keytheorems package

version 0.2.2

github.com/mbertucci47/keytheorems

Matthew Bertucci

November 11, 2024

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. 11 and `tcolorbox-no-titlebar`→P. 11 keys require tcolorbox, and the `numbered=unless-unique`→P. 7 key requires the unique package. A LaTeX kernel no older than 2023-06-01 is required; if older than 2024-06-01, nameref 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. 3
`\declaretheoremstyle`	→	`\newkeytheoremstyle`→P. 12
`\listoftheorems`	→	`\listofkeytheorems`→P. 14
`\listtheoremname`	→	`title`→P. 16 key
`\addtotheorempreheadhook`	→	`\addtotheoremhook`→P. 18
`\addtotheorempostheadhook`		
`\addtotheoremprefoothook`		
`\addtotheorempostfoothook`		
`restatable` environment	→	`store`→P. 5 key
`restatable*` environment	→	`store*`→P. 5 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. 14 and the note produced by the `continues`→P. 4 key. These texts can be manually customized with the `title`→P. 16 and `continues-code`→P. 3 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. 4 key. If English or an unknown language is used, defaults to `continuing from p.\,\pageref{#1}`. Currently (likely inaccurate!) translations exist for French, German, Italian, Portuguese, and Spanish.

`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. 17 option of `\listofkeytheorems`→P. 14. Note that this means a theorem body *cannot* contain verbatim material.

`store-sets-label` (initially unset)

Defines the `store`→P. 5 key to also set `label`→P. 4, i.e. it makes `store=`⟨*tag*⟩ equivalent to `store=`⟨*tag*⟩`,label=`⟨*tag*⟩. Similarly for `store*`→P. 5.

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. 7 key. All ⟨*options*⟩ are described in subsections 3.2 and 3.3.

```
% preamble
\newkeytheorem{theorem}

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

Theorem 1. *Some text*

`\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}[some heading]
Some text
\end{theorem}
```

Theorem 2 (some heading)**.** *Some text*

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=another heading]
Some more text
\end{theorem}
```

Theorem 3 (another heading)**.** *Some more text*

`short-note=⟨text⟩` (initially unset)

Alias `short-name`. This replaces the value of `note` when displayed in the list of theorems (`\listofkeytheorems`→P. 14).

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

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

```
\begin{theorem}[label=foo]
Some text
\end{theorem}
\ref{foo}
```

Theorem 4. *Some text*

4

`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, if it exists.

```
\begin{theorem}[continues=foo]
\dots and some more text.
\end{theorem}
```

Theorem 4 (continuing from p. 4)**.** *…and some more text.*

store*=⟨*tag*⟩ (initially unset)

Alias restate*. Stores the the theorem to be restated at any point in the document with \getkeytheorem→P. 14. 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=diagram]
Some commutative diagram:
\[\begin{tikzcd}[ampersand replacement=\&]
    X\times_S Y \ar[r] \ar[d] \& X \ar[d] \\
    Y \ar[r] \& S
\end{tikzcd}\]
\end{lemma}
\dots
\getkeytheorem{diagram}
```

Lemma 6. *Some commutative diagram:*

$$\begin{tikzcd} X \times_S Y \arrow[r] \arrow[d] & X \arrow[d] \\ Y \arrow[r] & S \end{tikzcd}$$

…

Lemma 6. *Some commutative diagram:*

$$\begin{tikzcd}
X \times_S Y \arrow[r] \arrow[d] & X \arrow[d] \\
Y \arrow[r] & S
\end{tikzcd}$$

`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. 13 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
\end{enumerate}
\end{observation}

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

Observation 1. *1. First item*

Observation 2.

1. First 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`$^{\rightarrow\text{P. 17}}$ for more details.

3.2 Keys also defined in thmtools

These are the `[`⟨*options*⟩`]` available to `\newkeytheorem`. Except for `name` and `style`$^{\rightarrow\text{P. 8}}$, each key below can also be used in `\newkeytheoremstyle`$^{\rightarrow\text{P. 12}}$. For more description, see the thmtools package.

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

Aliases `heading` and `title`.

```
% 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`)

For compatibility with thmtools, 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`.

```
% 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`.

```
% 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. 12, as well as any of the predefined amsthm styles: `plain`, `definition`, and `remark`.

```
% preamble
\newkeytheorem{remark}[style=remark]
% document
\begin{remark}
Some text
\end{remark}
```

Remark 1. Some text

`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 symbol □ is used.

```
% 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 ⟨*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`.

Refname=⟨*ref name*⟩ or {⟨*singular name*⟩,⟨*plural name*⟩} (initially ⟨*display name*⟩)

Same as `refname` 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}
\begin{theorem}
Consider \cref{abc,def}. \Autoref{abc} \dots
\end{theorem}
```

Proposition 1. *Some text*

Proposition 2. *Some more text*

Theorem 9. *Consider propositions 1 and 2. Proposition 1 …*

! 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

`leftmargin`=⟨*length*⟩
`rightmargin`=⟨*length*⟩

Sets the left (respectively, right) margin of the theorem relative to the text width. 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,
  leftmargin=1cm, rightmargin=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}
Some text
\end{corollary}
\begin{definition}[A nice definition]
Some more text
\end{definition}
```

Corollary 1.

Some text

Definition 1 (A nice definition)**.**

Some more text

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}
Some text
\end{boxcor}
```

Corollary 2. *Some text*

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=My Theorem,
  style=tcb-standard]

% document
\begin{mytheo}[This is my title]
Some theorem text
\end{mytheo}
```

My Theorem 1: This is my title

Some theorem text

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. Note that unlike amsthm's `\newtheoremstyle`, this command will error if a style has already been defined. To overwrite an existing style, there is the analogous `\renewkeytheoremstyle`. For completeness, there are also `\providekeytheoremstyle` and `\declarekeytheoremstyle`.

The defined style can be used with either the `style`→P. 8 key or the traditional `\theoremstyle`.

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

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. 3 can be used in `\newkeytheoremstyle`.

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

`break` (initially unset)

Do not use this with the `postheadspace` key.

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

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

Alias `headstyle`. Within ⟨*code*⟩, the commands `\NAME`, `\NUMBER`, and `\NOTE` correspond to the formatted parts of the theorem head.

> **!** In `headformat`, 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`, and `numberfont` will have no effect (of course, you can manually control these things inside the commands' arguments).

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

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

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

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

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

Do not use this with the `break` key.

`spaceabove`=⟨*length*⟩ (initially `\topsep`)

`spacebelow`=⟨*length*⟩ (initially `\topsep`)

> **!** With `tcolorbox`→P. 11 and `tcolorbox-no-titlebar`→P. 11, 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. 12. 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` take effect.

`numberfont`=⟨*font declarations*⟩ (initially)

For almost all theorem styles, it is recommended that you *do not* change this setting.

5 Restating theorems

When a theorem is given the `store`→P. 5 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` 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!* □

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. 16 and `indent`→P. 16 may be necessary.

`\keytheoremlistset{⟨options⟩}`

```
\listofkeytheorems
```

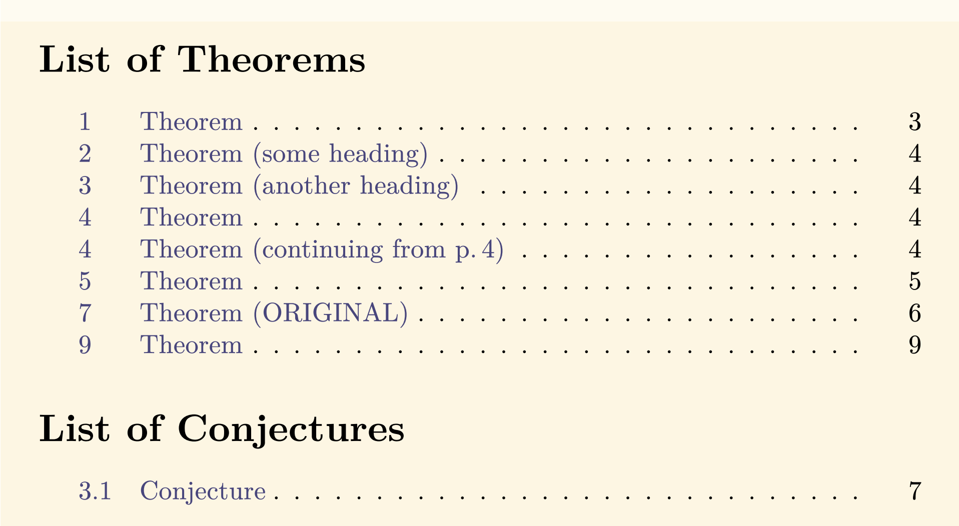
List of Theorems

6.1 Keys also defined in thmtools

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

`ignoreall` (initially unset)

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



List of Theorems

1 Theorem 3
2 Theorem (some heading) 4
3 Theorem (another heading) 4
4 Theorem 4
4 Theorem (continuing from p. 4) 4
5 Theorem 5
7 Theorem (ORIGINAL) 6
9 Theorem 9

List of Conjectures

3.1 Conjecture 7

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

For the AMS classes, this is initially `1.5pc`.

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

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

`showall` (initially set)

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

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

Defaults to "List of Theorems" if English or an unknown language is used. Currently French, German, Italian, Portuguese, and Spanish 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. 17.

`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.

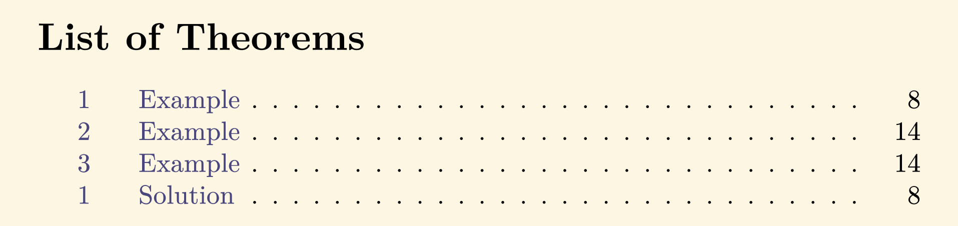
`no-continues=true|false` (initially `false`)

Suppresses the printing of theorems given the `continues`→P. 4 key in the list of theorems.

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

Suppresses the title of the list of theorems. Useful for custom ordering of the list.

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



List of Theorems

1 Example 8
2 Example 14
3 Example 14
1 Solution 8

`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.

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

Similar to `onlynamed`→P. 16, 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. 7 theorem key. This is the only way to fully customize which theorems appear in the list of theorems. Unlike with `show`→P. 16, you do not need to use `ignoreall`→P. 15 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

There are analogous commands to `\addcontentsline` and `\addtocontents` for adding entries or arbitrary code to the list of theorems.

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

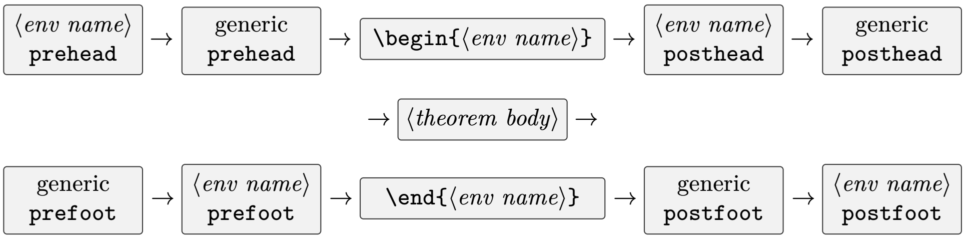
`\addtheoremcontentsline{⟨level⟩}{⟨text⟩}`

`\addtotheoremcontents{⟨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, keytheorems disables the `\label` and `\RecordProperties` commands in restated theorems.

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. 8, etc. is outermost, meaning executed first in `prehead` and `posthead` and last in `prefoot` and `postfoot`.

8 Miscellaneous notes

8.1 beamer support

The package contains some *highly experimental* code to support theorems with beamer, including overlays. Most style keys are disabled by the `default` beamer theorem template. More become functional by setting

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

in the preamble. Alternatively, you have full control of theorems by setting the class option `noamsthm`.

Note that by default beamer defines a set of theorems when the class is loaded. These can be overwritten with `\renewkeytheorem`→P. 3 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. 14 is disabled and a warning issued if used.

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

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.

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, 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/`). The current dev formats make amsthm 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. 11 and `tcolorbox-no-titlebar`→P. 11 keys. There is only partial support for `leftmargin`→P. 10 and `rightmargin`→P. 10, namely only the latter works.

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,
    testphase={phase-III,math,table,title,firstaid}
  }
```

At a minimum, the `testphase` modules `phase-III` and `firstaid` are required. The GitHub tests folder contains an example of a tagged PDF using keytheorems: `tagged-keytheorems-amsthmtest.tex`.

8.5 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.

`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 file `thmtest.tex` in the Github `tests` folder: `keytheorems-amsthmtest.tex`.

Index