`School .clo files`
`\ntAddToCoverTikz`	Inject Ti*k*Z code onto cover canvas	`School .clo files`
`\ntindex`	Typeset and index a word	`Chapters`
`\ifphddoc` / `\ifmscdoc`	Conditional on document type	`0-Config/4_files.tex, chapters`
`\ifdocfinal`	Conditional on document status	`0-Config/4_files.tex, chapters`
`\ifxeorlua`	Conditional on LaTeX engine	`0-Config/5_packages.tex`
`\NTDeclareHook`	Declare an extension hook	`School .clo files`
`\NTAddToHook`	Add code to an extension hook	`School .clo files`
`\NTRunHook`	Execute an extension hook	`Class internals`
`\shiftbookmarks`	Shift PDF bookmark levels	`nt-render-matter.sty`
`\ntselectlang`	Switch active language mid-document	`Abstract files, chapters`
`\debuggrid`	Overlay layout debug grid	`During development`
`\novathesis`	Template name in canonical style	`Anywhere in text`

B

The novathesis Cover Page System

This chapter documents the cover page engine used by the `novathesis` class. It is written for developers who need to add cover support for a new institution, or who want to understand or refactor an existing cover definition. End users who only want to configure metadata (title, author, committee) should work exclusively with `0-Config/3_cover.tex`.

The cover engine is implemented in:

- `novathesisFiles/StyFiles/nt-covers.sty` — a modern LaTeX3/expl3 implementation that provides a high-level declarative Application Programming Interface (API) for cover design.

The remainder of this chapter describes that API.

B.1 Architecture Overview

Every cover page is rendered through three independent layers applied in order:

1. **Background colour.** A solid `\pagecolor{}` fills the entire page. Set with:

   ```
   \thesiscover(id,bgcolor):={colorname}
   ```

2. **Background image.** A full-bleed graphic is stretched to `\paperwidth` × `\paperheight`. Set with `\thesiscover(id,image):={filename}`.

3. **Content layer.** A `tikzpicture` overlay anchors a large `minipage` at the top-left of the text area (offset by the cover margins). Inside this minipage, elements registered with `\ntaddtocover` are typeset top-to-bottom. TikZ elements registered with `tikz=true` or `\ntAddToCoverTikz` are collected and rendered in a *second* overlay `tikzpicture` drawn after the minipage, so they escape the minipage's coordinate space.

The cover text area is defined by four cover-specific margins (see Section B.3.1). All `\ntaddtocover` elements are stacked *vertically* inside this area, in the order they were registered.

B.2 Cover Identifiers

Each cover page has a two-part identifier `major-minor`:

ID	Description	When printed
`1-1`	Front cover — recto (right/odd page)	Always
`1-2`	Front cover — verso (back of cover)	When content is registered
`2-1`	Front page — recto	When `print/frontpage=true`
`2-2`	Front page — verso	When content is registered
`N-1`	Back cover — recto (reverse of back)	When content is registered
`N-2`	Back cover — verso (outside back cover)	Always

The *major* number (1, 2, N) identifies the cover pair; the *minor* number (1, 2) selects the recto or verso page within the pair. A comma-separated list of IDs (e.g. `1-1,2-1`) can be passed to any command that accepts cover IDs.

B.3 Setting Up the Cover Canvas

B.3.1 Cover-Specific Margins

Margins for the text area (the minipage placed on the content layer) are separate from the normal page margins. They are set with the `\margin(cover,...):={...}` family:

```
\SetMargin*(cover,top)={8.6cm}    % default top margin for all cover pages
\SetMargin*(cover,bottom)={1.5cm}
\SetMargin*(cover,left)={3.55cm}
\SetMargin*(cover,right)={2.0cm}
\SetMargin*(cover,2-1,top)={6.6cm} % override top margin for cover 2-1 only
```

The cover-ID-specific variant takes precedence when printing that particular page. Only the text area is affected; background images always fill the entire paper.

B.3.2 Background Colour and Image

These properties are stored in the `thesiscover` data family and retrieved through a prioritised lookup that checks the current document type and degree class before falling back to the generic key:

```
% Set a solid background colour for cover 1-1
\SetThesisCover*(1-1,bgcolor)={fctdarkblue}

% Per-doctype variant (overrides the generic key when doctype=phd)
\SetThesisCover*(phd,1-1,bgcolor)={fctdarkgreen}

% Set a background image (file must be findable by \hologo{LaTeX}/graphicx)
```

```
\SetThesisCover*(phd,1-1,image)={nova-fct-cover-green-front}

% Set the default text colour for the entire cover page
\SetThesisCover*(1-1,textcolor)={white}
```

Supported keys:

- `bgcolor` — name of a colour defined with `\definecolor`.
- `image` — file name (no extension; passed to `\includegraphics`). The image is automatically scaled to `\paperwidth` × `\paperheight`.
- `textcolor` — sets `\color{...}` for the entire content layer of that cover page.

The lookup cascade used internally is:

```
\@DOCTYPE, \@DEGREEACR, cover-id, key
\@DOCCLASS, \@DEGREEACR, cover-id, key
\@DEGREEACR, cover-id, key
\@DEGREEACR, key
\@DOCTYPE, cover-id, key
\@DOCCLASS, cover-id, key
cover-id, key
key
```

The first matching entry wins, so you can easily have a generic fallback with targeted per-doctype overrides.

B.4 The `\ntaddtocover` Command

`\ntaddtocover` is the primary interface for populating a cover page. It registers a *content block* — either a vertical box or a TikZ code snippet — for one or more cover pages.

Syntax:

```
\ntaddtocover[<attributes>]{<cover-id-list>}{<code>}
```

`[attributes]` Optional comma-separated key=value list. See Section B.4.1.

`{cover-id-list}` Comma-separated list of cover IDs (e.g. `1-1,2-1`) to which this element is appended.

`{code}` The content to typeset. Evaluated inside a `minipage` (unless `tikz=true`; see Section B.4.3).

Elements are rendered in the order they are registered. The first call produces the topmost block.

Minimal example — adding a title block:

```
\ntaddtocover[halign=l, height=3cm, valign=c]{1-1}{%
  \fontsize{20}{21}\selectfont
  \textbf{\GetDocTitle(\@LANG@COVER,main)}%
}
```

B.4.1 Attribute Reference

All attributes are optional. Defaults are shown in parentheses.

Element-level parameters (`\ntaddtocover[options]{covers}{content}`):

Attribute	Values	Default	Description
`status`	`working`, `provisional`, `final`	(all)	Filter by doc status
`degree`	`phd`, `msc`, `bsc`, …	(all)	Filter by degree type
`width`	*dim*	text width	Box width
`height`	*dim*	natural	Fixed box height
`height/phd`, `height/msc`	*dim*	—	Per-doctype height override
`minheight`	*dim*	`0pt`	Minimum box height
`hspace`	*dim*	`0pt`	Horizontal offset from left margin
`halign`	`l`, `c`, `r`, `j`	`c`	Horizontal content alignment
`valign`	`t`, `c`, `b`	`c`	Vertical content alignment (needs `height`)
`color`	*colorname*	(page default)	Foreground text colour
`bgcolor`	*colorname*	transparent	Box background colour
`vspace`	*dim*	`0pt`	Fixed vertical gap *before* the element
`vfill`	*integer*	—	Elastic gap before element (*n*×`fill`)
`tikz`	`true`, `false`	`false`	Treat content as raw TikZ overlay code
`newpar`	`true`, `false`	`true`	Emit `\par` after element

Filter attributes (the element is silently skipped when the filter does not match the current document):

`status={...}` Comma-separated list of document statuses for which this element should appear. Valid values: `working`, `provisional`, `final`. Empty (default) means *always*.

`degree={...}` Comma-separated list of document types for which this element should appear. Valid values: `phd`, `msc`, `bsc`, `plain`, `phdplan`, `phdprop`, `mscplan`. Empty (default) means *always*.

Box geometry attributes:

`width={<dim>}` Width of the box (default: full text area width minus `hspace`). Any valid TEX dimension.

`height={<dim>}` Fixed height of the box. When omitted, the box height matches its natural content height. Using a fixed height together with `valign` lets you vertically place content inside a reserved area.

`height/phd={<dim>}` Same as `height`, but only applied when the document type is PhD. Overrides `height` for PhD documents.

`height/msc={<dim>}` Same, but for MSc documents.

`minheight={<dim>}` Minimum height of the box, even when the content is shorter (default: `0pt`).

`hspace={<dim>}` Horizontal offset from the left margin (default: `0pt`). Useful for indenting elements that must not start at the margin.

Alignment attributes:

`halign=c|l|r|j` Horizontal alignment of content inside the box (default: `c`). Values: `c` = `\centering`, `l` = `\raggedright`, `r` = `\raggedleft`, `j` = justified (no alignment command).

`valign=c|t|b` Vertical alignment of content inside a fixed-height box (default: `c`). Ignored when no `height` is given.

Colour attributes:

`color={<colorname>}` Foreground text colour inside the box. Overrides `textcolor` from `\thesiscover`.

`bgcolor={<colorname>}` Background colour of the box (applied via `\colorbox`). Default: transparent.

Vertical spacing attributes:

Two attributes control the vertical gap *before* the element. Use `vspace` for an absolute dimension, or `vfill` for elastic (stretchable) spacing:

`vspace={<dim>}` Insert a fixed `\vspace*{<dim>}` above the element.

`vfill={<n>}` Insert an elastic `\vskip 0pt plus <n>fill` before the element. This creates a stretchable gap — a single `vfill=1` between two groups of elements pushes one group to the top and the other to the bottom. Two elements each with `vfill=1` distribute remaining space evenly.

`vspace/phd`, `vspace/msc` Per-doctype override of `vspace`. Takes precedence over the generic `vspace` when the document type matches.

`vspace/1-1`, `vspace/2-1` Per-cover-ID override. Takes precedence when the element is being rendered on that specific cover.

vspace/phd/1-1, vspace/msc/1-1, etc. Combined per-doctype per-cover override. Highest precedence.

Layout control attributes:

newpar=true|false Emit a \par after the element (default: true). Set newpar=false to place the next element on the same line.

tikz=true|false Treat the content block as raw TikZ code to be injected into the overlay TikZ picture rather than typeset inside the minipage (default: false). See Section B.4.3.

B.4.2 Stacking Elements Vertically

The most common cover layout is a vertical stack of boxes, similar to a minipage column. Each \ntaddtocover call appends one box to the column. The following design rules apply:

1. The element appears immediately after the previous one (zero gap by default).
2. Use vspace={<dim>} for a fixed gap and vfill={<n>} (integer) for a stretchable gap.
3. A single vfill=1 between two groups of elements pushes one group to the top and the other to the bottom. Use equal integer weights (e.g. two elements with vfill=1) for equal distribution.
4. Use height to reserve vertical space even when content may vary in size (e.g. optional subtitle lines). Combine with valign=c to centre the content vertically inside the reserved area.

Example — logo, title, elastic gap, date:

```
% Logo at the top of the text area
\ntaddtocover[halign=l]{1-1,2-1}{%
  \includegraphics[height=1.5cm]{my-school-logo}%
}

% Title - reserved box of fixed height so subtitle does not shift the author
\ntaddtocover[vspace=5mm, halign=l, height=4cm, valign=c]{1-1,2-1}{%
  \fontsize{18}{20}\selectfont
  \textbf{\GetDocTitle(\@LANG@COVER,main)}%
}

% Author
\ntaddtocover[halign=l]{1-1,2-1}{%
  \fontsize{14}{14}\selectfont
  \GetDocAuthor(name)%
```

```
}

% Elastic gap pushes the date to the bottom of the text area
\ntaddtocover[vfill=1, halign=l]{1-1,2-1}{%
  \fontsize{11}{11}\selectfont
  \PrintDateISO{\GetDocDate(submission,year)-\GetDocDate(submission,month)}%
}
```

Conditional content per document type:

Use the `degree` filter or the internal `\ifphddoc`/`\ifmscdoc` helpers when the cover layout differs between PhD and MSc documents:

```
\ntaddtocover[halign=l, height=3cm, valign=c]{1-1}{%
  \ifphddoc{%
    \fontsize{20}{21}\selectfont
    \textbf{\GetDocTitle(\@LANG@COVER,main)}%
  }{%
    \fontsize{17}{17}\selectfont
    \textbf{\GetDocAuthor(name)}%
  }%
}
```

Alternatively, register two separate elements with complementary `degree` filters and identical heights so they occupy the same space:

```
\ntaddtocover[degree=phd, height=3cm, valign=c]{1-1}{%
  \fontsize{20}{21}\selectfont \textbf{\GetDocTitle(\@LANG@COVER,main)}%
}
\ntaddtocover[degree=msc, height=3cm, valign=c]{1-1}{%
  \fontsize{17}{17}\selectfont \textbf{\GetDocAuthor(name)}%
}
```

B.4.3 Embedding TikZ Code via `tikz=true`

Setting `tikz=true` in the attribute list causes the content block to be collected as *overlay TikZ code* rather than box content. The block is executed as a `tikzpicture[remember picture, overlay]` covering the whole page, drawn after the text-layer minipage has been placed.

This means the coordinate origin is the `current page` node, not the text area, so you can position elements anywhere on the page with absolute coordinates. The `calc` TikZ library (`\usetikzlibrary{calc}`) is loaded by default and can be used freely.

```
% Place the university logo at the top-left corner (absolute coordinates)
\ntaddtocover[tikz]{1-1}{%
  \node[anchor=center]
    at ($(current page.north west) + (20mm,-20mm)$)
    {\includegraphics[width=21mm]{\GetUniversity(logo,neg)}};%
  % Vertical rule separator
  \draw[thick]
```

```
    ($(current page.north west) + (3.5cm,-1cm)$) -- ++(0,-2cm);%
}
```

All `tikz=true` blocks registered for the same cover are concatenated into a single `tikzpicture`, in registration order.

When to use `tikz=true` vs. plain box elements:

- Use plain box elements for text content whose position depends on the height of other elements (titles, author names, committee lists). The vertical-stacking model handles reflow automatically.
- Use `tikz=true` for graphics or decorations that need to be anchored at a fixed absolute position (logos at a corner, background watermarks, ruled lines).

B.5 The `\ntAddToCoverTikz` Command

`\ntAddToCoverTikz` is a convenience command for registering individual TikZ nodes or drawing commands. It mirrors the syntax of TikZ's own node and draw commands, inserting them into the same overlay picture as `\ntaddtocover[tikz]`.

Syntax:

```
\ntAddToCoverTikz<cmd>[opts](position)[rest]{cover-id-list}{code}
```

`<cmd>` Optional: a TikZ command other than `\node` (use angle brackets, e.g. `<\draw>`). When omitted, the command defaults to `\node`.

`[opts]` Optional TikZ options for the node or command (square brackets, default empty).

`(position)` Required: position of the node or command, in the TikZ coordinate system relative to `current page` (parentheses).

`[rest]` Optional: additional TikZ node syntax placed between the position and the final `{code}` argument (square brackets, default empty).

`{cover-id-list}` Comma-separated list of cover IDs.

`{code}` Node content (when using the default `\node` command) or the draw-path code (when using a custom command).

How the arguments map to TikZ output:

When no command is specified, the generated TikZ is:

```
\node[opts] at (position) rest {code};
```

When a command `<\draw>` is specified:

```
\draw[opts] (position) rest;
```

(note: `code` is ignored in command mode)

Examples:

```
% Default \node - place a logo
\ntAddToCoverTikz
  [anchor=north east]
  (current page.north east)
  {1-1,2-1}
  {\includegraphics[width=4cm]{my-logo}}

% Custom command: \draw
\ntAddToCoverTikz<\draw>
  [thick, white]
  ($(current page.north west) + (1cm,-4cm)$)
  [-- ++(\paperwidth-2cm, 0)]
  {1-1,2-1}
  {}
```

Note on `yscale=-1`. The overlay picture for `\ntAddToCoverTikz` uses `yscale=-1`, so the y-axis points *downward*. Coordinates like `(current page.north west) + (x,-y)` move right and *down* from the top-left corner. The plain `\ntaddtocover[tikz]` blocks use a *standard* overlay without `yscale`, so their coordinate conventions differ.

B.6 Cover Hooks

Hooks let you execute code at specific points in the cover rendering pipeline without modifying the `\ntaddtocover` element list. They are registered with `\NTAddToHook`:

```
\NTAddToHook{cover/1-1/pre}{\color{white}}
\NTAddToHook{cover/text/pre}{\sffamily}
\NTAddToHook{cover/post}{\pagecolor{white}} % reset after each cover
```

Available hook points (in execution order for a single cover page):

Hook name	When executed
`cover/pre`	Before any cover page
`cover/<id>/pre`	Before cover `<id>`
`cover/<id>/text/pre`	Before text-layer elements
`cover/<id>/text/post`	After text-layer elements
`cover/<id>/bgcolor/pre`	Before background colour is applied
`cover/<id>/bgcolor/post`	After background colour is applied
`cover/<id>/image/pre`	Before background image is rendered
`cover/<id>/image/post`	After background image is rendered
`cover/<id>/grid/pre`	Before debug grid (only in debug mode)
`cover/<id>/grid/post`	After debug grid
`cover/<id>/post`	After cover `<id>`
`cover/post`	After any cover page

Typical use cases:

- Switch to a sans-serif font or set a global colour for the whole cover page inside the `cover/<id>/pre` hook.
- Reset font or colour in `cover/post` to avoid bleeding into the rest of the document.
- Load cover-specific fonts (e.g. via `\setmainfont` for LuaLaTeX/XeLaTeX) in `cover/pre`.

B.7 Debugging a Cover Layout

Enable debug mode in `\ntsetup` in `0-Config/1_novathesis.tex`:

```
\ntsetup{debug={cover,spine}}
```

With `cover` in the debug set, the engine automatically calls `\debuggrid` over every cover page and draws thin frames around each text box placed by `\ntaddtocover`.

You can also call `\debuggrid` manually inside a cover element:

```
\ntaddtocover{1-1}{\debuggrid[cyan]}
```

The grid draws:

- 1 mm ticks (lightest),
- 5 mm ticks,
- 1 cm ticks,

- 5 cm ticks (darkest).

All lines respect the cover margins, so the grid origin is the top-left corner of the text area.

A fast iteration workflow:

```
# Rebuild automatically on every save, for the school being tuned
make watch NT="school=nova/fct"
```

The `NT` override selects the school without editing any configuration file, and `make watch` rebuilds on every save, giving fast feedback when tuning positions and margins.

B.8 Cover Rendering: Execution Order Summary

The following sequence describes what happens when `\ntprintcoversingle{1}{1}` (cover `1-1`) is called:

1. A fresh group is opened; `cover/pre` and `cover/1-1/pre` hooks are run.
2. A PDF bookmark is inserted.
3. `thispagestyleempty` suppresses headers and footers.
4. Background colour lookup: the engine searches the `thesiscover` data store for a matching `bgcolor` key and calls `\pagecolor`.
5. Background image lookup: if an `image` key is found, a full-page `tikzpicture` overlay renders it.
6. `cover/1-1/text/pre` hook.
7. Cover margins are read; a `tikzpicture` overlay anchors a `minipage` at the page's top-left, offset by those margins.
8. Inside the minipage, each element registered with `\ntaddtocover[…]{1-1}` is processed in order. For each element the engine: resolves `vspace`, typeset the content into a box, applies `bgcolor` wrapping, and emits the box.
9. `tikz=true` elements accumulated during step 8 are emitted in a second overlay `tikzpicture`.
10. `cover/1-1/text/post`, `cover/1-1/post`, and `cover/post` hooks are run.
11. The group is closed.

B.9 Step-by-Step: Adding Cover Support for a New School

This section walks through adding a complete cover definition. Substitute `myuniv` and `myfac` with the institution identifiers.

1. Create the directory tree:

```
novathesisFiles/Schools/myuniv/myfac/
novathesisFiles/Schools/myuniv/myfac/Images/
```

2. Create the defaults file `myuniv-myfac-defaults.clo`:

```
%!TEX root=../../../template.tex
%% myuniv/myfac/myuniv-myfac-defaults.clo

\typeout{NT FILE schools/myuniv/myfac/myuniv-myfac-defaults.clo}%


%%=========================================================
%% COLOURS
%%=========================================================

\definecolor{myfacblue}{RGB}{0, 64, 128}
\definecolor{myfacgold}{RGB}{190, 155, 80}


%%=========================================================
%% INSTITUTION STRINGS  (one line per supported language)
%%=========================================================

\SetUniversity*(en)={My University}
\SetUniversity*(pt)={Minha Universidade}
\SetSchool*(en)={My Faculty}
\SetSchool*(pt)={Minha Faculdade}

\SetSchool*(logo,en)={myuniv-myfac-logo-en}
\SetSchool*(logo,pt)={myuniv-myfac-logo-pt}
\IfMemVoidTF{School}{logo,\@LANG@COVER}
  {\SetSchool*(logo)={\GetSchool(logo,en)}}
  {\SetSchool*(logo)={\GetSchool(logo,\@LANG@COVER)}}


%%=========================================================
%% COMMITTEE ORDER
%%=========================================================

\SetCommitteeOrder*()={c,r,a,ca,m}%
```

```
%%=========================================================
%% LAYOUT
%%=========================================================

\SetMargin*(cover,top)={4.0cm}
\SetMargin*(cover,bottom)={2.0cm}
\SetMargin*(cover,left)={2.5cm}
\SetMargin*(cover,right)={2.5cm}


%%=========================================================
%% COVER BACKGROUNDS
%%=========================================================

\SetThesisCover*(1-1,bgcolor)={myfacblue}
\SetThesisCover*(1-1,textcolor)={white}
\SetThesisCover*(2-1,bgcolor)={white}
\SetThesisCover*(N-2,bgcolor)={myfacblue}


%%=========================================================
%% COVER ELEMENTS - cover 1-1 (front cover)
%%=========================================================

% Logo - absolutely positioned in the top-right area
\ntaddtocover[tikz]{1-1}{%
  \node[anchor=north east]
    at ($(current page.north east) + (-1cm,-1cm)$)
    {\includegraphics[height=2cm]{\GetSchool(logo)}};%
}

% Title
\ntaddtocover[halign=l, height=4cm, valign=c]{1-1}{%
  \fontsize{20}{22}\selectfont\bfseries
  \GetDocTitle(\@LANG@COVER,main)%
}

% Author
\ntaddtocover[halign=l]{1-1}{%
  \fontsize{14}{14}\selectfont
  \GetDocAuthor(name)%
}

% Degree name + date pushed to the bottom
\ntaddtocover[vfill=1, halign=l]{1-1}{%
  \fontsize{11}{11}\selectfont
  \GetDegreeName(\@LANG@COVER)\\
  \PrintDateISO{\GetDocDate(submission,year)-%
```

```
                \GetDocDate(submission,month)}%
}


%%=======================================================
%% COVER ELEMENTS - cover 2-1 (front page)
%%=======================================================

% Reproduce key elements on the front page
\ntaddtocover[halign=l]{2-1}{%
  \includegraphics[height=1.5cm]{\GetSchool(logo)}\\[4mm]%
  \fontsize{16}{18}\selectfont\bfseries
  \GetDocTitle(\@LANG@COVER,main)%
}

% Advisers
\ntaddtocover[vfill=2, halign=l]{2-1}{%
  \MemSet{adviserstringfont}{}{\bfseries}%
  \fontsize{11}{11}\selectfont
  \ntprintpersons{0.9}{2}{adviser}{\GetAdviserOrder()}%
}

% Committee (final status only)
\ntaddtocover[vfill=1, halign=l, status={final}]{2-1}{%
  \MemSet{committeetitlestringfont}{}{\bfseries}%
  \fontsize{10}{10}\selectfont
  \ntprintpersons{0.9}{2}{committee}{\GetCommitteeOrder()}%
}

% Date at the bottom
\ntaddtocover[vfill=1, halign=l]{2-1}{%
  \fontsize{11}{11}\selectfont
  \GetDegreeName(\@LANG@COVER)\\
  \PrintDateISO{\GetDocDate(submission,year)-%
                \GetDocDate(submission,month)}%
}
```

3. Add logos to `novathesisFiles/Schools/myuniv/myfac/Images/`. Supported formats: `.pdf` (preferred for vector logos), `.png`, `.jpg`.

4. Test the new school (no configuration file is edited):

```
make NT="school=myuniv/myfac"
```

5. Test all document types:

```
make NT="school=myuniv/myfac,doctype=phd,docstatus=final"
make NT="school=myuniv/myfac,doctype=msc,docstatus=final"
```

6. Enable debug mode during development to verify spacing:

```
\ntsetup{debug={cover}}
```

B.10 Real-World Examples

B.10.1 NOVA/FCT: Mixed LaTeX and TikZ

The FCT cover (`novathesisFiles/Schools/nova/fct/nova-fct-defaults.clo`) demonstrates a typical mixed strategy:

- All logos and graphical decorations (university logo, school logo, separator rule, department name) are placed with `\ntaddtocover[tikz]{...}`, giving pixel-precise positioning independent of the text flow.
- All text content (title, author, degree, date) uses plain `\ntaddtocover` elements stacked in the cover's text area, which starts at `margin(cover,top)=8.6cm` — well below the absolutely-positioned logo band.
- The front page (cover `2-1`) reuses the same elements but with a lighter margin (`6.6cm` top) and adds advisers and committee.
- PhD and MSc documents swap the order of title and author name using `\ifphddoc` conditionals inside shared elements.

Key cover-margin setting:

```
\SetMargin*(cover,top)={8.6cm}   % leaves room for the TikZ logo band
\SetMargin*(cover,2-1,top)={6.6cm} % front page has a smaller logo band
```

Background setup (colour varies by document class):

```
\def\@novafctcovercolor{red}          % default (BSc/other)
\ifdocclass{msc}{\def\@novafctcovercolor{blue}}{}
\ifdocclass{phd}{\def\@novafctcovercolor{green}}{}
\SetThesisCover*(\@DOCTYPE,1-1,image)=
    {nova-fct-cover-\@novafctcovercolor-front}
```

B.10.2 ULISBOA/IST: Pure LaTeX Box Stacking

The IST cover (`novathesisFiles/Schools/ulisboa/ist/ulisboa-ist-defaults.clo`) uses no TikZ for content elements at all. Every element — logo, school name, a decorative image, title, author, advisers, committee, date — is placed as a plain box element with explicit `height` and `vspace` values. This approach is simpler to reason about and produces a layout that tolerates variable-length content gracefully.

The layout sequence for cover `1-1`:

```
\ntaddtocover[halign=l, hspace=-2.25mm]{1-1,2-1}{logo image}
\ntaddtocover[vspace=5mm, hspace=23mm]{1-1,2-1}{school name text}
\ntaddtocover[vspace=1.62cm, height=5cm]{1-1,2-1}{decorative image}
\ntaddtocover[vspace=1.2cm,  height=1.35cm, valign=c]{1-1,2-1}{title}
\ntaddtocover[vspace=1.0cm,  height=1.05cm]{1-1,2-1}{author name}
\ntaddtocover[vspace=1.7cm,  height=3.0cm, valign=t]{1-1}{advisers}
\ntaddtocover[vfill=1,      height=1.6cm, valign=t]{1-1}{degree text}
\ntaddtocover[vfill=1, height=1.25cm, valign=t,
              status={working,provisional}]{1-1}{draft message}
\ntaddtocover[vfill=1, height=1.25cm, valign=t,
              status={final}]{1-1}{approved message}
\ntaddtocover[vfill=1, height=8.5mm, valign=b]{1-1,2-1}{date}
```

Each `height` reservation ensures that regardless of whether the optional subtitle, specialization line, or committee is present, the elements below remain at a predictable vertical position.

B.11 Quick Reference

Command / Setting	Purpose
`\margin(cover,top):={...}`	Cover text-area top margin
`\margin(cover,<id>,top):={...}`	Per-cover override
`\thesiscover(<id>,bgcolor):={...}`	Page background colour
`\thesiscover(<id>,image):={...}`	Full-page background image
`\thesiscover(<id>,textcolor):={...}`	Default text colour
`\ntaddtocover[attrs]{ids}{code}`	Register a text box element
`\ntaddtocover[tikz]{ids}{code}`	Register TikZ overlay code
`\ntAddToCoverTikz<cmd>[opts](…){ids}{code}`	Register a TikZ node/command
`\NTAddToHook{cover/<id>/pre}{...}`	Inject code before a cover
`\debuggrid[color]`	Render a measurement grid
`\ntsetup{debug={cover}}`	Enable cover debug mode
`make NT="school=<univ/school>"`	Build for a given school

C

The Build System

novathesis documents are built with `latexmk`. All the LaTeX-specific knowledge (default engine, `biber`, glossaries, auxiliary directory, clean-up lists) lives in the `latexmkrc` file at the project root, which `latexmk` loads automatically. The `Makefile` is a thin, convenient wrapper around it.

For everyday writing you only need `make`. If you cannot (or do not want to) use `make`, calling `latexmk` directly gives the same result (see Section C.4).

C.1 Everyday Building

```
make            # build with LuaLaTeX (recommended default)
make view       # build, then open the PDF
make watch      # rebuild automatically every time a file is saved
make clean      # remove build artifacts (keeps the PDF)
```

The main targets are:

Target	Effect
`make` (or `make lua`)	Build with LuaLaTeX
`make pdf`	Build with pdfLaTeX
`make xe`	Build with XeLaTeX
`make view`	Build, then open the PDF
`make watch`	Rebuild on every change (`latexmk -pvc`)
`make log`	Show the build log
`make clean`	Remove build artifacts (keeps the PDF)
`make distclean`	`clean` + remove PDF and SyncTeX files
`make help`	Show the built-in help
`make glsbib`	Convert 7.10.x glossary `.tex` entry files to `.bib` (one-off; see Appendix D)

All auxiliary files go into the `AUXDIR/` directory; on success, the PDF (and its SyncTeX file) are copied to the project root.

The behavior may be adjusted with variables appended to the command:

Variable	Effect
`FILE=name`	Root `.tex` file (default: auto-detect the single file containing `\documentclass`)
`NT="k=v,..."`	`\ntsetup` overrides (see Section C.2)
`V=1`	Verbose: show the raw LaTeX output
`BATCH=1`	Batch mode, never stops at errors (good for Continuous Integration (CI))
`FASTWRITES=0`	Load `morewrites` (skipped by default) for a register-heavy pdfLaTeX/XeLaTeX document (see Section C.3)
`TL=2024`	Use `/usr/local/texlive/2024` for this run
`VIEWER=...`	PDF viewer used by `make view`
`FLAGS=...`	Extra `latexmk` flags, passed through

```
make V=1                    # see everything LaTeX prints
make pdf BATCH=1            # pdfLaTeX, never stop at errors
make view VIEWER="open -a Skim"
```

C.2 Command-Line Overrides

Any `\ntsetup` option can be given at the command line, *without editing* `0-Config/1_novathesis.tex`. Command-line values take precedence over the values in the configuration files.

```
# Build for a different school/doctype/language, nothing is edited
make NT="doctype=msc,school=uminho/ec,lang=pt"

# Any \ntsetup key works
make NT="docstatus=final,print/index=true"
```

This is the recommended way to test other schools, to build final versions in CI, or to keep several variants of the same document without touching the configuration.

Internally, the `NT` variable defines the macro `\ntoverride` before the class loads. The overrides are applied right after `0-Config/1_novathesis.tex` is read, and the `school` option is then re-applied so that school-derived settings (e.g. the degree acronym, which depends on `doctype`) stay consistent, whatever the order of the keys.

C.3 Faster pdfLaTeX and XeLaTeX Builds

pdfTeX and XeTeX can only write to 16 files at a time (LuaTeX allows 65536). novathesis alone claims 9 of those 16 slots. The `morewrites` package removes the limit, but at a steep price: it makes a pdfLaTeX pass roughly nine times slower. Since 8.0 a typical document fits within the 16 slots, so **`morewrites` is not loaded by default**.

Beyond the 9 slots the class uses, each of the following costs one more, and only when you actually use it: `\listofalgorithms` and `\listoflistings` (when the list is printed), plus whatever your own packages need (`minted`, for instance). Glossaries, acronyms and symbols cost nothing at all since 8.0 — `bib2gls` passes everything through the `.aux` file. Even this manual, which exercises every one of those features, now fits.

Only a register-heavy document overshoots, and the build then stops with `No room for a new \write`. In that case, re-enable `morewrites`:

```
make pdf FASTWRITES=0
```

The switch is a no-op under LuaLaTeX, which never needs `morewrites`. Without `make`, define the macro before `\documentclass`:

```
pdflatex -shell-escape -jobname=template \
    '\def\ntmorewrites{}\input{template}'
```

C.4 Building Without `make`

Since `latexmkrc` carries the configuration, a plain `latexmk` call is enough:

```
latexmk -pdflua -shell-escape -file-line-error template
```

The overrides of Section C.2 are also available:

```
# With latexmk
latexmk -pdflua -shell-escape -usepretex \
    -pretex='\def\ntoverride{doctype=msc,lang=pt}' template

# With a plain engine
lualatex -shell-escape -jobname=template \
    '\def\ntoverride{doctype=msc,lang=pt}\input{template}'
```

C.5 Maintainer Tools

The remaining machinery is only relevant for novathesis development and regression testing. It lives in `Makefile.dev` (loaded automatically by the `Makefile` when present; not shipped in releases) and in the helper script `.Build/nt-variant.sh`.

C.5.1 Building One School Variant

```
make school SCHOOL=nova/fct                    # defaults from schools.conf
make school SCHOOL=uminho/ec TYPE=msc LNG=pt ENGINE=lua
```

The variant is built with the override mechanism of Section C.2, so the working copy is never modified. Each variant uses its own auxiliary directory under `AUXDIR/variants/`. The resulting PDF is named `univ-school-doctype-lang-engine.pdf` (e.g. `nova-fct-phd-en-lua.pdf`). Defaults come from `.Build/schools.conf`: the highest degree available for the school, and `lua` whenever the school supports it. If the build fails, the log file is copied next to where the PDF would go, with the same base name. `LNG` (not `LANG`) sets the language: `LANG` is the shell's own locale variable, so using it here would leak into the build and, for anything other than an installed locale, break LuaLaTeX.

`NT='<keys>'` appends extra `\ntsetup` keys to the doctype/school/lang/docstatus this target already computes, e.g. to print every SDG on this variant's cover:

```
make school SCHOOL=nova/fct \
    NT='print/sdgs/list={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17}'
```

Quote it: it typically contains commas and braces. This is the same variable name as the main `Makefile`'s `NT=` (Section C.2), but not the same behaviour: there, `NT=` *replaces* `\ntoverride` outright; here it only *appends* to the override this target builds for you.

C.5.2 Building the Full Matrix

```
make matrix                  # all variants
make matrix JOBS=4           # up to 4 units in parallel
make matrix FILTER=nova/fct  # restrict by prefix
↪  (university/school/sub-program)
make matrix FILTER=uminho,upl  # comma-separated prefixes
make matrix WARM=0           # one fresh aux dir per variant (no warm reuse)
make matrix NT='print/sdgs/list={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17}'
                              # extra \ntsetup keys, appended to every variant
```

`NT`, quoted, works exactly as in `make school` above: it appends extra `\ntsetup` keys to *every* variant in the matrix, rather than replacing `\ntoverride` like the main `Makefile`'s `NT=` does.

`make matrix` builds every school/doctype/language/engine combination listed in `.Build/schools.conf`. The `FILTER` variable restricts the matrix by prefix: a university (`nova`), a school (`nova/fct`, which includes sub-programs such as `nova/fct/cbbi`), or a comma-separated list of prefixes. All PDFs of one invocation go into `AUXDIR/matrix/YYYY-MM-DD@hh-mm-s` (a folder named with the invocation timestamp), together with the log/build output of any failed variants.

The file `schools.conf` is organized in *groups*, each starting with a header such as:

```
[order: school, doctype, lang; processor: pdf,lua]
nova/fct    [phd, msc]  [en, pt]
...
```

Each group × engine is a *unit*: a sequential chain of that group's variants and the quantum the scheduler dispatches. The semantics are:

- **Units build sequentially**, in the order set by the `order:` tuple (the last dimension varies fastest), so consecutive variants are as similar as possible. A group with `processor: pdf,lua` yields two units (one per engine).

- **Warm reuse** (`WARM=1`, the default). The variants of a unit share *one* auxiliary directory, so each reuses the previous one's `.aux/` `.bbl/` `.glstex` and converges in fewer passes (`biber` and `bib2gls` are skipped when their inputs are unchanged). This is safe because the per-variant `\ntoverride` is injected through a *tracked* file that `latexmk` watches (not the untracked `-pretex`), so changing school forces the correct rebuild and no variant can silently reuse the previous one's PDF. `make matrix WARM=0` restores a fresh (cold) aux directory per variant.

- **Cost-based scheduling.** Units are dispatched *longest-first*, using each variant's measured build time cached in `.Build/.matrix-costs.tsv` (refreshed after every run). This front-loads the heavy targets so the cores stay busy right to the end, with no need to hand-order `schools.conf`. An empty cache or an unseen variant falls back to group order (`NT_COST_DEFAULT` sets the assumed time for unknowns); delete the file to reset.

- **`JOBS` parallelizes units, never a unit's inside.** With `JOBS>1` whole units run concurrently, each under its own *jobname* (its aux-directory name), so per-jobname caches that tools write into the working directory — `minted`'s `_minted-<jobname>` and the like — cannot collide between concurrent units. Concurrent `biber` processes fail sporadically, so a `flock` wrapper (`.Build/shims/biber`) is enabled automatically when `JOBS>1` (or forced with `BIBER_LOCK=1`).

- **`FASTWRITES=1` by default**, so pdfLaTeX variants skip `morewrites` and run in about half the time. The matrix then does not exercise the `morewrites` path, so run `make matrix FASTWRITES=0` from time to time to keep it covered.

- **Each variant's glossaries are verified**, not just its exit status: `.Build/check-glossaries.py` asserts that no entry was dropped and that the lists are in sort-key order. A build that succeeds with a mangled glossary is reported as a failure.

C.5.3 Other Maintainer Targets

Target	Effect
`make zip`	Release archive `novathesis-<version>@<date>.zip` (excludes `Makefile.dev` and `.Build/`)
`make bump-patch`	Bump the patch number (e.g. 8.0.0 becomes 8.0.1)
`make bump-minor`	Bump the minor number (e.g. 8.0.0 becomes 8.1.0)
`make bump-major`	Bump the major number (e.g. 8.0.0 becomes 9.0.0)
`make needlua`	Regenerate the list of schools that require LuaLaTeX
`make gclean`	`git clean -fx` (keeps `Scripts` and `Fonts`)
`make help-dev`	Show the maintainer help

D

Migrating from 7.10.x

novathesis 8.0 changed how glossaries, acronyms and symbols are stored. If you are starting a new document, you can skip this appendix entirely — the template already ships in the new format. If you are upgrading a document you have been writing against 7.10.x or earlier, this is the only change that requires your attention.

D.1 What Changed, and Why

Glossary entries used to be written as LaTeX commands in `.tex` files, and processed by `makeglossaries`. They are now `.bib` entries, processed by `bib2gls`.

The reason is a hard limit in pdfTeX and XeTeX: a document can only write to 16 files at once, and the old glossary machinery claimed four of those slots. The template had to load the `morewrites` package to work around it, which made every pdfLaTeX pass roughly nine times slower. With `bib2gls` the glossaries need *no* write slots at all, so most documents no longer need `morewrites`: a full pdfLaTeX build of this manual went from 109 to 36 seconds. See Section C.3.

New requirement: Java

`bib2gls` is a Java program. It ships with TeX Live, but it needs a Java Runtime Environment (JRE) on your machine. Overleaf provides one. For a local installation, install any current JRE if `bib2gls --version` does not work.

D.2 The Migration, Step by Step

1. Convert your entry files.

```
make glsbib
```

This converts every glossary `.tex` file in `1-FrontMatter/` into a `.bib` file of the same name, leaving the originals untouched. It skips any file already converted, so it is safe to run twice.

2. Re-add your `sort` keys. This is the step people miss, so `make glsbib` counts them for you and prints a warning like:

```
convert  1-FrontMatter/symbols.tex  ->  1-FrontMatter/symbols.bib
   WARNING: 1-FrontMatter/symbols.tex had 8 sort key(s). convertgls2bib
            DROPS them; re-add them in 1-FrontMatter/symbols.bib or those
            entries will mis-sort.
```

The converter silently discards every `sort` key. Nothing breaks — the document still builds, at the same page count — but the affected entries come out in the wrong order. It bites exactly the entries that needed a sort key in the first place: symbols, and any entry whose name is a command. Open each converted file and restore them:

```
@symbol{mu,
  name        = {\ensuremath{\mu}},
  sort        = {mu},        % <- restore this
  description = {Mu},
  type        = {symbols}
}
```

3. Convert symbols to `@symbol`. The converter turns everything into `@entry`. Entries that are symbols should use `@symbol` instead, so `bib2gls` treats them as symbols rather than words.

4. Remove `\glsaddall`. If your document calls `\glsaddall`, delete the call. It has no equivalent in `bib2gls`, which decides inclusion per glossary: the template lists only the entries you actually use, in every list. To list every glossary and symbol entry whether or not it is cited (what `\glsaddall` used to do), set `\ntsetup{glossaries/addall=true}`.

5. Delete the old `.tex` entry files once the output looks right. They are no longer read. If one is still registered in `0-Config/4_files.tex`, the build stops with an error telling you so — that error is the safety net for this whole procedure, not a bug.

6. Rebuild and check the lists. Compare the glossary, acronym and symbol lists against your previous PDF: same entries, same order.

D.3 If Something Looks Wrong

Symptom	Cause
"Glossary source … is in the old (7.10.x) format"	A `.tex` entry file is still registered. Run `make glsbib`.
Entries in a strange order	Missing `sort` keys (step 2).
An entry is missing from a list	For acronyms this is normal — only the ones you cite are listed. For the glossary and symbols lists it is not; check the entry really is in the `.bib` file.
"Undefined control sequence \bibglsnew…"	`bib2gls` did not run. Check `bib2gls --version`, i.e. check Java.

Nothing outside the glossaries changed: your chapters, bibliography, cover configuration and `\ntsetup` options all work as before.

E

Detailed AI Taxonomy Descriptions

This appendix provides a detailed breakdown of the AIDpackageName taxonomy. For each category, we define the **Objective** (the high-level goal of using AI in this phase) and the **Scope** (what is generally included or excluded).

1. Conceptualization

Objective: To use AI as a thought partner for brainstorming, refining the research direction, and establishing the theoretical foundation before empirical work begins.

Scope: Includes ideation, hypothesis formation, and feasibility checks. Excludes the actual execution of experiments or data collection.

Keys:

Idea generation: Using AI to brainstorm new research topics, interdisciplinary connections, or novel angles on existing problems.

Objective refinement: Refining vague goals into concrete, actionable research objectives.

Research questions: Drafting and iterating on specific research questions (RQs) to ensure they are clear and answerable.

Feasibility check: Assessing whether the proposed study is viable regarding resources, time, and data availability.

Preliminary research: Conducting quick background checks or "pre-studies" to see if the idea has already been solved.

Simulation/Scenarios: Using AI to conceptualize theoretical models, simulate persona responses, or design hypothetical scenarios.

2. Literature Review

Objective: To accelerate the discovery, synthesis, and organization of existing knowledge.

Scope: Includes searching, summarizing, and translating papers. Excludes the final critical argumentation (which remains the author's responsibility).

Keys:

Search & Discovery: Using AI tools (e.g., semantic search) to find relevant papers that keyword searches might miss.

Summarization: Generating summaries or abstracts of long papers to quickly assess relevance.

Mapping: Visualizing connections, citation networks, or thematic clusters in the literature.

Pattern recognition: Identifying trends or recurring themes across a large corpus of text.

Gap identification: Using AI to suggest areas where current research is lacking or contradictory.

Translation: Translating foreign-language literature to make it accessible for the review.

3. Methodology

Objective: To assist in the structural design of the research study.

Scope: Includes experimental design and instrument creation. Excludes the physical conduct of experiments.

Keys:

Experimental design: Designing the logic, control groups, and variables of the study.

Prototyping: Creating early drafts of survey instruments, interview guides, or experimental apparatus designs.

Method selection: Suggesting appropriate statistical methods or qualitative frameworks for the data.

4. Software Development and Automation

Objective: To facilitate the creation, optimization, and maintenance of code used in the research.

Scope: Includes coding, debugging, and documentation.

Keys:

Code generation: Generating boilerplate code, scripts, or functions from natural language descriptions.

Optimization: Refactoring code for better performance or readability.

Debugging: Identifying syntax errors or logical bugs in scripts.

Automation: Writing scripts to automate file management, backups, or batch processing.

Algorithm design: Assisting in the logic and mathematical formulation of algorithms.

Documentation: Generating docstrings, comments, and README files for research software.

5. Data Management

Objective: To handle the data lifecycle from collection to reporting.

Scope: Includes cleaning, analysis, and synthetic generation. Excludes the fabrication of results (which is ethical misconduct, distinct from declared synthetic data).

Keys:

Collection: Writing scrapers or using AI agents to gather public data.

Validation: Checking data for consistency, outliers, or errors.

Cleaning: Automating the formatting, parsing, and repair of messy datasets.

Curation: Organizing and categorizing large datasets.

Analysis: Suggesting or performing statistical tests and interpreting raw outputs.

Visualization: Generating code for plots, graphs, and data dashboards.

Reporting: Summarizing data findings in textual or tabular format.

Labeling: Using LLMs to annotate or classify text/image datasets (zero-shot/few-shot labeling).

Synthesis: Generating synthetic datasets to preserve privacy or augment small samples.

Anonymization: Detecting and removing Personally Identifiable Information (PII).

Translation: Translating textual data (e.g., survey responses) into the analysis language.

6. Visuals and Multimedia

Objective: To create or enhance non-data visual elements.

Scope: Includes illustrative diagrams and image editing. Excludes scientific data plots (covered in Data Management).

Keys:

Generation: Creating conceptual images, illustrations, or diagrams from scratch.

Editing: Enhancing, cropping, or modifying existing images (e.g., removing background).

Charts/Infographics: Creating flowcharts, process diagrams, or high-level infographics.

7. Writing and Editing

Objective: To assist in the textual articulation of the research.

Scope: Includes drafting, polishing, and translation. Note: Authors remain accountable for accuracy.

Keys:

Drafting: Generating initial text for sections based on bullet points or notes.

Polishing: Correcting grammar, spelling, and punctuation.

Summarizing: Creating the abstract or plain-language summary.

Conclusions: Synthesizing the discussion into a final concluding statement.

Tone adjustment: Rewriting text to be more formal, concise, or accessible.

Translation: Translating the manuscript from the author's native language to the publication language.

References: Formatting citations and bibliography styles.

Presentation: Drafting slide decks or conference poster text.

Title generation: Brainstorming catchy and accurate titles for the work.

8. Ethics Review

Objective: To act as a check on the ethical integrity of the work.

Scope: Includes bias detection and risk assessment.

Keys:

Bias detection: Scanning text or study designs for potential cultural or gender bias.

Risk assessment: Identifying potential dual-use concerns or societal risks.

Compliance: Checking against specific ethical guidelines or checklists.

Confidentiality: Ensuring no private data is inadvertently leaked in the text.

9. Quality Assurance (Supervisor Role)

Objective: To use AI as a critical reviewer or "devil's advocate."

Scope: Includes simulated peer review and limitation checking.

Keys:

Simulated Peer Review: Using AI as a reviewer to identify unclear sections or potential critiques.

Field Trends & Consistency: Checking the manuscript against current research trends and ensuring internal consistency.

Limitations: Identifying weaknesses or limitations in the study that the authors missed.

Publication strategy: Suggesting suitable journals or conferences and helping with the submission strategy.

Publication support: Assisting in drafting cover letters, responses to reviewers, or other publication-related materials.

I

Lorem Ipsum Annex

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

I.1 A section A

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

I.2 A section B

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum

quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

I.2.1 A section B.1

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

I.2.2 A section B.2

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Colophon



This document was typeset with xeLaTeX and the novathesis template (v8.5.0), set in New PX (Palatino), with New PX Math for math.

J. M. Lourenço. *novathesis — The Manual: A LaTeX template for theses and dissertations*. Version 8.5.0. 2026. DOI: 10.5281/zenodo.21938603 This is *not* a PDF/A archival copy.

Build 12cc9022, November 14, 2026.

A LaTeX class for writing theses and dissertations, with ready-made covers and typographic rules for more than twenty institutions.

This manual documents installation, the configuration files, school and cover selection, bibliography styles, page layout, and every option the class accepts. It is written for authors who want to start writing on the first day, and for the maintainers who add new schools.

DOCUMENTATION

novathesis.github.io

SOURCE

github.com/joaomlourenco/novathesis

DOI

10.5281/zenodo.21938603

LICENCE

LPPL 1.3c



João M. Lourenço

NOVA UNIVERSITY LISBON

AUGUST 2026

v8.5.0