keytheorems package

version 0.3.3

github.com/mbertucci47/keytheorems

Matthew Bertucci

December 11, 2025

Abstract

An expl3-implementation of a key-value interface to amsthm, implementing most of the functionality provided by thmtools. Several issues encountered with thmtools are avoided (see the README for a list) and a few new features are added.

Contents

1 Dependencies

The package depends on the aliascnt, amsthm, refcount, and translations packages. The `tcolorbox`→P. 13 and `tcolorbox-no-titlebar`→P. 13 keys require tcolorbox, and the `numbered=unless-unique`→P. 8 key requires the unique package. A LaTeX kernel no older than 2025-06-01 is required.

2 Global options

`\keytheoremset{`⟨*options*⟩`}`

Every key in this section can be given as an option to `\usepackage` or in `\keytheoremset`, with the exception that `continues-code`→P. 3 can only be used in the latter.

2.1 Compatibility options

`overload` (initially unset)

Redefines `\newtheorem` to internally use the keytheorems machinery. The syntax remains the same. This is automatically set by `thmtools-compat`.

`thmtools-compat` (initially unset)

For compatibility with thmtools syntax. For most documents,

```
\usepackage[thmtools-compat]{keytheorems}
```

should be a drop-in replacement for `\usepackage{amsthm,thmtools}`. The option defines the commands in the left column below. The right column lists the corresponding keytheorems replacement that should be used in new documents.

thmtools command		keytheorems replacement
`\declaretheorem`	→	`\newkeytheorem`→P. 4
`\declaretheoremstyle`	→	`\newkeytheoremstyle`→P. 15
`\listoftheorems`	→	`\listofkeytheorems`→P. 19
`\listtheoremname`	→	`title`→P. 22 key
`\addtotheorempreheadhook` `\addtotheorempostheadhook` `\addtotheoremprefoothook` `\addtotheorempostfoothook`	→	`\addtotheoremhook`→P. 24
`restatable` environment	→	`store`→P. 6 key
`restatable*` environment	→	`store*`→P. 6 key

Also defined are the `shaded` and `thmbox` keys, implemented internally with tcolorbox rather than the shadethm and thmbox packages, respectively.

2.2 Other global options

`auto-translate=true|false` (default `true`, initially `true`)

If `false`, keytheorems does not automatically translate the title text used for `\listofkeytheorems` →P. 19 and the note produced by the `continues` →P. 5 key. These texts can be manually customized with the `title` →P. 22 and `continues-code` keys, respectively.

`continues-code=`⟨*code with* `#1`⟩

(initially `\GetTranslation{keythms_continues}\pageref{#1}`)

The code used to typeset the note produced by the `continues` →P. 5 key. If English or an unknown language is used, defaults to `continuing from p.\,\pageref{#1}`. Currently (likely inaccurate!) translations exist for several European languages.

`predefined={`⟨*options*⟩`}` (initially unset)

This is a convenience key, similar to ntheorem's `standard` option, that predefines a set of theorems that, unless `auto-translate` is set to `false`, are translated into the current language if translations exist. The predefined theorems are

- `plain` style: `conjecture`, `corollary`, `lemma`, `proposition`, `theorem`;
- `definition` style: `axiom`, `definition`, `example`;
- `remark` style: `remark`.

If your language does not have translations, please feel free to open a GitHub pull request.

These theorems are provided at the end of the preamble (specifically, in the `begindocument` hook) with `\providekeytheorem` →P. 4 so will not overwrite user-defined environments with the same name. By default, the predefined theorems share a counter and do not have a parent counter. These settings can be changed by calling `siblings=false` (alias `sharenumbers=false`) and/or `parent=`⟨*counter*⟩, respectively, in ⟨*options*⟩.

```
\usepackage[
  predefined={parent=section}
  ]{keytheorems}

% or equivalently
\usepackage{keytheorems}
\keytheoremset{predefined={parent=section}}
```

`qed-symbol=`⟨*symbol*⟩ (initially `\openbox`)

Redefines `\qedsymbol` to be ⟨*symbol*⟩.

`restate-counters={`⟨*comma-list of counters*⟩`}` (initially `{equation}`)

Additional counters whose values are preserved when a theorem is restated. This key does not reset the list, so you don't need to include `equation` in ⟨*comma-list*⟩.

`store-all` (initially unset)

Tells keytheorems to grab the body of each theorem so it can later be printed with the `print-body` →P. 23 option of `\listofkeytheorems` →P. 19. Note that this means a theorem body *cannot* contain verbatim material.

`store-sets-label` (initially unset)

Defines the `store` →P. 6 key to also set `label` →P. 5, i.e., it makes `store=`⟨*tag*⟩ equivalent to `store=`⟨*tag*⟩`,label=`⟨*tag*⟩. Similarly for `store*` →P. 6.

3 Defining theorems

`\newkeytheorem{`⟨*env name*⟩`}[`⟨*options*⟩`]`

Defines a theorem environment ⟨*env name*⟩ which itself takes a few options (see subsection 3.1). You can also declare multiple theorems at once by replacing ⟨*env name*⟩ with a comma-list of names, e.g.,

`\newkeytheorem{theorem,lemma,proposition}[`⟨*options*⟩`]`.

By default, the theorem's printed name is a title-cased ⟨*env name*⟩. This can be changed with the `name` → P. 8 key. All ⟨*options*⟩ are described in subsections 3.2 and 3.3.

```
% preamble
\newkeytheorem{theorem}

% document
\begin{theorem}
There are infinitely many prime numbers.
\end{theorem}
```

Theorem 1. *There are infinitely many prime numbers.*

`\renewkeytheorem{`⟨*env name*⟩`}[`⟨*options*⟩`]`
`\providekeytheorem{`⟨*env name*⟩`}[`⟨*options*⟩`]`
`\declarekeytheorem{`⟨*env name*⟩`}[`⟨*options*⟩`]`

Sometimes a package or class defines theorems that need to be overwritten by the user. For this case, keytheorems provides `\renewkeytheorem` which redefines ⟨*env name*⟩ or errors if it is not defined. For completeness, also provided are `\providekeytheorem` and `\declarekeytheorem`. The former only defines ⟨*env name*⟩ if it is not already defined; the latter always overwrites ⟨*env name*⟩.

3.1 Keys available to theorem environments

As in amsthm, theorems can take an optional argument that contains a note or heading.

```
\begin{theorem}[Bertrand's postulate]
For every $n\geq 1$, there is a prime number $p$ with $n<p\leq 2n$.
\end{theorem}
```

Theorem 2 (Bertrand's postulate)**.** *For every $n \geq 1$, there is a prime number p with $n < p \leq 2n$.*

Alternatively, the optional argument may contain any of the following keys.

`note=`⟨*text*⟩ (initially unset)

Alias `name`. This is the key-value equivalent of the optional argument described above. This syntax, however, allows the argument to contain other keys.

```
\begin{theorem}[note=Legendre's formula]
The number $n!$ contains the prime factor $p$ exactly
  \[ \sum_{k\geq 1} \Bigl\lfloor\frac{n}{p^k}\Bigr\rfloor \]
```

```
times.
\end{theorem}
```

Theorem 3 (Legendre's formula)**.** *The number $n!$ contains the prime factor p exactly*

$$\sum_{k\geq 1}\left\lfloor \frac{n}{p^k} \right\rfloor$$

times.

`short-note`=⟨*text*⟩ (initially unset)

Alias `short-name`. This replaces the value of `note` →P. 4 when displayed in the list of theorems (`\listofkeytheorems` →P. 19).

`label`=⟨*label name*⟩ (initially unset)

This is the key-value equivalent of `\begin{theorem} \label{`⟨*label name*⟩`}`.

```
\begin{theorem}[label=bezout,note=Bézout's identity]
Let $a$ and $b$ be integers. Then there exist integers $x$ and $y$
such that $ax+by=\gcd(a,b)$.
\end{theorem}
See \zcref{bezout}.
```

Theorem 4 (Bézout's identity)**.** *Let a and b be integers. Then there exist integers x and y such that $ax + by = \gcd(a, b)$.*

See theorem 4.

`manual-num`=⟨*text*⟩ (initially unset)

Use this to override the printed number of a theorem. It is useful for making "starred" versions of other theorems, perhaps to represent a reformulated or more difficult version.

```
\begin{theorem}[manual-num=\ref*{bezout}*]
Let $a_1,\dots,a_n$ be integers. Then there exist integers
$x_1,\dots,x_n$ such that $a_1x_1+\dots+a_nx_n=\gcd(a_1,\dots,a_n)$.
\end{theorem}
\begin{theorem}[manual-num=\faRocket] % requires fontawesome5
Don't confuse your readers by changing the numbering without good
reason.
\end{theorem}
```

Theorem 4*. *Let $a_1, \dots, a_n$ be integers. Then there exist integers $x_1, \dots, x_n$ such that $a_1x_1 + \cdots + a_nx_n = \gcd(a_1, \dots, a_n)$.*

Theorem 🚀. *Don't confuse your readers by changing the numbering without good reason.*

`continues*`=⟨*label name*⟩ (initially unset)

Pick up a theorem where you left off. The theorem number remains the same. The printed text can be customized with the `continues-code` →P. 3 option. The starred version also copies the theorem `note` →P. 4 and `short-note` if they are nonempty.

```
\begin{theorem}[continues=bezout]
Moreover, the integers of the form $az+bt$ are exactly the multiples
of $\gcd(a,b)$.
\end{theorem}
\begin{theorem}[continues*=bezout]
Moreover, the integers of the form $az+bt$ are exactly the multiples
of $\gcd(a,b)$.
\end{theorem}
```

Theorem 4 (continuing from p. 5)**.** *Moreover, the integers of the form* $az+bt$ *are exactly the multiples of* $\gcd(a,b)$*.*

Theorem 4 (Bézout's identity, continuing from p. 5)**.** *Moreover, the integers of the form* $az+bt$ *are exactly the multiples of* $\gcd(a,b)$*.*

`store*`=⟨*tag*⟩ (initially unset)

Alias `restate*`. Stores the the theorem to be restated at any point in the document with `\getkeytheorem` → P. 17. With the starred version, counters and labels are taken from the copy called with `\getkeytheorem`, so in this case can only be restated once. This allows you, for example, to write all theorems and proofs in the appendix and call `\getkeytheorem` at the appropriate time mid-document. For the numbering to be correct, the unstarred key will need at most two runs and the starred key at most three runs.

```
\begin{theorem}[store=blub]
A theorem worth restating.
\end{theorem}
More brilliant mathematics.
\getkeytheorem{blub}
```

Theorem 5. *A theorem worth restating.*

More brilliant mathematics.

Theorem 5. *A theorem worth restating.*

A theorem given this key *cannot* contain verbatim material or other unexpected catcodes such as a tikz-cd diagram. The latter issue can be averted with the `ampersand-replacement` key.

```
% preamble
\usepackage{tikz}
\usetikzlibrary{cd}

% document
\begin{lemma}[store=fiberprod]
For any $S$-schemes $X$ and $Y$, there exists a scheme $X\times_S Y$
with morphisms to $X$ and $Y$ such that the diagram
\[\begin{tikzcd}[ampersand replacement=\&]
    X\times_S Y \ar[r] \ar[d] \& X \ar[d] \\
    Y \ar[r] \& S
  \end{tikzcd}\]
commutes and is universal with respect to this property.
```

```
\end{lemma}
\dots
\getkeytheorem{fiberprod}
```

Lemma 6. *For any S-schemes X and Y, there exists a scheme $X \times_S Y$ with morphisms to X and Y such that the diagram*

$$\begin{array}{ccc} X \times_S Y & \longrightarrow & X \\ \downarrow & & \downarrow \\ Y & \longrightarrow & S \end{array}$$

commutes and is universal with respect to this property.

…

Lemma 6. *For any S-schemes X and Y, there exists a scheme $X \times_S Y$ with morphisms to X and Y such that the diagram*

$$\begin{array}{ccc} X \times_S Y & \longrightarrow & X \\ \downarrow & & \downarrow \\ Y & \longrightarrow & S \end{array}$$

commutes and is universal with respect to this property.

`restate-keys={`⟨*list of keys*⟩`}` (initially unset)

Allows passing different keys to the restated theorem. At the moment this is only useful with the `note` →P. 4 key.

```
\begin{theorem}[
  store=rktest,
  note=Original,
  restate-keys={note=Restated},
  ]
Wow, yet another theorem.
\end{theorem}
\getkeytheorem{rktest}
```

Theorem 7 (Original)**.** *Wow, yet another theorem.*

Theorem 7 (Restated)**.** *Wow, yet another theorem.*

`listhack=true|false` (initially `false`)

Meant only to be used with the `break` →P. 15 style key for a theorem starting with a list. Compare:

```
% preamble
\newkeytheoremstyle{breaksty}{break}
\newkeytheorem{observation}[style=breaksty]

% document
\begin{observation}
```

```
\begin{enumerate}
\item First item
\item Second item
\end{enumerate}
\end{observation}

\begin{observation}[listhack=true]
\begin{enumerate}
\item First item
\item Second item
\end{enumerate}
\end{observation}
```

Observation 1. *1. First item*

2. Second item

Observation 2.

1. First item

2. Second item

Note that the value `true` must be explicitly set so that `listhack` is not interpreted as the note text.

`seq`=⟨*name*⟩ (initially unset)

Adds the theorem to a custom sequence ⟨*name*⟩ that can then be listed with `\listofkeytheorems[seq=`⟨*name*⟩`]`. See `seq` → P. 23 for more details.

3.2 Keys also defined in thmtools

These are the `[`⟨*options*⟩`]` available to `\newkeytheorem`. Except for `name` and `style` → P. 9, each key below can also be used in `\newkeytheoremstyle` → P. 15. For more description, see the thmtools package.

`name`=⟨*display name*⟩ (initially title-cased ⟨*env name*⟩)

Aliases `heading` and `title`. Sets the displayed name of the theorem.

```
% preamble
\newkeytheorem{mythm}[name=Some Name]

% document
\begin{mythm}
Some text
\end{mythm}
```

Some Name 1. *Some text*

`numbered`=`true`|`false`|`unless-unique` (default `true`, initially `true`)

Determines if the theorem is numbered. With the value `unless-unique`, there are two cases. If no `parent` → P. 9 is given, a theorem is not numbered if it is the only theorem of its type and numbered otherwise. If a `parent` → P. 9 is given, the same is true but considered within a single value of the parent counter.

For compatibility with thmtools, `numbered` also accepts the values `yes`, `no`, and `unless unique`.

```
% preamble
\newkeytheorem{theorem*}[name=Theorem,numbered=false]

% document
\begin{theorem*}
An unnumbered theorem.
\end{theorem*}
```

Theorem. *An unnumbered theorem.*

`parent`=⟨*counter*⟩ (initially unset)

Aliases `numberwithin` and `within`. Sets the parent counter for the theorem counter, i.e., the displayed theorem number is of the form ⟨*parent counter*⟩`.`⟨*theorem counter*⟩ and the theorem counter is reset to 1 whenever the parent counter is incremented.

```
% preamble
\newkeytheorem{conjecture}[parent=section]

% document
\begin{conjecture}
The first number is the section.
\end{conjecture}
```

Conjecture 3.1. *The first number is the section.*

`sibling`=⟨*counter*⟩ (initially unset)

Aliases `numberlike` and `sharenumber`. Sets the sibling counter for the theorem counter, i.e., the sibling and theorem counters are incremented and reset simultaneously.

```
% preamble
\newkeytheorem{lemma}[sibling=theorem]

% document
\begin{lemma}
This shares its counter with \texttt{theorem}.
\end{lemma}
```

Lemma 8. *This shares its counter with* `theorem`.

`style`=⟨*style name*⟩ (initially unset)

Accepts any ⟨*style name*⟩ defined by `\newkeytheoremstyle` → P. 15, as well as any of the predefined amsthm styles: `plain`, `definition`, and `remark`.

```
% preamble
\newkeytheorem{remark}[style=remark]

% document
\begin{remark}
It's nice to distinguish remarks from definitions and theorems.
```

```
\end{remark}
```

Remark 1. It's nice to distinguish remarks from definitions and theorems.

`preheadhook`=⟨*code*⟩ (initially unset)
`postheadhook`=⟨*code*⟩ (initially unset)
`prefoothook`=⟨*code*⟩ (initially unset)
`postfoothook`=⟨*code*⟩ (initially unset)

Details in section 7.

```
% preamble
\newkeytheorem{test}[
  preheadhook=PREHEAD,
  postheadhook=POSTHEAD,
  prefoothook=PREFOOT,
  postfoothook=POSTFOOT,
  ]
% document
\begin{test}
Some text
\end{test}
```

PREHEAD

Test 1. *POSTHEADSome text PREFOOT*

POSTFOOT

`qed`=⟨*symbol*⟩ (default `\qedsymbol`, initially unset)

Adds ⟨*symbol*⟩ to the end of the theorem body. If no value is given, the current value of `\qedsymbol` is used (one can redefine this or set it with `qed-symbol` → P. 3). By default, this is □.

```
% preamble
\newkeytheorem{example}[qed]
\newkeytheorem{solution}[qed=$\clubsuit$]
% document
\begin{example}
Some text.
\end{example}
\begin{solution}
Some more text.
\end{solution}
```

Example 1. *Some text.* □

Solution 1. *Some more text.* ♣

`refname`=⟨*ref name*⟩ or `{`⟨*singular name*⟩`,`⟨*plural name*⟩`}`

(initially `\text_lowercase:n {`⟨*display name*⟩`}`)

If a single string, then the name used by hyperref's `\autoref`, cleveref's `\cref`, and

zref-clever's `\zcref`. If two strings separated by a comma, then the second string is the plural form used by `\cref` and `\zcref`.

`Refname`=⟨*ref name*⟩ or `{`⟨*singular name*⟩`,`⟨*plural name*⟩`}`

(initially `\text_titlecase_first:n {`⟨*display name*⟩`}`)

Same as `refname` →P. 10 but for `\Autoref`, `\Cref`, and `\zcref` with any of its capitalizing options. Note that `\Autoref` is defined by keytheorems, but requires hyperref to work. As with `\autoref`, there is also a starred version `\Autoref*` that suppresses the hyperlink.

```
% preamble
\newkeytheorem{prop}[
  name=Proposition,
  refname={proposition,propositions},
  Refname={Proposition,Propositions},
  ]
% document
\begin{prop}[label=abc]
Some text.
\end{prop}
\begin{prop}[label=def]
Some more text.
\end{prop}
Consider \zcref{abc,def}. \Autoref{abc} \dots
```

Proposition 1. *Some text.*

Proposition 2. *Some more text.*

Consider propositions 1 and 2. Proposition 1 …

Both cleveref and zref-clever define default reference names for some commonly used counters like `theorem`, `lemma`, etc. For technical reasons, unless explicit values for `refname` →P. 10 and `Refname` are given, keytheorems does not try to change these defaults at all in the case of cleveref and only the singular name in the case of zref-clever. The easiest way to get exactly the output you want is to just explicitly set `refname` →P. 10 and `Refname`.

! The `cleveref` →CTAN package has not been updated since 2018 and contains several incompatibilities with the LaTeX kernel. These are often patched by the LaTeX team, but further incompatibilities are likely to arise with each future update. For this reason, I recommend moving to `zref-clever` →CTAN. It offers all the same features as cleveref and is actively maintained.

3.3 Keys added by keytheorems

`counter-format`=⟨*code*⟩ (initially unset)

Syntactic sugar that essentially does `\renewcommand{\the`⟨*counter*⟩`}{`⟨*code*⟩`}`. The ⟨*code*⟩ should not contain any unexpandable tokens such as formatting commands. Formatting should be taken care of in the style keys `headfont` →P. 15 and `numberfont` →P. 17. If used with an unnumbered theorem, a warning is issued.

```
% preamble
\newkeytheorem{mainthm}[
  name=Theorem,
  counter-format=\Alph*,
  ]

% document
\begin{mainthm}
The first main result, distinguished by using letters.
\end{mainthm}
\begin{mainthm}
And here is the second main result.
\end{mainthm}
```

Theorem A. *The first main result, distinguished by using letters.*

Theorem B. *And here is the second main result.*

The `*` following `\Alph` means "use the current counter", a syntax originally introduced in enumitem's `label` key. This is available with a LaTeX kernel 2025-06-01 or later. For older kernels, the above example could be implemented equivalently with `counter-format=\Alph{mainthm}`.

`leftmargin`=⟨*length*⟩
`rightmargin`=⟨*length*⟩
`margin`=⟨*length*⟩ (initially `0pt`)

Sets the left (respectively, right) margin of the theorem relative to the text width. The `margin` key sets both simultaneously. This sets the theorem apart from the text, similar to a block quote. The code was adapted from Enrico Gregorio's TeX Stack Exchange answers:

- How to change margins in enunciation (theorem-like environment)?
- A theoremstyle with complete indentation using amsthm

```
% preamble
\newcommand{\marginthmtext}{%
  We need some text to show off theorems with margins. }
\newkeytheorem{quotethm}[name=Quote Theorem,margin=1cm]
\newkeytheorem{indentedthm}[name=Indented Theorem,leftmargin=1cm]

% document
\marginthmtext\marginthmtext\marginthmtext

\begin{quotethm}
\marginthmtext\marginthmtext\marginthmtext
\end{quotethm}

\marginthmtext\marginthmtext\marginthmtext

\begin{indentedthm}
\marginthmtext\marginthmtext\marginthmtext
\end{indentedthm}
```

We need some text to show off theorems with margins. We need some text to show off theorems with margins. We need some text to show off theorems

with margins.

> **Quote Theorem 1.** *We need some text to show off theorems with margins. We need some text to show off theorems with margins. We need some text to show off theorems with margins.*

We need some text to show off theorems with margins. We need some text to show off theorems with margins. We need some text to show off theorems with margins.

> **Indented Theorem 1.** *We need some text to show off theorems with margins. We need some text to show off theorems with margins. We need some text to show off theorems with margins.*

`tcolorbox={`⟨*tcolorbox options*⟩`}` (initially unset)

This key specifies that the theorem be placed inside a `tcolorbox` environment with ⟨*options*⟩. The theorem head is typeset as a `tcolorbox` title; to avoid this see `tcolorbox-no-titlebar`.

```
% preamble
\tcbset{
  defstyle/.style={
    arc=0mm,
    colback=blue!5!white,
    colframe=blue!75!black
    },
  }
\newkeytheorem{corollary}[tcolorbox]
\newkeytheorem{definition}[style=definition,tcolorbox={defstyle}]

% document
\begin{corollary}
Products exist in the category of schemes over $S$.
\end{corollary}
\begin{definition}[Dedekind domains]
A \emph{Dedekind domain} is an integrally closed, Noetherian domain of
dimension one.
\end{definition}
```

Corollary 1.

Products exist in the category of schemes over S.

Definition 1 (Dedekind domains)**.**

A *Dedekind domain* is an integrally closed, Noetherian domain of dimension one.

`tcolorbox-no-titlebar={`⟨*tcolorbox options*⟩`}` (initially unset)

Same usage as `tcolorbox` but the theorem head is typeset as usual, not as a `tcolorbox` title.

```
% preamble
\newkeytheorem{boxcor}[
  tcolorbox-no-titlebar={colback=red!10},
  name=Corollary,
  sibling=corollary,
  ]

% document
\begin{boxcor}[Cauchy's theorem]
Let $G$ be a finite group and $p$ a prime dividing the order of $G$.
Then $G$ contains an element of order $p$.
\end{boxcor}
```

Corollary 2 (Cauchy's theorem)**.** *Let G be a finite group and p a prime dividing the order of G. Then G contains an element of order p.*

tcolorbox offers its own comprehensive `theorems` library. If all of your theorems are to be tcolorboxes, I highly recommend using it instead of this package! However, if only some of your theorems will use a tcolorbox, you may want to replicate the styles of `\NewTcbTheorem`. Here is an example that emulates tcolorbox's `standard` theorem style.

```
% preamble
\tcbset{
  thmstyle/.style={
    colback=green!5,
    colframe=green!35!black},
  }
\newkeytheoremstyle{tcb-standard}{
  tcolorbox=thmstyle,
  headpunct={},
  notebraces={}{},
  noteseparator={: },
  notefont=\bfseries,
  bodyfont=\normalfont,
  }
\newkeytheorem{mytheo}[
  name=Theorem,
  style=tcb-standard,
  ]

% document
\begin{mytheo}[Quillen-Suslin]
Every finitely generated projective module over a polynomial ring is free.
\end{mytheo}
```

Theorem 1: Quillen-Suslin

Every finitely generated projective module over a polynomial ring is free.

4 Theorem styles

`\newkeytheoremstyle{`⟨*name*⟩`}{`⟨*options*⟩`}`

This is keytheorems' version of thmtools' `\declaretheoremstyle`. Since it makes little sense to define a style with no keys, we've made the ⟨*options*⟩ argument mandatory. The defined style can be used with either the `style` → P. 9 key or the traditional `\theoremstyle`. Note that unlike amsthm's `\newtheoremstyle`, this command will error if a style has already been defined.

`\renewkeytheoremstyle{`⟨*env name*⟩`}{`⟨*options*⟩`}`
`\providekeytheoremstyle{`⟨*env name*⟩`}{`⟨*options*⟩`}`
`\declarekeytheoremstyle{`⟨*env name*⟩`}{`⟨*options*⟩`}`

To overwrite an existing style, there is the analogous `\renewkeytheoremstyle`. For completeness, also provided are `\providekeytheoremstyle` and `\declarekeytheoremstyle`.

4.1 Keys also defined in thmtools

The following keys have the same meaning and syntax as the corresponding thmtools keys. In addition to the list below, most of the keys available to `\newkeytheorem` → P. 4 can be used in `\newkeytheoremstyle`.

`bodyfont=`⟨*font declarations*⟩ (initially `\itshape`)

Sets the font declarations for the theorem body. This does not affect the theorem heading; see `headfont`.

`break` (initially unset)

Causes the theorem body to be typeset on a new line after the heading. Do not use this with the `postheadspace` → P. 16 key.

`headfont=`⟨*font declarations*⟩ (initially `\bfseries`)

Sets the font declarations for the theorem heading, which includes the theorem name, number, and note. The heading is typeset in a TEX group, so these declarations will not affect the body of the theorem. Note, however, that the declarations in the `numberfont` → P. 17 and `notefont` → P. 16 keys may reset some font features set in `headfont`. By default, the number is typeset in upright shape and the note in both upright shape and medium weight.

`headformat=margin|swapnumber|`⟨*code using* `\NAME`*,* `\NUMBER`*, and* `\NOTE`⟩

(initially `\NAME\␣\NUMBER\NOTE`)

Alias `headstyle`. Allows the user to completely control the appearance of the theorem heading. Within ⟨*code*⟩, the commands `\NAME`, `\NUMBER`, and `\NOTE` correspond to the formatted parts of the theorem head. For compatibility with thmtools, the user is in charge of specifying the separator (usually a space) preceding `\NUMBER`; the separator preceding `\NOTE` is controlled by `noteseparator` → P. 17.

The special values `margin` and `swapnumber` are demonstrated below.

```
% preamble
\newkeytheoremstyle{marginstyle}{headformat=margin}
\newkeytheoremstyle{swapnumberstyle}{headformat=swapnumber}
\newkeytheorem{marginthm}[
  name=Margin Theorem,
```

```
    style=marginstyle,
    ]
  \newkeytheorem{swapnumberthm}[
    name=Swapnumber Theorem,
    style=swapnumberstyle,
    ]

  % document
  \begin{marginthm}
  A theorem whose number juts into the margin.
  \end{marginthm}
  \begin{swapnumberthm}
  A theorem whose name and number are swapped.
  \end{swapnumberthm}
```

1 Margin Theorem. *A theorem whose number juts into the margin.*

1 Swapnumber Theorem. *A theorem whose name and number are swapped.*

! In `headformat` →P. 15, you may also use the traditional amsthm commands `\thmname`, `\thmnumber`, and `\thmnote`, where `#1` is the theorem name, `#2` the number, and `#3` the note. keytheorems expands the head spec inside `\text_expand:n` so for these commands to work properly, the package adds them to `\l_text_expand_exclude_tl`. Note also that if you use these lower-level commands, the style keys `notebraces`, `notefont`, `noteseparator` →P. 17, and `numberfont` →P. 17 will have no effect (of course, you can manually control these things inside the commands' arguments).

`headindent`=⟨*length*⟩ (initially `0pt`)

Sets the distance between the left margin and the theorem heading.

`headpunct`=⟨*code*⟩ (initially `{.}`)

Sets the punctuation that ends the theorem heading.

`notebraces`=`{`⟨*left brace*⟩`}{`⟨*right brace*⟩`}` (initially `{(}{)}`)

Sets the delimiters surrounding the theorem note if present.

`notefont`=⟨*font declarations*⟩ (initially `\fontseries\mddefault\upshape`)

Sets the font declarations for the theorem note. The note is also affected by `headfont` →P. 15; the initial value of `notefont` happens to negate the initial value of `headfont` by resetting the font weight to medium, however any other settings in `headfont` will propagate to the note.

`postheadspace`=⟨*skip expr*⟩ (initially `5pt plus 1pt minus 1pt`)

Sets the distance between the theorem heading and body. Do not use this with the `break` →P. 15 key.

`spaceabove`=⟨*skip expr*⟩ (initially `\topsep`)

Sets the vertical space before the theorem.

`spacebelow`=⟨*skip expr*⟩ (initially `\topsep`)

Sets the vertical space after the theorem.

! With tcolorbox →P. 13 and tcolorbox-no-titlebar →P. 13, the spaceabove and spacebelow keys are internally passed to tcolorbox's before skip and after skip. When no explicit spaceabove or spacebelow values are given, tcolorbox defaults are used instead of \topsep.

4.2 Keys added by keytheorems

inherit-style=⟨*style name*⟩ (initially unset)

Inherit the keys of any style declared with \newkeytheoremstyle →P. 15. Additionally, the three styles predefined by amsthm are possible values: plain, definition, and remark.

noteseparator=⟨*code*⟩ (initially ␣)

The code inserted before the note, and printed only if there is a note. This is executed *before* the font commands set by notefont →P. 16 take effect.

numberfont=⟨*font declarations*⟩ (initially \upshape)

Sets the font declarations for the theorem number. For almost all theorem styles, it is recommended that you *do not* change this setting. As with notefont →P. 16, the number font is affect by headfont →P. 15, though here the initial value only changes the shape to upright.

! For the AMS classes amsart, amsbook, and amsproc, as well as the amsart-based acmart and aomart, the initial key values are slightly different those listed in sections 4.1 and 4.2 in order to match those class's defaults. See subsection 8.2 for details.

5 Restating theorems

When a theorem is given the store →P. 6 key, the contents of the theorem are saved and written to a .thlist file. At the start of the next run, this file is input at the beginning of the document and allows you to retrieve the stored theorems at any point, before or after the original theorem.

\getkeytheorem[⟨*property*⟩]{⟨*tag*⟩}

Retrieves the theorem given the key store=⟨*tag*⟩ or store*=⟨*tag*⟩. An optional ⟨*property*⟩ can be given to retrieve only the corresponding part of the theorem. Currently only the property body is implemented, which retrieves the (unformatted) body of the theorem.

```
\getkeytheorem{mytag}

\begin{example}[store=mytag]
Fascinating example.
\end{example}

\getkeytheorem[body]{mytag}
```

Example 2. *Fascinating example.* □

Example 2. *Fascinating example.* □

Fascinating example.

`\IfRestatingTF{`⟨*true code*⟩`}{`⟨*false code*⟩`}`
`\IfRestatingT{`⟨*true code*⟩`}`
`\IfRestatingF{`⟨*false code*⟩`}`

Executes ⟨*true code*⟩ if being retrieved with `\getkeytheorem` → P. 17 and ⟨*false code*⟩ if in the original theorem. This is reversed if `store*` is used.

```
\begin{example}[store=hmm]
I am the \IfRestatingTF{restated}{original} example!
\end{example}

\getkeytheorem{hmm}
```

Example 3. *I am the original example!* □

Example 3. *I am the restated example!* □

5.1 Counters within a restated theorem

Counters used within a theorem can be stored and set to their original value when restated. By default, this is done for the `equation` counter, however more counters can be added with `restate-counters` → P. 3. This is necessary if, as is often the case, equations use the `section` or `subsection` counter as part of their numbering.

```
% preamble
\counterwithin{equation}{section}
\keytheoremset{restate-counters={section}}

% document
\begin{theorem}[store=Fermat]
For $n\geq 3$, there are no nontrivial integer solutions to the equation
\begin{equation}
x^n+y^n=z^n.
\end{equation}
\end{theorem}

\getkeytheorem{Fermat}
```

Theorem 9. *For $n \geq 3$, there are no nontrivial integer solutions to the equation*

$$x^n + y^n = z^n. \tag{5.1}$$

Theorem 9. *For $n \geq 3$, there are no nontrivial integer solutions to the equation*

$$x^n + y^n = z^n. \tag{5.1}$$

In this example, removing `\keytheoremset{restate-counters={section}}` would not change the restated equation number because it is being restated in the same section. If that line were removed, the restated equation number would be different (wrong) if restated in a different section.

5.2 Restating theorems from an external file

`\externaltheorems[`⟨*prefix*⟩`]{`⟨*file name*⟩`}`

This is keytheorems' version of the xr package's `\externaldocument`. It allows the user to restate theorems from another document's `.thlist` file. Say you have a file `mycoolpaper.tex`,

```
% mycoolpaper.tex
\documentclass{article}
\usepackage{keytheorems}
\newkeytheorem{theorem}
\begin{document}
\begin{theorem}[store=cooltheorem]
My cool theorem.
\end{theorem}
\end{document}
```

and you'd like to restate the theorem with tag `cooltheorem` in another file `myothercoolpaper.tex` with the same numbering as in the original paper. Since your new paper probably also has cool theorems that you may want to tag as `cooltheorem`, you'd like to give all restatable theorems from `mycoolpaper.tex` a prefix when retrieved with `\getkeytheorem` →P. 17, say "`orig:`". Just call `\externaltheorems[orig:]{mycoolpaper}` after loading keytheorems in the new document. Then any stored theorem from `mycoolpaper.tex` can be retrieved with `\getkeytheorem{orig:`⟨*tag*⟩`}`.

```
% myothercoolpaper.tex
\documentclass{article}
\usepackage{keytheorems}
\externaltheorems[orig:]{mycoolpaper}
\newkeytheorem{theorem}
\begin{document}
\getkeytheorem{orig:cooltheorem}
\end{document}
```

It is important that the `theorem` environment is defined in both documents.

6 Listing theorems

`\listofkeytheorems[`⟨*options*⟩`]`

Similar to `\listoffigures` or `\listoftables` but for theorems. For memoir and the AMS classes, keytheorems tries to copy the formatting of these commands as defined by the class. For other classes, manual adjustments to `numwidth` →P. 21 and `indent` →P. 22 may be necessary.

`\keytheoremlistset{`⟨*options*⟩`}`

The ⟨*options*⟩ of `\listofkeytheorems` may also be set globally with this command.

```
\listofkeytheorems
```

List of Theorems

6.1 Keys also defined in thmtools

`ignore={`⟨*comma-list of env names*⟩`}` (initially unset)

Environments in ⟨*comma-list*⟩ are filtered out from the list of theorems.

`ignoreall` (initially unset)

Applies `ignore` to all theorems. This is usually followed by the keys `show`→ P. 21, `onlynamed`→ P. 21, and/or `onlynumbered`→ P. 23.

```
\listofkeytheorems[ignoreall,show=theorem]
\listofkeytheorems[
  ignoreall, show=conjecture,
  title=List of Conjectures
  ]
```

List of Theorems

List of Conjectures

`numwidth=`⟨*length*⟩ (initially `2.3em`)

Sets the width that theorem numbers occupy in the list of theorems. This is needed when theorems have long parent counter representations, e.g., Theorem 3.7.4.1. For the AMS classes, this is initially `1.5pc`.

`onlynamed={`⟨*comma-list of env names*⟩`}` (initially unset)

Environments in ⟨*comma-list*⟩ are only printed if the theorem was given an optional note. If no value is given, then this applies to all theorems.

`show={`⟨*comma-list of env names*⟩`}` (initially all theorems)

The counterpart to `ignore`→P. 20. Environments in ⟨*comma-list*⟩ are shown in the list of theorems. Note that this is only needed if a theorem type was previously ignored, usually with `ignoreall`→P. 20.

`showall` (initially set)

Applies `show` to all theorems.

`swapnumber=true|false` (initially `false`)

Swaps the position of the theorem name and number in the list of theorems. This will override `format-code`→P. 22 if true.

```
\listofkeytheorems[ignoreall,show=lemma]
\listofkeytheorems[ignoreall,show=lemma,swapnumber]
```

List of Theorems

List of Theorems

`title`=⟨*text*⟩ (initially `\GetTranslation{keythms_listof_title}`)

Defaults to "List of Theorems" if English or an unknown language is used. Currently several European languages have (likely inaccurate!) translations. A translation can be added with a GitHub pull request or manually with

`\DeclareTranslation{`⟨*lang*⟩`}{keythms_listof_title}{`⟨*text*⟩`}`.

6.2 Keys added by keytheorems

`format-code`=⟨*code with* `#1`, `#2`, *and* `#3`⟩ (initially `\numberline{#2}#1#3`)

Allows full control over the format for list entries. The theorem name is `#1`, the number is `#2`, and the (formatted) note is `#3`. The note formatting is still controlled by `note-code` → P. 23.

`indent`=⟨*length*⟩ (initially `1.5em`)

Sets the left indent of items in the list of theorems. For memoir and the AMS classes, the indent is initially `0pt`. It is not recommended to change this unless your class has different defaults not already covered.

`no-chapter-skip=true|false` (initially `false`)

By default a small vertical space is inserted between each chapter's chunk of theorems. Setting this key to `true` removes this space.

`chapter-skip-length`=⟨*length*⟩ (initially `10pt`)

Controls the amount of space inserted between chunks of theorems belonging to each chapter.

`no-continues=true|false` (initially `false`)

Suppresses the printing of theorems given the `continues` → P. 5 key in the list of theorems.

`no-title=true|false` (initially `false`)

Suppresses the title of the list of theorems. This is useful for custom ordering of the list.

```
\keytheoremlistset{ignoreall}
\listofkeytheorems[show=example]
\listofkeytheorems[show=solution,no-title]
```

List of Theorems

`no-toc=true|false` (initially `false`)

With the standard classes, lists of figures/tables are not added to the table of contents by default. The same is true for `\listofkeytheorems`, and with those classes this key does nothing. However some classes, notably memoir and the AMS classes, do add lists to the table of contents. With these classes, this key suppresses the addition of the list of theorems to the table of contents.

`note-code=`⟨*code with* `#1`⟩ (initially `{␣(#1)}`)

Formats the optional note in the list of theorems.

`one-col=true|false` (initially `false`)

In `twocolumn` mode, some classes such as the standard book typeset lists of figures, etc., in one column. keytheorems does not do this by default; this key forces the list into `onecolumn` mode. Note that this is *not* compatible with the application mentioned under `no-title` →P. 22 since switching between `\onecolumn` and `\twocolumn` starts a new page. You can, however, achieve the same effect manually:

```
\keytheoremlistset{ignoreall}
\onecolumn
\listofkeytheorems[show=example]
\listofkeytheorems[show=solution,no-title]
\twocolumn
```

If `no-chapter-skip` →P. 22 is false (the default), you may want to add some vertical space between the two lists.

`onlynumbered={`⟨*comma-list of env names*⟩`}` (initially unset)

Similar to `onlynamed` →P. 21, but lists only those theorems which are numbered. This is useful if you'd like to exclude things like unnumbered definitions and remarks from the list of theorems.

`print-body` (initially unset)

Instead of listing the theorem headings, the theorems are restated with their body text. Not very useful without the `store-all` →P. 3 load-time option.

`seq=`⟨*name*⟩ (initially unset)

Used to list only the theorems added to the custom sequence ⟨*name*⟩ with the `seq` →P. 8 theorem key. This is the only way to fully customize which theorems appear in the list of theorems. Unlike with `show` →P. 21, you do not need to use `ignoreall` →P. 20 to prevent theorems not in ⟨*name*⟩ from being printed.

`title-code=`⟨*code with* `#1`⟩ (initially `\section*{#1}`)

If `\chapter` is defined, then initially this is instead `\chapter*{#1}`. This key has no effect if used with an AMS class because these classes hard-code the section heading into `\@starttoc`.

6.3 Adding code to list of theorems

`\addtheoremcontentsline{`⟨*level*⟩`}{`⟨*text*⟩`}`

This command is analogous to `\addcontentsline` and has the same usage, that is, manually adding entries to the list of theorems.

`\addtotheoremcontents{`⟨*code*⟩`}`

This command is analogous to `\addtocontents` and has the same usage, that is, manually adding arbitrary code to the list of theorems.

! You *must* use these commands rather than the analogous ones mentioned because the `.thlist` file is also used to define restated theorems and cannot contain unexpected code.

7 Theorem hooks

`\addtotheoremhook[`⟨*env name*⟩`]{`⟨*hook name*⟩`}{`⟨*code*⟩`}`

The ⟨*hook name*⟩ can be `prehead`, `posthead`, `prefoot`, `postfoot`, or `restated`. If no ⟨*env name*⟩ is given, the ⟨*code*⟩ is added to the "generic" hook, i.e., applied to all theorems. As in thmtools, the order of hooks is as follows:

⟨*env name*⟩ `prehead` → generic `prehead` → `\begin{`⟨*env name*⟩`}` → ⟨*env name*⟩ `posthead` → generic `posthead` → ⟨*theorem body*⟩ → generic `prefoot` → ⟨*env name*⟩ `prefoot` → `\end{`⟨*env name*⟩`}` → generic `postfoot` → ⟨*env name*⟩ `postfoot`

The `restated` hook is applied at the start of theorems retrieved with the command `\getkeytheorem`, after the `prehead` hook. This can be useful for disabling commands such as `\footnote` in the restated theorems, e.g.,

```
\addtotheoremhook{restated}{\renewcommand\footnote[2][]{}}
```

By default, the `restated` hook disables the `\glossary`, `\index`, `\label`, and `\RecordProperties` commands.

In thmtools, the `prefoot` and `postfoot` hooks always prepend code, i.e., the code

```
\addtotheorempostfoothook{A}
\addtotheorempostfoothook{B}
```

results in `BA` after the theorem. With keytheorems, code is added in the order declared, meaning

```
\addtotheoremhook{postfoot}{A}
\addtotheoremhook{postfoot}{B}
```

results in `AB` after the theorem. This is the behavior of the LaTeX kernel hooks that keytheorems uses under the hood.

Code added using the hook keys `preheadhook` → P. 10, etc. is outermost, meaning executed first in `prehead` and `posthead` and last in `prefoot` and `postfoot`. Furthermore, if present, the `qed` → P. 10 symbol is placed *before* the `prefoot` hook.

8 Miscellaneous notes

8.1 beamer support

The package contains some *highly experimental* code to support theorems with beamer, including overlays. The support differs based on whether or not the beamer class option `noamsthm` is used.

User feedback is necessary to make this code fully compatible. Please report issues on the Github page!

8.1.1 Without `noamsthm`

Most style keys are disabled by the `default` beamer theorem template. More become functional by setting

```
\setbeamertemplate{theorems}[ams style]
```

in the preamble. For full control of theorems, load the class option `noamsthm` and see section 8.1.2.

Note that by default beamer defines a set of theorems when the class is loaded. These can be overwritten with `\renewkeytheorem` →P. 4 or disabled entirely with the `notheorems` class option.

Due to complications with overlays, writing contents of theorems to the `thlist` file is disabled. This means theorems can only be restated *after* their original statement. Furthermore, `\listofkeytheorems` →P. 19 is disabled and a warning issued if used.

8.1.2 With `noamsthm`

If `noamsthm` is loaded, you can use keytheorems exactly as you would in any other document except that `\listofkeytheorems` →P. 19 is disabled. This includes no limitations on storing and restating theorems that exist without `noamsthm`.

With or without `noamsthm`, overlays are supported with syntax

`\begin{`⟨*theorem name*⟩`}<`⟨*overlay spec*⟩`>[`⟨*options*⟩`]`
⟨*contents*⟩
`\end{`⟨*theorem name*⟩`}`

8.2 Support for other classes

As mentioned in section 4, the initial style key values set by keytheorems are adjusted for the AMS classes amsart, amsbook, and amsproc, the amsart-based acmart and aomart, and jlreq. You can find the exact changed values in the support files `keythms-`⟨*class*⟩`-support.tex`.

These class support files also contain code to adapt to class' formatting of lists-of as mentioned in section 6; changes are made for the AMS classes, memoir, IEEEtran, and jlreq.

Lastly, in addition to the beamer support outlined in section 8.1, support is provided for overlays in the experimental tagging-compatible presentation class ltx-talk. See section 8.4 for more details.

8.3 Support for font packages

Some font packages, all by Michael Sharpe, offer a `theoremfont` option that redefines the `plain` style body font to have italic text with upright figures, punctuation, and delimiters. keytheorems detects this option and sets its initial style values accordingly. The supported packages are baskervillef, cochineal, libertinust1math, newpxtext, newtxtext, scholax, stickstootext, and XCharter.

8.4 Support for tagged PDF

The LaTeX team has been working hard to support the creation of tagged PDFs (see https://latex3.github.io/tagging-project/). In the current format, amsthm is compatible with the kernel tagging code. Most of keytheorems is supported too, and anything that doesn't work should be reported. Explicitly not supported are the tcolorbox →P. 13 and tcolorbox-no-titlebar →P. 13 keys.

To produce a tagged PDF, add \DocumentMetadata in the first line of your document (additional instructions are found on the Tagging Project website). An example invocation might look like

```
\DocumentMetadata
  {
    lang=en-US,
    pdfversion=2.0,
    pdfstandard=ua-2,
    tagging=on,
    % tagging-setup={math/setup=mathml-SE}, % optional
  }
```

As mentioned above, support is also provided for the tagging-compatible ltx-talk presentation class. You may use keytheorems exactly as you would in a normal tagged document, with the added support of overlays. The syntax for overlays is the same as for beamer, shown in section 8.1.2. Also as for beamer, the \listofkeytheorems →P. 19 command is disabled.

8.5 subfiles compatibility

Because of how the subfiles package works, it is not possible to restate theorems from one subfile in another without an extra step. The issue boils down to the same reason the xr package is needed for labels and references to work between subfiles. Thankfully, the fix is fairly easy with \externaltheorems →P. 19. Just add \externaltheorems[⟨*prefix*⟩]{⟨*main file name*⟩} to the preamble of your main file. Then, after compiling your main file at least once, theorems can be restated in subfiles with \getkeytheorem{⟨*prefix*⟩⟨*tag*⟩}.

Here is a complete example.

```
% mainfile.tex
\documentclass{article}
\usepackage{keytheorems}
\usepackage{subfiles}

\externaltheorems[main:]{mainfile}
\newkeytheorem{theorem}

\begin{document}
\subfile{subfile1.tex}
\subfile{subfile2.tex}
\end{document}

% subfile1.tex
\documentclass[mainfile.tex]{subfiles}
\begin{document}
\section{subfile1}
\begin{theorem}
```

```
Some theorem from the first subfile.
\end{theorem}
\getkeytheorem{main:secondthm}
\end{document}

% subfile2.tex
\documentclass[mainfile.tex]{subfiles}
\begin{document}
\section{subfile2}
\begin{theorem}[store=secondthm]
Some theorem from the second subfile.
\end{theorem}
\end{document}
```

8.6 Public coding interfaces

`\l_keythms_thmuse_envname_tl`

Inside theorem environments and in all theorem hooks, you have access to the theorem's environment and counter name in this token list variable.

`\keythms_getthm_theorem:nnnnn` {⟨*name*⟩} {⟨*number*⟩} {⟨*restate counters*⟩} {⟨*keys*⟩} {⟨*body text*⟩}
`\keythms_getthm_body:nnn` {⟨*name*⟩} {⟨*restate counters*⟩} {⟨*body text*⟩}

These are the commands called by `\getkeytheorem` → P. 17 when restating stored theorems. They can be useful for customizing the behavior of the list of theorems with a printed body; see `\keythms_listof_listcmd:nnnnnnn`.

`\keythms_listof_listcmd:nnnnnnn` {⟨*name*⟩} {⟨*number*⟩} {⟨*Href*⟩} {⟨*page*⟩} {⟨*restate counters*⟩} {⟨*keys*⟩} {⟨*body text*⟩}

This is the command that controls how theorems are printed in the list of theorems. See `https://tex.stackexchange.com/a/747257/208544` for a concrete use-case in tandem with `\keythms_getthm_theorem:nnnnn`.

keytheorems/allthms/⟨*hook name*⟩
keytheorems/⟨*envname*⟩**/**⟨*hook name*⟩

These are the "real" names for the hooks described in section 7. They can be useful with `\AddToHookNext` or the kernel's label mechanism for hooks.

9 Further examples

More examples will be added soon – rather, eventually… For now, you can find a keytheorems adaptation of amsthm's classic `thmtest.tex` in the Github `tests` folder: `keytheorems-amsthmtest.tex`. There is also a version for tagged PDF: `tagged-keytheorems-amsthmtest.tex`.

Index