keytheorems package

version 0.0.8

github.com/mbertucci47/keytheorems

Matthew Bertucci

May 24, 2024

Abstract

An experimental expl3-implementation of a key-value interface to amsthm, implementing most of the functionality provided by thmtools. Likely contains many bugs; use at your own risk!

Contents

1 Dependencies

Without using the `tcolorbox` → P. 6 or `tcolorbox-no-titlebar` → P. 7 options, the package loads the aliascnt, amsthm, refcount, and translations packages.

2 Global Options

`\keytheoremset{`⟨*options*⟩`}`

Every key in this section can be given as a package option with `\usepackage[`⟨*options*⟩`]{keytheorems}` or in `\keytheoremset{`⟨*options*⟩`}`, with the exception that `continues-code` → P. 2 can only be used in the latter.

`overload` (initially unset)

Redefines `\newtheorem` to internally use the keytheorems machinery. The syntax remains the same.

`thmtools-compat` (initially unset)

For compatibility with thmtools syntax. Currently defines the following commands:

thmtools command	keytheorems replacement
`\declaretheorem`	`\newkeytheorem`
`\declaretheoremstyle`	`\newkeytheoremstyle` → P. 7
`\listoftheorems`	`\listofkeytheorems` → P. 9
`\addtotheorempreheadhook`	
`\addtotheorempostheadhook`	`\addtotheoremhook` → P. 11
`\addtotheoremprefoothook`	
`\addtotheorempostfoothook`	
`restatable` environment	`store` → P. 3 key

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

`store-all` (initially unset)

Tells keytheorems to grab the body of each theorem so it can later be printed with the `print-body` → P. 10 option of `\listofkeytheorems` → P. 9. Note that this means a theorem body *cannot* contain verbatim material.

`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*⟩.

`continues-code=`⟨*code with* `#1`⟩ (initially `\GetTranslation{keythms_continues}\pageref{#1}`)

The code used to typeset the note produced by the `continues` → P. 3 key. If English or an unknown language is used, defaults to `continuing from p.\,\pageref{#1}`. Currently (likely inaccurate!) translations exist for French, German, and Spanish.

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

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

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

If `false`, keytheorems does not automatically translate the title text for `\listofkeytheorems` → P. 9 and the note produced by the `continues` → P. 3 key. These texts can be manually customized with the `title` → P. 10 and `continues-code` keys, respectively.

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. 4 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*

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}[some heading]
Some text
\end{theorem}
\begin{theorem}[note=another heading]
Some more text
\end{theorem}
```

Theorem 3 (some heading)**.** *Some text*

Theorem 4 (another heading)**.** *Some more text*

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

Alias `short-name`. This replaces the value of `note` when displayed in `\listofkeytheorems` → P. 9.

`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 5. *Some text*

5

`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. 2 option. The starred version also copies the theorem note, if it exists.

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

Theorem 5 (continuing from p. 3)**.** *. . . 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. 8.

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

Theorem 6. *A theorem worth restating.*

More brilliant mathematics.

Theorem 6. *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 7. *Some commutative diagram:*

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

...

Lemma 7. *Some commutative diagram:*

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

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

Meant only to be used with the `break` → P. 8 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` → P. 10 for more details.

3.2 Keys inherited from thmtools

These are the `[`⟨*options*⟩`]` available to `\newkeytheorem`. Except for `name` and `style` → P. 5, each key below can also be used in `\newkeytheoremstyle` → P. 7. For more description, see the thmtools package.

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

Aliases `title` and `heading`.

```
% 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. 7, 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

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

If a single string, then the name used by hyperref's `\autoref` and cleveref's `\cref`. 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` and `\Cref`. Note that `\Autoref` is defined by keytheorems, but requires hyperref to work.

```
% 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 ...*

qed=⟨*symbol*⟩ (default `\openbox`, 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* ♣

3.3 Keys added by keytheorems

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 → P. 7.

```
% 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}
Some more text
\end{definition}
```

Corollary 1.

Some text

Definition 1.

Some more text

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

Same usage as tcolorbox → P. 6 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*

4 Theorem Styles

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

This is keytheorems' version of thmtools' \declaretheoremstyle[⟨*options*⟩]{⟨*name*⟩}. 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 is also \declarekeytheoremstyle and \providekeytheoremstyle.

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

4.1 Keys inherited from 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. 2 can be used in \newkeytheoremstyle.

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

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

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

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

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

headpunct=⟨*code*⟩ (initially `{.}`)

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

Do not use this with the `break` key.

break (initially unset)

Do not use this with the `postheadspace` key.

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

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

headstyle=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.

4.2 Keys added by keytheorems

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

Inherit the keys of any style declared with `\newkeytheoremstyle` → P. 7. Additionally, the three styles predefined by amsthm are possible values: `plain`, `definition`, and `remark`.

5 Restating Theorems

When a theorem is given the `store` → P. 3 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*⟩. 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*⟩`}`

Executes ⟨*true code*⟩ if being retrieved with `\getkeytheorem` and ⟨*false code*⟩ if in the original theorem.

```
\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*⟩`]`

`\keytheoremlistset{`⟨*options*⟩`}`

```
\listofkeytheorems
```

List of Theorems

6.1 Keys inherited from thmtools

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

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

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

`onlynamed={`⟨*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

List of Conjectures

`showall` (initially set)

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

Defaults to "List of Theorems" if English or an unknown language is used. Currently French, German, 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*⟩`}`.

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

6.2 Keys added by keytheorems

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

Similar to `onlynamed` → P. 9, 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.

`seq=`⟨*name*⟩ (initially unset)

Used to list only the theorems added to the custom sequence ⟨*name*⟩ with the `seq` → P. 4 theorem key. This is the only way to fully customize which theorems appear in the list of theorems.

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

If `\chapter` is defined, then initially this is instead `\chapter*{#1}`.

`no-title` (initially unset)

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

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

Formats the optional note in 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. 2 load-time option.

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

Suppresses the printing of theorems given the `continues` → P. 3 key in the list of theorems.

`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=`⟨*dimension*⟩ (initially `10pt`)

Controls the amount of space inserted between chunks.

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*⟩`}`

⟨*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 `\getkeytheorem`, before 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.

Right now, code added using the hook keys `preheadhook` →P. 5, etc. is outermost, meaning executed first in `prehead` and `posthead` and last in `prefoot` and `postfoot`. This may change if I think of good reasons to do so...

8 Implementation

```
\ProvidesExplPackage{keytheorems}{2024-05-24}{0.0.8}{l3keys interface to amsthm}

%%% TESTING
 \debug_on:n { all }
%%% END TESTING

\RequirePackage{aliascnt} % for sibling theorems
\RequirePackage{amsthm}
% ^ ams classes have way of ignoring this so don't need to check if they're loaded
\RequirePackage{refcount} % for \getrefnumber
\RequirePackage{translations} % for translating "List of Theorems"

%%%%%%%%%%%%%%%%%%%%%%%%
%%% Error Messages %%%
%%%%%%%%%%%%%%%%%%%%%%%%

\msg_new:nnn { keytheorems } { thmtools-before }
  {
    keytheorems~is~not~compatible~with~thmtools.~
    Try~replacing~\protect\usepackage{thmtools}~with~
    \protect\usepackage[thmtools-compat]{keytheorems}.
  }
\msg_new:nnn { keytheorems } { thmtools-after }
  {
    keytheorems~is~not~compatible~with~thmtools.~
    This~will~not~work~as~you~think!~
    Try~replacing~\protect\usepackage{thmtools}~with~
    \protect\usepackage[thmtools-compat]{keytheorems}.
  }
\msg_new:nnn { keytheorems } { no-stored-theorem }
  {
    No~stored~theorem~'#1'~found!~
    Try~compiling~again.~If~that~doesn't~work,~
    check~the~spelling~of~'#1'.
  }
\msg_new:nnn { keytheorems } { undefined-thm-hook }
  {
    No~theorem~hook~'#1'.~Check~the~spelling.~
    Should~be~one~of~'prehead',~'posthead',~'prefoot',~'postfoot',~or'restated'.
  }
\msg_new:nnn { keytheorems } { no-Autorefname }
  {
    No~Autoref~name~for~'#1'.
  }
\msg_new:nnn { keytheorems } { thmstyle-undefined }
  {
    Theorem~style~'#1'~undefined.~
    Use~\protect\newkeytheoremstyle\space instead.
  }
\msg_new:nnn { keytheorems } { thmstyle-defined }
  {
    Theorem~style~'#1'~already~defined.~
    Use~\protect\renewkeytheoremstyle\space instead.
  }

% Error if thmtools loaded since compilation hangs.
```

```
% If loaded thmtools loaded after, produce warning.
\IfPackageLoadedTF { thmtools }
  {
    \msg_fatal:nn { keytheorems } { thmtools-before }
  }
  {
    \hook_gput_code:nnn { package/thmtools/before } { . }
      {
        \msg_warning:nn { keytheorems } { thmtools-after }
      }
  }

%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Declare Variables %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

\tl_new:N \l__keythms_tmpa_tl

\bool_new:N \g__keythms_listof_writefile_bool
\bool_gset_false:N \g__keythms_listof_writefile_bool
\bool_new:N \l__keythms_thm_numbered_bool
\bool_new:N \l__keythms_thm_unlessunique_bool
\bool_new:N \l__keythms_thmuse_listhack_bool
\bool_new:N \l__keythms_thmuse_restating_bool
\clist_new:N \g__keythms_restatecounters_clist
\clist_new:N \l__keythms_thmstyle_savedkeys_clist
\iow_new:N \g__keythms_listof_stream
\prop_new:N \g__keythms_thmnames_prop
\prop_new:N \g__keythms_thmuse_othercounters_prop
\prop_new:N \l__keythms_restate_counters_prop
\tl_new:N \l__keythms_thm_currentthmstyle_tl
\tl_new:N \l__keythms_thm_defaultkeys_tl
\tl_new:N \l__keythms_thm_envname_tl
\tl_new:N \l__keythms_thmstyle_defaultkeys_tl
\tl_new:N \l__keythms_thmstyle_lnotebrace_tl
\tl_new:N \l__keythms_thmstyle_rnotebrace_tl
\tl_new:N \l__keythms_thmuse_envname_tl
\tl_new:N \g__keythms_thmuse_temprestatedata_tl

\newcounter{keythms_restate_dummyctr}
\cs_gset:Npn \theHkeythms_restate_dummyctr
  { restate.\arabic{keythms_restate_dummyctr} }
\cs_gset:Npn \thekeythms_restate_dummyctr { }
\newcounter{keythms_continues_dummyctr}
\cs_gset:Npn \theHkeythms_continues_dummyctr
  { continues.\arabic{keythms_continues_dummyctr} }
\cs_gset:Npn \thekeythms_continues_dummyctr { }
\newcounter{keythms_unnumbered_dummyctr}
\cs_gset:Npn \theHkeythms_unnumbered_dummyctr
  { unnumbered.\arabic{keythms_unnumbered_dummyctr} }
\cs_gset:Npn \thekeythms_unnumbered_dummyctr { }

\cs_generate_variant:Nn \hook_gput_code:nnn { nnV }
\cs_generate_variant:Nn \keys_precompile:nnN { nv, nVc }

% for detecting ams classes
\prg_new_conditional:Npnn \keythms_if_amsclass: { T, TF }
  {
```

```
    \IfClassLoadedTF { amsart } { \prg_return_true: }
      {
        \IfClassLoadedTF { amsbook } { \prg_return_true: }
          {
            \IfClassLoadedTF { amsproc } { \prg_return_true: }
              { \prg_return_false: }
          }
      }
  }

%%%%%%%%%%%%%%
%%% Styles %%%
%%%%%%%%%%%%%%

% \__keythms_thmstyle_setbraces:nn { <left brace> } { <right brace> }
\cs_new_protected:Npn \__keythms_thmstyle_setbraces:nn #1#2
  {
    \tl_set:Nn \l__keythms_thmstyle_lnotebrace_tl { #1 }
    \tl_set:Nn \l__keythms_thmstyle_rnotebrace_tl { #2 }
  }
\cs_new:Npn \keythms_thmstyle_savethmkey_reqval:n #1
  {
    \clist_put_right:No \l__keythms_thmstyle_savedkeys_clist
      { \l_keys_key_str = { #1 } }
  }
\cs_new:Npn \keythms_thmstyle_savethmkey_optval:n #1
  {
    \tl_if_empty:NTF \l_keys_value_tl
      {
        \clist_put_right:No \l__keythms_thmstyle_savedkeys_clist
          { \l_keys_key_str }
      }
      {
        \clist_put_right:No \l__keythms_thmstyle_savedkeys_clist
          { \l_keys_key_str = { #1 } }
      }
  }

\keys_define:nn { keytheorems/thmstyle }
  {
    spaceabove    .tl_set:N = \l__keythms_thmstyle_spaceabove_tl,
    spacebelow    .tl_set:N = \l__keythms_thmstyle_spacebelow_tl,
    bodyfont      .tl_set:N = \l__keythms_thmstyle_bodyfont_tl,
    headindent    .tl_set:N = \l__keythms_thmstyle_headindent_tl,
    headfont      .tl_set:N = \l__keythms_thmstyle_headfont_tl,
    headpunct     .tl_set:N = \l__keythms_thmstyle_headpunct_tl,
    postheadspace .tl_set:N = \l__keythms_thmstyle_postheadspace_tl,
    break         .meta:n   = { postheadspace = \newline }, % add error if postheadspace set
    break         .value_forbidden:n = true,
    notefont      .tl_set:N = \l__keythms_thmstyle_notefont_tl,
    notebraces    .code:n   = \exp_after:wN \__keythms_thmstyle_setbraces:nn #1,
    headstyle     .choice:,
    headstyle / margin .code:n =
      {
        \cs_set:Nn \keythms_thmstyle_headcmd:nnn
          { \makebox[0pt][r]{\NUMBER\ }\NAME\NOTE }
      },
    headstyle / swapnumber .code:n =
```

```
      {
        \cs_set:Nn \keythms_thmstyle_headcmd:nnn { \NUMBER\ \NAME\NOTE }
      },
    headstyle / unknown .cs_set:Np = \keythms_thmstyle_headcmd:nnn #1#2#3,
    headformat    .meta:n = { headstyle = #1 },
    inherit-style .choice:,
    inherit-style / plain .meta:n = {},
    inherit-style / definition .meta:n = { bodyfont = \normalfont },
    inherit-style / remark .meta:n =
      {
        headfont = \itshape,
        bodyfont = \normalfont,
        spaceabove = 0.5\topsep,
        spacebelow = 0.5\topsep,
      },
    % thm keys that are saved for later
    numbered      .code:n = \keythms_thmstyle_savethmkey_optval:n { #1 },
    parent        .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    numberwithin  .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    within        .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    sibling       .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    numberlike    .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    sharenumber   .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    preheadhook   .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    postheadhook  .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    prefoothook   .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    postfoothook  .code:n = \keythms_thmstyle_savethmkey_reqval:n { #1 },
    qed           .code:n = \keythms_thmstyle_savethmkey_optval:n { #1 },
    tcolorbox     .code:n = \keythms_thmstyle_savethmkey_optval:n { #1 },
    tcolorbox-no-titlebar .code:n = \keythms_thmstyle_savethmkey_optval:n { #1 },
  }

\cs_new_protected:Nn \keythms_thmstyle_thmname:n { \thmname{#1} }
\cs_new_protected:Nn \keythms_thmstyle_thmnumber:n { \thmnumber{#1} }
\cs_new_protected:Nn \keythms_thmstyle_thmnote:n { \thmnote{#1} }

%% NOTE: if these are used, user is in charge of spacing with \NAME and \NUMBER
%% QUESTION: should these be moved into def of \newkeytheoremstyle?
\cs_new:Npn \NAME { \keythms_thmstyle_thmname:n { ##1 } }
\cs_new:Npn \NUMBER
  {
    \keythms_thmstyle_thmnumber:n { \textup { ##2 } }
  }
\cs_new:Npn \NOTE
  {
    \keythms_thmstyle_thmnote:n
      { ~ \group_begin: % group so notefont doesn't affect headpunct
        \exp_not:V \l__keythms_thmstyle_notefont_tl
        \l__keythms_thmstyle_lnotebrace_tl ##3 \l__keythms_thmstyle_rnotebrace_tl
        \group_end:
      }
  }

\cs_new:Npn \keythms_thmstyle_headcmd_default:nnn #1#2#3
  {
    \keythms_thmstyle_thmname:n { #1 }
    \keythms_thmstyle_thmnumber:n
      { \tl_if_empty:nF{#1}{~} \exp_not:N \textup{#2} }
```

```
      % ^ this \tl_if_empty has no effect...
    \keythms_thmstyle_thmnote:n
      { ~ \group_begin: % group so notefont doesn't affect headpunct
        \exp_not:V \l__keythms_thmstyle_notefont_tl
        \l__keythms_thmstyle_lnotebrace_tl #3 \l__keythms_thmstyle_rnotebrace_tl
        \group_end:
      }
  }

%%% <SURELY A BETTER WAY>
\cs_new_protected:Npn \__keythms_thmstyle_definekeylist:nn #1#2
  {
    \clist_const:cn { c__keythms_thmstyle_defaultkeys_ #1 _clist } { #2 }
  }

\cs_new_protected:Npn \__keythms_thmstyle_setdefaultkeys:n #1
  {
    \keys_precompile:nvN { keytheorems/thmstyle }
      { c__keythms_thmstyle_defaultkeys_ #1 _clist }
      \l__keythms_thmstyle_defaultkeys_tl
  }

\__keythms_thmstyle_definekeylist:nn { default }
  {
    spaceabove    = \topsep,
    spacebelow    = \topsep,
    bodyfont      = \itshape,
    headindent    = 0pt,
    headfont      = \bfseries,
    headpunct     = {.},
    postheadspace = 5pt plus 1pt minus 1pt,
    notefont      = \fontseries\mddefault\upshape,
    notebraces    = {(}{)},
    headstyle     = \keythms_thmstyle_headcmd_default:nnn{#1}{#2}{#3},
  }
\__keythms_thmstyle_definekeylist:nn { amsart }
  {
    spaceabove    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    spacebelow    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    bodyfont      = \itshape,
    headindent    = 0pt,
    headfont      = \bfseries,
    headpunct     = {.},
    postheadspace = 5pt plus 1pt minus 1pt,
    notefont      = \fontseries\mddefault\upshape,
    notebraces    = {(}{)},
    headstyle     = \keythms_thmstyle_headcmd_default:nnn{#1}{#2}{#3},
  }
\__keythms_thmstyle_definekeylist:nn { amsproc }
  {
    spaceabove    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    spacebelow    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    bodyfont      = \itshape,
    headindent    = \parindent,
    headfont      = \scshape,
    headpunct     = {.},
    postheadspace = 5pt plus 1pt minus 1pt,
    notefont      = \fontseries\mddefault\upshape,
```

```
    notebraces    = {(}{)},
    headstyle     = \keythms_thmstyle_headcmd_default:nnn{#1}{#2}{#3},
  }
\__keythms_thmstyle_definekeylist:nn { amsbook }
  {
    spaceabove    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    spacebelow    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    bodyfont      = \itshape,
    headindent    = \parindent,
    headfont      = \scshape,
    headpunct     = {.},
    postheadspace = 5pt plus 1pt minus 1pt,
    notefont      = \fontseries\mddefault\upshape,
    notebraces    = {(}{)},
    headstyle     = \keythms_thmstyle_headcmd_default:nnn{#1}{#2}{#3},
  }
\__keythms_thmstyle_definekeylist:nn { acmart }
  {
    spaceabove    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    spacebelow    = .5\baselineskip plus .2\baselineskip minus .2\baselineskip,
    bodyfont      = \@acmplainbodyfont,
    headindent    = \@acmplainindent,
    headfont      = \@acmplainheadfont,
    headpunct     = {.},
    postheadspace = .5em,
    notefont      = \@acmplainnotefont,
    notebraces    = {(}{)},
    headstyle     = \keythms_thmstyle_headcmd_default:nnn{#1}{#2}{#3},
  }

\IfClassLoadedTF { amsart }
  {
    \IfClassLoadedTF { acmart } % acmart loads amsart
      {
        \__keythms_thmstyle_setdefaultkeys:n { acmart }
      }
      {
        \__keythms_thmstyle_setdefaultkeys:n { amsart }
        \keys_define:nn { keytheorems/thmstyle }
          {
            inherit-style / remark .meta:n =
              {
                headfont = \itshape,
                bodyfont = \normalfont,
              }
          }
      }
  }
  {
    \IfClassLoadedTF { amsbook }
      {
        \__keythms_thmstyle_setdefaultkeys:n { amsbook }
        \keys_define:nn { keytheorems/thmstyle }
          {
            inherit-style / remark .meta:n =
              {
                bodyfont = \normalfont,
              },
```

```
            }
        }
        {
          \IfClassLoadedTF { amsproc }
            {
              \__keythms_thmstyle_setdefaultkeys:n { amsproc }
              \keys_define:nn { keytheorems/thmstyle }
                {
                  inherit-style / remark .meta:n =
                    {
                      bodyfont = \normalfont,
                    },
                }
            }
            {  \__keythms_thmstyle_setdefaultkeys:n { default } }
        }
  }
%%% </SURELY A BETTER WAY>

\NewDocumentCommand \newkeytheoremstyle { m m }
  {
    \cs_if_free:cTF { th@ #1 }
      { \keythms_thmstyle_declarestyle:nn { #1 } { #2 } }
      { \msg_error:nnn { keytheorems } { thmstyle-defined } { #1 } }
  }
\NewDocumentCommand \renewkeytheoremstyle { m m }
  {
    \cs_if_free:cTF { th@ #1 }
      { \msg_error:nnn { keytheorems } { thmstyle-undefined } { #1 } }
      { \keythms_thmstyle_declarestyle:nn { #1 } { #2 } }
  }
\NewDocumentCommand \providekeytheoremstyle { m m }
  {
    \cs_if_free:cT { th@ #1 }
      { \keythms_thmstyle_declarestyle:nn { #1 } { #2 } }
  }
\NewDocumentCommand \declarekeytheoremstyle { m m }
  {
    \keythms_thmstyle_declarestyle:nn { #1 } { #2 }
  }

\@onlypreamble \newkeytheoremstyle
\@onlypreamble \renewkeytheoremstyle
\@onlypreamble \providekeytheoremstyle
\@onlypreamble \declarekeytheoremstyle

\cs_new_eq:NN \keythms_thmstyle_new:nnnnnnnnn \newtheoremstyle
\cs_generate_variant:Nn \keythms_thmstyle_new:nnnnnnnnn { nVVVVVVVe }

\cs_new_protected:Npn \keythms_thmstyle_declarestyle:nn #1#2
  {
    \clist_clear:N \l__keythms_thmstyle_savedkeys_clist
    \tl_use:N \l__keythms_thmstyle_defaultkeys_tl
    \keys_set:nn { keytheorems/thmstyle } { #2 }
    \keythms_thmstyle_new:nVVVVVVe { #1 }
      \l__keythms_thmstyle_spaceabove_tl
      \l__keythms_thmstyle_spacebelow_tl
      \l__keythms_thmstyle_bodyfont_tl
```

```
      \l__keythms_thmstyle_headindent_tl
      \l__keythms_thmstyle_headfont_tl
      \l__keythms_thmstyle_headpunct_tl
      \l__keythms_thmstyle_postheadspace_tl
      { \text_expand:n { \keythms_thmstyle_headcmd:nnn{##1}{##2}{##3} } }
    % Define new inherit-style key
    \keys_define:nn { keytheorems/thmstyle }
      { inherit-style / #1 .meta:n = { #2 } }
    \tl_if_exist:cF { l__keythms_thmstyle_ #1 _savedkeys_tl }
      { \tl_new:c { l__keythms_thmstyle_ #1 _savedkeys_tl } }
    \keys_precompile:nVc { keytheorems/thm }
      \l__keythms_thmstyle_savedkeys_clist
      { l__keythms_thmstyle_ #1 _savedkeys_tl }
  }

%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Defining Theorems %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

% FIX: reimplement these without \NewDocumentCommand and \SplitArgument

% \keythms_thm_setrefnames:n { <envname> } { <refname> or <sing,plural> }
\NewDocumentCommand \keythms_thm_setrefnames:nn
  { m >{\SplitArgument{1}{,}} m }
  { \__keythms_thm_setrefnames_aux:nnn{#1}#2 }
\cs_new_protected:Npn \__keythms_thm_setrefnames_aux:nnn #1#2#3
  {
    \cs_set:cpn { #1 autorefname } { #2 }
    \IfPackageLoadedTF { cleveref }
      {
        \tl_if_novalue:nTF { #3 }
          { \crefname{#1}{#2}{\textbf{??~(pl.~#2)}} }
          { \crefname{#1}{#2}{#3} }
      }
      { }
  }
\cs_generate_variant:Nn \keythms_thm_setrefnames:nn { nV }

% \keythms_thm_setRefnames:n { <envname> } { <refname> or <sing,plural> }
\NewDocumentCommand \keythms_thm_setRefnames:nn
  { m >{\SplitArgument{1}{,}} m }
  { \__keythms_thm_setRefnames_aux:nnn{#1}#2 }
\cs_new_protected:Npn \__keythms_thm_setRefnames_aux:nnn #1#2#3
  {
    \cs_set:cpn { #1 Autorefname } { #2 }
    \IfPackageLoadedTF { cleveref }
      {
        \tl_if_novalue:nTF { #3 }
          { \Crefname{#1}{#2}{\textbf{??~(pl.~#2)}} }
          { \Crefname{#1}{#2}{#3} }
      }
      { }
  }
\cs_generate_variant:Nn \keythms_thm_setRefnames:nn { nV }

\keys_define:nn { keytheorems/thm }
  {
    name           .tl_set:N  = \l__keythms_thm_name_tl,
```

```
    title           .meta:n    = { name = #1 },
    heading         .meta:n    = { name = #1 },
    refname         .tl_set:N  = \l__keythms_thm_refname_tl,
    Refname         .tl_set:N  = \l__keythms_thm_Refname_tl,
    numbered        .choice:,
    numbered / true .code:n    = \bool_set_true:N \l__keythms_thm_numbered_bool,
    numbered / false .code:n   = \bool_set_false:N \l__keythms_thm_numbered_bool,
    numbered / yes .meta:n     = { numbered = true },
    numbered / no  .meta:n     = { numbered = false },
    numbered / unless-unique .code:n =
      {
        \bool_set_true:N \l__keythms_thm_unlessunique_bool
      },
    numbered / unless~unique .meta:n = { numbered = unless-unique },
    numbered        .default:n = true,
    parent          .tl_set:N  = \l__keythms_thm_parent_tl,
    numberwithin    .meta:n    = { parent = #1 },
    within          .meta:n    = { parent = #1 },
    sibling         .tl_set:N  = \l__keythms_thm_sibling_tl,
    numberlike      .meta:n    = { sibling = #1 },
    sharenumber     .meta:n    = { sibling = #1 },
    style           .tl_set:N  = \l__keythms_thm_style_tl,
    style           .groups:n  = { style-comes-first },
    preheadhook     .tl_set:N  = \l__keythms_thm_preheadhook_tl,
    postheadhook    .tl_set:N  = \l__keythms_thm_postheadhook_tl,
    prefoothook     .tl_set:N  = \l__keythms_thm_prefoothook_tl,
    postfoothook    .tl_set:N  = \l__keythms_thm_postfoothook_tl,
    qed             .tl_set:N  = \l__keythms_thm_qed_tl,
    qed             .default:n = \c_novalue_tl,
    % ^ distinguish between 'qed' and 'qed={}'
    tcolorbox       .tl_set:N  = \l__keythms_thm_tcbkeys_tl,
    tcolorbox       .default:n = {},
    tcolorbox-no-titlebar .meta:n =
      {
        tcolorbox={
          notitle,
          before~upper={
            \group_begin:
            \__keythms_thm_tcboxtemphead:
            \group_end:
            },
          #1
          }
      },
    tcolorbox-no-titlebar .default:n = {},
  }

% what below is unnecessary? Commenting everything but \def causes spaceabove,
% etc. to not work
\cs_new:Npn \__keythms_thm_storedeferredthmhead:n #1
  {
    % \if@inlabel \indent \par \fi % eject a section head if one is pending
    % \if@nobreak
      % \adjust@parskip@nobreak
    % \else
    % \addpenalty\@beginparpenalty
    \addvspace\@topsep
    \addvspace{-\parskip}
```

```
    % \fi
    % \global\@inlabeltrue
    % \everypar\dth@everypar
    \cs_set:Npn \__keythms_thm_tcboxtemphead: { \normalfont #1 }
    \ignorespaces
  }

\keys_precompile:nnN { keytheorems/thm }
  {
    name         = \q_no_value,
    refname      = \q_no_value,
    Refname      = \q_no_value,
    numbered     = true,
    parent       = {},
    sibling      = {},
    style        = {},
    preheadhook  = {},
    postheadhook = {},
    prefoothook  = {},
    postfoothook = {},
    qed          = \q_no_value,
    tcolorbox    = \q_no_value,
  }
  \l__keythms_thm_defaultkeys_tl

\cs_new_protected:Npn \__keythms_thm_makethmhooks:n #1
  {
    \hook_new:n { keytheorems/#1/prehead }
    \hook_new:n { keytheorems/#1/posthead }
    \hook_new_reversed:n { keytheorems/#1/prefoot }
    \hook_new_reversed:n { keytheorems/#1/postfoot }
    \hook_new:n { keytheorems/#1/restated }
  }

% Make generic theorem hooks
\__keythms_thm_makethmhooks:n { allthms }

% \newkeytheorem{<name>}{<keys>}
\NewDocumentCommand \newkeytheorem { m O{} }
  {
    \clist_map_inline:nn { #1 } % define multiple theorems at once
      { \keythms_thm_newkeythm:nn { ##1 } { #2 } }
  }

\@onlypreamble \newkeytheorem

% to prevent error when plain, remark, or definition style used
\tl_new:N \l__keythms_thmstyle_plain_savedkeys_tl
\tl_new:N \l__keythms_thmstyle_remark_savedkeys_tl
\tl_new:N \l__keythms_thmstyle_definition_savedkeys_tl

% \keythms_thm_newkeythm:nn { <envname> } { <keys> }
\cs_new_protected:Npn \keythms_thm_newkeythm:nn #1#2
  {
    % Store envname
    \tl_set:Nn \l__keythms_thm_envname_tl { #1 }
    % Make unless-unique false by default (can't precompile this)
    \bool_set_false:N \l__keythms_thm_unlessunique_bool
```

```
    % Set default keys
    \tl_use:N \l__keythms_thm_defaultkeys_tl
    % First set style so we can pick up additional thm keys, then overwrite if necessary
    \keys_set_groups:nnn { keytheorems/thm } { style-comes-first } { #2 }
    \tl_if_empty:NF \l__keythms_thm_style_tl
      {
        % Store theorem style
        \tl_set:Ne \l__keythms_thm_currentthmstyle_tl { \the\thm@style }
        % Temporarily set theorem style
        \__keythms_theoremstyle:n { \l__keythms_thm_style_tl }
        % If thm keys given in style, call now (possibly overwritten in next step)
        % but don't error if user uses a style defined with just \newtheoremstyle
        \tl_if_exist:cT { l__keythms_thmstyle_ \l__keythms_thm_style_tl _savedkeys_tl }
          { \tl_use:c { l__keythms_thmstyle_ \l__keythms_thm_style_tl _savedkeys_tl } }
      }
    % Set env-specific keys
    \keys_set:nn { keytheorems/thm } { #2 }
    % Set up env-specific hooks
    \__keythms_thm_makethmhooks:n { #1 }
    % Add to env-specific hooks (use label so code given in keys is outermost)
    % NOTE: faster to check if empty than add empty code to hook
    \tl_if_empty:NF \l__keythms_thm_preheadhook_tl
      {
        \hook_gput_code:nnV { keytheorems/#1/prehead }
          { keythms_hook_keys } \l__keythms_thm_preheadhook_tl
      }
    \tl_if_empty:NF \l__keythms_thm_postheadhook_tl
      {
        \hook_gput_code:nnV { keytheorems/#1/posthead }
          { keythms_hook_keys } \l__keythms_thm_postheadhook_tl
      }
    \tl_if_empty:NF \l__keythms_thm_prefoothook_tl
      {
        \hook_gput_code:nnV { keytheorems/#1/prefoot }
          { keythms_hook_keys } \l__keythms_thm_prefoothook_tl
      }
    \tl_if_empty:NF \l__keythms_thm_postfoothook_tl
      {
        \hook_gput_code:nnV { keytheorems/#1/postfoot }
          { keythms_hook_keys } \l__keythms_thm_postfoothook_tl
      }
    % Set name if none given
    \quark_if_no_value:NT \l__keythms_thm_name_tl % use quark so name={} is valid
      {
        % use e so \text_titlecase called only once per theorem definition,
        %  not each time the theorem is used
        \tl_set:Ne \l__keythms_thm_name_tl
          { \text_titlecase_first:n { #1 } }
      }
    % associate formatted name with envname in prop list
    \prop_gput:NnV \g__keythms_thmnames_prop { #1 } \l__keythms_thm_name_tl
    % Call correct \newtheorem variant
    \bool_if:NTF \l__keythms_thm_unlessunique_bool
      {
        % [unq] is required since aux is read at begindocument
        % (technically right before) which is after theorem is defined
        \RequirePackage[unq]{unique}
        \tl_if_empty:NTF \l__keythms_thm_parent_tl
```

```
      {
        \hook_gput_code:nnn { keytheorems/#1/prehead }
          { keythms_hook_keys } { \setuniqmark { #1 } }
        \ifuniq{ #1 }
          { \bool_set_false:N \l__keythms_thm_numbered_bool }
          { \bool_set_true:N \l__keythms_thm_numbered_bool }
        \bool_if:NTF \l__keythms_thm_numbered_bool
          {
            \tl_if_empty:NTF \l__keythms_thm_sibling_tl
              {
                \__keythms_thm_new:nV { #1 } \l__keythms_thm_name_tl
              }
              {
                \exp_args:NnV \newaliascnt { #1 } \l__keythms_thm_sibling_tl
                \__keythms_thm_new_sibling:nVn { #1 }
                  \l__keythms_thm_name_tl { #1 }
                \aliascntresetthe { #1 }
              }
          }
          {
            \__keythms_thm_new_nonumber:nV { #1 } \l__keythms_thm_name_tl
            \hook_gput_code:nnn { keytheorems/#1/prehead } { keythms_hook_keys }
              {
                \keythms_if_restating:F
                  { \refstepcounter{ keythms_unnumbered_dummyctr } }
              }
          }
      }
      {
        \__keythms_thm_new_uuwithparent:nVV { #1 }
          \l__keythms_thm_name_tl \l__keythms_thm_parent_tl
      }
  }
  {
    \bool_if:NTF \l__keythms_thm_numbered_bool
      {
        \tl_if_empty:NTF \l__keythms_thm_parent_tl
          {
            \tl_if_empty:NTF \l__keythms_thm_sibling_tl
              {
                \__keythms_thm_new:nV { #1 } \l__keythms_thm_name_tl
              }
              {
                \exp_args:NnV \newaliascnt { #1 } \l__keythms_thm_sibling_tl
                \__keythms_thm_new_sibling:nVn { #1 }
                  \l__keythms_thm_name_tl { #1 }
                \aliascntresetthe { #1 }
              }
          }
          {
            \__keythms_thm_new_parent:nVV { #1 }
              \l__keythms_thm_name_tl \l__keythms_thm_parent_tl
          }
      }
      {
        \__keythms_thm_new_nonumber:nV { #1 } \l__keythms_thm_name_tl
        \hook_gput_code:nnn { keytheorems/#1/prehead } { keythms_hook_keys }
          {
```

```
                \keythms_if_restating:F
                  { \refstepcounter{ keythms_unnumbered_dummyctr } }
              }
          }
      }
    % Store theorem def and redefine it with keys
    \keythms_keyify_theorem:n { #1 }
    % define \<env>autorefname and \<env>Autorefname, might be redefined next
    \exp_args:NnV \cs_set:cpn { #1 autorefname } \l__keythms_thm_name_tl
    \exp_args:NnV \cs_set:cpn { #1 Autorefname } \l__keythms_thm_name_tl
    % Set ref names
    \quark_if_no_value:NF \l__keythms_thm_refname_tl
      { \keythms_thm_setrefnames:nV { #1 } \l__keythms_thm_refname_tl }
    \quark_if_no_value:NF \l__keythms_thm_Refname_tl
      { \keythms_thm_setRefnames:nV { #1 } \l__keythms_thm_Refname_tl }
    % Set up qed if needed
    \quark_if_no_value:NF \l__keythms_thm_qed_tl
      {
        \exp_args:Nno \__keythms_thm_qedcode:nn { #1 } { \l__keythms_thm_qed_tl }
      }
    % Set up tcolorbox if needed
    \quark_if_no_value:NF \l__keythms_thm_tcbkeys_tl
      {
        \exp_args:Nno \__keythms_thm_tcboxcode:nn { #1 }
          { \l__keythms_thm_tcbkeys_tl }
      }
    % Set default list-of display command
    \__keythms_listof_show_aux:n { #1 }
    % Set theorem style back to original state if needed
    \tl_if_empty:NF \l__keythms_thm_style_tl
      {
        \__keythms_theoremstyle:V \l__keythms_thm_currentthmstyle_tl
      }
  }

\cs_new_protected:Npn \__keythms_thm_tcboxcode:nn #1#2
  {
    \RequirePackage{tcolorbox}
    \hook_gput_code:nnn { keytheorems/#1/prehead }
      { keythms_tcbox }
      {
        \cs_set_eq:NN \deferred@thm@head \__keythms_thm_storedeferredthmhead:n
        \cs_set_eq:NN \Hy@theorem@makelinktarget \use_none:n
        % ^ don't like playing with hyperref internals... but don't see around
        %   it because hyperref tries to add to para hook which doesn't work
        %   when title set up the way we do it
        \cs_set:Npn \thm@space@setup { \thm@preskip=0pt \thm@postskip=0pt }
        % ^ to match tcolorbox defaults; shouldn't interfere with user styles
      }
    \hook_gset_rule:nnnn { keytheorems/#1/posthead }
      { keythms_tcbox } { before } { keythms_hook_keys }
    \hook_gset_rule:nnnn { keytheorems/#1/prefoot }
      { keythms_tcbox } { after } { keythms_hook_keys }
    \hook_gset_rule:nnnn { keytheorems/#1/prefoot }
      { keythms_tcbox } { after } { keythms_qed }
    \bool_if:NTF \l__keythms_thm_numbered_bool
      {
        \hook_gput_code:nnn { begindocument } { . }
```

```
            {
              \IfPackageLoadedTF{cleveref}
                { % hyperref doesn't patch \@thm if cleveref loaded
                  \hook_gput_code:nnn { keytheorems/#1/posthead }
                    { keythms_tcbox }
                    {
                      \begin{tcolorbox}[
                        savedelimiter=#1,
                        title={ \__keythms_thm_tcboxtemphead: },
                        #2]
                    }
                }
                {
                  \hook_gput_code:nnn { keytheorems/#1/posthead }
                    { keythms_tcbox }
                    {
                      \begin{tcolorbox}[
                        savedelimiter=#1,
                        title={ \__keythms_thm_tcboxtemphead: },
                        phantom={ \MakeLinkTarget*{\@currentHref} }, % fix hyperlinking
                        #2]
                    }
                }
            }
        }
        {
          \hook_gput_code:nnn { keytheorems/#1/posthead }
            { keythms_tcbox }
            {
              \begin{tcolorbox}[
                savedelimiter=#1,
                title={ \__keythms_thm_tcboxtemphead: },
                #2]
            }
        }
      \hook_gput_code:nnn { keytheorems/#1/prefoot }
        { keythms_tcbox } { \end{tcolorbox} }
    }
  \cs_new_protected:Npn \__keythms_thm_qedcode:nn #1#2
    {
      \hook_gput_code:nnn { keytheorems/#1/posthead }
        { keythms_qed }
        {
          \exp_args:No \tl_if_novalue:nF { #2 } { \protected@edef\qedsymbol{#2} }
          \pushQED{\qed}
        }
      \hook_gput_code:nnn { keytheorems/#1/prefoot }
        { keythms_qed }
        {
          \exp_args:No \tl_if_novalue:nF { #2 } { \protected@edef\qedsymbol{#2} }
          \popQED
        }
    }

  \cs_new_eq:NN \__keythms_theoremstyle:n \theoremstyle
  \cs_generate_variant:Nn \__keythms_theoremstyle:n { V }

  % \newtheorem variants
```

```
\cs_new_eq:NN \__keythms_thm_new:nn \newtheorem
\cs_generate_variant:Nn \__keythms_thm_new:nn { nV }

\cs_new_protected:Npn \__keythms_thm_new_nonumber:nn #1#2
  { \__keythms_thm_new:nn*{#1}{#2} }
\cs_generate_variant:Nn \__keythms_thm_new_nonumber:nn { nV }

\cs_new_protected:Npn \__keythms_thm_new_parent:nnn #1#2#3
  { \__keythms_thm_new:nn{#1}{#2}[#3] }
\cs_generate_variant:Nn \__keythms_thm_new_parent:nnn { nVV }

\cs_new_protected:Npn \__keythms_thm_new_sibling:nnn #1#2#3
  { \__keythms_thm_new:nn{#1}[#3]{#2} }
\cs_generate_variant:Nn \__keythms_thm_new_sibling:nnn { nV }

\cs_new_protected:Npn \__keythms_thm_new_uuwithparent:nnn #1#2#3
  {
    \cs_undefine:c { keythms_orig_nonumber_#1 } % for renew, declare
    \__keythms_thm_new_nonumber:nn { keythms_orig_nonumber_#1 } { #2 }
    \__keythms_thm_new_parent:nnn { #1 } { #2 } { #3 }
    \DeclareEnvironmentCopy { keythms_orig_withparent_#1 } { #1 }
    \renewenvironment { #1 } % opt arg is implicit
      {
        \setuniqmark{ #1. \use:c {the #3} }
        \ifuniq{ #1. \use:c {the #3} }
          {
            \keythms_if_restating:F
              { \refstepcounter{ keythms_unnumbered_dummyctr } }
            \begin{keythms_orig_nonumber_#1}
          }
          {
            \begin{keythms_orig_withparent_#1}
          }
      }
      {
        \ifuniq{ #1. \use:c {the #3} }
          { \end{keythms_orig_nonumber_#1} }
          { \end{keythms_orig_withparent_#1} }
      }
  }
\cs_generate_variant:Nn \__keythms_thm_new_uuwithparent:nnn { nVV }

% for getting notes with continues*, use nameref if available, otherwise ltproperties
\hook_gput_code:nnn { begindocument } { . }
  {
    \IfPackageLoadedTF { nameref }
      {
        \cs_new:Npn \__keythms_thmuse_recordnote: { } % nameref takes care of this
        \cs_new:Npn \__keythms_getrecordednote:n #1
          {
            \getrefbykeydefault{ #1 }{ name }{ }
          }
      }
      { % needs https://github.com/latex3/latex2e/issues/1200 fixed
        \property_new:nnnn { keytheorems/recordednote } { now } { }
          { \l__keythms_thmuse_note_tl }
        \cs_new:Npn \__keythms_getrecordednote:n #1
          {
```

```
            \property_ref:nn { keythms_recordednote_#1 }
              { keytheorems/recordednote }
          }
        \cs_new:Npn \__keythms_thmuse_recordnote:
          {
            \tl_if_empty:NF \l__keythms_thmuse_note_tl
              {
                \RecordProperties
                  { keythms_recordednote_\l__keythms_thmuse_label_tl }
                  { keytheorems/recordednote }
              }
          }
      }
  }

\keys_define:nn { keytheorems/thmuse }
  {
    label      .tl_set:N = \l__keythms_thmuse_label_tl,
    note       .tl_set:N = \l__keythms_thmuse_note_tl,
    name       .meta:n   = { note = #1 },
    % ^ for compatibility. "name" is ambiguous and doesn't match amsthm language
    short-note .code:n   = {}, % these do nothing at point of use
    short-name .code:n   = {}, % ^ worthwhile compatibility?
    continues  .tl_set:N = \l__keythms_thmuse_contlabel_tl,
    continues* .code:n   =
      {
        \keys_set:nn { keytheorems/thmuse } { continues = #1 }
        \protected@edef \l__keythms_tmpa_tl { \__keythms_getrecordednote:n{#1} }
        \tl_if_empty:NF \l__keythms_tmpa_tl
          {
            \keys_set:nn { keytheorems/thmuse }
              { note = \l__keythms_tmpa_tl }
          }
      },
    store      .tl_set:N = \l__keythms_thmuse_store_tl,
    %store      .default:n = \q_no_value, % ={name} causes issues
    restate    .meta:n   = { store = #1 },
    % ^ thmtools compatibility
    listhack   .choice:, % need equals sign
    listhack / true .code:n = \bool_set_true:N \l__keythms_thmuse_listhack_bool,
    listhack / false .code:n = \bool_set_false:N \l__keythms_thmuse_listhack_bool,
    listhack   .initial:n = false,
    seq        .code:n = {},
  }

\cs_new_protected:Npn \keythms_keyify_theorem:n #1
  { % #1 = theorem name
    \DeclareEnvironmentCopy { keythms_orig_#1 } { #1 }
    \DeclareDocumentEnvironment { keythms_grab_#1 } { m O{} +b }
      { % ##1 = keys, ##2 = note, ##3 = theorem body
        \__keythms_thm_prehead_code:n { #1 }
        \begin{keythms_orig_#1}[{##2}]
        \clist_map_inline:Nn \g__keythms_restatecounters_clist
          {
            \prop_gput:Nne \g__keythms_thmuse_othercounters_prop { ####1 }
              { \the\value{####1} }
          }
        \__keythms_thm_posthead_code:n { #1 }
```

```
    % below needs to come after posthead so that correct \@currentHref
    % is stored for tcolorbox theorems
    \__keythms_thm_addcontentsdata:nnnn { #1 }
      { \prop_to_keyval:N \g__keythms_thmuse_othercounters_prop }
      { ##1 } { ##3 }
    \__keythms_thm_tempstorerestatedata:nnn { #1 } { ##1 } { ##3 }
    ##3
    \__keythms_thm_prefoot_code:n { #1 }
    \end{keythms_orig_#1}
    \__keythms_thm_postfoot_code:n { #1 }
  }
  {}
  % NOTE: have to do a lot of shenanigans to make sure the begin/end of grabbed
  %       theorem env captures only the body and no package code.
  %       This is the price of on-the-fly redefining the env to grab body
  \RenewDocumentEnvironment { #1 } { ={note} O{} }
    {
      \keys_set:nn { keytheorems/thmuse } { ##1 }
      \tl_if_empty:NF \l__keythms_thmuse_store_tl
        {
          \bool_gset_true:N \g__keythms_listof_writefile_bool
          \cs_set_eq:NN \__keythms_withhooks_begin:nn \__keythms_grab_begin:nn
          \cs_set_eq:NN \__keythms_withhooks_begin:nnn \__keythms_grab_begin:nnn
          \cs_set_eq:NN \__keythms_withhooks_begin:nnV \__keythms_grab_begin:nnV
          \cs_set_eq:NN \__keythms_withhooks_end:n \__keythms_grab_end:n
        }
      \__keythms_thm_prehead_continues_code:n { #1 }
      \tl_if_empty:NTF \l__keythms_thmuse_note_tl
        { \__keythms_withhooks_begin:nn { #1 } { ##1 } }
        {
          \__keythms_withhooks_begin:nnV { #1 } { ##1 }
            \l__keythms_thmuse_note_tl
        }
    }
    {
      \__keythms_withhooks_end:n { #1 }
      \tl_if_empty:NF \l__keythms_thmuse_store_tl
        {
          \cs_if_exist:cF
            { __keythms_getthm_ \l__keythms_thmuse_store_tl _theorem }
            {
              \cs_new:cpe
                { __keythms_getthm_ \l__keythms_thmuse_store_tl _theorem }
                {
                  \exp_not:N \__keythms_getthm_theorem:nnnnn
                  \exp_not:o { \g__keythms_thmuse_temprestatedata_tl }
                }
              \cs_new:cpe
                { __keythms_getthm_ \l__keythms_thmuse_store_tl _body }
                {
                  \exp_not:N \__keythms_getthm_body:nn
                  \exp_args:No \exp_not:o
                    {
                      \exp_after:wN \__keythms_use_iii_v_braced:nnnnn
                        \g__keythms_thmuse_temprestatedata_tl
                    }
                }
            }
```

```
            }
        }
  }
\cs_new:Npn \__keythms_use_iii_v_braced:nnnnn #1#2#3#4#5 { {#3}{#5} }

\cs_new_protected:Npn \__keythms_withhooks_begin:nn #1#2
  { % #1 = theorem name, #2 = keys
    \__keythms_thm_prehead_code:n { #1 }
    \begin{keythms_orig_#1}
    \__keythms_thm_posthead_code:n { #1 }
    \__keythms_thm_addcontentsdata:nnnn { #1 } { } { #2 } { }
    \ignorespaces % I hope this is alright
  }
\cs_new_protected:Npn \__keythms_withhooks_begin:nnn #1#2#3
  { % #1 = theorem name, #2 = keys, #3 = note
    \__keythms_thm_prehead_code:n { #1 }
    \begin{keythms_orig_#1}[{#3}]
    \__keythms_thm_posthead_code:n { #1 }
    \__keythms_thm_addcontentsdata:nnnn { #1 } { } { #2 } { }
    \ignorespaces % I hope this is alright
  }
\cs_generate_variant:Nn \__keythms_withhooks_begin:nnn { nnV }
\cs_new_protected:Npn \__keythms_withhooks_end:n #1
  {
    \__keythms_thm_prefoot_code:n { #1 }
    \end{keythms_orig_#1}
    \__keythms_thm_postfoot_code:n { #1 }
  }
\cs_new_protected:Npn \__keythms_grab_begin:nn #1#2
  { % #1 = theorem name, #2 = keys
    \begin{keythms_grab_#1}{#2}
  }
\cs_new_protected:Npn \__keythms_grab_begin:nnn #1#2#3
  { % #1 = theorem name, #2 = keys, #3 = note
    \begin{keythms_grab_#1}{#2}[{#3}]
  }
\cs_generate_variant:Nn \__keythms_grab_begin:nnn { nnV }
\cs_new_protected:Npn \__keythms_grab_end:n #1 { \end{keythms_grab_#1} }

\cs_new_protected:Npn \__keythms_orig_begin:n #1 { \begin{keythms_orig_#1} }
\cs_new_protected:Npn \__keythms_orig_begin:nn #1#2
  { \begin{keythms_orig_#1}[{#2}] }
\cs_generate_variant:Nn \__keythms_orig_begin:nn { nV }
\cs_new_protected:Npn \__keythms_orig_end:n #1 { \end{keythms_orig_#1} }

\cs_new:Npn \__keythms_thm_prehead_code:n #1
  { % #1 = theorem name
    \tl_set:Nn \l__keythms_thmuse_envname_tl { #1 }
    \hook_use:n { keytheorems/#1/prehead }
    \hook_use:n { keytheorems/allthms/prehead }
  }
%% this below has to be separate from prehead_code above since we need to add
%% continues-code to note before retrieving it in \__keythms_withhooks_begin:nnV
\cs_new:Npn \__keythms_thm_prehead_continues_code:n #1
  { % #1 = theorem name
    \tl_if_empty:NF \l__keythms_thmuse_contlabel_tl
      {
        \tl_if_empty:NF \l__keythms_thmuse_note_tl
```

```
            { \tl_put_right:Nn \l__keythms_thmuse_note_tl { , ~ } }
          \tl_put_right:Ne \l__keythms_thmuse_note_tl
            { \__keythms_thmuse_continues:V \l__keythms_thmuse_contlabel_tl }
          \cs_set:cpn { the #1 }
            {
              \getrefnumber { \l__keythms_thmuse_contlabel_tl }
            }
          \cs_set_eq:cN { c@ #1 } \c@keythms_continues_dummyctr
          \cs_set_eq:cN { theH #1 } \theHkeythms_continues_dummyctr
          %\cs_set_eq:NN \setuniqmark \use_none:n % not the right fix
        }
    }
  \cs_new:Npn \__keythms_thm_posthead_code:n #1
    { % #1 = theorem name
      \hook_use:n { keytheorems/#1/posthead }
      \hook_use:n { keytheorems/allthms/posthead }
      \tl_if_empty:NF \l__keythms_thmuse_label_tl
        {
          \label{ \l__keythms_thmuse_label_tl }
          \__keythms_thmuse_recordnote:
        }
      \bool_if:NT \l__keythms_thmuse_listhack_bool
        { % straight from thm-amsthm.sty
          \leavevmode
          \vspace{-\baselineskip}%
          \par
          \everypar{\setbox\z@\lastbox\everypar{}}%
        }
    }
  \cs_new:Npn \__keythms_thm_prefoot_code:n #1
    { % #1 = theorem name
      \hook_use:n { keytheorems/allthms/prefoot }
      \hook_use:n { keytheorems/#1/prefoot }
    }
  \cs_new:Npn \__keythms_thm_postfoot_code:n #1
    { % #1 = theorem name
      \hook_use:n { keytheorems/allthms/postfoot }
      \hook_use:n { keytheorems/#1/postfoot }
    }
  \cs_new:Npn \__keythms_thm_addcontentsdata:nnnn #1#2#3#4
    { % #1 = theorem name, #2 = stored counters, #3 = keys, #4 = body
      \keythms_listof_chaptervspacehack:
      \iow_shipout:Ne \@auxout
        {
          \exp_not:N \@writefile { thlist }
            {
              \KeyThmsSavedTheorem{ #1 }
                { \@currentlabel }
                { \@currentHref }
                { \thepage }
                { #2 }
                { \exp_not:n { #3 } } % do we want any expansion here, perhaps
                { \exp_not:n { #4 } } % with \text_expand:n ?
            }
        }
    }
  \cs_new:Npn \__keythms_thm_tempstorerestatedata:nnn #1#2#3
    { % #1 = theorem name, #2 = keys, #3 = body
```

```
    \tl_gset:Ne \g__keythms_thmuse_temprestatedata_tl % needs to be global to get out of env
      {
        { #1 }
        { \@currentlabel }
        { \prop_to_keyval:N \g__keythms_thmuse_othercounters_prop }
        { \exp_not:n { #2 } } % do we want any expansion here, perhaps
        { \exp_not:n { #3 } } % with \text_expand:n ?
      }
  }

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Retrieving Theorem Data %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\cs_new_protected:Npn \KeyThmsSavedTheorem #1#2#3#4#5#6#7 % 7th arg is body
  { \use:c { __keythms_thmitem_#1:nnnnnn } {#2}{#3}{#4}{#5}{#6}{#7} }

\keys_define:nn { keytheorems/storeatbegin }
  {
    store   .tl_set:N = \l__keythms_storeatbegin_store_tl,
    restate .meta:n   = { store=#1 },
    unknown .code:n   = { } % do nothing with unknown keys
  }

\cs_new_protected:Npn \KeyThmsContentsLine #1 { #1 }
\NewDocumentCommand \addtheoremcontentsline { m m }
  {
    \addtocontents { thlist }
      {
        \KeyThmsContentsLine
          { % copied from def of \addcontentsline
            \protect\contentsline{#1}{#2}{\thepage}{}
            \protected@file@percent
          }
      }
  }
\NewDocumentCommand \addtotheoremcontents { m }
  {
    \addtocontents { thlist }
      {
        \KeyThmsContentsLine { #1 }
      }
  }

\hook_gput_code:nnn { begindocument } { . }
  {
    \group_begin:
    \cs_set_eq:NN \KeyThmsContentsLine \use_none:n
    \cs_set_eq:NN \KeyThmsAddvspace \use_none:n
    \cs_set_protected:Npn \KeyThmsSavedTheorem #1#2#3#4#5#6#7
      {
        \group_begin:
        \keys_set:nn { keytheorems/storeatbegin } { #6 }
        \tl_if_empty:NF \l__keythms_storeatbegin_store_tl
          {
            \cs_new:cpn
              { __keythms_getthm_ \l__keythms_storeatbegin_store_tl _theorem }
              {
```

```
                \__keythms_getthm_theorem:nnnnn
                  {#1}{#2}{#5}{#6}{#7}
              }
            \cs_new:cpn
              { __keythms_getthm_ \l__keythms_storeatbegin_store_tl _body }
              {
                \__keythms_getthm_body:nn {#5}{#7}
              }
          }
        \group_end:
      }
    \file_if_exist_input:n { \c_sys_jobname_str.thlist }
    \group_end:
  }

\prg_new_conditional:Npnn \keythms_if_restating: { T, F, TF }
  {
    \bool_if:NTF \l__keythms_thmuse_restating_bool
      { \prg_return_true: }
      { \prg_return_false: }
  }
\NewDocumentCommand \IfRestatingTF { } { \keythms_if_restating:TF }

\cs_new_protected:Npn \__keythms_getthm_theorem:nnnnn #1#2#3#4#5
  { % #1 = name, #2 = number, #3 = restate counters, #4 = keys, #5 = theorem body
    \group_begin:
    \bool_set_true:N \l__keythms_thmuse_restating_bool
    \prop_set_from_keyval:Nn \l__keythms_restate_counters_prop { #3 }
    \prop_map_inline:Nn \l__keythms_restate_counters_prop
      {
        \tl_set:ce { l_keythms_restate_current_##1_tl } { \the\value{##1} }
        \setcounter { ##1 } { ##2 }
        % ^ FIX: what if eq's numbered by section, theorem, etc.? The
        %        thmtools code is opaque.... Or maybe should be up to the
        %        user to say "restate-counters={section,chapter,...}".
        \cs_set:cpn { theH ##1 } { \use:c { the ##1 } . \theHkeythms_restate_dummyctr }
      }
    \tl_if_empty:nTF { #2 }
      { \refstepcounter{keythms_restate_dummyctr} } % for unnumbered theorems
      {
        \cs_set:cpn { the #1 } { #2 }
        \cs_set_eq:cN { c@ #1 } \c@keythms_restate_dummyctr
        \cs_set_eq:cN { theH #1 } \theHkeythms_restate_dummyctr
        % ^ why are the last two line here? We shouldn't be referencing
        %   restated theorems. Think it's a remnant of thmtools
        % WRONG: needed to make numbering correct after restated theorem.
        % not sure about theH. <- this is needed to prevent duplicate anchors
      }
    \renewcommand\label[2][]{} % disable \label (opt arg in case cleveref loaded)
    \cs_set_eq:NN \ltx@label \use_none:n % disable \ltx@label
    \cs_set_eq:NN \property_record:nn \use_none:nn % disable \RecordProperties
    \cs_set_eq:NN \setuniqmark \use_none:n % work with numbered=unless-unique
    % QUESTION: also disable \hyper@@anchor? \MakeLinkTarget?
    \keys_set:nn { keytheorems/thmuse } { #4 }
    \hook_use:n { keytheorems/#1/restated }
    \hook_use:n { keytheorems/allthms/restated }
    \__keythms_thm_prehead_continues_code:n { #1 }
    \__keythms_thm_prehead_code:n { #1 }
```

```
    \tl_if_empty:NTF \l__keythms_thmuse_note_tl
      { \__keythms_orig_begin:n { #1 } }
      { \__keythms_orig_begin:nV { #1 } \l__keythms_thmuse_note_tl }
    \__keythms_thm_posthead_code:n { #1 }
    #5
    \__keythms_thm_prefoot_code:n { #1 }
    \__keythms_orig_end:n { #1 }
    \__keythms_thm_postfoot_code:n { #1 }
    \prop_map_inline:Nn \l__keythms_restate_counters_prop
      {
        \exp_args:Nnc \setcounter { ##1 }
          { l_keythms_restate_current_##1_tl }
      }
    \group_end:
  }

\cs_new_protected:Npn \__keythms_getthm_body:nn #1#2
  { % #1 = restate counters, #2 = theorem body
    \group_begin:
    \bool_set_true:N \l__keythms_thmuse_restating_bool
    \prop_set_from_keyval:Nn \l__keythms_restate_counters_prop { #1 }
    \prop_map_inline:Nn \l__keythms_restate_counters_prop
      {
        \tl_set:ce { l_keythms_restate_current_##1_tl } { \the\value{##1} }
        \setcounter { ##1 } { ##2 }
        % ^ FIX: what if eq's numbered by section, theorem, etc.? The
        %        thmtools code is opaque.... Or maybe should be up to the
        %        user to say "restate-counters={section,chapter,...}".
        \cs_set:cpn { theH ##1 } { \use:c { the ##1 } . \theHkeythms_restate_dummyctr }
      }
    \refstepcounter{keythms_restate_dummyctr}
    \renewcommand\label[2][]{} % disable \label (opt arg in case cleveref loaded)
    \cs_set_eq:NN \ltx@label \use_none:n % disable \ltx@label
    \cs_set_eq:NN \property_record:nn \use_none:nn % disable \RecordProperties
    \hook_use:n { keytheorems/#1/restated }
    \hook_use:n { keytheorems/allthms/restated }
    #2
    \prop_map_inline:Nn \l__keythms_restate_counters_prop
      {
        \exp_args:Nnc \setcounter { ##1 }
          { l_keythms_restate_current_##1_tl }
      }
    \group_end:
  }

% \getkeytheorem[<property>]{<tag>}
\NewDocumentCommand \getkeytheorem { o m }
  {
    \cs_if_exist:cTF { __keythms_getthm_#2_theorem }
      {
        \IfNoValueTF { #1 }
          { \use:c { __keythms_getthm_#2_theorem } }
          { \use:c { __keythms_getthm_#2_#1 } }
      }
      {
        \textbf{??}
        \msg_warning:nnn { keytheorems } { no-stored-theorem } { #2 }
      }
```

```
  }

%%%%%%%%%%%%%%%%%%%%%%%
%%% Theorem Hooks %%%
%%%%%%%%%%%%%%%%%%%%%%%

%%% \addtotheoremhook[<envname>]{<hook>}{<code>}
\NewDocumentCommand \addtotheoremhook { o m +m }
  {
    \__hook_if_declared:nTF { keytheorems/allthms/#2 }
      {
        \IfNoValueTF { #1 }
          { \hook_gput_code:nnn { keytheorems/allthms/#2 } { . } { #3 } }
          { \hook_gput_code:nnn { keytheorems/#1/#2 } { . } { #3 } }
      }
      {
        \msg_error:nnn { keytheorems } { undefined-thm-hook } { #2 }
      }
  }

% NOTE: I think it's OK we use the internal \__hook_if_declared:nTF above
%       since we don't need to worry about the user creating new theorem hooks
%       so, as we're only checking the existence of hooks created by us, it's OK.

%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of Theorems %%%
%%%%%%%%%%%%%%%%%%%%%%%%

\keys_define:nn { keytheorems/listof }
  {
    numwidth    .dim_set:N = \l__keythms_listof_numwidth_dim,
    numwidth    .initial:n = 2.3em,
    ignore      .code:n    =
      {
        \hook_gput_code:nnn { begindocument/before } { keytheorems }
          { \keythms_listof_ignore:n { #1 } }
      },
    show        .code:n    =
      {
        \hook_gput_code:nnn { begindocument/before } { keytheorems }
          { \keythms_listof_show:n { #1 } }
      },
    onlynamed   .code:n    =
      {
        \hook_gput_code:nnn { begindocument/before } { keytheorems }
          { \keythms_listof_onlynamed:n { #1 } }
      },
    onlynamed   .default:n = \q_no_value,
    onlynumbered   .code:n    =
      {
        \hook_gput_code:nnn { begindocument/before } { keytheorems }
          { \keythms_listof_onlynumbered:n { #1 } }
      },
    onlynumbered   .default:n = \q_no_value,
    ignoreall   .code:n    =
      {
        \hook_gput_code:nnn { begindocument/before } { keytheorems } % in case called before theorem
          {
```

```
            \prop_map_inline:Nn \g__keythms_thmnames_prop
              { \__keythms_listof_ignore_aux:n { ##1 } }
          }
      },
    showall    .code:n    =
      {
        \hook_gput_code:nnn { begindocument/before } { keytheorems } % in case called before theorem
          {
            \prop_map_inline:Nn \g__keythms_thmnames_prop
              { \__keythms_listof_show_aux:n { ##1 } }
          }
      },
    title      .tl_set:N   = \l__keythms_listof_title_tl,
    title      .initial:n  = \GetTranslation{keythms_listof_title},
    swapnumber .bool_set:N = \l__keythms_listof_swapnumber_bool,
    swapnumber .initial:n  = false,
    title-code .cs_set:Np  = \__keythms_listof_titlecmd:n #1,
    no-title   .bool_set:N = \l__keythms_listof_notitle_bool,
    no-title   .initial:n  = false,
    print-body .code:n     =
      {
        \cs_set:Nn \keythms_listof_listcmd:nnnnnnn
          {
            \tl_if_empty:nF { ##7 }
              {
                \__keythms_getthm_theorem:nnnnn
                  {##1}{##2}{##5}{##6}{##7}
              }
          }
        \cs_set_eq:NN \KeyThmsContentsLine \use_none:n
        % ^ I assume we want this?
        \cs_set_eq:NN \KeyThmsAddvspace \use_none:n
      },
    note-code  .cs_set:Np = \__keythms_listof_notecmd:n #1,
    note-code  .initial:n = { ~ (#1) },
    no-continues .bool_set:N = \l__keythms_listof_nocont_bool,
    no-continues .initial:n  = false,
    no-chapter-skip .bool_set:N = \l__keythms_listof_nochapskip_bool,
    no-chapter-skip .initial:n  = false,
    chapter-skip-length .dim_set:N = \keythms@listof@chaptervspace@dim,
    chapter-skip-length .initial:n = 10pt,
  }

\hook_gput_code:nnn { begindocument } { . } % redefine these keys at begindocument
  {
    \keys_define:nn { keytheorems/listof }
      {
        ignore    .code:n    = \keythms_listof_ignore:n { #1 },
        show      .code:n    = \keythms_listof_show:n { #1 },
        onlynamed .code:n    = \keythms_listof_onlynamed:n { #1 },
        onlynamed .default:n = \q_no_value,
        onlynumbered .code:n    = \keythms_listof_onlynumbered:n { #1 },
        onlynumbered .default:n = \q_no_value,
        ignoreall .code:n    =
          {
            \prop_map_inline:Nn \g__keythms_thmnames_prop
              { \__keythms_listof_ignore_aux:n { ##1 } }
          },
```

```
        showall   .code:n    =
          {
            \prop_map_inline:Nn \g__keythms_thmnames_prop
              { \__keythms_listof_show_aux:n { ##1 } }
          },
        seq       .code:n = \keythms_listof_showseq:n { #1 },
      }
  }

\NewDocumentCommand \keytheoremlistset { m }
  {
    \keys_set:nn { keytheorems/listof } { #1 }
  }

\cs_new_protected:Npn \keythms_listof_ignore:n #1
  {
    \clist_map_inline:nn { #1 } { \__keythms_listof_ignore_aux:n { ##1 } }
  }
\cs_new_protected:Npn \__keythms_listof_ignore_aux:n #1
  {
    \cs_set_protected:cpn { __keythms_thmitem_#1:nnnnnn } ##1##2##3##4##5##6
      { }
  }

\cs_new_protected:Npn \keythms_listof_show:n #1
  {
    \clist_map_inline:nn { #1 } { \__keythms_listof_show_aux:n { ##1 } }
  }
\cs_new_protected:Npn \__keythms_listof_show_aux:n #1
  {
    \cs_set_protected:cpn { __keythms_thmitem_#1:nnnnnn } ##1##2##3##4##5##6
      {
        \__keythms_listof_listcmd_setup:nn { ##5 }
          {
            \keythms_listof_listcmd:nnnnnnn
              {#1}{##1}{##2}{##3}{##4}{##5}{##6}
          }
      }
  }

\cs_new_protected:Npn \keythms_listof_onlynamed:n #1
  {
    \quark_if_no_value:nTF { #1 }
      {
        \prop_map_inline:Nn \g__keythms_thmnames_prop
          { \__keythms_listof_onlynamed_aux:n { ##1 } }
      }
      {
        \clist_map_inline:nn { #1 }
          { \__keythms_listof_onlynamed_aux:n { ##1 } }
      }
  }
\cs_new_protected:Npn \__keythms_listof_onlynamed_aux:n #1
  {
    \cs_set_protected:cpn { __keythms_thmitem_#1:nnnnnn } ##1##2##3##4##5##6
      {
        \__keythms_listof_listcmd_setup:nn { ##5 }
          {
```

```
            \tl_if_empty:NF \l__keythms_listofheading_note_tl
              {
                \keythms_listof_listcmd:nnnnnnn
                  {#1}{##1}{##2}{##3}{##4}{##5}{##6}
              }
          }
      }
  }

\cs_new_protected:Npn \keythms_listof_onlynumbered:n #1
  {
    \quark_if_no_value:nTF { #1 }
      {
        \prop_map_inline:Nn \g__keythms_thmnames_prop
          { \__keythms_listof_onlynumbered_aux:n { ##1 } }
      }
      {
        \clist_map_inline:nn { #1 }
          { \__keythms_listof_onlynumbered_aux:n { ##1 } }
      }
  }
\cs_new_protected:Npn \__keythms_listof_onlynumbered_aux:n #1
  {
    \cs_set_protected:cpn { __keythms_thmitem_#1:nnnnnn } ##1##2##3##4##5##6
      {
        \__keythms_listof_listcmd_setup:nn { ##5 }
          {
            \tl_if_empty:nF { ##1 }
              {
                \keythms_listof_listcmd:nnnnnnn
                  {#1}{##1}{##2}{##3}{##4}{##5}{##6}
              }
          }
      }
  }

\cs_new_protected:Npn \keythms_listof_showseq:n #1
  {
    \prop_map_inline:Nn \g__keythms_thmnames_prop
      { \__keythms_listof_showseq_aux:nn { #1 } { ##1 } }
  }
\cs_new_protected:Npn \__keythms_listof_showseq_aux:nn #1#2
  { % #1 = seq name, #2 = theorem name
    \cs_set_protected:cpn { __keythms_thmitem_#2:nnnnnn } ##1##2##3##4##5##6
      {
        \__keythms_listof_listcmd_setup:nn { ##5 }
          {
            \tl_if_eq:NnT \l__keythms_listofheading_seq_tl { #1 }
              {
                \keythms_listof_listcmd:nnnnnnn
                  {#2}{##1}{##2}{##3}{##4}{##5}{##6}
              }
          }
      }
  }

% Seems unnecessary to repeat all this for reading the keyvals from seq.
% In thmtools they just hook the "thmitem" definition into the theorem declaration.
```

```
%% NOTE ON ABOVE: this gives more flexibility to define different kinds of lists.
%% See acro.sty for template idea.
\keys_define:nn { keytheorems/listofheading }
  {
    note       .tl_set:N = \l__keythms_listofheading_note_tl,
    name       .meta:n   = { note = #1 },
    short-note .tl_set:N = \l__keythms_listofheading_shortnote_tl,
    short-name .meta:n   = { short-note = #1 },
    continues  .tl_set:N = \l__keythms_listofheading_contlabel_tl,
    continues* .code:n   =
      {
        \keys_set:nn { keytheorems/listofheading } { continues = #1 }
        \protected@edef \l__keythms_tmpa_tl { \__keythms_getrecordednote:n{#1} }
        \tl_if_empty:NF \l__keythms_tmpa_tl
          {
            \keys_set:nn { keytheorems/listofheading }
              { note = \l__keythms_tmpa_tl }
          }
      },
    seq     .tl_set:N = \l__keythms_listofheading_seq_tl,
    unknown .code:n = { } % do nothing with unknown keys
    % ^ this is OK because we have total control over possible keys; if invalid
    %   key is given to theorem then an error will be raised there
  }

\cs_new:Npn \__keythms_listof_printheading:
  {
    \tl_if_empty:NTF \l__keythms_listofheading_shortnote_tl
      {
        \tl_if_empty:NF \l__keythms_listofheading_note_tl
          { \__keythms_listof_notecmd:n { \l__keythms_listofheading_note_tl } }
      }
      {
        \__keythms_listof_notecmd:n { \l__keythms_listofheading_shortnote_tl }
      }
  }

\cs_new:Npn \__keythms_listof_default_listcmd:nnnnnnn #1#2#3#4#5#6#7
  {
    \contentsline{ #1 }
      {
        \bool_if:NTF \l__keythms_listof_swapnumber_bool
          {
            \prop_item:Nn \g__keythms_thmnames_prop { #1 } ~ #2
          }
          {
            \numberline{ #2 }
            \prop_item:Nn \g__keythms_thmnames_prop { #1 }
          }
        \__keythms_listof_printheading:
      }
      { #4 }{ #3 }
  }

% NOTE: We still need to do this setup for [print-body] so that onlynamed works
\cs_new:Npn \__keythms_listof_listcmd_setup:nn #1#2
  { % #1 = keys, #2 = list command
    \group_begin:
```

```
    \keys_set:nn { keytheorems/listofheading } { #1 }
    \tl_if_empty:NTF \l__keythms_listofheading_contlabel_tl
      { #2 }
      {
        \bool_if:NF \l__keythms_listof_nocont_bool
          {
            \tl_if_empty:NF \l__keythms_listofheading_note_tl
              {
                \tl_put_right:Nn \l__keythms_listofheading_note_tl { , ~ }
              }
            \tl_put_right:Nn \l__keythms_listofheading_note_tl
              {
                \__keythms_thmuse_continues:V \l__keythms_listofheading_contlabel_tl
              }
            #2
          }
      }
    \group_end:
  }

% set default listcmd
\cs_new_eq:NN \keythms_listof_listcmd:nnnnnnn
  \__keythms_listof_default_listcmd:nnnnnnn

\cs_if_exist:NTF \chapter
  { \cs_set:Npn \__keythms_listof_titlecmd:n #1 { \chapter*{#1} } }
  { \cs_set:Npn \__keythms_listof_titlecmd:n #1 { \section*{#1} } }

\hook_gput_code:nnn { begindocument } { . }
  { % try to detect ams classes
    \keythms_if_amsclass:TF
      {
        \prop_map_inline:Nn \g__keythms_thmnames_prop
          {
            \cs_set:cpn { l@ #1 }
              {
                \@tocline{ 0 }{ 3pt plus 2pt }{ 0pt }
                  { \l__keythms_listof_numwidth_dim }{ }
              }
          }
      }
      {
        \prop_map_inline:Nn \g__keythms_thmnames_prop
          {
            \cs_set:cpn { l@ #1 }
              {
                \@dottedtocline{ 1 }{ 1.5em }
                  { \l__keythms_listof_numwidth_dim }
              }
          }
      }
  }

\keythms_if_amsclass:TF
  {
    \keys_define:nn { keytheorems/listof } % adjust to class
      {
        numwidth .initial:n = 1.5pc,
```

```
      }
    \NewDocumentCommand \listofkeytheorems { O{} }
      { % title command not customizable here
        \bool_gset_true:N \g__keythms_listof_writefile_bool
        \group_begin:
        \keys_set:nn { keytheorems/listof } { #1 }
        \bool_if:NT \l__keythms_listof_nochapskip_bool
          {
            \cs_set_eq:NN \KeyThmsAddvspace \use_none:n
          }
        \legacy_if_set_false:n { @filesw }
        \bool_if:NTF \l__keythms_listof_notitle_bool
          {
            \@starttoc{ thlist }{ }
          }
          { % ams classes don't expand title enough
            \protected@edef \l__keythms_tmpa_tl { \l__keythms_listof_title_tl }
            \@starttoc{ thlist }{ \l__keythms_tmpa_tl }
          }
        \group_end:
      }
  }
  {
    \NewDocumentCommand \listofkeytheorems { O{} }
      {
        \bool_gset_true:N \g__keythms_listof_writefile_bool
        \group_begin:
        \keys_set:nn { keytheorems/listof } { #1 }
        \bool_if:NT \l__keythms_listof_nochapskip_bool
          {
            \cs_set_eq:NN \KeyThmsAddvspace \use_none:n
          }
        \bool_if:NF \l__keythms_listof_notitle_bool
          {
            \__keythms_listof_titlecmd:n { \l__keythms_listof_title_tl }
            \@mkboth % QUESTION: should this go in titlecmd ?
              { \MakeUppercase \l__keythms_listof_title_tl }
              { \MakeUppercase \l__keythms_listof_title_tl }
          }
        \legacy_if_set_false:n { @filesw }
        \@starttoc{ thlist }
        \group_end:
      }
  }
% ^ unlike thmtools we don't use the class's style of \listoffigures because
%   we want control over title-code, no-title, etc. But this means we have to guess
%   things like marks, sectioning command, etc.

\hook_gput_code:nnn { enddocument } { . }
  {
    \bool_if:NTF \g__keythms_listof_writefile_bool
      {
        \legacy_if:nT { @filesw }
          {
            \iow_new:N \tf@thlist
            \iow_open:Nn \tf@thlist { \c_sys_jobname_str.thlist }
          }
      }
```

```
      { % if .thlist file left over from previous run but not needed, clear it
        \file_if_exist:nT { \c_sys_jobname_str.thlist }
          {
            \iow_open:Nn \g_tmpa_iow { \c_sys_jobname_str.thlist }
            \iow_close:N \g_tmpa_iow
          }
      }
  }

% chaptervspacehack (code translated from thmtools)
\cs_new_eq:NN \KeyThmsAddvspace \addvspace
\int_new:N \g_keythms_listof_prevchapter_int
\int_gset:Nn \g_keythms_listof_prevchapter_int { 1 }
% ^ if this is zero, bad things happen if title-code is changed; anyways don't
%   need addvspace at top
\cs_new:Npn \keythms_listof_chaptervspacehack: { }
\cs_if_exist:cT { c@chapter }
  {
    \cs_if_eq:NNF \c@chapter \relax
      {
        \cs_set:Npn \keythms_listof_chaptervspacehack:
          {
            \int_compare:nNnF { \value{chapter} } = { \g_keythms_listof_prevchapter_int }
              {
                \addtocontents{ thlist }
                  {
                    \protect\KeyThmsAddvspace
                      { \keythms@listof@chaptervspace@dim }
                  }
                \int_gset:Nn \g_keythms_listof_prevchapter_int { \value{chapter} }
              }
          }
      }
  }

%%%%%%%%%%%%%%%%
%%% \Autoref %%%
%%%%%%%%%%%%%%%%

\ProvideDocumentCommand { \Autoref } { s m }
  {
    \group_begin:
    \cs_set_eq:NN \HyRef@testreftype \__keythms_Autoref_testreftype:w
    \IfBooleanTF { #1 } { \autoref*{#2} } { \autoref{#2} }
    \group_end:
  }

\cs_new:Npn \__keythms_Autoref_testreftype:w #1.#2\\
  {
    \cs_if_exist:cTF { #1 Autorefname }
      {
        \cs_set:Npe \HyRef@currentHtag
          {
            \exp_not:N \use:c { #1 Autorefname }
            \exp_not:N \c_space_token
          }
      }
      { \msg_warning:nnn { keytheorems } { no-Autorefname } { #1 } }
```

```
  }

%%%%%%%%%%%%%%%%%%%%%
%%% Global Keys %%%
%%%%%%%%%%%%%%%%%%%%%

\keys_define:nn { keytheorems }
  {
    restate-counters .code:n =
      {
        \clist_map_inline:nn { #1 }
          { \tl_new:c { l_keythms_restate_current_##1_tl } }
        \clist_gput_right:Nn \g__keythms_restatecounters_clist { #1 }
      },
    restate-counters .initial:n = equation,
    continues-code   .cs_set:Np = \__keythms_thmuse_continues:n #1,
    continues-code   .initial:n =
      { % not sure how best to handle this translation
        \GetTranslation{keythms_continues}\pageref{#1}
      },
    qed-symbol       .cs_set_protected:Np = \qedsymbol,
    overload         .code:n = \__keythms_overload_code:,
    overload         .value_forbidden:n = true,
    overload         .usage:n = preamble,
    thmtools-compat  .code:n = \__keythms_thmtoolscompat_code:,
    thmtools-compat  .value_forbidden:n = true,
    thmtools-compat  .usage:n = preamble,
    store-all        .code:n = \__keythms_storeall_code:,
    store-all        .value_forbidden:n = true,
    store-all        .usage:n = preamble,
    auto-translate   .bool_gset:N = \g__keythms_autotranslate_bool,
    auto-translate   .initial:n = true,
  }

\cs_generate_variant:Nn \__keythms_thmuse_continues:n { V }

% \keytheoremset{<options>}
\NewDocumentCommand \keytheoremset { m }
  {
    \keys_set:nn { keytheorems } { #1 }
  }

\cs_new_protected:Npn \__keythms_overload_code:
  {
    \RenewDocumentCommand { \newtheorem } { smomo }
      {
        \IfBooleanTF { ##1 }
          { \keythms_thm_newkeythm:nn { ##2 } { name=###4, numbered=no } }
          {
            \IfNoValueTF { ##3 }
              {
                \IfNoValueTF { ##5 }
                  { \keythms_thm_newkeythm:nn { ##2 } { name=###4 } }
                  { \keythms_thm_newkeythm:nn { ##2 } { name=###4, parent=##5 } }
              }
              { \keythms_thm_newkeythm:nn { ##2 } { name=###4, sibling=##3 } }
          }
      }
```

```
  }

\cs_new_protected:Npn \__keythms_thmtoolscompat_code:
  {
    \ProvideDocumentCommand { \declaretheoremstyle } { O{} m }
      {
        \declarekeytheoremstyle { ##2 } { ##1 }
      }
    \ProvideDocumentCommand { \declaretheorem } { O{} m }
      {
        \newkeytheorem { ##2 } [ ##1 ]
      }
    \ProvideDocumentEnvironment { restatable } { O{} m m }
      {
        \begin{##2}[##1,store=##3]
      }
      {
        \end{##2}
        \cs_new:cpn { ##3 }
          { % make \foo and \foo* identical
            \peek_meaning_remove:NTF *
              { \use:c { __keythms_getthm_ ##3 _theorem } }
              { \use:c { __keythms_getthm_ ##3 _theorem } }
          }
      }
    \ProvideDocumentCommand { \listoftheorems } { } { \listofkeytheorems }
    \ProvideDocumentCommand { \addtotheorempreheadhook } { o m }
      {
        \IfNoValueTF { ##1 }
          { \addtotheoremhook { prehead } { ##2 } }
          { \addtotheoremhook [ ##1 ] { prehead } { ##2 } }
      }
    \ProvideDocumentCommand { \addtotheorempostheadhook } { o m }
      {
        \IfNoValueTF { ##1 }
          { \addtotheoremhook { posthead } { ##2 } }
          { \addtotheoremhook [ ##1 ] { posthead } { ##2 } }
      }
    \ProvideDocumentCommand { \addtotheoremprefoothook } { o m }
      {
        \IfNoValueTF { ##1 }
          { \addtotheoremhook { prefoot } { ##2 } }
          { \addtotheoremhook [ ##1 ] { prefoot } { ##2 } }
      }
    \ProvideDocumentCommand { \addtotheorempostfoothook } { o m }
      {
        \IfNoValueTF { ##1 }
          { \addtotheoremhook { postfoot } { ##2 } }
          { \addtotheoremhook [ ##1 ] { postfoot } { ##2 } }
      }
    \clist_new:N \l__keythms_tcbshaded_keys_clist
    \clist_new:N \l__keythms_tcbthmbox_keys_clist
    \keys_define:nn { keytheorems/thm/shaded }
      {
        textwidth   .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { width=##1 },
        bgcolor     .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { colback=##1 },
        rulewidth   .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { boxrule=##1 },
        rulecolor   .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { colframe=##1 },
```

```
      margin      .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { boxsep=##1 },
      padding     .meta:n = { margin=##1 },
      leftmargin  .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { left~skip=##1 }
      rightmargin .code:n = \clist_put_right:Nn \l__keythms_tcbshaded_keys_clist { right~skip=##1
    }
  \keys_define:nn { keytheorems/thm/thmbox }
    {
      L .code:n =
        {
          \clist_put_right:Nn \l__keythms_tcbthmbox_keys_clist
            { keythms_tcbthmbox_L }
        },
      M .code:n =
        {
          \clist_put_right:Nn \l__keythms_tcbthmbox_keys_clist
            { keythms_tcbthmbox_M }
        },
      S .code:n =
        {
          \clist_put_right:Nn \l__keythms_tcbthmbox_keys_clist
            { keythms_tcbthmbox_S }
        },
      underline .choice:,
      underline / true .code:n = {},
      underline / false .code:n =
        {
          \clist_put_right:Nn \l__keythms_tcbthmbox_keys_clist
            { boxed~title~style={bottomrule=0pt} }
        },
      underline .default:n = true,
      nounderline .meta:n = { underline=false },
      cut .choice:,
      cut / true .code:n = {},
      cut / false .code:n =
        {
          \clist_put_right:Nn \l__keythms_tcbthmbox_keys_clist { unbreakable }
        },
      cut .default:n = true,
      nocut .meta:n = { cut=false },
      thickness .code:n = % could also add keys to clist with changed dimens; which is better?
        {
          \hook_gput_code:nnn { keytheorems/\l__keythms_thm_envname_tl/prehead }
            { keythms_tcbox }
            { \dim_set:Nn \l_keythms_tcbthmbox_thickness_dim { ##1 } }
        },
      leftmargin .code:n =
        {
          \hook_gput_code:nnn { keytheorems/\l__keythms_thm_envname_tl/prehead }
            { keythms_tcbox }
            { \dim_set:Nn \l_keythms_tcbthmbox_leftmargin_dim { ##1 } }
        },
      rightmargin .code:n =
        {
          \hook_gput_code:nnn { keytheorems/\l__keythms_thm_envname_tl/prehead }
            { keythms_tcbox }
            { \dim_set:Nn \l_keythms_tcbthmbox_rightmargin_dim { ##1 } }
        },
      hskip .code:n =
```

```
      {
        \hook_gput_code:nnn { keytheorems/\l__keythms_thm_envname_tl/prehead }
          { keythms_tcbox }
          { \dim_set:Nn \l_keythms_tcbthmbox_hskip_dim { ##1 } }
      },
    vskip .code:n =
      {
        \hook_gput_code:nnn { keytheorems/\l__keythms_thm_envname_tl/prehead }
          { keythms_tcbox }
          { \dim_set:Nn \l_keythms_tcbthmbox_vskip_dim { ##1 } }
      },
  }
\dim_new:N \l_keythms_tcbthmbox_thickness_dim
\dim_set:Nn \l_keythms_tcbthmbox_thickness_dim { 0.6pt }
\dim_new:N \l_keythms_tcbthmbox_leftmargin_dim
\dim_set:Nn \l_keythms_tcbthmbox_leftmargin_dim { 0.7\parindent } % use \parindent? thmbox does
\dim_new:N \l_keythms_tcbthmbox_rightmargin_dim
\dim_set:Nn \l_keythms_tcbthmbox_rightmargin_dim { 0pt }
\dim_new:N \l_keythms_tcbthmbox_hskip_dim
\dim_set:Nn \l_keythms_tcbthmbox_hskip_dim { 0.2em }
\dim_new:N \l_keythms_tcbthmbox_vskip_dim
\dim_set:Nn \l_keythms_tcbthmbox_vskip_dim { 0.2em }
\keys_define:nn { keytheorems/thm }
  {
    shaded .code:n =
      {
        \clist_clear:N \l__keythms_tcbshaded_keys_clist
        \keys_set:nn { keytheorems/thm/shaded } { ##1 }
        % FIX: surely a better way to do this
        \RequirePackage{tcolorbox}
        \pgfkeysifdefined{/tcb/keythms_tcbshaded_default/.@cmd} % even worth it?
          {}
          {
            \tcbset % wish I could do this outside of key but can't assume tcb loaded
              {
                keythms_tcbshaded_default/.style=
                  {
                    sharp~corners = all,
                    boxrule = 0pt,
                    left = 0pt, right = 0pt,
                    top = 0pt, bottom = 0pt,
                    parbox = false,
                  }
              }
          }
        \keys_set:ne { keytheorems/thm }
          {
            tcolorbox-no-titlebar =
              {
                keythms_tcbshaded_default,
                \l__keythms_tcbshaded_keys_clist
              }
          }
      },
    thmbox .code:n = % adapted from https://tex.stackexchange.com/a/236230/208544
      {
        \clist_clear:N \l__keythms_tcbthmbox_keys_clist
        % \bool_set_true:N \l__keythms_tcbthmbox_underline_bool
```

```
\keys_set:nn { keytheorems/thm/thmbox } { ##1 }
% FIX: surely a better way to do this
\RequirePackage{tcolorbox}
\tcbuselibrary{skins,breakable}
\pgfkeysifdefined{/tcb/keythms_tcbthmbox_default/.@cmd} % even worth it?
  {}
  {
    \tcbset{
      keythms_tcbthmbox_default/.style={
        enhanced,
        breakable,
        sharp~corners=all,
        right=\l_keythms_tcbthmbox_hskip_dim,
        left=\l_keythms_tcbthmbox_hskip_dim,
        top=\l_keythms_tcbthmbox_vskip_dim,
        bottom=\l_keythms_tcbthmbox_vskip_dim,
        coltitle=black,
        frame~engine=empty,
        interior~titled~engine=empty,
        interior~engine=empty,
        extras~broken={
          frame~engine=empty,
          interior~titled~engine=empty,
          interior~engine=empty
        },
        parbox=false,
        % even though frame isn't drawn, makes spacing correct
        boxrule=0.5\l_keythms_tcbthmbox_thickness_dim,
        attach~boxed~title~to~top~left={
          xshift=-\l_keythms_tcbthmbox_leftmargin_dim,
          },
        boxed~title~style={
          empty,
          size=minimal,
          bottom=0.7ex,
          top=0ex,
          % ditto
          bottomrule=0.5\l_keythms_tcbthmbox_thickness_dim,
          },
        left~skip=\l_keythms_tcbthmbox_leftmargin_dim,
        right~skip=\l_keythms_tcbthmbox_rightmargin_dim,
        overlay~unbroken={
          \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
            (title.south~west)
            --
            (title.south~east);
          \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
            (frame.north~west)
            |-
            ([xshift=10mm]frame.south~west);
          },
        overlay~first={
          \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
            (title.south~west)
            --
            (title.south~east);
          \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
            (frame.north~west)
```

```
      --
      (frame.south~west);
    },
  overlay~middle={
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~west)
      --
      (frame.south~west);
    },
  overlay~last={
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~west)
      |-
      ([xshift=10mm]frame.south~west);
    }
  },
keythms_tcbthmbox_L/.style={
  overlay~unbroken={
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (title.south~west)
      --
      (title.south~east);
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~west)
      |-
      (frame.south~east)
      -|
      (frame.north~east);
    },
  overlay~first={
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (title.south~west)
      --
      (title.south~east);
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~west)
      --
      (frame.south~west);
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~east)
      --
      (frame.south~east);
    },
  overlay~middle={
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~west)
      --
      (frame.south~west);
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~east)
      --
      (frame.south~east);
    },
  overlay~last={
    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
      (frame.north~west)
      |-
      (frame.south~east)
```

```
                        -|
                        (frame.north~east);
                    }
                  },
                keythms_tcbthmbox_M/.style={},
                keythms_tcbthmbox_S/.style={ % first and middle same as M
                  overlay~unbroken={
                    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
                      (title.south~west)
                      --
                      (title.south~east);
                    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
                      (frame.north~west)
                      --
                      (frame.south~west);
                    },
                  overlay~last={
                    \draw[line~width=\l_keythms_tcbthmbox_thickness_dim]
                      (frame.north~west)
                      --
                      (frame.south~west);
                    }
                  },
                }
            }
          \keys_set:ne { keytheorems/thm }
            {
              tcolorbox =
                {
                  keythms_tcbthmbox_default,
                  \l__keythms_tcbthmbox_keys_clist
                }
            }
        },
      thmbox .default:n = M,
    }
  }

\cs_new_protected:Npn \__keythms_storeall_code:
  {
    \cs_set_eq:NN \__keythms_withhooks_begin:nn \__keythms_grab_begin:nn
    \cs_set_eq:NN \__keythms_withhooks_begin:nnn \__keythms_grab_begin:nnn
    \cs_set_eq:NN \__keythms_withhooks_begin:nnV \__keythms_grab_begin:nnV
    \cs_set_eq:NN \__keythms_withhooks_end:n \__keythms_grab_end:n
  }

\hook_gput_code:nnn { begindocument/before } { . }
  { % use 'provide' in case user defines their own translation in preamble
    \ProvideTranslationFallback { keythms_listof_title } { List~of~Theorems }
    \ProvideTranslationFallback { keythms_continues } { continuing~from~p.\, }
    \bool_if:NT \g__keythms_autotranslate_bool
      {
        \ProvideTranslation { English } { keythms_listof_title } { List~of~Theorems }
        \ProvideTranslation { English } { keythms_continues } { continuing~from~p.\, }
        % from DeepL; I don't know these languages!
        \ProvideTranslation { French } { keythms_listof_title } { Liste~des~théorèmes }
        \ProvideTranslation { French } { keythms_continues } { suite~de~la~p.\, }
        \ProvideTranslation { German } { keythms_listof_title } { Liste~der~Theoreme }
```

```
        \ProvideTranslation { German } { keythms_continues } { weiter~von~Seite~ }
        \ProvideTranslation { Spanish } { keythms_listof_title } { Lista~de~teoremas }
        \ProvideTranslation { Spanish } { keythms_continues } { continúa~de~la~p.\, }
      }
  }

\ProcessKeyOptions[keytheorems]

\file_input_stop:

%% Another idea: only print "up-to-now" theorems, or by section
%%               This is complicated. See etoc. And acro (\acbarrier).

%% Another: use style key for multiple theorems, like
%    \keytheoremset{style={<keys>}}
%    \newkeytheorem{thm1}
%    \newkeytheorem{thm2}...
% Well maybe not. Because should these style keys also be applied to a called
% \newkeytheoremstyle? Or only used for that? Question is: should user be able to style theorems
% without an explicit call to \newkeytheorem{<thm>}[style=<style>]?

%% Another: link theorem to restated (see TeX.sx)

%% Ideas/Issues:
% -- Fix equation, etc. numbering in restated theorems when numbered by
%      chapter, section, etc. Or leave it up to user to add these counters
%      with restate-counters?
% ✓✓ (done) listhack
% -- Proof hooks? Other proof customization? New proof-like envs? feature creep...
% -✓ (partial) \newtheoremstyle overwrites existing styles. Should \newkeytheoremstyle check
%      if style exists? Relevant for plain, remark, definition. Could provide
%      \renewkeytheoremstyle. Further, if there is use for (re)defining theorems
%      mid-document, could make cmds usable after preamble by removing package loading
%      in keys
%    Slightly more complicated than anticipated, need to deal with hooks
% -- Currently numbered=unless-unique + parent is incompatible with restate, but this is
%      true also with thmtools.
% ✓✓ (fixed) Idea for qed, tcolorbox in style: in thm, just check if already set and adjust accordi
% ✓✓ (done) What about \zlabel and other "label" commands in restated theorem? Should
%      there be an interface for disabling them?
% -- unless-unique more general: https://tex.stackexchange.com/a/705572/208544
% -- Rename "preheadhook" etc. to "prehead"?
% -- \listofkeytheorems does not print restated theorems. Should an option be added
%      to do this? No, right?
% ✓✓ (done) Language support? At least for list of theorems title
% ✓✓ (done) Interface for adding entries to list of theorems. In fact, should
%      benchmark the "add to seq then write at end of file" approach vs.
%      traditional \addcontentsline approach
% -✓ (partial) "short name" key for theorems to replace thmtools'
%      "name={[short name]name}". Should we also support this syntax?
%      Progress: key implemented but not yet compat with continues
% -- And what about "restate={[options]foo}" syntax? How useful is this?
% ✓✓ (done) Should "continues(*)" theorems appear in list of theorems? Decide and add option to
%      change default.
% ✓✓ (done) Add "shaded" and "thmbox" keys that use tcolorbox under the hood.
%      In thmtools-compat or available always? Or as "library"?
%      Currently thmbox has white background, want transparent but has issue:
%      https://tex.stackexchange.com/a/706216/208544
```

```
% ✓✓ (done) Should more hook operations be available in user commands, like labels and removing?
%    Answer: no, I don't think so. \addtotheoremhook is for simple things.
%    Programmers can use the usual hook interface with more verbose labels
% ✓✓ (done) Document how to disable things like footnotes in restated theorems.
% ✓✓ (done) Should indent be suppressed after tcolorbox theorems? It is for thmtools' shaded and th
%      Just setting \tcbset{after=\par\@endpetrue} doesn't work since there is code in between but
%      \AddToHook{keytheorems/<env>/postfoot}[keythms_hook_keys]{\par\@endpetrue} seems to work
%    Answer: I say no, since it's not suppressed after regular theorems.
% -✓ (partial) break + tcolorbox-no-titlebar adds too much space if theorem starts with list
%      listhack fixes this.
% ✓✓ (done) rename package to "keytheorems"?
% ✓✓ (done) should all code added to hooks have the "keytheorems" label as opposed to "."?
% ✓✓ (fixed) seq approach doesn't work well with \includeonly
% ✓✓ (done) chaptervspacehack
% -- Should there be a way to add content lines to \listofkeytheorems with print-body?
%      Currently we disable code added with \addtotheoremcontents
% ✓✓ (done) hyperref does not jump to correct location for tcolorbox theorems
% -- use \MakeLinkTarget instead of dummy counter?
% ✓✓ (done) should default theorem style be changed if ams class is loaded? Yes,
%      and need general mechanism for setting up class defaults
% ✓✓ need to clear .thlist file if no restate or list of
% -- Beamer issues:
%    - tcolorbox theorems not compatible with beamer
%    - cannot give action spec to theorems
% -- Idea: delay making hooks until begindocument to avoid using internal cmd
% ✓✓ (done) fix acmart defaults
% ✓✓ (done) use different dummy counters for unnumbered, continues, and restate
% ✓✓ (done) seq for custom lists of theorems
% ✓✓ make code more modular
% -- option to restate without writing to file
% ✓✓ (done) code with brackets fail in note key, e.g. \cite[bla]{ref}
% ✓✓ (done) with tcolorbox theorems, spaceabove and spacebelow should always be set to 0pt
% ✓✓ no-auto-translate or something like that
```

Index