The mermin-equations package*

Evan Berkowitz

2026/11/18

Abstract

The mermin-equations package helps physicists write equations the way N. David Mermin asks in "What's Wrong with These Equations?" [1]. An equation is labeled by the noun phrases that describe it, as many as the prose needs, with `\aka`, and cited by one of those phrases with `\eq`, which prints the phrase and the number together. Citing an equation by a bare number becomes the harder thing to write.

1 Introduction

In his 1989 column [1] Mermin proposed three rules for equations in physics writing. Fisher's rule asks that every displayed equation be numbered, whether or not the text refers to it yet, because readers, referees, and later papers will. The Good Samaritan rule asks that an equation be cited by a phrase as well as a number, so that the reader need not page back to learn what "(11)" says: one writes "the anomaly condition (11)", never a bare "(11)" and never "Eq. (11) shows". An equation number locates; the noun phrase says what is there. The third rule, that mathematics is prose, asks that a display be punctuated as part of its sentence, so that the paragraph reads straight through the equation.

LaTeX makes the Good Samaritan rule harder than it need be. A label is an arbitrary key, so nothing ties a citation to the phrase that should accompany it, and amsmath allows a numbered line exactly one `\label`, although the right phrase depends on why the equation is being cited. The same line may be "the anomaly condition" in one paragraph and "the Pythagorean condition" in another. This package makes the phrase the key. A line is labeled by each phrase that describes it, and `\eq` cites it by one of them, printing the phrase and the number.

2 Usage

`\aka` `\eq`

Load the package with `\usepackage{mermin-equations}`. Then `\aka{`⟨*phrase*⟩`}`, for "also known as", attaches ⟨*phrase*⟩ as a key to the current equation, or to the current line of a multi-line display, in addition to its `\label`. It may be used any number of times on a line, before or after the `\label`, or on a line that has no `\label` at all. Every key refers to the same number and, with hyperref, to the same anchor. `\eq{`⟨*phrase*⟩`}` prints ⟨*phrase*⟩, a tie, and the equation number in parentheses, linked by hyperref to the equation.

*This document corresponds to mermin-equations v0.1.1, dated 2026/11/18.

Consider two-flavor chiral gauge theories in two dimensions, whose gauge anomaly cancels when the left- and right-handed charges obey

$$L_1^2 + L_2^2 = R_1^2 + R_2^2. \tag{1}$$

The source of this display, and of a paragraph that cites it, reads

```
\begin{equation}
  L_1^2+L_2^2=R_1^2+R_2^2.
  \label{anomaly condition}
  \aka{Pythagorean condition}
\end{equation}

The 3450 model, with charges $(3,4\,\vert\,5,0)$, satisfies
the \eq{anomaly condition}.
Read as a statement about integers, the same line is
the \eq{Pythagorean condition}, and $3^2+4^2=5^2$ is its
smallest nontrivial solution.
```

and typesets as

> The 3450 model, with charges $(3,4\,|\,5,0)$, satisfies the anomaly condition (1). Read as a statement about integers, the same line is the Pythagorean condition (1), and $3^2 + 4^2 = 5^2$ is its smallest nontrivial solution.

We advise leaving the article out of the phrase and in the prose. Capitalization is the reason. A sentence that opens with a citation needs "The", one that cites in the middle needs "the", and `The \eq{anomaly condition}` handles both, because the capital falls on the article, which the sentence supplies. With the article inside the key, the capitalized and lowercase forms would be different labels. Keeping it out also leaves the sentence free to say "this", "that", or nothing at all. This is only advice: a label such as `the anomaly condition` works like any other key.

Lines of an `align` are labeled the same way, one set of phrases per line, and a line may be labeled by `\aka` alone:

```
\begin{align}
  S &= \int d^2x\,\mathcal{L} \label{action} \aka{fermion action} \\
  P_{L,R} &= \tfrac12(1\pm\gamma_5) \aka{chiral projectors}
\end{align}
```

2.1 Phrases and keys

Labels written with an `eq:` prefix, as many papers already have, need no change: when no label ⟨*phrase*⟩ exists, `\eq` uses `eq:`⟨*phrase*⟩. For example, `\eq{shift relation}` cites the line labeled `\label{eq:shift relation}`; so does `\eq{eq:shift relation}`, and both print "shift relation", without the prefix. If `shift relation` labels a section, say, and `eq:shift relation` an equation, `\eq` cites the equation; if both label equations, different ones, `\eq` cites the first and warns.

`\eq` prints its argument as the phrase. A phrase that cannot be a key, because it contains mathematics or a command, is given where the label is defined, as

the optional argument of `\aka`, and every citation then uses the plain key. For example, a compact boson on a lattice may be given the Villain action

$$S_\Gamma = \frac{\beta}{2} \sum_\ell (d\phi - 2\pi n)_\ell^2 \,. \tag{2}$$

The source of this display, and of a paragraph that cites it by both phrases, reads

```
\begin{equation}
  S_\Gamma=\frac{\beta}{2}\sum_{\ell}\left(d\phi-2\pi n\right)_\ell^2.
  \aka[$\Gamma$ action]{Gamma action}
  \aka{Villain model}
\end{equation}

The integer field $n$ makes the \eq{Gamma action} invariant
under $\phi\to\phi+2\pi k$ with $n\to n+dk$.
Summed over $n$, the \eq{Villain model} is a periodic function
of each $d\phi_\ell$.
```

and typesets as

> The integer field n makes the Γ action (2) invariant under $\phi \to \phi + 2\pi k$ with $n \to n + dk$. Summed over n, the Villain model (2) is a periodic function of each $d\phi_\ell$.

A key that is not printable text, such as `E_field`, needs a phrase in the same way. Without one `\eq` prints the key verbatim and, once the label exists, warns. To give a phrase to a line that already has a `\label`, add `\aka` with the same key, `\label{E_field}\aka[$E$ field]{E_field}`, or replace the `\label` with it.

2.2 How `\aka` behaves

Because `\label` is left alone, two `\label`s on one line remain the error amsmath has always reported; only `\aka` adds keys. Citing a key with `\ref`, `\eqref`, `\pageref`, or the cleveref command `\cref` works exactly as for a key given to `\label`. Outside a display `\aka` is simply `\label`, so it can also give a section, a figure, or a theorem a second name, and it may stand inside a heading or caption just as `\label` may. `\eq`, however, is for equations only, since it prints a parenthesized number, and it warns when its key names anything else.

On a line without a number, such as one with `\notag` or in `equation*`, `\aka` does what `\label` would do there, which is seldom what is wanted, and `\eq` cannot tell: on an align line it labels the next numbered line, even one with its own `\label`, or nothing on the last line; in gather its number is empty; and in `equation*` it takes the section's number. Put it on a numbered line.

2.3 Mermin's rules as warnings

Two of Mermin's rules become warnings, which never stop compilation. *Fisher's rule*, "number all displayed equations", warns at every amsmath display, empheq, or eqnarray that typesets no number: `equation*`, `\[...\]`, or an align whose every line is `\notag`. (Other packages' unnumbered displays, such as breqn's `dmath*`,

and plain TEX's `$$...$$`, are not checked.) A derivation that is numbered once, on its last line, is one equation and is numbered:

```
\begin{align}
  f &= g \nonumber\\
    &= h \label{f=h}
\end{align}
```

The mathtools option `showonlyrefs`, which numbers only the equations that are cited, is the practice Mermin calls Occam's rule, a heresy; with it, one warning says so. The *Good Samaritan rule*, "when referring to an equation identify it by a phrase as well as a number", warns at every `\eqref`, `\cref`, `\Cref`, or `\labelcref` of an equation, even one that words precede, as in `the Dirac equation~\eqref{Dirac}`: `\eq` gives the phrase and the number together, from one place. Page references such as `\cpageref` are not checked, and neither is plain `\ref`, which other commands use internally; so `Eq.~(\ref{x})`, a common REVTeX style, passes unremarked. The options `nofisher` and `nosamaritan` turn the warnings off.

The warnings appear in the log file, and in the summaries of tools such as `latexmk`, like this:

```
Package mermin-equations Warning: 'equation*' display without a number.
(mermin-equations)                Fisher's rule: number every display
(mermin-equations)                (option nofisher) on input line 4.

Package mermin-equations Warning: 'chain' is cited by number alone.
(mermin-equations)                Good Samaritan rule: name it, as \eq does
(mermin-equations)                (option nosamaritan) on input line 12.
```

The first names the environment (`\[...\]` is `equation*`) and the line where the display ends; the second names the key and the line of the citation. With `showonlyrefs` a single warning appears at `\begin{document}`:

```
Package mermin-equations Warning: mathtools' showonlyrefs numbers only the
(mermin-equations)                equations that are cited, which Mermin calls
(mermin-equations)                Occam's rule; Fisher's rule numbers them all
(mermin-equations)                (option nofisher silences this).
```

2.4 Long names

`\alsolabel` `\eqcite` Short names are easily taken. `\alsolabel` and `\eqcite` are the same commands as `\aka` and `\eq`, under names no other package uses. Several packages, among them nccmath, define their own `\eq`, and a document may define its own `\aka`. mermin-equations then leaves the existing command alone, says so in a warning, and the document uses the long name instead.

3 Compatibility

3.1 Requirements and loading

The package requires a LATEX format from 2021-11-15 or later, for the hooks it uses, and loads amsmath. Load amsmath with options, such as `fleqn`, before

mermin-equations, as before any package that loads it. hyperref and cleveref may be loaded before or after mermin-equations; cleveref, as always, after amsmath. arXiv compiles papers with its own fixed TEX Live installations, which do not receive new packages, so a paper submitted there should include `mermin-equations.sty` among its files.

3.2 Displays

The package works in the amsmath environments equation, align, gather, multline, split (with the keys given in the enclosing equation), and subequations, and in empheq; in breqn's `dmath`, `dseries`, and `dgroup`; in the cases package's `numcases` and `subnumcases`; in IEEEtrantools' `IEEEeqnarray`; in subeqnarray; and in the kernel's eqnarray. With the mathtools option `showonlyrefs`, a line is numbered when any of its keys is cited.

3.3 Other documents

With xr or xr-hyper, `\eq{S-`⟨*key*⟩`}` cites an equation of a document loaded with `\externaldocument[S-]`, and prints its phrase, or its key without the prefix; the `eq:` fallback reaches it too, so `\eq{S-sum rule}` cites `\label{eq:sum rule}` there. A document that does not use mermin-equations has no record of which labels are equations, so `\eq` treats all of its labels as equations. Load mermin-equations before calling `\externaldocument`: a document read earlier is imported without its phrases or its record of which labels are equations, so its keys print with the prefix, the `eq:` fallback does not reach it, and `\eq` warns about every one.

A paper that adopts mermin-equations while `\includeonly` leaves out chapters may see such warnings for equations in those chapters, until each is compiled once with the package.

3.4 Headings, bookmarks, and running heads

`\aka`, like `\label`, vanishes from tables of contents, running heads, and PDF bookmarks. In a bookmark `\eq` gives its phrase and number as plain text. A phrase with mathematics is dropped from bookmarks, as mathematics is from any bookmark; hyperref's `\texorpdfstring` gives a plain-text form, as in `\aka[\texorpdfstring{$E$}{E} field]{E_field}`.

In a heading or running head that a class sets in capitals with `\MakeUppercase`, the phrase is left as written, from the LaTeX release of 2022-06 on, since it is printed from its key; for a running head, give the heading a short form without `\eq`, as in `\section[...]{...}`. REVTeX sets the section headings themselves in capitals, and the phrase stays lowercase there ("THE anomaly condition (1)"); `\eqref` in the same place fails altogether, printing "(??)". Classes that use the primitive `\uppercase`, such as mnras and aastex631, capitalize the key itself, which breaks `\eq` there as it breaks `\eqref`; wrap such a citation in a command defined in the preamble.

3.5 Classes and tools

In beamer, `\aka<`⟨*overlay*⟩`>` passes the overlay to `\label`, as `\label<`⟨*overlay*⟩`>` does. In a display the overlay of `\aka` or `\label` is dropped, and the label's anchor

is on the frame's first slide.

The iopart class conflicts with amsmath, which this package loads; when iopart is the class, the package applies IOP's usual remedy, undefining `equation*` first.

latexdiff shows a changed `\eq` only when told that it is safe to keep in deleted text, with `--append-safecmd=eq,eqcite`; do not add `aka`, which would bring deleted labels back. An `\aka` in text latexdiff marks as deleted is ignored.

showkeys and showlabels show the first key of a display line, not the keys `\aka` adds to it.

3.6 Not supported

autonum copies `\label` before this package can attach its extra keys, and numbers a line by its first key only. witharrows' `DispWithArrows` keeps its labels where `\aka` cannot reach them.

4 Testing

The package's repository, https://github.com/evanberkowitz/mermin, also holds a test suite and the issue tracker. Running `make test` there typesets every environment above, with the keys given before, after, and instead of `\label`, in four package configurations, and checks that each key in the resulting `.aux` file carries its line's number and anchor. It also checks the text `\eq` prints, its warnings, and each of the interactions above. The suite runs with pdfLaTeX, LuaLaTeX, and XeLaTeX on the current TeX Live, and with pdfLaTeX on TeX Live 2021, 2023, and 2025.

References

[1] N. D. Mermin, "What's Wrong with These Equations?", *Physics Today* **42**(10), 9 (1989), doi:10.1063/1.2811173.

5 Implementation

The implementation has eleven parts. Parts 5.1–5.3 set up the package and the records it keeps in the `.aux` file. Parts 5.4–5.6 define `\alsolabel` and the machinery that gives a display line several keys. Part 5.7 imports the records of another document through xr. Parts 5.8–5.10 define `\eqcite` and the short names, and make both behave in headings and bookmarks. Part 5.11 turns two of Mermin's rules into warnings.

The narrative is here; the code carries short comments wherever a construction looks odd or exists for one particular package, kernel version, or engine, so that a reader of the bare `mermin-equations.sty` is not lost. Two words are kept apart: a key is *marked* when it is recorded as an equation, and an `.aux` file is *signed* when it was written with this package.

5.1 Loading

The package needs amsmath, whose display machinery it extends. IOP's iopart class defines `equation*` itself, which amsmath then refuses to redefine; IOP's documented remedy is to undefine it first, and the package does so when iopart is the class and amsmath is not yet loaded.

```
⟨*package⟩
% iopart.cls defines equation*, and amsmath would stop with "Command
% \equation* already defined"; this is IOP's own workaround.  amsmath's
% \@xp is not defined yet, hence \expandafter.
\@ifclassloaded{iopart}{%
  \@ifpackageloaded{amsmath}{}{%
    \expandafter\let\csname equation*\endcsname\relax
    \expandafter\let\csname endequation*\endcsname\relax}}{}
\RequirePackage{amsmath}
```

Two options turn off the warnings of part 5.11.

```
\newif\ifmermin@fisher \mermin@fishertrue
\newif\ifmermin@samaritan \mermin@samaritantrue
\DeclareOption{nofisher}{\mermin@fisherfalse}
\DeclareOption{nosamaritan}{\mermin@samaritanfalse}
\ProcessOptions\relax
```

5.2 Utilities

`\mermin@warn` `\mermin@warn@noline` Every warning goes through these, so the package's name is written once. A warning's lines are kept under 45 characters: the prefix `(mermin-equations)` and its padding take 34 of the 79 that TEX prints before it breaks a line of the log.

```
\def\mermin@warn{\PackageWarning{mermin-equations}}
\def\mermin@warn@noline{\PackageWarningNoLine{mermin-equations}}
```

`\mermin@firstchar` `\mermin@ifeqprefix` Two expandable tests on a key look at its characters rather than its tokens. `\mermin@firstchar{`⟨*text*⟩`}` expands to the first character of ⟨*text*⟩, as a character of category 12. `\mermin@ifeqprefix{`⟨*key*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}` tests whether ⟨*key*⟩ begins with `eq:`. Both detokenize first, for two reasons. The raw key is never scanned as a delimited argument, which inside an alignment would end the cell at an `&` in the key; and the test does not depend on categories, which babel languages change: French makes the colon active in the document body. `\if` compares character codes only, so the category-12 characters of the detokenized key match the letters typed here.

```
% Detokenized, so that an & in a key cannot end an alignment cell and
% babel's active characters still compare as characters.
\def\mermin@firstchar#1{%
  \@xp\mermin@firstchar@\detokenize{#1}\relax\mermin@nil}
% Without the \relax an empty key would make #1 \mermin@nil, and #2
% would run away looking for it.
\def\mermin@firstchar@#1#2\mermin@nil{#1}
% The three \relax pad keys shorter than three characters.
\def\mermin@ifeqprefix#1{%
  \@xp\mermin@ifeqprefix@\detokenize{#1}\relax\relax\relax\mermin@nil}
\def\mermin@ifeqprefix@#1#2#3#4\mermin@nil{%
  \ifnum\if e#1\if q#2\if :#3 1\else 0\fi\else 0\fi\else 0\fi=1
    \@xp\@firstoftwo
```

```
  \else
    \@xp\@secondoftwo
  \fi}
```

`\mermin@ifunprintable` `\mermin@verbatim` A key that is printed as text, because it has no typeset phrase, must not contain characters that typesetting would misread. `\mermin@ifunprintable` tests for them; `\mermin@verbatim` prints a key character by character in typewriter type, whose fonts have every ASCII character. Braces are not in the list: `\in@` cannot find a brace character, and a key with braces prints, only without them.

```
\edef\mermin@specials{% < > | print as other glyphs in OT1 fonts
  \detokenize{_^$&<>|}\@percentchar\@backslashchar}
\def\mermin@ifunprintable#1{%
  \edef\mermin@tmp{\detokenize{#1}}%
  \let\mermin@bad\@empty
  \@xp\@tfor\@xp\mermin@c\@xp:\@xp=\mermin@specials\do{%
    \edef\reserved@a{\noexpand\in@{\mermin@c}{\mermin@tmp}}%
    \reserved@a
    \ifin@\def\mermin@bad{x}\fi}%
  \ifx\mermin@bad\@empty\@xp\@secondoftwo\else\@xp\@firstoftwo\fi}
\def\mermin@verbatim#1{{\normalfont\ttfamily\detokenize{#1}}}
```

5.3 Records in the `.aux` file

Three kinds of record travel through the `.aux` file, like labels, so that a citation may come before its equation:

- the typeset phrase given to `\alsolabel`, as `\mermin@newphrase` with the arguments ⟨*key*⟩ and ⟨*phrase*⟩, which defines `\mermin@ph@`⟨*key*⟩;
- `\mermin@equation{`⟨*key*⟩`}`, which marks ⟨*key*⟩ as an equation by defining `\mermin@eq@`⟨*key*⟩, so that `\eqcite` can warn about anything else;
- `\mermin@signature`, once per file, which signs the file as written with this package and sets `\mermin@signed`.

Until a signed file has been read, `\eqcite` does not warn about non-equations: a paper that has just adopted the package has an `.aux` file full of equation labels and no marks for them.

`\mermin@record` `\mermin@writephrase` `\mermin@markeq` Records do not depend on the page, so they are written immediately. The argument is expanded as `\label` expands its key, robust commands excepted, so that a key such as `sec\thesection:mass` is recorded as the same key that `\label` writes.

```
\def\mermin@record#1{%
  \if@filesw
     % Expand the key, robust commands excepted, as \label's write does
     % (which expands once more at shipout), but write now.
    \begingroup
      \let\protect\@unexpandable@protect
      \immediate\write\@auxout{#1}%
    \endgroup
  \fi}
\def\mermin@writephrase#1#2{%
```

```
  \mermin@record{\string\mermin@newphrase{#1}{\unexpanded{#2}}}}
\def\mermin@markeq#1{\mermin@record{\string\mermin@equation{#1}}}
```

`\mermin@auxheader` `\mermin@equation` `\mermin@signature`

At the beginning of the document the package writes, to each `.aux` file, `\providecommand`s for the three records, so that a document that stops loading the package still reads its old files, and then the signature. Files of chapters read with `\include` get the same lines: under `\includeonly` their old files are read without being rewritten.

```
\def\mermin@auxheader#1{%
  \immediate\write#1{%
    \string\providecommand\string\mermin@newphrase[2]{}%
    \string\providecommand\string\mermin@equation[1]{}%
    \string\providecommand\string\mermin@signature{}}%
   % The signature gets a line of its own: xr looks only at the first
   % token of each line.
  \immediate\write#1{\string\mermin@signature}}
\AtBeginDocument{\if@filesw\mermin@auxheader\@auxout\fi}
 % Under \includeonly, the old .aux files of excluded chapters are read
 % but not rewritten, so each needs its own \providecommand lines.
\AddToHook{include/before}{\if@filesw\mermin@auxheader\@partaux\fi}
\def\mermin@signature{\global\let\mermin@signed\@empty}
\def\mermin@equation#1{\@xp\gdef\csname mermin@eq@#1\endcsname{}}
```

`\mermin@newphrase` `\mermin@checkphrases`

When LaTeX rereads the `.aux` file at the end of the document to look for changed labels, a changed phrase also asks for another run. So does a removed one: the keys that had phrases when the run began are listed in `\mermin@phkeys`, and after the reread any of them that was not seen again sets the same flag. Phrases are compared by their characters only.

```
\let\mermin@phkeys\@empty
\def\mermin@newphrase#1#2{%
   % The end-of-document reread is recognized by \@setckpt, which the
   % kernel (and REVTeX) makes \@gobbletwo only then.  \@newl@bel is the
   % obvious test, but acronym changes it during the reread.
  \ifx\@setckpt\@gobbletwo
    \global\@namedef{mermin@seen@#1}{}%
     % Detokenized: unicode-math changes the category of letters such as
     % a literal Gamma after the .aux file is first read, and comparing
     % tokens would ask for a rerun forever.
    \edef\reserved@a{\detokenize{#2}}%
    \edef\reserved@b{%
      \detokenize\@xp\@xp\@xp{\csname mermin@ph@#1\endcsname}}%
    \ifx\reserved@a\reserved@b\else\@tempswatrue\fi
  \else
    \@xp\gdef\csname mermin@ph@#1\endcsname{#2}%
    \g@addto@macro\mermin@phkeys{\do{#1}}%
  \fi}
\def\mermin@checkphrases{%
   % Runs inside \enddocument's group, so \do is redefined only there.
  \def\do##1{\@ifundefined{mermin@seen@##1}{\@tempswatrue}{}}%
  \mermin@phkeys}
 % afteraux runs after the reread and before the kernel's rerun warning.
\AddToHook{enddocument/afteraux}{\if@filesw\mermin@checkphrases\fi}
```

5.4 The command \alsolabel

\alsolabel Where LaTeX disables \label, as it does in the text written to a table of contents or a running head, \alsolabel must disappear with it, together with its arguments. \alsolabel is therefore expandable: it looks at what \label means and expands either to a gobbler or to the real command, \mermin@alsolabel, which is robust. Releases before 2025-06-01 disable \label as \@gobble or \relax, as hyperref, memoir, and REVTeX do on any release; later releases as \@gobble@om, or \@gobble@with@sphack@om where the text is typeset, which \alsolabel then uses itself; and the caption package as \caption@gobble.

\alsolabel also disappears in text that latexdiff marks as deleted. latexdiff keeps a deleted display, as displaymath between \DIFdelbegin and \DIFdelend, and comments out the \labels in it, but its list of label commands is fixed and cannot be extended; a deleted \alsolabel would define its key a second time, possibly with the wrong number. The package therefore sets \ifmermin@deleted between the two markers. latexdiff defines them in the preamble, after \usepackage{mermin-equations}, so they are wrapped only at begindocument/end.

```
\newif\ifmermin@deleted
\AddToHook{begindocument/end}{%
  % latexdiff defines its markers in the preamble, after
  % \usepackage{mermin-equations}.  They are made \protected so that,
  % in a moving argument such as a heading, they are written as
  % themselves and still set the flag when the heading is typeset
  % again.
  \ifdefined\DIFdelbegin
    \let\mermin@DIFdelbegin\DIFdelbegin
    \let\mermin@DIFdelend\DIFdelend
    \protected\def\DIFdelbegin{%
      \global\mermin@deletedtrue\mermin@DIFdelbegin}%
    \protected\def\DIFdelend{%
      \mermin@DIFdelend\global\mermin@deletedfalse}%
  \fi}
\def\alsolabel{%
  \ifmermin@deleted \mermin@aka@use\mermin@gobble@aka\fi
   % \@gobble and \relax: kernels before 2025-06, and hyperref,
   % memoir, and REVTeX on any kernel.
  \ifx\label\@gobble \mermin@aka@use\mermin@gobble@aka\fi
  \ifx\label\relax \mermin@aka@use\mermin@gobble@aka\fi
   % \@gobble@om and \@gobble@with@sphack@om: kernels from 2025-06.
  \ifx\label\@gobble@om \mermin@aka@use\mermin@gobble@aka\fi
  \ifx\label\@gobble@with@sphack@om
    \mermin@aka@use\@gobble@with@sphack@om\fi
   % caption disables \label this way, in its single-line check and in
   % subcaption's list of figures.
  \ifx\label\caption@gobble \mermin@aka@use\mermin@gobble@aka\fi
  \mermin@alsolabel}
 % \mermin@aka@use ends the test that matched, drops the remaining tests
 % up to \mermin@alsolabel, and leaves #1 in its place.
\def\mermin@aka@use#1#2\mermin@alsolabel{\fi#1}
\NewExpandableDocumentCommand\mermin@gobble@aka{d<> o m}{}
```

\mermin@alsolabel The real command records the optional phrase and then calls whatever \label

means at that point, with the flag `\ifmermin@also` raised. In a display `\label` is `\label@in@display`, which part 5.6 wraps to read the flag; elsewhere `\label` acts as usual, and `\alsolabel` gives a section or a figure a second name. The measuring passes of amsmath gobble or defer `\label`, and `\alsolabel` inherits the same treatment. A beamer overlay, `<`⟨*overlay*⟩`>`, goes to beamer's `\label` outside displays, and is dropped inside them. `\mermin@doclabel` is the document's `\label`, saved at `begindocument/end` after hyperref and beamer have set it.

```
\newif\ifmermin@also
\AddToHook{begindocument/end}{% after hyperref and beamer set \label
  \let\mermin@doclabel\label}
\NewDocumentCommand\mermin@alsolabel{d<> o m}{%
   % \label ends with \@esphack, but inside the group below it cannot
   % remove the space after the command in running text; the command
   % therefore has its own \@bsphack and \@esphack.
  \@bsphack
  \IfValueT{#2}{\mermin@writephrase{#3}{#2}}%
  \begingroup
    \mermin@alsotrue
     % cleveref's multline trial pass calls this hook directly instead
     % of \label@in@display; see \mermin@cref@entry.
    \ifdefined\mermin@cref@old
      \let\cref@old@label@in@display\mermin@also@stored
    \fi
    \IfValueTF{#1}%
       % Only beamer's \label takes an overlay: the document's \label,
       % or \mermin@marklabel wrapping it.  amsmath's display \label
       % does not.
      {\ifx\label\mermin@doclabel
         \label<#1>{#3}%
       \else\ifx\label\mermin@marklabel
         \label<#1>{#3}%
       \else
         \label{#3}%
       \fi\fi}%
      {\label{#3}}%
  \endgroup
  \@esphack}
```

5.5 Which keys are equations

`\mermin@markargs` `\mermin@keyof`

A key is marked as an equation where a display registers it, and not where labels are written: `\ltx@label`, which writes them, is reached from outside displays too. A key arrives as `{`⟨*key*⟩`}`, as `[`⟨*type*⟩`]{`⟨*key*⟩`}` with cleveref's optional argument, or as amsmath stores it without cleveref, the bare ⟨*key*⟩. `\mermin@keyof` sets `\mermin@rawkey` to the key in any of these forms.

```
\def\mermin@markargs#1{%
   % The \iffalse braces raise TeX's alignment brace count without
   % opening a group, so that an & in a bare key, scanned as a delimited
   % argument inside align, does not end the cell.
  \iffalse{\fi
  \mermin@keyof#1\mermin@nil
  \@xp\mermin@markeq\@xp{\mermin@rawkey}%
```

```
  \iffalse}\fi}
 % [type] is cleveref's optional argument of \label.
\def\mermin@keyof{\@ifnextchar[\mermin@keyof@opt\mermin@keyof@plain}
\def\mermin@keyof@plain#1\mermin@nil{\def\mermin@rawkey{#1}}
\def\mermin@keyof@opt[#1]#2\mermin@nil{\def\mermin@rawkey{#2}}
```

`\mermin@wraplabel` `\mermin@marklabel` Some equations are labeled by an ordinary `\label`, not by `\label@in@display`: the label of a subequations group, numbered like "(12)", and the labels in breqn's displays, in the cases package's `numcases`, in IEEEtrantools' `IEEEeqnarray`, in subeqnarray, and in the kernel's eqnarray. Inside these environments `\label` is `\mermin@marklabel`, which marks its key and calls the `\label` it replaced. The amsmath displays within them install their own `\label`, and an environment nested in another, such as `dmath` in `dgroup`, finds `\label` already wrapped and leaves it.

```
 % For environments that label equations with the ordinary \label.
\def\mermin@wraplabel{%
  \ifx\label\mermin@marklabel\else
    \let\mermin@marklabel@orig\label
    \let\label\mermin@marklabel
  \fi}
 % beamer's \label takes an overlay, <...>, before the optional
 % argument.
\def\mermin@marklabel{%
  \@ifnextchar<\mermin@marklabel@ov
    {\@ifnextchar[\mermin@marklabel@opt\mermin@marklabel@plain}}
\def\mermin@marklabel@plain#1{%
  \mermin@markeq{#1}\mermin@marklabel@orig{#1}}
\def\mermin@marklabel@opt[#1]#2{%
  \mermin@markeq{#2}\mermin@marklabel@orig[#1]{#2}}
\def\mermin@marklabel@ov<#1>#2{%
  \mermin@markeq{#2}\mermin@marklabel@orig<#1>{#2}}
\def\mermin@markenv#1{\AddToHook{env/#1/begin}{\mermin@wraplabel}}
\@for\mermin@env:=%
    subequations,% amsmath
    dmath,dseries,dgroup,% breqn
    numcases,subnumcases,% cases
    IEEEeqnarray,IEEEeqnarray*,% IEEEtrantools
    eqnarray,% LaTeX kernel
    subeqnarray% subeqnarray
  \do{\@xp\mermin@markenv\@xp{\mermin@env}}
 % subeqnarray labels each row with \slabel, which writes \newlabel
 % itself.
\AddToHook{env/subeqnarray/begin}{%
  \let\mermin@slabel@orig\slabel
  \def\slabel#1{\mermin@markeq{#1}\mermin@slabel@orig{#1}}}
```

5.6 Extra keys on a display line

amsmath does not write a display label when it meets `\label`. It stores the key in `\df@label` and writes it with `\ltx@label{`⟨*contents of* `\df@label`⟩`}` once the line's tag, and so `\@currentlabel` and the hyperref anchor, are known. The package keeps the extra keys of a line in a queue and writes them immediately after that primary label, so they receive the same number and anchor.

Three complications shape the code. First, cleveref stores keys in a different form, wrapping the key in braces or preceding it with an optional argument, so every queued key is stored in exactly the form `\label@in@display` would have stored it. Second, amsmath and empheq typeset some displays twice, once to measure them, and may discard a primary label without writing it. The queue is therefore tagged with the primary label it belongs to and written only when that label is, and a key is never queued twice. Third, `\alsolabel` may come before `\label` or without it. It then occupies `\df@label` itself, *promoted*, and a later `\label` reclaims the slot and moves the earlier key into the queue.

`\mermin@queue` `\mermin@tag` `\mermin@promoted` The queue holds extra keys as `\do{`⟨*stored form*⟩`}`. The tag is the `\df@label` the queue belongs to. `\mermin@promoted` is nonempty while `\alsolabel` occupies `\df@label`. All three are global, like `\df@label` itself, because the cells of an alignment are groups.

```
\gdef\mermin@queue{}    % the line's extra keys, as \do{<stored form>}
\gdef\mermin@tag{}      % the \df@label the queue belongs to
\gdef\mermin@promoted{} % an \aka key occupying \df@label, if any
```

`\mermin@store` `\mermin@markstore` `\mermin@canonical` `\mermin@store` calls the `\label@in@display` that was in force before the package wrapped it, which stores the key in `\df@label`; `\mermin@markstore` also marks the key as an equation. Their argument is the complete argument list, braces included. `\mermin@canonical{`⟨*key*⟩`}` sets `\mermin@form` to the form in which that `\label@in@display` would store ⟨*key*⟩, by letting it store the key into an emptied `\df@label` and restoring the old contents.

```
\def\mermin@store#1{%
  \begingroup
    % Undo, locally, the redirection of cleveref's inner hook.
    \ifdefined\mermin@cref@old
      \let\cref@old@label@in@display\mermin@cref@old
    \fi
    \mermin@orig@label@in@display#1%
  \endgroup}
\def\mermin@markstore#1{\mermin@markargs{#1}\mermin@store{#1}}
\def\mermin@canonical#1{%
  % Let the original store #1 into an empty \df@label, read the stored
  % form back, and restore \df@label: cleveref and amsmath store keys
  % differently, and queued keys must match the form exactly.
  \global\let\mermin@saved\df@label
  \global\let\df@label\@empty
  \mermin@store{{#1}}%
  \global\let\mermin@form\df@label
  \global\let\df@label\mermin@saved}
```

`\mermin@enqueue` This queues `\mermin@form` for the current `\df@label`. A queue tagged with a different label is stale and is discarded first. A form already in the queue is skipped: a display typeset twice runs its `\alsolabel`s twice, and `\text` typesets its argument four times. With the mathtools option `showonlyrefs` a line is numbered only if its primary label was cited, which mathtools records as `\MT_r_`⟨*label*⟩; a citation of a queued key counts for the line too.

```
\def\mermin@enqueue{%
  \ifx\mermin@tag\df@label\else
    \gdef\mermin@queue{}%
```

```
    \global\let\mermin@tag\df@label
  \fi
  \iffalse{\fi % alignment guard, as in \mermin@markargs
   % mathtools' showonlyrefs numbers a line whose label is cited, which
   % it records as \MT_r_<label>: a cited queued key counts too.
  \@xp\mermin@keyof\mermin@form\mermin@nil
  \@ifundefined{MT_r_\mermin@rawkey}{}%
    {\global\@namedef{MT_r_\df@label}{}}%
  \iffalse}\fi
  \begingroup
    \let\mermin@found\@empty
    \def\do##1{\def\mermin@tmp{##1}%
      \ifx\mermin@tmp\mermin@form \def\mermin@found{x}\fi}%
    \mermin@queue
    \ifx\mermin@found\@empty
      \toks@\@xp{\mermin@form}%
      \xdef\mermin@queue{%
        \unexpanded\@xp{\mermin@queue}\noexpand\do{\the\toks@}}%
    \fi
  \endgroup}
```

`\mermin@also@in@display` `\mermin@also@stored`

`\alsolabel` in a display converts its key to stored form. A key the line's label already has adds nothing, so that `\label{K}\aka[`⟨*phrase*⟩`]{K}` gives an existing label a phrase. If the line has no label yet, the key is promoted; otherwise it is queued.

```
\def\mermin@also@in@display#1{%
  \mermin@canonical{#1}%
  \@xp\mermin@also@stored\@xp{\mermin@form}}
\def\mermin@also@stored#1{%
  \mermin@markargs{#1}%
  \def\mermin@tmp{#1}%
  \ifx\mermin@tmp\df@label   % the line already has this key
  \else\ifx\df@label\@empty % no \label yet: promote this key
    \gdef\df@label{#1}%
    \global\let\mermin@promoted\df@label
    \global\let\mermin@tag\df@label
    \gdef\mermin@queue{}%
  \else
    \gdef\mermin@form{#1}%
    \mermin@enqueue
  \fi\fi}
```

`\mermin@real@in@display`

A real `\label` normally passes straight through, and so still reports a second `\label` on one line as an error. If instead the line's label was promoted from `\alsolabel`, the real key takes the slot and the promoted key joins the queue, unless it is the same key. A promotion left over from a discarded measuring pass no longer matches `\df@label` and is cleared.

```
\def\mermin@real@in@display#1{%
  \ifx\mermin@promoted\@empty
    \mermin@markstore{#1}%
  \else\ifx\mermin@promoted\df@label
     % The line's label is a promoted \aka key: the real key takes the
     % slot, and the promoted one is queued unless it is the same key.
```

```
    \global\let\mermin@form\df@label
    \ifx\mermin@tag\df@label\else\gdef\mermin@queue{}\fi
    \global\let\df@label\@empty
    \mermin@markstore{#1}%
    \global\let\mermin@tag\df@label
    \ifx\mermin@form\df@label\else\mermin@enqueue\fi
    \gdef\mermin@promoted{}%
  \else
     % A promotion left over from a discarded measuring pass.
    \gdef\mermin@promoted{}%
    \mermin@markstore{#1}%
  \fi\fi}
```

`\mermin@label@in@display` `\mermin@cref@entry`

A real `\label` in a display arrives at `\mermin@label@in@display`, with cleveref's optional argument passed through. beamer's display `\label` takes an overlay, which is dropped.

cleveref's multline trial pass bypasses `\label@in@display` and hands its inner hook, `\cref@old@label@in@display`, the stored key: braced, {⟨*key*⟩}, with amsmath before 2.17z, but bare with amsmath 2.17z (TeX Live 2025 and later). `\mermin@cref@entry` accepts either, or [⟨*type*⟩]{⟨*key*⟩}.

```
\def\mermin@label@in@display{%
  \@ifnextchar<\mermin@label@in@display@ov
    {\@ifnextchar[\mermin@label@in@display@opt
                  \mermin@label@in@display@plain}}
\def\mermin@label@in@display@ov<#1>{% beamer: drop the overlay
  \mermin@label@in@display}
\def\mermin@label@in@display@opt[#1]#2{%
  \mermin@real@in@display{[#1]{#2}}}
\def\mermin@label@in@display@plain#1{\mermin@real@in@display{{#1}}}
 % The forms are told apart by first character, never by scanning a
 % bare key, which may hold an &.  \@charlb is a brace character of
 % category 12.
\def\mermin@cref@entry#1{%
  \if\mermin@firstchar{#1}[\@xp\@firstoftwo\else\@xp\@secondoftwo\fi
    {\mermin@cref@entry@opt#1\mermin@nil}%
    {\if\mermin@firstchar{#1}\@charlb
       \@xp\@firstoftwo\else\@xp\@secondoftwo\fi
       {\mermin@real@in@display{#1}}%
       {\mermin@real@in@display{{#1}}}}}
\def\mermin@cref@entry@opt[#1]#2\mermin@nil{%
  \mermin@real@in@display{[#1]{#2}}}
```

`\ltx@label` `\label@in@display`

hyperref and cleveref install their own `\ltx@label` and `\label@in@display` in `\AtBeginDocument` code. The package wraps them in the `begindocument/end` hook, which runs after all such code, so that load order does not matter. The new `\ltx@label` writes the primary label as before and then, if the queue belongs to it, every queued key except the primary one, which an `\aka{A}\aka{K}\label{K}` line would otherwise write twice. The new `\label@in@display` sends `\alsolabel` and `\label` their separate ways.

```
 % begindocument/end runs after hyperref's and cleveref's
 % \AtBeginDocument code, which define their own versions of both.
\AddToHook{begindocument/end}{%
  \let\mermin@ltx@label\ltx@label
```

```
\def\ltx@label#1{%
  \mermin@ltx@label{#1}%
  \def\mermin@tmp{#1}%
  \ifx\mermin@tmp\mermin@tag
    \begingroup
       % Skip a queued key equal to the label just written, as on an
       % \aka{A}\aka{K}\label{K} line.
      \def\do##1{\def\mermin@tmp{##1}%
        \ifx\mermin@tmp\mermin@tag\else\mermin@ltx@label{##1}\fi}%
      \mermin@queue
    \endgroup
    \gdef\mermin@queue{}\gdef\mermin@tag{}\gdef\mermin@promoted{}%
  \fi}%
\let\mermin@orig@label@in@display\label@in@display
 % cleveref's multline trial pass calls its inner hook directly, not
 % \label@in@display; give it the same treatment, which \mermin@store
 % undoes locally.
\ifdefined\cref@old@label@in@display
  \let\mermin@cref@old\cref@old@label@in@display
  \def\cref@old@label@in@display#1{\mermin@cref@entry{#1}}%
\fi
\def\label@in@display{%
  \ifmermin@also
    \expandafter\mermin@also@in@display
  \else
    \expandafter\mermin@label@in@display
  \fi}}
```

5.7 Documents read through xr

`\mermin@xr@test` `\mermin@xr@import`

xr and xr-hyper import another document's labels, as for a paper and its supplement, by reading that document's `.aux` file a line at a time with `\XR@test`, which keeps only `\newlabel` and `\bibcite` lines. A wrapper also imports the records, with the external document's prefix added to each key. For every equation key it imports, it stores the text that `\eqcite` prints when the key has no phrase, the key without the prefix (and without `eq:`), in `\mermin@xrdef@`⟨*prefixed key*⟩; and for a key `eq:`⟨*rest*⟩ it defines the `eq:` fallback `\mermin@xreq@`⟨*prefix*⟩⟨*rest*⟩, which `\eqcite`'s resolver calls. A document that does not use the package records no equations; its file is not signed, and all its labels are imported as equations rather than warned about. The wrapper is installed whether xr or xr-hyper is loaded before or after this package, and again if both are; an `\externaldocument` called before the package is loaded is beyond its reach.

```
\long\def\mermin@xr@test#1#2#3#4\XR@{%
  % xr-hyper's \protected@xdef expands the titles stored in the other
  % file here, in the preamble, before the short names exist; a title
  % may hold \eq.  (\aka never reaches the file: nameref removes it.)
\ifdefined\eq\else\let\eq\eqcite\fi
\ifx#1\mermin@signature
  \def\mermin@xr@signed{}%
\else\ifx#1\mermin@equation
  \mermin@xr@import{#2}%
\else\ifx#1\mermin@newphrase
  \@xp\gdef\csname mermin@ph@\XR@prefix#2\endcsname{#3}%
```

```
  \else\ifx#1\newlabel
     % An unsigned file, from a document without this package, has no
     % marks: treat all its labels as equations rather than warn about
     % each.
    \ifdefined\mermin@xr@signed\else\mermin@xr@import{#2}\fi
  \fi\fi\fi\fi
  \mermin@xr@orig#1{#2}{#3}#4\XR@}
\def\mermin@xr@import#1{%
  \@xp\gdef\csname mermin@eq@\XR@prefix#1\endcsname{}%
  \mermin@ifeqprefix{#1}%
    {\@xp\mermin@xr@eqkey\@xp{\@gobblethree#1}}%
    {\@xp\xdef\csname mermin@xrdef@\XR@prefix#1\endcsname
       {\unexpanded{#1}}}}
 % For a key eq:<rest>: the text is <rest>, and
 % \mermin@xreq@<prefix><rest> takes the resolver's continuation and
 % passes it <prefix>eq:<rest>.
\def\mermin@xr@eqkey#1{%
  \@xp\xdef\csname mermin@xrdef@\XR@prefix eq:#1\endcsname
    {\unexpanded{#1}}%
  \@xp\xdef\csname mermin@xreq@\XR@prefix#1\endcsname##1{%
    ##1{\XR@prefix\unexpanded{eq:#1}}}}
\def\mermin@xr@patch{%
   % Wrap whatever \XR@test is now, unless it is already wrapped: xr
   % and then xr-hyper each define their own.
  \ifx\XR@test\mermin@xr@test\else
    \let\mermin@xr@orig\XR@test
    \let\XR@test\mermin@xr@test
  \fi}
\AddToHook{package/xr/after}{\mermin@xr@patch}
\AddToHook{package/xr-hyper/after}{\mermin@xr@patch}
\@ifpackageloaded{xr}{\mermin@xr@patch}{}
\@ifpackageloaded{xr-hyper}{\mermin@xr@patch}{}
```

5.8 Citing with `\eqcite`

`\mermin@resolve` `\mermin@resolve{`⟨*argument*⟩`}{`⟨*continuation*⟩`}` finds the key ⟨*argument*⟩ cites: the argument itself if it is a label, else `eq:`⟨*argument*⟩, else the `eq:` key of another document that `xr` imported, else the argument, which is then undefined. It expands to ⟨*continuation*⟩`{`⟨*key*⟩`}`. It is expandable, so that the same code serves bookmarks, and passes the key as an argument rather than storing it, so that nothing depends on a macro that a group might end: `ulem` typesets each word in a group of its own. `\@ifundefined` treats a label made `\relax` by an earlier failed `\ref` as undefined.

One case comes first: when the argument labels something that is not an equation, a section say, and `eq:`⟨*argument*⟩ labels an equation, `\eqcite` cites the equation. `\mermin@ifpreferprefix` tests for it, and only in a document whose `.aux` file is signed, since only then are the equations known.

```
\def\mermin@ifpreferprefix#1{%
   % True if #1 labels a non-equation and an eq: equation exists: here
   % (both its label and its mark), or in a document read through xr.
   % Only in a signed document, where non-equations are known.
  \ifdefined\mermin@signed\@xp\@firstoftwo\else\@xp\@secondoftwo\fi
    {\@ifundefined{r@#1}{\@secondoftwo}%
```

```
      {\@ifundefined{mermin@eq@#1}%
         {\@ifundefined{mermin@xreq@#1}%
            {\@ifundefined{r@eq:#1}{\@secondoftwo}%
               {\@ifundefined{mermin@eq@eq:#1}%
                  {\@secondoftwo}{\@firstoftwo}}}%
            {\@firstoftwo}}%
         {\@secondoftwo}}}%
    {\@secondoftwo}}
\def\mermin@resolve#1#2{%
  \mermin@ifpreferprefix{#1}%
    {\@ifundefined{r@eq:#1}%
       {\csname mermin@xreq@#1\endcsname{#2}}% through xr
       {#2{eq:#1}}}%
    {\mermin@resolve@{#1}{#2}}}
\def\mermin@resolve@#1#2{%
  \@ifundefined{r@#1}%
    {\@ifundefined{r@eq:#1}%
       {\@ifundefined{mermin@xreq@#1}%
          {#2{#1}}%
          {\csname mermin@xreq@#1\endcsname{#2}}}%
       {#2{eq:#1}}}%
    {#2{#1}}}
```

`\eqcite` `\mermin@cite` `\mermin@check@ambiguous`

`\eqcite` resolves its argument. `\mermin@cite` receives the argument and the key. It warns if the key is a label but not an equation, and, through `\mermin@check@ambiguous`, if the argument and `eq:`⟨*argument*⟩ are both labels of different equations, as when a revision adds `\aka{mass}` to one equation while an older `\eq{mass}` meant `\label{eq:mass}` on another. It then prints the phrase, a tie, and the number from `\eqref`, which hyperref makes a link. In math mode the whole citation is set as text.

```
\NewDocumentCommand\eqcite{m}{%
  % Alignment guard, as in \mermin@markargs: \eq{A & B} in a cell of
  % align or tabular builds csnames from the key.
  \iffalse{\fi
  \mermin@resolve{#1}{\mermin@cite{#1}}%
  \iffalse}\fi}
\def\mermin@cite#1#2{%
  \mermin@ifpreferprefix{#1}{}{\mermin@check@ambiguous{#1}}%
  \@ifundefined{r@#2}{}{%
    \ifdefined\mermin@signed
      \@ifundefined{mermin@eq@#2}%
        {\mermin@warn{%
           `#1' is not an equation\MessageBreak
           label, yet \mermin@eqcs\space prints a parenthesized%
           \MessageBreak
           equation number; for a figure, section, or\MessageBreak
           theorem, use \string\ref\space or \string\cref}}%
        {}%
    \fi}%
  \ifmmode\@xp\text\else\@xp\@firstofone\fi
    {\mermin@phrase{#1}{#2}~\mermin@keepflag{\mermin@eqref{#2}}}}
\def\mermin@check@ambiguous#1{%
  \@ifundefined{r@#1}{}{\@ifundefined{r@eq:#1}{}{%
    % \ifx compares the two labels' number and page: keys of one line
```

```
     % agree.
    \@xp\ifx\csname r@#1\@xp\endcsname\csname r@eq:#1\endcsname
    \else
      \mermin@warn{%
        '#1' and 'eq:#1' label different\MessageBreak
        equations;\MessageBreak
        \mermin@eqcs\space cites '#1'}%
    \fi}}
```

`\mermin@phrase` `\mermin@warn@unprintable` The phrase is the typeset phrase recorded for the key, or else the key's text: for a key imported through xr, the key without the prefix, and otherwise the argument without a leading `eq:`. The key's text is printed verbatim if it is not printable text. Once the label exists, a verbatim key also warns, saying where to give it a phrase; before the first run defines it, every key is quietly verbatim if need be, because its phrase may simply not be known yet. `\mermin@keytext{`⟨*argument*⟩`}{`⟨*key*⟩`}{`⟨*continuation*⟩`}` chooses the key's text and passes it to ⟨*continuation*⟩ as an argument, unexpanded; bookmarks use it too. `\eqcite`'s continuation is `\def\mermin@text`: expanding the text instead would fail on accent commands such as `\"` in a key, and would expand macros that should print as written.

```
\def\mermin@keytext#1#2#3{%
  \@ifundefined{mermin@xrdef@#2}%
    {\mermin@ifeqprefix{#1}%
       {\@xp\mermin@apply\@xp{\@gobblethree#1}{#3}}%
       {#3{#1}}}%
     % Expand the \csname, then the macro it names, once each.
    {\@xp\@xp\@xp\mermin@apply
       \@xp\@xp\@xp{\csname mermin@xrdef@#2\endcsname}{#3}}}
\def\mermin@apply#1#2{#2{#1}}
\def\mermin@phrase#1#2{%
  \@ifundefined{mermin@ph@#2}%
    {\mermin@keytext{#1}{#2}{\def\mermin@text}%
     \@xp\mermin@ifunprintable\@xp{\mermin@text}%
       {\@ifundefined{r@#2}{}%
          {\@xp\mermin@warn@unprintable\@xp{\mermin@text}{#1}{#2}}%
        \@xp\mermin@verbatim\@xp{\mermin@text}}%
       {\mermin@text}}%
    {\csname mermin@ph@#2\endcsname}}
 % #1 is the text printed, #2 the argument, #3 the key.
\def\mermin@warn@unprintable#1#2#3{%
  \@ifundefined{mermin@xrdef@#3}%
    {\mermin@warn{'#2' has no typeset phrase\MessageBreak
       and is not text, so it is printed verbatim;\MessageBreak
       where it is defined, write\MessageBreak
       \mermin@akacs[<phrase>]{#3}}}%
    {\mermin@warn{'#2' has no typeset phrase\MessageBreak
       and is not text, so it is printed verbatim;\MessageBreak
       give it a phrase in the document\MessageBreak
       that defines '#1'}}}
```

5.9 Headings, bookmarks, and running heads

\mermin@pdf@eq hyperref's bookmarks need an expandable \eqcite, which gives the phrase or key text and the number as plain text. It uses the same resolver, and the same choice of text as \mermin@phrase, but expandably and without the verbatim form.

```
\def\mermin@pdf@eq#1{\mermin@resolve{#1}{\mermin@pdf@cite{#1}}}
\def\mermin@pdf@cite#1#2{%
  \@ifundefined{mermin@ph@#2}%
    {\mermin@keytext{#1}{#2}\@firstofone}%
    {\csname mermin@ph@#2\endcsname}%
  \space(\ref{#2})}
```

\mermin@stripaka nameref takes the name of a section or a caption from its title, through gettitlestring. That package's default method, which does not expand the title, removes \label from it but knows nothing of \alsolabel, which would stay in the name and define its key again wherever the name is typeset. The package wraps the method to remove \alsolabel too, with its arguments: spaces around it go with it at the end of the title, but in the middle they stay, so that `Chiral \aka{chiral}fermions` keeps its space. \aka is removed only when it is this package's. (The expanding method needs nothing: \alsolabel expands to nothing there, since that method disables \label.)

```
\ExplSyntaxOn
 % The pieces of one command: its name, an optional <overlay>, an
 % optional [phrase], and the {key}.  A phrase or key may hold brace
 % groups one level deep, as in [\emph{x}] or {a{b}}; a lazy .*? would
 % instead run on to the title's last closing brace.
\tl_const:Nn \c__mermin_overlay_tl { (?: \< \c[^BE][^\>]* \> )? }
\tl_const:Nn \c__mermin_phrase_tl
  { (?: \[ (?: \c[^BE][^\]] | \cB. \c[^BE]* \cE. )* \] )? }
\tl_const:Nn \c__mermin_key_tl
  { \cB. (?: \c[^BE]. | \cB. \c[^BE]* \cE. )* \cE. }
\cs_new:Npn \__mermin_aka_one:n #1
  {
    \exp_not:n { \c{(?:#1)} }
    \exp_not:V \c__mermin_overlay_tl
    \exp_not:V \c__mermin_phrase_tl
    \exp_not:V \c__mermin_key_tl
  }
 % A run of the commands at the end of the title goes with the spaces
 % around it; one anywhere else goes alone, and its spaces stay.
\cs_new:Npn \__mermin_aka_pattern:n #1
  {
    \exp_not:n { \s* (?: } \__mermin_aka_one:n {#1}
    \exp_not:n { \s* )+ \Z | } \__mermin_aka_one:n {#1}
  }
\exp_args:NNx \regex_const:Nn \c__mermin_aka_regex
  { \__mermin_aka_pattern:n { aka|alsolabel|mermin@alsolabel } }
\exp_args:NNx \regex_const:Nn \c__mermin_long_regex
  { \__mermin_aka_pattern:n { alsolabel|mermin@alsolabel } }
\cs_new_protected:Npn \mermin@stripaka
  {
    \ifx \aka \alsolabel
      \regex_replace_all:NnN \c__mermin_aka_regex { }
```

```
        \GetTitleStringResult
    \else
      \regex_replace_all:NnN \c__mermin_long_regex { }
        \GetTitleStringResult
    \fi
  }
\ExplSyntaxOff
\AddToHook{begindocument/end}{%
   % nameref, through gettitlestring, takes section and caption names
   % from their titles; this is its default, non-expanding method.
  \ifdefined\GetTitleStringNonExpand
    \let\mermin@gts@orig\GetTitleStringNonExpand
    \def\GetTitleStringNonExpand#1{%
      \mermin@gts@orig{#1}\mermin@stripaka}%
  \fi}
```

In text that `\MakeUppercase` changes, such as running heads, the argument of `\eqcite` is a key and must not change; nor may the key of `\mermin@alsolabel`. The LaTeX kernel from 2022-06 keeps the argument of every command in `\l_text_case_exclude_arg_tl` as written.

```
% l_text_case_exclude_arg_tl exists in LaTeX 2022-06 and later.
\AddToHook{begindocument/end}{%
  \@ifundefined{l_text_case_exclude_arg_tl}{}{%
    \@xp\g@addto@macro\csname l_text_case_exclude_arg_tl\endcsname
      {\eqcite\mermin@alsolabel}%
    \ifx\eq\eqcite
      \@xp\g@addto@macro\csname l_text_case_exclude_arg_tl\endcsname
        {\eq}%
    \fi}}
```

5.10 The short names

`\eq` `\aka` The short names `\eq` and `\aka` may already be taken, `\eq` for instance by nccmath. Each is therefore assigned only at `\begin{document}`, after every package and the preamble have had the chance to define it, and only if it is still free; otherwise the existing command is kept and the user is told to write the long name. Each short name is a copy of the long one made with `\let`, so it behaves like it in moving arguments, and `\ifx\aka\alsolabel` tells whether `\aka` is this package's. The warnings then name the commands the document can use. The bookmark versions of the commands are installed at the same time.

```
\def\mermin@akacs{\string\alsolabel}
\def\mermin@eqcs{\string\eqcite}
\def\mermin@shortname#1#2{%
  \@ifundefined{#1}%
    {\@xp\let\csname #1\endcsname#2}%
    {\mermin@warn@noline{\@backslashchar#1
      was already defined, by a package or\MessageBreak
      the preamble, and is left alone;\MessageBreak
      write \string#2\space instead}}}
% env/document/begin, not begindocument/before or later: beamer builds
% the PDF title, which may hold \eq, in this same hook (ordered after
% this code by the rule below), and its themes typeset section titles
% in \AtBeginDocument code.
```

```
\AddToHook{env/document/begin}[mermin-equations]{%
  \mermin@shortname{eq}\eqcite
  \mermin@shortname{aka}\alsolabel
  \ifx\aka\alsolabel\def\mermin@akacs{\string\aka}\fi
  \ifx\eq\eqcite\def\mermin@eqcs{\string\eq}\fi
  \ifdefined\pdfstringdefDisableCommands
    \pdfstringdefDisableCommands{%
       % Test the short names before the long ones change.
      \ifx\aka\alsolabel\let\aka\mermin@gobble@aka\fi
      \ifx\eq\eqcite\let\eq\mermin@pdf@eq\fi
      \let\alsolabel\mermin@gobble@aka
      \let\eqcite\mermin@pdf@eq
       % latexdiff's markers, which this package makes robust.
      \let\DIFdelbegin\@empty
      \let\DIFdelend\@empty
       % Neither of Mermin's rules is for bookmarks.
      \let\eqref\mermin@eqref
      \ifdefined\mermin@maketag\let\maketag@@@\mermin@maketag\fi}%
  \fi}
\DeclareHookRule{env/document/begin}
  {mermin-equations}{before}{beamer}
```

5.11 Mermin's rules as warnings

`\ifmermin@numbered` `\mermin@fisher@install` Fisher's rule: number every displayed equation. A display counts as numbered when it typesets at least one number, automatic or given by `\tag`; a derivation whose intermediate lines are `\notag` is one equation, and is numbered. Every number that amsmath typesets passes through `\maketag@@@`, and every one the kernel's eqnarray typesets through `\@eqnnum`; both raise a flag, which each display lowers when it begins and reads when it has ended. The flag is global, because the number is typeset inside the cells of an alignment, which are groups. A citation by `\eqref` inside a display typesets a number through `\maketag@@@` too, so the citation commands keep the flag as they found it, with `\mermin@keepflag`. empheq typesets its inner display twice, the second time without `\maketag@@@`, so it is checked as a whole instead. A display that latexdiff marks as deleted is not checked, and on beamer's later overlay slides of a frame neither rule warns again. With the mathtools option `showonlyrefs` a display is numbered only if it is cited, the practice Mermin calls Occam's rule; then one warning says so, and displays are not checked one by one.

```
\newif\ifmermin@numbered
\def\mermin@fisher@begin{\global\mermin@numberedfalse}
\def\mermin@fisher@after#1{%
  \ifmermin@numbered\else\ifmermin@deleted\else
  \ifnum\mermin@slide>\@ne\else
    \mermin@warn{%
      `#1' display without a number.\MessageBreak
      Fisher's rule: number every display\MessageBreak
      (option nofisher)}%
  \fi\fi\fi}
% beamer typesets a frame once per overlay slide; warn on the first.
% A number, tested with a real \ifnum: a macro expanding to \ifnum
% would not be counted as a conditional when TeX skips a branch.
```

```
\def\mermin@slide{%
  \ifdefined\beamer@slideinframe\beamer@slideinframe\else1\fi}
\def\mermin@fisherenv#1{%
  \AddToHook{env/#1/begin}{\mermin@fisher@begin}%
  \AddToHook{env/#1/after}{\mermin@fisher@after{#1}}}
 % A macro of its own: written inline, a false \ifmermin@fisher would
 % take the showonlyrefs test's \else and \fi as its own.
\def\mermin@fisher@install{%
     % mathtools' showonlyrefs: \MH_if_boolean_show_only_refs: is
     % \iftrue or \iffalse.
    \@ifundefined{MH_if_boolean_show_only_refs:}{\@secondoftwo}%
      {\csname MH_if_boolean_show_only_refs:\endcsname
         \@xp\@firstoftwo\else\@xp\@secondoftwo\fi}%
      {\mermin@warn@noline{%
         mathtools' showonlyrefs numbers only the\MessageBreak
         equations that are cited, which Mermin calls\MessageBreak
         Occam's rule; Fisher's rule numbers them all\MessageBreak
         (option nofisher silences this)}}%
      {\let\mermin@maketag\maketag@@@
       \def\maketag@@@{\global\mermin@numberedtrue\mermin@maketag}%
       \let\mermin@eqnnum\@eqnnum
       \def\@eqnnum{\global\mermin@numberedtrue\mermin@eqnnum}%
        % \[ is equation* under amsmath, and so is displaymath, which
        % would otherwise be checked twice.
       \@for\mermin@env:=%
           equation,equation*,align,align*,gather,gather*,%
           multline,multline*,flalign,flalign*,alignat,alignat*,%
           eqnarray,eqnarray*%
         \do{\@xp\mermin@fisherenv\@xp{\mermin@env}}%
        % empheq typesets its inner environment twice, the second time
        % with \maketag@@@ replaced by a \vphantom: check empheq as a
        % whole, and silence the inner environment's own check.
       \AddToHook{env/empheq/begin}{%
         \mermin@fisher@begin
         \let\mermin@fisher@begin\relax
         \let\mermin@fisher@after\@gobble}%
       \AddToHook{env/empheq/after}{\mermin@fisher@after{empheq}}}}
\AddToHook{begindocument/end}{%
  \ifmermin@fisher\@xp\mermin@fisher@install\fi}
```

`\mermin@eqref` `\mermin@samaritan` The Good Samaritan rule: when referring to an equation, identify it by a phrase as well as a number. The warning is given once per citation: not again in amsmath's measuring pass, in caption's trial typesetting, on a later beamer slide, or when the table of contents or list of figures is read back. `\eqref`, and cleveref's `\cref` and `\Cref`, warn when they cite an equation by number alone; `\eqcite` calls the original `\eqref`, saved as `\mermin@eqref`, and so never warns. Only equations are checked, and only in a signed document, where they are known. Plain `\ref` is not checked: other commands, `\eqref` among them, use it internally.

```
 % A first copy now, for beamer, which typesets its title, perhaps with
 % \eq, before begindocument/end; the copy is refreshed there.
\NewCommandCopy\mermin@eqref\eqref
 % An \eqref in a display typesets its number through \maketag@@@ too;
 % \mermin@keepflag runs #1 without letting it count as the display's
 % own number.
```

```
\def\mermin@keepflag#1{%
  \ifmermin@numbered\@xp\@firstofone\else\@xp\mermin@keepflag@\fi{#1}}
\def\mermin@keepflag@#1{#1\global\mermin@numberedfalse}
\def\mermin@samaritan#1{%
   % Warn once: not in amsmath's measuring pass, nor caption's
   % single-line check (\label is \caption@gobble there), nor on a later
   % beamer slide; the table of contents is handled below.
  \ifx\label\caption@gobble\else\ifmeasuring@\else
  \ifnum\mermin@slide>\@ne\else
  \ifdefined\mermin@signed
    \@ifundefined{r@#1}{}{\@ifundefined{mermin@eq@#1}{}{%
      \mermin@warn{%
        `#1' is cited by number alone.\MessageBreak
        Good Samaritan rule: name it, as \mermin@eqcs\space does%
        \MessageBreak
        (option nosamaritan)}}}%
  \fi\fi\fi\fi}
\AddToHook{begindocument/end}{%
   % A command copy, not \let, since \eqref is robust.  mathtools'
   % showonlyrefs, or the document, may have redefined \eqref by now;
   % the copy keeps their version.
  \DeclareCommandCopy\mermin@eqref\eqref
  \ifmermin@samaritan
     % \protected, not \DeclareRobustCommand: that would redefine the
     % inner command `\eqref ', which a \let copy of \eqref made earlier
     % (\let\oldeqref\eqref, or refcheck's) still calls, and the two
     % would call each other forever.  Bookmarks use the original.
    \protected\def\eqref#1{%
      \mermin@samaritan{#1}\mermin@keepflag{\mermin@eqref{#1}}}%
     % cleveref's \cref, \Cref, and \labelcref, starred or not, pass
     % their list of keys to \@cref, with their own name as #1; so do
     % its page references (\cpageref, and varioref's \vref), which
     % cite a page, not an equation, and are not checked.
    \ifdefined\@cref
      \let\mermin@cref\@cref
      \def\@cref#1#2{%
        \in@{,#1,}{,cref,Cref,labelcref,}%
        \ifin@
          \@for\mermin@k:=#2\do{\@xp\mermin@samaritan\@xp{\mermin@k}}%
        \fi
        \mermin@cref{#1}{#2}}%
    \fi
     % Reading back the table of contents or list of figures would warn
     % again for each citation in a title or caption.
    \let\mermin@starttoc\@starttoc
    \def\@starttoc#1{%
      {\let\mermin@samaritan\@gobble\mermin@starttoc{#1}}}%
  \fi}
⟨/package⟩
```

Change History

v0.1

General: Initial version, providing `\aka` (`\alsolabel`) and `\eq` (`\eqcite`). 1

v0.1.1

General: Renamed from mermin to mermin-equations at CTAN's request; the commands are unchanged. 1

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\< 473
\> 473
\@auxout 50, 64
\@backslashchar 34, 529
\@bsphack 133, 134
\@charlb 288, 293
\@cref 646, 649, 650, 651
\@eqnnum 591, 592
\@esphack . 131, 133, 156
\@firstofone 420, 465, 617
\@firstoftwo 29, 42, 291, 294, 377, 383, 384, 583
\@for 187, 595, 654
\@gobble 111, 113, 606, 662
\@gobble@om . . . 115, 116
\@gobble@with@sphack@om 115, 117, 118
\@gobblethree . 351, 436
\@gobbletwo 73, 75
\@ifclassloaded 5
\@ifpackageloaded 6, 371, 372
\@ifundefined . . 90, 230, 378, 379, 380, 381, 382, 389, 394, 395, 396, 409, 411, 423, 434, 443, 446, 453, 464, 517, 527, 581, 626
\@namedef 76, 231
\@ne 562, 624
\@newl@bel 73
\@partaux 67
\@percentchar 34
\@secondoftwo 31, 42, 291, 294, 377, 378, 381, 383, 385, 386, 581, 583
\@setckpt 72, 75
\@starttoc . . . 660, 661
\@tempswatrue . . . 83, 90
\@tfor 38
\@unexpandable@protect 49
\[. 475, 593
\] 475
_ 478, 487, 489, 490, 493, 495

A

\aka *1*, 202, 264, 309, 333, 498, 524
\alsolabel *4*, 94, 498, 524, 539, 540, 545, 547
\AtBeginDocument 64, 300, 536

B

\beamer@slideinframe 572

C

\c . . . 473, 474, 475, 476, 477, 480, 481, 482, 483, 492, 494, 499, 502
\caption@gobble 121, 621, 623
\cB 475, 477
\cE 475, 477
\cpageref 647
\Cref 645
\cref 417, 645
\cref@old@label@in@display 141, 207, 320, 321, 322
\cs 478, 487, 496
\csname 7, 8, 69, 82, 85, 340, 349, 352, 358, 360, 390, 398, 426, 438, 440, 450, 466, 518, 521, 528, 582

D

\DeclareCommandCopy 637
\DeclareHookRule . . 556
\DeclareOption . . 12, 13
\DeclareRobustCommand 639
\detokenize . . 20, 26, 34, 36, 43, 80, 82
\df@label 201, 202, 213, 214, 216, 217, 219, 220, 222, 224, 231, 250, 251, 252, 253, 254, 263, 266, 267, 268, 270, 271
\DIFdelbegin 101, 102, 104, 550
\DIFdelend 103, 106, 551

E

\edef . . 33, 36, 39, 80, 81
\emph 471
\endcsname . 7, 8, 69, 82, 85, 340, 349, 352, 358, 360, 390, 398, 426, 440, 450, 466, 518, 521, 528, 582
\enddocument 89
\eq *1*, 333, 334, 402, 520, 522, 524, 611
\eqcite . *4*, 334, 401, 519, 520, 525, 538, 541, 546, 548
\eqref 553, 612, 613, 634, 635, 637, 640, 641, 643
\equation 3
\exp 480, 481, 482, 483, 489, 490, 492, 494
\ExplSyntaxOff 506
\ExplSyntaxOn 468

G

\g@addto@macro 86, 518, 521
\GetTitleStringNonExpand 510, 511, 512
\GetTitleStringResult 500, 503

I

\if 28, 291, 293

\if@filesw 45, 64, 67, 93
\iffalse 158, 161, 164, 226, 232, 404, 406, 580
\ifin@ 41, 653
\ifmeasuring@ 623
\ifmermin@also 127, 325
\ifmermin@deleted 94, 110, 561
\ifmermin@fisher 10, 576, 609
\ifmermin@numbered 558, 617
\ifmermin@samaritan 11, 638
\ifmmode 420
\ifnum . 28, 562, 569, 624
\iftrue 580
\IfValueT 135
\IfValueTF 143
\immediate . . 50, 57, 63
\in@ 39, 652
\includeonly 65

L

\label@in@display 139, 299
\labelcref 645
\long 330
\ltx@label 299

M

\maketag@@@ . . . 554, 589, 590, 601, 613
\mermin@aka@use 110, 113, 114, 116, 118, 121, 123, 125
\mermin@akacs 457, 524, 540
\mermin@also@in@display 244, 326
\mermin@also@stored 141, 244
\mermin@alsolabel 122, 124, 125, 127, 519
\mermin@alsotrue . . 137
\mermin@apply 436, 439, 441
\mermin@auxheader . 56
\mermin@bad . . 37, 41, 42
\mermin@c 38, 39
\mermin@canonical 203, 245
\mermin@check@ambiguous 401
\mermin@checkphrases 70
\mermin@cite 401
\mermin@cref . . 650, 656
\mermin@cref@entry 139, 278, 322
\mermin@cref@entry@opt 292, 297
\mermin@cref@old . . . 140, 206, 207, 321
\mermin@deletedfalse 107
\mermin@deletedtrue 105
\mermin@DIFdelbegin 102, 105
\mermin@DIFdelend 103, 107
\mermin@doclabel 129, 147
\mermin@enqueue 221, 258, 271
\mermin@env 187, 194, 595, 599
\mermin@eqcs . . 414, 431, 525, 541, 629
\mermin@eqnnum 591, 592
\mermin@eqref 421, 553, 610
\mermin@equation 55, 56, 337
\mermin@firstchar 17, 291, 293
\mermin@firstchar@ 20, 23
\mermin@fisher@after . 560, 575, 606, 607
\mermin@fisher@begin . 559, 574, 604, 605
\mermin@fisher@install 558
\mermin@fisherenv 573, 599
\mermin@fisherfalse 12
\mermin@fishertrue . 10
\mermin@form . . 219, 229, 236, 239, 246, 257, 266, 271
\mermin@found 234, 236, 238
\mermin@gobble@aka 110, 113, 114, 116, 121, 126, 545, 547
\mermin@gts@orig 511, 513
\mermin@ifeqprefix 17, 350, 435
\mermin@ifeqprefix@ 26, 27
\mermin@ifpreferprefix 373, 388, 408
\mermin@ifunprintable 33, 445
\mermin@k 654
\mermin@keepflag . . . 421, 614, 616, 644
\mermin@keepflag@ 617, 618
\mermin@keyof . 157, 229
\mermin@keyof@opt 166, 168
\mermin@keyof@plain 166, 167
\mermin@keytext 433, 444, 465
\mermin@label@in@display 278, 328
\mermin@label@in@display@opt 280, 284
\mermin@label@in@display@ov 279, 282
\mermin@label@in@display@plain 281, 286
\mermin@ltx@label 302, 304, 311
\mermin@maketag 554, 589, 590
\mermin@markargs 157, 211, 226, 248, 402
\mermin@markenv 186, 194
\mermin@markeq 44, 163, 181, 183, 185, 199
\mermin@marklabel 145, 149, 169
\mermin@marklabel@opt 179, 182
\mermin@marklabel@orig . 172, 181, 183, 185
\mermin@marklabel@ov 178, 184
\mermin@marklabel@plain 179, 180
\mermin@markstore . . 203, 262, 269, 276
\mermin@newphrase . . . 54, 58, 70, 339

\mermin@nil 20, 21, 23, 26, 27, 162, 167, 168, 229, 292, 297
\mermin@numberedfalse 559, 618
\mermin@numberedtrue 590, 592
\mermin@orig@label@in@display 209, 316
\mermin@pdf@cite 462, 463
\mermin@pdf@eq 462, 546, 548
\mermin@phkeys 70, 86, 91
\mermin@phrase 421, 433
\mermin@promoted . . . 200, 253, 261, 263, 272, 275, 314
\mermin@queue 200, 223, 237, 240, 241, 255, 267, 312, 314
\mermin@rawkey 163, 167, 168, 230
\mermin@real@in@display 260, 285, 286, 295, 296, 298
\mermin@record 44
\mermin@resolve 373, 405, 462
\mermin@resolve@ 392, 393
\mermin@samaritan . 610
\mermin@samaritanfalse 13
\mermin@samaritantrue 11
\mermin@saved . 216, 220
\mermin@shortname 526, 538, 539
\mermin@signature 56, 335
\mermin@signed 68, 377, 410, 625
\mermin@slabel@orig 198, 199
\mermin@slide 562, 571, 624
\mermin@specials 33, 38
\mermin@starttoc 660, 662
\mermin@store . 203, 318
\mermin@stripaka . . 468
\mermin@tag 200, 222, 224, 254, 267, 270, 306, 311, 314
\mermin@text . . 444, 445, 447, 448, 449
\mermin@tmp 36, 39, 235, 236, 249, 250, 305, 306, 310, 311
\mermin@verbatim 33, 448
\mermin@warn 15, 412, 428, 454, 458, 563, 627
\mermin@warn@noline 15, 529, 584
\mermin@warn@unprintable 433
\mermin@wraplabel . 169
\mermin@writephrase 44, 135
\mermin@xr@eqkey 351, 357
\mermin@xr@import . 330
\mermin@xr@orig 347, 366
\mermin@xr@patch 362, 369, 370, 371, 372
\mermin@xr@signed 336, 345
\mermin@xr@test . . . 330
\mermin@xreq@ 355
\MessageBreak . 413, 415, 416, 429, 430, 454, 455, 456, 458, 459, 460, 530, 531, 564, 565, 585, 586, 587, 628, 630
\MH 579
\MT 228

N

\NewCommandCopy . . . 612
\NewDocumentCommand 130, 401
\NewExpandableDocumentCommand 126
\newlabel 195, 341
\normalfont 43

O

\oldeqref 641

P

\PackageWarning . . . 15
\PackageWarningNoLine 16
\pdfstringdefDisableCommands 542, 543
\ProcessOptions . . . 14
\protect 49
\protected 97, 104, 106, 639, 643
\protected@xdef . . . 331
\providecommand 58, 59, 60, 66

R

\ref 417, 467
\regex 492, 494, 499, 502
\relax 7, 8, 14, 20, 21, 24, 26, 111, 114, 605
\reserved@a 39, 40, 80, 83
\reserved@b 81, 83

S

\s 489, 490
\slabel . . . 195, 198, 199
\space 414, 417, 431, 467, 532, 629
\string 54, 55, 58, 59, 60, 63, 417, 524, 525, 532, 540, 541

T

\text 420
\tl 473, 474, 476
\ttfamily 43

U

\usepackage 97

V

\vphantom 601
\vref 647

W

\write 50, 57, 63

X

\XR@ 330, 347
\XR@prefix 340, 349, 352, 358, 360, 361
\XR@test 363, 365, 366, 367

Z

\Z 490