

THE CSTHM PACKAGE: THEOREM ENVIRONMENTS FOR COMPUTER SCIENCE

AGNI DATTA

ABSTRACT. The `csthm` package delivers a robust and accessible suite of theorem-like environments tailored to the needs of computer science and related technical disciplines. It provides three distinct visual styles, modern (`fancy`), standard (`normal`), and traditional (`oldschool`), each supporting both numbered and unnumbered forms to address diverse formal documentation requirements. Designed for minimal configuration, the package integrates seamlessly with established tools such as `amsthm`, `thmtools`, `mdframed`, and `cleveref`, facilitating advanced cross-referencing, customizable headers, and polished presentation with minimal user intervention. By balancing flexibility and simplicity, `csthm` enables efficient creation of well-structured and visually consistent theorem environments without introducing unnecessary complexity, making it particularly suitable for users prioritizing streamlined usage and uniform formatting in technical manuscripts. This release is a major revision, superseding prior versions and benefiting from more extensive testing. Nonetheless, as with any evolving software, ongoing refinement is anticipated. User feedback regarding any issues or bugs is welcome to ensure prompt resolution. While the traditional (`oldschool`) style remains the most widely adopted, continued improvements are especially needed for the `fancy` and `normal` styles.

CONTENTS

1. Introduction	3
1.1. Motivation	3
1.2. Package Overview	3
1.3. Package Installation	3
1.4. Loading the Package	4
2. Dependencies	4
3. Environment Categories	5
3.1. Complete Environment List	5
3.2. Starred Variants	6
3.3. Special Environments (Case Environment Only)	6
4. Customization	6
4.1. Accent Color	6
4.2. QED Symbols	7
4.3. Cross-referencing	7
4.4. Other Hacks	7
5. Style Examples	7
5.1. Visual Differences	8
5.2. When to Use Each Style	8
6. Implementation Notes	9
6.1. Technical Details	9
6.2. Compatibility	9
7. Troubleshooting	9
7.1. Common Issues	9
8. Version History	10
9. License	10
10. Contact and Support	10

1. INTRODUCTION

1.1. Motivation. The `csthm` package addresses a common need in academic writing within computer science and mathematics: the clear and structured presentation of formal content such as theorems, definitions, propositions, lemmas, and proofs. Standard packages like `amsthm` and `ntheorem` offer some level of control, but they often require manual configuration for stylistic consistency and do not provide fine-grained typographic affordances out of the box. `csthm` fills this gap by offering a complete, visually cohesive framework tailored to the stylistic conventions and practical requirements of formal theoretical writing. The package was designed with particular attention to use cases in theoretical computer science, logic, discrete mathematics, and adjacent areas, where a rich structure of environment types, symbolically meaningful headers, and clear semantic grouping is both common and expected. By offering multiple visual styles and tightly integrated cross-referencing support, `csthm` allows authors to focus on content while maintaining a professional and readable formal layout throughout the document.

1.2. Package Overview. The `csthm` package offers a comprehensive and customizable infrastructure for typesetting a wide range of formal environments, including theorems, corollaries, claims, axioms, principles, conjectures, observations, remarks, warnings, and more. It seamlessly integrates into typical \LaTeX workflows, eliminating the need for extensive customization or intricate technical configuration. Each environment is visually distinct, featuring a unique bullet or marker, and adheres to one of three predefined styles that govern font choices, framing, and color usage. Building upon the `amsthm`, `thmtools` foundation, the package replaces and extends its stylistic model, ensuring syntactic consistency across environments while maintaining semantic distinction through visual indicators. Additionally, it supports per-environment numbering and section-based numbering schemes, and is fully compatible with standard packages like `hyperref` and `cleveref`. Notable features include:

Style Themes: Three selectable themes (`normal`, `fancy`, `oldschool`) that define the visual presentation of all environments.

Environment Collection: A complete set of predefined environments, each with distinct semantic and visual features suitable for various formal content.

Customization: User-configurable headers, symbols, and accent colors via simple commands, requiring no modification of the style file.

Cross-Referencing: Full integration with `cleveref` for automatic, type-aware cross-references without redundant labels.

Numbering Variants: Support for both numbered and unnumbered environments using the standard asterisk-suffix syntax (e.g., `theorem*`).

Ease of Use: To minimize the setup process, we recommend selecting a style and loading the package. All functionalities will be immediately accessible.

The package aims to be declarative, not prescriptive. While it provides a rich set of environments and visual forms, it avoids overengineering and encourages clarity and brevity in formal exposition.

1.3. Package Installation. To begin using the `csthm` package, you must first ensure that the file `csthm.sty` is correctly installed within your local \TeX environment. The most straightforward approach is to place this file in the same directory as your working document. Alternatively, you may choose to install it globally within your \TeX distribution's local tree structure, depending on your setup and preferences. For most users, it may be convenient to install the package via your distribution's package manager (such as `tlmgr` for \TeX Live or `MiKTeX Console` for MiKTeX). If you prefer manual installation or are working in an environment without package management, you can directly obtain the package from CTAN at <https://ctan.org/pkg/csthm>.

1.4. Loading the Package. Once the package is available to your system, it must be loaded in the document preamble using the `\usepackage` command. The `csthm` package provides three distinct visual styles, selectable through an optional argument. One of the following lines should be included at the top of your document, depending on your desired formatting:

```
\usepackage[fancy]{csthm}      % Modern framed style
\usepackage[normal]{csthm}     % Standard sans-serif style
\usepackage[oldschool]{csthm}  % Classic bold/scshape style
```

Each option presents a unique aesthetic that caters to different presentation styles or typographic conventions. We provide a brief overview of the available style options below. These options control the visual layout of theorem-like environments defined by the package. Each option modifies both the typographic appearance and, to some extent, the framing and symbolization.

- `fancy`: A modern presentation style that uses the `mdframed` and `TikZ` packages to produce colored boxes with subtle styling, suited for contemporary documents and slides.
- `normal`: A clean and conventional format featuring sans-serif fonts and symbolic labels; appropriate for documents where typographic neutrality is desired.
- `oldschool`: A traditional layout that emphasizes bold headings and small-caps formatting, reminiscent of classic mathematical texts.

2. DEPENDENCIES

The `csthm` package internally relies on a small suite of well-established \LaTeX packages to enable its features and styles. These dependencies are standard across most modern \TeX distributions and should already be present in typical installations. However, it is useful to be aware of them, particularly if you encounter compilation errors or are configuring a custom setup. The following packages are required:

- `amsmath`, `amssymb`, `amsthm`: These foundational packages are essential for mathematical typesetting. They provide core environments and symbol sets that underpin virtually all mathematical documents written in \LaTeX . The `csthm` package builds directly on top of these.
- `sansmath`: This package enables the use of sans-serif fonts in mathematical environments. It is specifically required to render mathematics in stylistic alignment with the `normal` style option provided by `csthm`, which opts for a sans-serif aesthetic throughout.
- `enumitem`: For customizing and controlling the layout of lists, `enumitem` is necessary. Within `csthm`, it is used to adjust spacing, labeling, and indentation of itemized structures embedded within theorem environments.
- `thmtools`: This package extends the standard theorem infrastructure in \LaTeX providing more flexible and powerful ways to define, customize, and manage theorem-like environments. It is used internally to ensure consistency and modularity in how environments behave across different styles.
- `mdframed`: This package is required only when the `fancy` style is selected. It enables the construction of framed and colored boxes around theorem environments using highly customizable framed layouts. If you do not intend to use the `fancy` style, this dependency may not be invoked.
- `xcolor`: This package is used to provide enhanced color support. It works in conjunction with `mdframed` to style the backgrounds, borders, and headings of theorems when using the `fancy` style.
- `cleveref`: For smart and automatic cross-referencing of theorems, lemmas, and related structures, `cleveref` is employed. It should be used alongside `hyperref`, since it depends on hyperlink functionality to resolve and format references correctly in the compiled document.

3. ENVIRONMENT CATEGORIES

The `csthm` package defines a broad collection of theorem-like environments, each equipped with a distinct visual identity. Every environment is styled with a unique symbolic prefix or bullet, designed to enhance clarity and typographic structure. In general, for each defined environment, both numbered and unnumbered variants are available. The unnumbered forms are invoked by appending an asterisk (*) to the environment name (for example, `theorem`).

3.1. Complete Environment List. The environments provided by the package are organized into four primary categories. The table below lists all supported environments along with their corresponding styles.

Environment	Description
Theorem-like Environments	
<code>theorem</code>	Mathematical theorems and major results
<code>assertion</code>	Strong claims requiring proof
<code>assumption</code>	Foundational assumptions
<code>axiom</code>	Fundamental principles
<code>claim</code>	Statements to be proven
<code>conclusion</code>	Final results or outcomes
<code>conjecture</code>	Unproven but believed statements
<code>corollary</code>	Direct consequences of theorems
<code>fact</code>	Established truths
<code>folklore</code>	Well-known unattributed results
<code>hypothesis</code>	Tentative explanations
<code>lemma</code>	Auxiliary results
<code>postulate</code>	Fundamental assumptions
<code>property</code>	Characteristics of objects
<code>proposition</code>	Mathematical statements
Definition-like Environments	
<code>application</code>	Practical applications
<code>construction</code>	Step-by-step constructions
<code>convention</code>	Agreed-upon standards
<code>definition</code>	Formal definitions
<code>experiment</code>	Experimental procedures
<code>notation</code>	Symbolic conventions
<code>problem</code>	Problem statements
<code>protocol</code>	Algorithmic procedures
<code>result</code>	Computational or experimental results
Remark-like Environments	
<code>commentary</code>	Extended commentary
<code>example</code>	Illustrative examples
<code>exercise</code>	Practice problems
<code>motivation</code>	Motivational discussions
<code>notationabuse</code>	Notation abuse warnings

continued on next page

Environment	Description
<code>note</code>	Informal notes
<code>observation</code>	Empirical observations
<code>question</code>	Open questions
<code>remark</code>	General remarks
Highlight-like Environments	
<code>guideline</code>	Best practice guidelines
<code>highlight</code>	Important points
<code>important</code>	Critical information
<code>insight</code>	Key insights
<code>keypoint</code>	Essential points
<code>recall</code>	Review material
<code>summary</code>	Content summaries
<code>takeaway</code>	Main takeaways
<code>tip</code>	Helpful tips
<code>warning</code>	Important warnings
Proof Environment	
<code>proof</code>	Mathematical proofs

3.2. Starred Variants. Every environment has an unnumbered starred variant (e.g., ‘theorem’, ‘definition’). Use it when you desire a header without a numerical designation. Please note that cross-references may not function correctly.

3.3. Special Environments (Case Environment Only). As an added utility, the package provides a simple pre-defined environment for structured case analysis. This is intended purely as a convenience and can be used as follows:

```
\begin{case}
  \item First case analysis
  \item Second case analysis
\end{case}
```

This environment is beneficial for presenting succinct, enumerated case breakdowns within proofs or definitions. Although it is included for completeness, it is anticipated that most users are already familiar with defining custom itemized or bullet-style environments tailored to their specific stylistic preferences. The inclusion here is intentionally minimal.

4. CUSTOMIZATION

4.1. Accent Color. The `csthm` package allows for minimal but effective customization, primarily through a consistent accent color used across theorem headers, symbols, and framed environments (when applicable). To modify this accent color, you may use the following command in your document preamble:

```
\setaccentcolor{blue} % or any xcolor-supported color
```

Any color name (theoretically) supported by the `xcolor` package is valid. This color will propagate through the styling of theorem titles and visual markers, maintaining a coherent thematic identity throughout your document. Changing the color is immediate and requires no additional configuration.

4.2. QED Symbols. If you wish to adjust the QED symbol displayed at the end of proof environments, you can override its definition directly. The default symbol uses the selected accent color and is typeset in script style. You may redefine it using:

```
\renewcommand{\qedsymbol}{\color{\accentcolor}\square$}
```

This definition ensures that the QED symbol visually matches the rest of the document’s styled elements. You are free to replace the symbol with any other mathematical character or construct, but the default choice emphasizes consistency and clarity.

4.3. Cross-referencing. When `hyperref` is loaded, the package automatically loads `cleveref`, enabling intelligent and context-aware cross-referencing throughout the document. This integration ensures that references to theorems, definitions, proofs, and other environments are formatted consistently and include appropriate labels without additional user intervention.

4.4. Other Hacks. While `csthm` is designed to be simple and declarative, ideally allowing users to load the package, choose a style, and proceed without friction, the underlying `.sty` file does expose several internal customizations for advanced users. These are intentionally undocumented here, as the package is not meant to serve as a general-purpose customization toolkit. Its philosophy favors minimal user intervention over maximal flexibility.

If you are inclined to experiment with deeper adjustments, such as modifying the structure of theorem environments, changing symbol behavior, or redefining internal macros, those hooks are available in the source, though no official interface is provided. In such cases, it may be more appropriate to use lower-level packages like `thmtools`, `ntheorem`, or `amsthm` directly, which offer finer-grained control and a broader API surface for theorem management.

That said, tweaks like accent color and QED symbol adjustments are intended to cover the vast majority of user needs with minimal setup. Any further modification is left to the especially astute or curious reader.

5. STYLE EXAMPLES

For more thorough guidance and to witness the rendered output across diverse environments, users are advised to refer to the official documentation available within the `\doc` directory of CTAN. The `csthm` package is meticulously designed to seamlessly integrate formal environments within \LaTeX documents, facilitating a smooth transition from conventional theorem packages. Its user-friendly interface retains familiarity while introducing enhanced stylistic control and semantic clarity.

Below is an exemplar displaying the declaration of a definition environment, followed by a theorem environment that references the former using the `cleveref` package. This showcases the straightforward syntax and integrated cross-referencing capabilities:

```
\begin{definition}[Prime Number]\label{def:prime}
  A prime number is a natural number greater than $1$ that has no positive divisors
  other than 1 and itself.
\end{definition}

\begin{theorem}[Prime Number Theorem]\label{thm:prime-number}
  If $p$ is a prime as defined in~\cref{def:prime}, then the distribution of primes
  among the natural numbers follows an asymptotic density given by
  \[
    \pi(x) \sim \frac{x}{\log x} \text{ as } x \rightarrow \infty,
  \]
  where $\pi(x)$ denotes the prime-counting function.
\end{theorem}
```

```
\begin{proof}
```

The proof of the Prime Number Theorem involves complex analysis and properties of the Riemann zeta function; see standard references for full details.

```
\end{proof}
```

Additional environments follow the same usage pattern. For instance, one may define lemmas, corollaries, remarks, or examples simply by invoking the corresponding environment with optional titles and labels:

```
\begin{lemma}[Euclid's Lemma]\label{lem:euclid}
```

If a prime divides the product of two integers, it must divide at least one of those integers.

```
\end{lemma}
```

```
\begin{corollary}\label{cor:inf-primes}
```

There exist infinitely many primes.

```
\end{corollary}
```

```
\begin{remark}
```

This corollary follows immediately from Euclid's classical proof.

```
\end{remark}
```

```
\begin{example}
```

The number 2 is the smallest prime number.

```
\end{example}
```

This structured approach promotes clarity and rigor, ensuring that all formal statements are visually distinct and logically connected through precise labeling and referencing, thus adhering to the highest standards expected in academic and technical writing.

5.1. Visual Differences. The `csthm` package offers three distinctive visual styles, each carefully designed to address different presentation contexts and aesthetic preferences. These styles provide meaningful visual cues that help readers quickly identify the nature of formal content within a document.

Fancy Style: This style incorporates colored left borders alongside subtle background shading and utilizes sans-serif fonts for headings. It is particularly suited for modern technical documents and presentations, where a dynamic and visually engaging layout can enhance comprehension and maintain reader interest over extended material.

Normal Style: Prioritizing clarity and simplicity, the normal style applies symbolic prefixes paired with clean sans-serif fonts. It avoids the use of color, making it an excellent choice for documents intended for print or environments where color reproduction may be inconsistent. This style balances formality and readability without compromising structural clarity.

Oldschool Style: Reflecting the typographic traditions of classical mathematical texts, the oldschool style uses boldface and small-caps formatting. This approach appeals to authors and readers who prefer a more conventional and time-tested visual presentation, aligning with formal publications and traditional academic standards.

5.2. When to Use Each Style. Selecting an appropriate style depends on the intended medium, audience, and desired visual impact. Understanding the contexts where each style excels allows authors to present their work most effectively.

- **Fancy:** Best suited for modern technical documents, online publications, and presentation slides. The enhanced color and framing elements aid in directing the reader's attention and emphasizing key formal components.

- **Normal:** Ideal for standard academic papers, theses, or reports, especially when color printing is limited or unavailable. This style ensures that formal content remains visually distinct without relying on color cues.
- **Oldschool:** Appropriate for formal mathematical treatises, classical journals, and orthodox publications. It upholds conventional typographic standards, providing a familiar and authoritative appearance.

6. IMPLEMENTATION NOTES

6.1. Technical Details. The `csthm` package leverages the powerful and flexible `thmtools` package to implement all theorem-like environments, making extensive use of the `\declaretheorem` command. This approach ensures that the environments are defined with precision, consistency, and full configurability, allowing for robust customization downstream. In particular, the `fancy` style introduces advanced framed environments which require the `mdframed` package operating on the `TikZ` backend; this dependency enables visually distinctive boxed environments with colored borders and subtle background shades. By default, the numbering scheme is configured to reset at section boundaries via the `numberwithin=section` option, reflecting common academic conventions and aiding navigation through hierarchical document structures. Additionally, the package automates the insertion of QED symbols at the end of proof-like environments, reducing the need for manual commands and enhancing consistency.

- All theorem-like environments are meticulously implemented using `thmtools`' `\declaretheorem` facility to guarantee extensibility and uniform behavior.
- The `fancy` style depends critically on `mdframed` with the `TikZ` backend to render modern framed environments with aesthetic precision.
- Numbering of environments defaults to a section-based scheme (`numberwithin=section`) to maintain logical document structure.
- QED symbols are seamlessly inserted at the conclusion of proof environments, requiring no additional user intervention.

6.2. Compatibility. Designed with broad applicability in mind, `csthm` ensures compatibility across a wide array of \LaTeX document classes commonly employed in academic and technical writing. It integrates smoothly with standard classes such as `article`, `book`, and `report`, which form the backbone of general-purpose \LaTeX documents. Furthermore, it supports AMS document classes including `amsart` and `amsbook`, which are standard choices for rigorous mathematical manuscripts. Although `csthm` strives to coexist peacefully with other theorem-related packages, users should be aware that conflicts may occur, especially if multiple packages attempt to redefine similar environments; such situations necessitate careful package loading order and, in some cases, manual conflict resolution.

- Compatible with all major standard \LaTeX document classes: `article`, `book`, and `report`.
- Fully supports AMS classes such as `amsart` and `amsbook`, frequently used in mathematical and theoretical computer science writing.
- Potential conflicts with other theorem or styling packages may arise and must be resolved by adjusting package loading order or disabling overlapping features.

7. TROUBLESHOOTING

7.1. Common Issues. Despite efforts to streamline usage, users may encounter typical issues when deploying the `csthm` package. Foremost among these is the requirement that exactly one style option must be specified at package loading; failure to do so or specifying multiple conflicting styles results in errors that prevent proper compilation. Color-related problems often trace back to improper loading or configuration of the `xcolor` package, particularly if incompatible driver options are selected or omitted.

The fancy style, relying on mdframed with the TikZ backend, will fail silently or generate errors if these dependencies are missing or misconfigured. Reference management is another common source of issues; to leverage cleveref effectively, the hyperref package must be loaded early in the preamble, before other packages that handle theorem environments or cross-referencing.

No style specified: The package mandates that exactly one style option be passed; omitting this or providing multiple styles causes compilation failure.

Color issues: Problems with color rendering typically indicate that xcolor is not loaded correctly, or incompatible driver options are in use.

Frame problems: The fancy style depends on mdframed and TikZ; verify these are installed and properly loaded to avoid errors.

Reference issues: For reliable cross-referencing, load hyperref before any other packages that interact with references or theorem environments, ensuring cleveref functions as intended.

8. VERSION HISTORY

This section chronicles the progressive development of the csthm package, detailing significant updates, feature additions, and improvements that have shaped its current form.

v1.0 (2024/01/01): Initial release.

v1.1 (2024/05/15): Added support for cleveref.

v1.2 (2024/08/31): Released on CTAN.

v1.3 (2025/01/16): Minor updates:

- Added starred versions of all environments.
- Enhanced theorem styling.
- Added new environments.
- Improved customization options.

v2.1 (2025/06/30): Major overhaul:

- Comprehensive documentation rewrite and expansion.
- Refined new style options with improved visual consistency.
- Enhanced testing and stability.
- Additional customization commands for advanced users.

9. LICENSE

This package is released under the terms of the L^AT_EX Project Public License (LPPL), version 1.3c or later. This license ensures that the package remains free to use, modify, and distribute under clearly defined conditions that protect both authors and users. The full text of the license is available at the L^AT_EX Project's official website: [L^AT_EX Project Public License](#).

10. CONTACT AND SUPPORT

For any issues, suggestions, or general feedback, users are encouraged to contact the package maintainer through various channels. Bug reports and feature requests can be submitted via the project's GitHub repository, where the package is actively maintained and updated. Direct communication through email is also fine for personalized support or inquiries (reply times may vary and may be longer if you encounter issues with my spam filter). The package is distributed on CTAN, providing easy access and version tracking for all users.

Email: agnidatta.org@gmail.com , for direct correspondence and support.

GitHub: <https://github.com/agnidatta/csthm> , to report issues or contribute.

CTAN: <https://ctan.org/pkg/csthm> , official package distribution and documentation.