

The phfthm package¹

Philippe Faist philippe.faist@bluewin.ch

March 01, 2021

¹This document corresponds to `phfthm v1.2`, dated 2021/03/01. It is part of the `phfqitlx` package suite, see <https://github.com/phfaist/phfqitlx>.

phfthm—Goodies for theorems and proofs.

1	Introduction	2
1.1	Theorem environments	2
1.2	Proof environments	2
1.3	Theorem-heading environments	3
2	Quick start and package options	3
2.1	Predefined theorem environments	3
2.2	The proof environment	6
2.3	The theorem-heading environment	7
3	Theorem environments	7
3.1	Define theorem environments manually	7
3.2	Loading theorem sets manually	9
3.3	Theorem hooks	10
4	Proof environments	11
4.1	Manually define a proof environment	12
4.2	Proof hooks	14
5	Pairing theorems to proofs and proof-reference mechanism	15
5.1	On the theorem side	16
5.2	On the proof side	17
5.3	Customizing appearance of the proof reference text	18
6	Theorem-heading definition-like environments	19
6.1	Define theorem-heading environments manually	19
6.2	Available hooks for theorem-heading environments	19
7	Implementation	20
7.1	Generic Internal Stuff	20
7.2	Definitions for theorem environments	20
7.2.1	<code>\phfMakeTheorem</code> : definition of a new theorem environment	20
7.2.2	Default hooks for theorems	23
7.2.3	Proof-ref mechanism (on the theorem side)	24
7.3	Definitions for proof environments	25
7.3.1	Define a proof environment: <code>\phfMakeProofEnv</code>	26
7.3.2	Common hooks for proofs	30

7.3.3	Default display environment for proofs	30
7.3.4	Parsing the proof argument	31
7.4	Implementation of the proof-ref machinery	32
7.4.1	Small general stuff	32
7.4.2	Utilities for interacting with \autoref labels	33
7.4.3	Default proof-ref style, with basic machinery	35
7.4.4	Other proof-ref styles: margin and marginbottom	37
7.5	Thmheading definition-like environments	39
7.5.1	Manually define a thmheading environment	39
7.6	Theorem sets	41
7.7	Package option handling	43
7.7.1	Declaring the package options	43
7.7.2	Parsing the package options	46
7.7.3	Execute package options-controlled actions	46
	Change History	50
	Index	50

■ 1 Introduction

The phfthm package provides enhanced theorem and proof environments, based on the amsthm original versions. It allows for hooks to be placed, adds some default goodies and is highly customizable.

There are three generic types of environments provided: theorem environments, proof environments and “thmheading” environments.

1.1 Theorem environments

Theorem environments look like this:

Theorem 1 (Gauss). *For a closed surface S enclosing a volume V , we have*

$$\oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) dV . \quad (1)$$

1.2 Proof environments

A proof environment might look like this:

Proof of Theorem 1. The proof of the theorem goes here.
 ... ■

The enhanced theorem and proof environments provided by this package allow to pair theorems with proofs, automatically generating references from one to the other (see section 5).

1.3 Theorem-heading environments

Finally, theorem-heading environments are formatted like theorems, but the heading title is set as an argument to the environment. These environments are a nice alternative for definitions, and look like this:

Trace Distance. *The ‘trace distance’ between ρ and σ is defined as*

$$\delta(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1, \quad (2)$$

where $\|\cdot\|_1$ is the Schatten-1 norm.

■ 2 Quick start and package options

Example: Load the rich theorem set, with separate counters, and with proof-ref mechanism on and always displaying the proof reference in the margin:

```
\usepackage[thmset=rich,sepcounters=true,proofref={always,margin}]{phfthm}
```

By default, some styles are tweaked a bit so that they appear nicely as documented below (for example, by using a filled square instead of a simple square for end-of-proof QED markers). Use the package option `resetstyle` to instruct `phfthm` not to proceed to these style adjustments; this allows you to enable features individually and selectively:

```
\usepackage[resetstyle,prooftitleitbf=true]{phfthm}
```

2.1 Predefined theorem environments

You may load predefined theorem sets via the package option `thmset`. Theorem sets group common environments used in mathematical works such as Theorem, Proposition, Definition, etc.

Some package options control the way these environments are defined. If you would like more refined control over the appearance of these environments, or over which environments are defined, you may consider calling `\phfLoadThmSet` manually or defining individual environments with `\phfMakeTheorem`.

Possible theorem sets are:

`thmset=` , `thmset=empty`

Do not define any environment at package loading. You may of course invoke `\phfLoadThmSet` or `\phfMakeTheorem` manually at any later point.

`thmset=simple`

theorem Define the environments theorem, proposition, lemma and corollary
proposition as theorem-like environments, and definition as a definition-like environment.
lemma
corollary

`thmset=default`

definition Define the environments theorem, proposition, lemma, corollary,
conjecture conjecture, remark as theorem-like environments, and definition
remark as a definition-like environment.

`thmset=shortnames`

In case you prefer to type less: the same environments are defined as the default set, but with shorter names. Define the environments `thm`, `prop`, `lem`, `cor`, `conj`, `rem` as theorem-like environments, and `defn` as a definition-like environment.

`thmset=rich`

idea Provides the same environments as the default theorem set, as well
question as in addition: idea, question, claim, observation, and problem as
claim theorem-like environments.
observation

problem You may also load a theorem set at a later point after loading the `phfthm` package by invoking the `\phfLoadThmSet` macro, see [subsection 3.2](#).

Further package options modify the style of the theorem-like and definition-like environments defined via the `thmset` package option:

`theoremstyle=<theorem style name>`

Use this package option to specify which style to use for theorem-like environments when loading the theorem set specified via the `thmset` package option. The theorem style name should be one of `plain`, `definition`, `remark`, or any other `\newtheoremstyle`-defined theorem style (see documentation of `amsthm`).

`definitionstyle=<theorem style name>`

Use this package option to specify which style to use for definition-like environments when loading the theorem set specified via the `thmset` package option. The theorem style name should be one of `plain`, `definition`, `remark`, or any other `\newtheoremstyle`-defined theorem style (see documentation of `amsthm`).

Further options control various aspects of the environments defined by `thmset`.

`sepcounters=<true | false>`

Each theorem environment defined with the `thmset` package option will use a separate counter if this option is set; otherwise (the default), there is a single counter which is shared by all those theorem environments.

`parentcounter=<counter name>`

Theorem counters will reset each time the parent counter is increased. Use this option for per-chapter or per-section numbering (use parent counter `parentcounter=chapter` or `parentcounter=section`). See also the `countpersection` package option.

The `proofref` package option allows to specify a comma-separated list of attributes to apply to the proof reference (“proof on page XYZ”) displayed along with the theorem. The following attributes may be specified:

`proofref={}`, `proofref={default}`

Do not change the default proof reference appearance.

`proofref=false`

Deactivate the proof-ref mechanism.

`proofref={margin, ...}`

The proof reference is displayed in the margin, instead of after the theorem.

`proofref={marginbottom, ...}`

The proof reference is displayed in the margin, but aligned with the bottom of the theorem statement.

`proofref={longref, ...}`

The proof reference is displayed as a full sentence (“The proof of this *Theorem Name* can be found on page *XYZ*”).

`proofref={off, ...}`

Turn off the proof reference mechanism completely for theorems defined with the `thmset` option.

`proofref={always, ...}`

Always display the proof reference, even if the proof is on the same page or on a nearby page.

Note: this option has a global effect.

`proofref={onyifveryfar, ...}`

The proof reference is only displayed if the proof is at least two pages back, or four pages ahead.

Note: this option has a global effect.

NOTE

The two package options `proofref={always}` and `proofref={onlyifveryfar}` apply to *all* theorem environments which use the proof-ref mechanism, whether they have already been defined or not (see `\phfProofrefPageBackTolerance` and `\phfProofrefPageAheadTolerance`). All the other above options apply only to the theorem environments defined via the `thmset` package option.

2.2 The proof environment

`proof` By default, the `phfthm` package overrides the `proof` environment with a the package's own enhanced version. If you want to preserve the original *AMS* environment, you should use the `proofenv=false` package option.

`proofenv=<true | false>`

If set to `true`, then define an enhanced `proof` environment when loading this package. This will override any previously existing `proof` environment such as *AMS*.

If set to `false`, no action is taken at package loading time. You should then directly use the `\phfMakeProofEnv` macro to define `proof` environments.

If you want finer control over how the `proof` environment is defined, or if you want to customize its appearance, you should use the `\phfMakeProofEnv` macro directly (subsection 4.1).

If you set `proofenv=true`, there are a couple package options which alter the way the `proof` displays:

`smallproofs=<true | false>`

If set to `true`, then proofs display in a smaller font.

`qedsymbolblacksquare=<true | false>`

If set to `true`, the QED end-of-proof symbol (usually “□” with `amsthm`) is replaced by a filled square (“■”).

`prooftitleitbf=<true | false>`

If set to `true`, then the `proof` title (“Proof” or “Proof of Theorem 1”) is typeset in bold italic font.

2.3 The theorem-heading environment

`thmheading` By default, the `thmheading` environment is provided by the `phfthm` package:

```
\begin{thmheading}{Trace Distance}
  The ‘trace distance’ between  $\rho$  and  $\sigma$  is defined as
  \begin{equation}
    \delta(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1,
  \end{equation}
  where  $\|\cdot\|_1$  is the Schatten-1 norm.
\end{thmheading}
```

Trace Distance. *The ‘trace distance’ between ρ and σ is defined as*

$$\delta(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1, \quad (3)$$

where $\|\cdot\|_1$ is the Schatten-1 norm.

You may also use `\label` and `\ref` as usual (`\ref` simply displays the given title).

Some package options control the way this environment is defined.

`thmheading=(true | false)`

Define the environment `\begin{thmheading}... \end{thmheading}` when loading the `phfthm` package, with reasonable default settings.

`thmheadingstyle=(theorem style)`

If `thmheading=true` was specified, you may use this option to specify the theorem style to use for the `thmheading` environment. Possible values are AMS theorem style names (e.g. the base styles `plain`, `definition` or `remark`), or any other style defined with `\newtheoremstyle`.

If you want to define theorem-heading environments manually, see [subsection 6.1](#).

■ 3 Theorem environments

A theorem environment is based on the environment furnished by `amsthm`’s `\newtheorem` command, but with added goodies.

3.1 Define theorem environments manually

`\phfMakeTheorem` If you don’t want to load a full theorem set ([subsection 2.1](#)), you can define

theorem environments individually with `\phfMakeTheorem`:

```
\phfMakeTheorem[<key-value options>]{<theorem environment name>}  
{<theorem name>}
```

This command defines a new environment (given as the first mandatory argument) which behaves as a theorem and is displayed as given by the second mandatory argument. For example, we might call `\phfMakeTheorem{theorem}{Theorem}` to define the environment `\begin{theorem}... \end{theorem}` which displays “*Theorem N...*”

The possible key-value options for the optional argument are:

`counter=<LATEX counter | (empty)>`

The name of the L^AT_EX counter to use for the theorem environment. If this is empty, then a new counter will be created which is specific to this theorem environment (the default). If not empty, then the theorem environment uses the given counter (or an alias thereof, see [aliascounter](#)).

If a counter is specified, the counter should already be defined with L^AT_EX’s `\newcounter`.

`aliascounter=<true | false>`

In some cases (e.g. if you’re using `hyperref`’s `\autoref`), it is important to have counters specific to each theorem environment (so you get “Theorem 5” or “Proposition 5” right). However, you may want different theorem environments to share a same logical counter (Say “Definition 1”, “Definition 2”, “Theorem 3”, “Proposition 4”). In this case, you should specify [aliascounter=true](#).

When this option is on, then first we define an alias counter of the one given to the [counter](#) option, and then use the alias for the theorem environment. The alias is declared using the `aliascnt` package. The alias counter is automatically set up correctly for using `\autoref`.

Note that the [aliascounter](#) option only has an effect if the [counter](#) option is set to some non-empty value. If [counter](#) is set to a non-empty value, then [aliascounter](#) defaults to true.

`thmstyle=<theorem style | (empty)>`

The theorem style to use to define this theorem environment. The value of this option should be a valid argument to *AMS*’s `\theoremstyle`. If you leave this empty (the default), then the theorem style is not set explicitly and whatever default style is used.

`defnostar=<true | false>`

Set this to true if you want the corresponding non-starred theorem environment to be defined, e.g. `\begin{theorem}... \end{theorem}`.

Normal (non-starred) versions of the environments have an associated theorem number, as you expect by default.

defstar= \langle true | false \rangle

Set this to true if you want the corresponding starred theorem environment to be defined, e.g. `\begin{theorem*}... \end{theorem*}`.

Starred versions of the environments do not have an associated theorem number.

proofref= \langle true | false \rangle

Enable or disable the proof-ref mechanism for this theorem environment (enabled by default).

proofrefstyle= \langle proof-ref style \rangle

The style to use for the proof references. Here you may specify how the proof ref appears, for example (in the margin, long sentence, ...). Possible styles are proofrefstyle=default (the default), proofrefstyle=margin (display the proof ref in the margin of the page), proofrefstyle=marginbottom (display the proof ref in the margin of the page aligned with the bottom of the theorem statement) and proofrefstyle=longref (as by default but with a full sentence). See [subsection 5.3](#) for how to further customize the appearance of the proof reference.

For example, you may use the following command invocation to define a theorem environment named “Remark” implemented as `\begin{remark}... \end{remark}`, also with a starred version `\begin{remark*}... \end{remark*}`, using the plain AMS theorem style, and without the proof-ref mechanism:

```
\phfMakeTheorem[defstar=true,defnstar=true,thmstyle=plain,counter=,%  
proofref=false]{remark}{Remark}
```

3.2 Loading theorem sets manually

`\phfLoadThmSet` You may load theorem sets at any time via the macro `\phfLoadThmSet`. This may be useful, for example, to load theorem sets only after you have defined a custom theorem style. The syntax of `\phfLoadThmSet` is:

```
\phfLoadThmSet{ $\langle$ options to \phfMakeTheorem for theorem-like  
environments $\rangle$ }{ $\langle$ options to \phfMakeTheorem for definition-like  
environments $\rangle$ }{ $\langle$ name of theorem set to load $\rangle$ }
```

The first and second argument to this macro are tokens to expand in front of `\phfMakeTheorem` for theorem-like or definition-like environments. For example:

```
\newcounter{mythmcounter}  
\newtheoremstyle{mythmstyle}{...}
```

```
\newtheoremstyle{mydefnstyle}{...}
\phfLoadThmSet{[thmstyle=mythmstyle,counter=mythmcounter]}
  {[thmstyle=mydefnstyle,counter=mythmcounter]}{rich}
```

WARNING

The first and second arguments to `\phfLoadThmSet` must either be empty, or be enclosed in square braces.

<pre>\theoremname \propositionname \lemmaname \corollaryname \conjecturename \remarkname \definitionname \ideaname \questionname \claimname \problemname</pre>	<p>The title of the theorem environments defined in theorem sets use the same scheme as figures, tables, etc. with regard to translations and babel: they use <code>\theoremname</code>, <code>\propositionname</code>, etc.</p> <p>This package is language agnostic (with titles defined by default in English), and does not provide the titles for other languages. In order to support language switching with <code>babel</code> and <code>\selectlanguage</code>, you should add the relevant names to the corresponding <code>\captions</code>(<i>language name</i>) macro, for example:</p> <pre>\usepackage[french,...]{babel} ... \addto\captionsfrench{% \def\theoremname{Th\'eor\`eme}% \def\propositionname{Proposition}% \def\lemmaname{Lemme}% \def\corollaryname{Corollaire}% \def\conjecturename{Conjecture}% \def\remarkname{Remarque}% \def\definitionname{D\'efinition}% \def\ideaname{Id\'ee}% \def\questionname{Question}% \def\claimname{Affirmation}% \def\problemname{Probl\`eme}% } ... \selectlanguage{french} ...</pre>
--	--

3.3 Theorem hooks

Any theorem environment automatically calls some hooks. There are hooks available per theorem environment as well as generic for all theorem environments.

`\phfthm@hook@start@thmname` The hook `\phfthm@hook@start@<theorem environment name>{<theorem title>}` is called at the start of the environment. More precisely, it is called inside the original `amsthm` base environment; that is, after the heading was generated. It takes one mandatory argument, the optional title provided to the theorem

environment which may be empty. By default, the hook defers to the global hook `\phfthm@hook@startcommonnostar`.

`\phfthm@hook@start@thmname*` The hook `\phfthm@hook@start@⟨starred theorem environment name⟩` is completely analogous, and is called for the starred environment. The only difference is that by default, it defers its call to `\phfthm@hook@startcommonstar`.

`\phfthm@hook@startcommonnostar` The hook `\phfthm@hook@startcommonnostar{⟨theorem environment name⟩}{⟨theorem optional given title⟩}` collects the default definitions for non-starred environments (none by default) and continues to defer to `\phfthm@hook@startcommon{⟨theorem environment name⟩}{⟨theorem optional given title⟩}`. Analogously, the macro `\phfthm@hook@startcommonstar{⟨theorem environment name⟩}{⟨theorem optional given title⟩}` groups commands for starred environments (typically doesn't take care of `\label` stuff) and also defers to `\phfthm@hook@startcommon`.

`\phfthm@hook@end@thmname` The end hooks work very much analogously. `\phfthm@hook@end@⟨theorem environment name⟩` and `\phfthm@hook@end@⟨starred theorem environment name⟩` are called respectively for the non-starred and starred version of that theorem environment, and by default they defer to the common `\phfthm@hook@endcommonnostar{⟨theorem environment name⟩}` or `\phfthm@hook@endcommonstar{⟨theorem environment name⟩}`. Both these hooks defer their calls to `\phfthm@hook@endcommon{⟨theorem environment name⟩}`.

`\phfthm@hook@afterlabel@thmname` For theorems using the proof-reference mechanism, i.e. for which `\proofref=true` was specified to `\phfMakeTheorem` and which uses the `\label` hack (section 5), there is an additional hook. The hook `\phfthm@hook@afterlabel@⟨theorem environment name⟩` is called just after the `\label` command corresponding to the theorem is encountered (this should always be at the *beginning* of the theorem, see section 5). Depending on the proof-ref style, this hook may be used to generate the proof reference text (for example, with the `margin` proof-ref style). The hook is called after the theorem label is set. The label itself can be recovered from the value of the macro `\phfthm@val@thmlabel`. By default, that hook calls the common hook `\phfthm@hook@afterlabelcommon{⟨theorem environment name⟩}`. (After the first occurrence of the command `\label`, the latter's definition is restored.)

`\phfthm@hook@afterlabelcommon`

■ 4 Proof environments

Proof environments typeset mathematical proofs. The proof environment(s) provided by `phfthm` give some added functionality with respect to the *AMS*-default proof environment, such as supporting the proof-reference mechanism described in section 5.

A proof environment might look like the following:

Proof of Theorem 5. Let $\mathcal{T}_{X \rightarrow X'}$ be any trace nonincreasing completely positive map such that $\mathcal{T}_{X \rightarrow X'}(\Gamma_X)$ lies within the support of $\Gamma_{X'}$. Define the normalized state $\gamma_X = \Gamma_X / \text{tr} \Gamma_X$.

Now consider this and that ... ■

The proof environments defined by this package wrap a given proof display environment (such as *AMS'* (amsthm's) or IEEEtran's original proof environment) by adding functionality in the form of hooks. In the following, we refer to the "underlying proof display environment" as the original environment which is wrapped. It may be any \LaTeX environment whose task is to format the proof nicely.

4.1 Manually define a proof environment

`\phfMakeProofEnv` You may use the macro `\phfMakeProofEnv` to declare a new proof environment. The syntax is:

```
\phfMakeProofEnv [⟨key-value options⟩] {⟨proof environment name⟩}
```

This defines a new environment with the given name, which may be used to display proofs to theorems. The options may be:

`displayenv=⟨name of \LaTeX environment⟩`

Set a \LaTeX environment to use to actually format and display the proof. (The `\phfMakeProofEnv` command itself doesn't care about how the proof is displayed or formatted; rather it adds a goodies infrastructure in which stuff can be plugged in and provides options for such goodies.)

You may specify here the name of a \LaTeX environment, or give the special value `displayenv=*` to indicate the default appearance provided by `phfthm`, or leave the value `displayenv=` to signify that no underlying display environment should be invoked. (The latter may be useful if you are plugging a `\phfMakeProofEnv`-generated environment into a larger environment which already takes care of the display.)

`defaultproofname=⟨default proof title⟩`

Specify here the title to use (e.g. "Proof") if no argument was given to the proof environment. If you do not specify any `defaultproofname`, or pass an empty value, then the value of `\proofname` is used.

`parselabel=(true | false)`

Specify whether the environment should parse its argument for some special information. If set to `true`, then the proof argument is passed on to a command (specified by the `parselabelcmd` option).

`parselabelcmd=⟨ \LaTeX macro⟩`

If `parselabel` is set to `true`, then specify here a \LaTeX command which parses whatever it wants from the proof environment's argument.

The macro should set the `\phfthm@val@displayargs` macro to tokens which will be expanded just after the invocation of the proof environment's display environment (`displayenv`). It should also set `\phfthm@val@proofoflabel` (if appropriate) to the label corresponding to the theorem for which this is the proof of.

By default, the command `\phfthm@proof@parselabel` is used, which parses the proof environment's argument for a reference to a theorem in the context of a proof-ref mechanism (see section 5). The label is parsed to see if it is of the form `[*thm:reference]`, where `thm:reference` is the label pinned to a theorem.

override=`<true | false>`

Whether to override any existing environment with the same name as the new proof environment. If `true` is specified here, then `\renewenvironment` is used to define the proof environment, otherwise a simple `\newenvironment` is used.

internalcounter=`<name of LATEX counter>`

The name of the internal counter the proof environment should use. The count number is not displayed (by default at least), but it is only used to pin down anchors for PDF hyperlinks.

The counter should already be defined with `\newcounter`.

proofofname=`<LATEX macro>`

Specify here a macro which will be called with a single argument. The macro produces the text to display when the proof environment is parsed as the proof of a specific theorem or proposition (or other theorem environment). The argument which will be given to it is the title of what the proof is of (e.g. "Theorem 3"). Typically, the macro should produce something like "Proof of Theorem XYZ."

By default, the global macro `\proofofname` is used.

`\proofname` Text to use to display "Proof." This should be already defined by the L^AT_EX system, and babel should already provide translations in different languages.

`\proofofname` The globally defined macro `\proofofname` specifies the default way of displaying "Proof of Theorem 5." It is originally defined as something like

```
\newcommand\proofofname[1]{\proofname{ } of #1}
```

You may override this to obtain something fancier, of you wish to display the document in a different language. For instance, you might want to display ("Démonstration (Théorème 5)" in french:

```
\def\proofofnameinparentheses#1{\proofname\ (#1)}
\addto\captionsfrench{%
```

```

\def\proofname{D\'emonstration}%
\let\proofofname\proofofnameinparenteses%
}
...
\selectlanguage{french} ...

```

4.2 Proof hooks

The proof hooks are relatively straightforward. All hooks presented here take no argument.

Information about the argument of the proof, both the raw argument and the possibly parsed reference, are available as macros to some of the hooks (but don't change these values unless you know what you're doing). The macro `\phfthm@val@proofarg` contains the raw argument to the proof environment, and is available to all hooks. If you use the default proof environment argument parsing (which you must have enabled when calling `\phfMakeProofEnv`), then additionally the macros `\phfthm@val@prooftitle` and `\phfthm@val@proofofname` are available containing, respectively, the label of the theorem which is referenced, and the displayable reference to it (e.g. "Theorem 5"). The last two macros are available to all hooks except the first one (`..@start`).

The hooks named `..@start...` are called within the call to `\begin{<proof environment>}`.

`\phfthm@hookproof@..@start`

The hook named `\phfthm@hookproof@<environment name>@start` is called at the very beginning of the proof environment.

`\phfthm@hookproof@..@startafterdisplay`

The hook named `\phfthm@hookproof@<environment name>@startafterdisplay` is invoked immediately after the beginning of the underlying "display" environment (the environment used to display the proof contents).

`\phfthm@hookproof@..@startlast`

The hook named `\phfthm@hookproof@<environment name>@startlast` is called after we are sure that an anchor has been pinned down for the proof. This hook is called last within the commands in `\begin{<proof environment>}`.

`\phfPinProofAnchor`

By the way, the macro `\phfPinProofAnchor` may be used within the hooks to pin down an anchor for referring to the proof (especially via the proof-ref mechanism). Just call it anywhere appropriate (a good idea is calling it after leaving v-mode before displaying the title, in order to avoid placing it just before a page break). If you do not call this macro, it is automatically called for you just before the `..@startlast` hook.

The two following hooks are called within the call to `\end{<proof environment>}`.

`\phfthm@hookproof@...@end` The hook `\phfthm@hookproof@<environment name>@end` is called before the proof display environment is closed.

`\phfthm@hookproof@...@final` The hook `\phfthm@hookproof@<environment name>@final` is called after the proof environment display is finished, as the very last.

All proof hooks call are defined by default to defer their call to a common hook. The common hooks each take one argument (the proof environment name). They are named `\phfthm@hookproof@startcommon{<environment name>}`, `\phfthm@hookproof@startafterdisplaycommon{<environment name>}`, `\phfthm@hookproof@startlastcommon{<environment name>}`, `\phfthm@hookproof@endcommon{<environment name>}`, and `\phfthm@hookproof@finalcommon{<environment name>}`. They are all defined to be empty by default.

■ 5 Pairing theorems to proofs and proof-reference mechanism

One of the goodies provided by the `phfthm` package is the proof-ref mechanism, where in a theorem environment, the text “see proof on page ...” is displayed to direct the reader to the location of the corresponding proof. The mechanism is deactivated by default, but can be enabled with a simple package option.

This only works if the proof is given the label of the corresponding theorem or proposition. For example:

```
\begin{theorem}[Gauss]
  \label{thm:Gauss}
  For a closed surface $$$ enclosing a volume $V$, we have
  \begin{equation}
    \oint_S \vec u \cdot d\vec S = \int_V (\vec \nabla \cdot \vec u) \, dV \ .
  \end{equation}
\end{theorem}

...

\begin{proof}[*thm:Gauss]
  ...
\end{proof}
```

The above example might produce the following output:

Theorem 17 (Gauss). For a closed surface S enclosing a volume V , we have

$$\oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) dV. \quad (41)$$

(Proof on page XXX.)

...

Proof of Theorem 17. ...

A positive side-effect is that in your LaTeX source, if you consistently label your proofs then you always know what theorem a proof refers to. If you would like to simply output “Proof.” instead of “Proof of Theorem X.”, for instance, because the proof immediately succeeds the theorem in question, you may use the syntax `[**thm:label]` instead. In this case, the proof-reference mechanism will know that this is the proof of the theorem labeled `thm:label` and will avoid undefined references. For example, the following code:

```
\begin{theorem}[Gauss]
  \label{thm:Gauss}
  For a closed surface $$$ enclosing a volume $$V$, we have
  \begin{equation}
    \oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) dV.
  \end{equation}
\end{theorem}
\begin{proof}[**thm:Gauss]
  ...
\end{proof}
```

might output:

Theorem 17 (Gauss). For a closed surface S enclosing a volume V , we have

$$\oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) dV. \quad (41)$$

Proof. ...

5.1 On the theorem side

On the theorem side, the proof-ref mechanism works by hacking into the definition of L^AT_EX’s `\label`. The `\label` command should be placed first within the theorem (see example above). It is important, in theorems which use the proof-ref mechanism (on by default), to always have a corresponding label: Indeed, you may experience weird results if you don’t have a theorem label, but then have labels for other objects in the theorem such as equations or itemize items.

Once the corresponding proof is detected (a proof environment with an optional argument of the form `[*thm:the-label]` for the same label `thm:the-label`

as specified to the theorem, see [subsection 5.2](#)), then a text is generated (by default “Proof on page ...”) and placed after the theorem. The appearance of this text is customizable ([subsection 5.3](#)).

More precisely, the hack with the `\label` command works as follows: At the beginning of the theorem, the `\label` command is redefined so that at its first occurrence, it stores its argument as the theorem's label to use for the proof reference, it then pins down a \TeX label as the original `\label` command would do, and finally it calls the `...@afterlabel` theorem hook (see [3.3](#)). After the first occurrence of `\label`, the `\label` command is restored to its original \TeX meaning in case there are other objects within the theorem which are to be referred to.

TIP

The `\label` hack is only active within theorem environments where the proof-ref mechanism has been enabled. Outside these environments, the `\label` macro retains its original \TeX definition.

`\noproofref` If, for any reason, you do not want to make sure you don't have any text “Proof on page ...” appearing (for example there is no corresponding proof because the theorem is obvious), then you should call `\noproofref` immediately inside the theorem:

```
\begin{theorem}
  \noproofref
  Theorem text ...
\end{theorem}
```

The command `\noproofref` temporarily disables the proof-ref mechanism (and restores `\label` to \TeX 's original meaning) for the current theorem.

5.2 On the proof side

On the proof side, you just need to specify for which theorem this is the proof of. For that (unless you override the defaults and plug in your own magic parsing; see [subsection 4.1](#)), you should specify an optional argument to the proof which is of the following form: `\begin{proof}[*<label>]`, where `<label>` is the label name you have associated with the theorem in question (see example above).

This has two effects: it sets the proof to display “*Proof of...*,” and also does some background dark magic to display, at the location of the corresponding theorem, some text like “*Proof on page...*,” where the page number corresponds to the page on which this proof is located.

You may also use the syntax `\begin{proof}[*<label>]`, which has the effect of producing the simple output “Proof.” (this is desirable e.g. if the proof immediately follows the theorem statement) but which still informs the proof-reference

mechanism that this proof is associated with the given theorem labeled by $\langle label \rangle$. This can avoid undefined references, and can simplify your job if you want to move proofs around in your document.

(It would be nice, of course, to have the package automatically output “Proof” only if the proof immediately follows the theorem and output “Proof of XXX” if the proof is further away. Unfortunately this is hard to detect in \LaTeX and I currently have no plans to try to implement this.)

5.3 Customizing appearance of the proof reference text

Here we explain the workings of the `\phfthm@proofrefstyle@...` macros and how they are called. It allows you to define new proof-ref styles, for example.

When the option `proofref=true` is given to `\phfMakeTheorem` to define an environment (say `mytheoremenv`), then the hook `phfthm@hook@start@mytheoremenv` will automatically include the following calls:

- The macro `\phfthm@proofrefstyle@<proof-ref style>@setup` is called (for the proof-ref style given via the `proofenvstyle=<style name>` key-value option to `\phfMakeTheorem`);
- The macro `\phfthm@def@label@thmlabel` is invoked, implementing the hack on the `\label` macro;
- The macro `\phfthm@proofref@impl@start` is called. This macro is expected to be defined after calling `\phfthm@proofrefstyle@<proof-ref style>@setup`.

Furthermore, the `\phfthm@hook@afterlabel@mytheoremenv` hook will include a call to `\phfthm@proofref@impl@afterlabel{<label of the theorem>}`. Again, the latter macro is expected to be defined after calling `\phfthm@proofrefstyle@<proof-ref style>@setup`.

Finally, the hook `\phfthm@hook@end@mytheoremenv` includes a call to `\phfthm@proofref@impl@end{<label of the theorem>}`. Again, the latter macro is expected to be defined after calling `\phfthm@proofrefstyle@<proof-ref style>@setup`.

Hence, to define a new proof-ref style, you simply need to define a macro called `\phfthm@proofrefstyle@<PROOF-REF-STYLE-NAME>@setup`. This macro should include commands to locally define the macros `\phfthm@proofref@impl@start`, `\phfthm@proofref@impl@afterlabel`, and `\phfthm@proofref@impl@end`.

Different proof-ref styles may work similarly and want to share most of the code. A good idea is to build up on the default proof-ref style, which is highly modular and can be instantiated in different flavors. For an example, check the margin

proof-ref style which does precisely that. For more documentation, check out the implementation of the default proof-ref style in [subsection 7.4.3](#).

■ 6 Theorem-heading definition-like environments

A theorem-heading environment is an environment which displays in the same way as a theorem environment, but where the title may be any text (say, “Trace Distance” instead of, e.g., “Theorem 5”).

By default, the `phfthm` package provides the `thmheading` environment (see [subsection 2.3](#)).

6.1 Define theorem-heading environments manually

`\phfMakeThmheadingEnvironment` A new theorem-heading environment can be defined by calling `\phfMakeThmheadingEnvironment`. The syntax is:

```
\phfMakeThmheadingEnvironment [key-value options] {environment name}
```

The key-value options may be any combination of the following:

`thmstyle`=*<theorem style name>*

The theorem style to use to display the environment. You may specify here any default AMS style (`plain`, `remark` or `definition`), or any other `\newtheoremstyle`-defined style.

`internalcounter`=*<name of counter>*

The name of a counter which will internally track environment instances. By default, a common internal counter is used for all theorem-heading environments (named `phfthmheadingcounter`). The counter must be already defined (see L^AT_EX’s `\newcounter`).

You can also use `\label` and `\ref` (the latter simply displays the given title).

6.2 Available hooks for theorem-heading environments

The hook `\phfthm@hook@thmheading@...@start{<title>}` is invoked at start, within an internal environment created with `\newtheorem`. This hook accepts one argument, the title of the theorem-heading.

The hook `\phfthm@hook@thmheading@...@end` is called at the end, but still within the internal theorem environment.

Replace the dots with the name of the theorem-heading environment (such as `thmheading`).

By default, these hooks simply call the common hooks `\phfthm@hook@thmheading@start` and `\phfthm@hook@thmheading@end`. These common hooks are empty by default.

■ 7 Implementation

First, load some packages. General toolboxes:

```
1 \RequirePackage{xkeyval}
2 \RequirePackage{etoolbox}
```

To define alias counters for theorems, load `aliascnt`:

```
3 \RequirePackage{aliascnt}
```

And finally, load the *AMS* math and theorem (`amsmath`, `amsthm`) packages:

```
4 \RequirePackage{amsmath}
5 \RequirePackage{amsthm}
```

7.1 Generic Internal Stuff

`\phfthm@internal@execattribs` Internal command: execute all definitions given in list of attributes. This was copy-pasted from a similar definition in the `phfnote` package.

#1 = prefix to look for attributes

#2 = name of what #1 represents, to use in message in case attribute is not found

#3 = list of attributes

```
6 \def\phfthm@internal@execattribs#1#2#3{%
7   \@for\next:=#3\do{%
8     \ifcsname #1\next\endcsname%
9     \csname #1\next\endcsname%
10    \else%
11      \PackageWarning{phfthm}{Unknown #2: '\next'. Ignoring.}
12    \fi
13  }
14 }
```

7.2 Definitions for theorem environments

7.2.1 `\phfMakeTheorem`: definition of a new theorem environment

First, define some key-value syntax accepted by the `\phfMakeTheorem` command.

```

15 \define@cmdkey{phfmkthm}{counter}{}
16 \define@boolkey{phfmkthm}{aliascounter}[true]{}
17 \define@cmdkey{phfmkthm}{thmstyle}{}
18 \define@boolkey{phfmkthm}{defnostar}[true]{}
19 \define@boolkey{phfmkthm}{defstar}[true]{}
20 \define@boolkey{phfmkthm}{proofref}[true]{}
21 \define@cmdkey{phfmkthm}{parentcounter}{}
22 \define@cmdkey{phfmkthm}{proofrefstyle}{}

```

`\phfMakeTheorem` Define a new theorem environment. The syntax is `\phfMakeTheorem` [*options*] {*theorem environment name*} {*Theorem Display Name*}. For example: `\phfMakeTheorem[counter=thmcounter]{prop}{Proposition}`

```

23 \newcommand\phfMakeTheorem[3] [] {% }

```

Handle the [options]. First, ensure that the defaults are set, and then, parse the input.

```

24 \KV@phfmkthm@aliascountertrue%
25 \def\cmdKV@phfmkthm@counter{}%
26 \def\cmdKV@phfmkthm@thmstyle{}%
27 \KV@phfmkthm@defnostartrue%
28 \KV@phfmkthm@defstartrue%
29 \KV@phfmkthm@proofreftrue%
30 \def\cmdKV@phfmkthm@parentcounter{}%
31 \def\cmdKV@phfmkthm@proofrefstyle{default}%
32 \setkeys{phfmkthm}{#1}%

```

Now, react to whatever was given in the options.

Set the theorem style, if requested.¹

```

33 \if\relax\detokenize\expandafter{\cmdKV@phfmkthm@thmstyle}\relax%
34 \else%
35   \theoremstyle{\cmdKV@phfmkthm@thmstyle}%
36 \fi%

```

If requested, define the default, unstarred version of the theorem. Use `\newtheorem` for that, which we make sure to call appropriately depending on whether a separate counter is requested or not. Make sure also to define `\dotsautorefname` for `\autoref`. If an alias counter is requested, create it and pass that one to `\newtheorem`.

At this point, we create a theorem named `phfthm@...` using `\newtheorem` (because we still want to add calls to hooks).

```

37 \ifKV@phfmkthm@defnostar%
38   \if\relax\detokenize\expandafter{\cmdKV@phfmkthm@counter}\relax%

```

¹The construct `\if\relax\detokenize{...}\relax` tests whether ... is empty: see <http://tex.stackexchange.com/a/53091/32188>

—in case we use a separate given counter (if `counter=` is given empty or not specified). Also check if we want a parent counter (per-section or per-chapter numbering) and take that into account:

```

39     \if\relax\detokenize\expandafter{\cmdKV@phfkmthm@parentcounter}\relax%
40     \edef\x{\noexpand\newtheorem{phfthm@#2}{#3}}%
41     \else
42     \edef\x{\noexpand\newtheorem{phfthm@#2}{#3}[\expandonce\cmdKV@phfkmthm@parentcounter]}%
43     \fi
44     \x
45     \csdef{phfthm@#2autorefname}{#3}%
46 \else%
47     \ifKV@phfkmthm@aliascounter%

```

—in case we make a distinct alias counter, eg. for use with `\autoref`:

```

48     \newaliascnt{#2}{\cmdKV@phfkmthm@counter}%
49     \newtheorem{phfthm@#2}[#2]{#3}%
50     \aliascntresetthe{#2}%
51     \csdef{#2autorefname}{#3}%
52 \else%

```

—in case we directly instruct `\newtheorem` to use the other counter (does not work with `\autoref`):

```

53     \newtheorem{phfthm@#2}[\cmdKV@phfkmthm@counter]{#3}%
54 \fi%
55 \fi%

```

And also define the actual theorem environment, adding calls to hooks.

```

56 \newenvironment{#2}[1][ ]{%
57 \begin{phfthm@#2}[##1]}%
58 \begingroup%
59 \csname phfthm@hook@start@#2\endcsname{##1}%
60 }{%
61 \csname phfthm@hook@end@#2\endcsname%
62 \endgroup%
63 \end{phfthm@#2}%
64 }%

```

Define hooks specific to this theorem with sensible defaults. If `proof-ref` is on, call the appropriate callbacks. Then, call the common hooks (see `\phfthm@hook@startcommonnostar`, `\phfthm@hook@afterlabelcommon` and `\phfthm@hook@endcommonnostar`, detailed in [subsection 3.3](#)).

```

65 \csedef{phfthm@hook@start@#2}##1{%
66 \ifKV@phfkmthm@proofref%
67 \expandafter\noexpand%
68 \csname phfthm@proofrefstyle@\cmdKV@phfkmthm@proofrefstyle @setup\endcsname%
69 \noexpand\phfthm@def@label@thmlabel{#2}%

```

```

70     \noexpand\phfthm@proofref@impl@start%
71     \fi%
72     \noexpand\phfthm@hook@startcommonstar{#2}{##1}%
73   }%
74   \csedef{phfthm@hook@afterlabel@#2}{%
75     \ifKV@phfmkthm@proofref%
76       \noexpand\phfthm@proofref@expandthmlabeltoarg%
77       \noexpand\phfthm@proofref@impl@afterlabel%
78     \fi%
79     \noexpand\phfthm@hook@afterlabelcommon{#2}%
80   }%
81   \csedef{phfthm@hook@end@#2}{%
82     \ifKV@phfmkthm@proofref%
83       \noexpand\phfthm@proofref@expandthmlabeltoarg%
84       \noexpand\phfthm@proofref@impl@end%
85     \fi%
86     \noexpand\phfthm@hook@endcommonstar{#2}%
87   }%
88   \fi%

```

If requested, define the starred version of the theorem. We call `\newtheorem*` to define the base theorem environment (which we call `phfthm@. . .`), after which as above we define the actual environment which also calls the relevant hooks.

```

89   \ifKV@phfmkthm@defstar%
90     \newtheorem*{phfthm@#2*}{#3}%
91     \newenvironment{#2*}[1][ ]{%
92       \begin{phfthm@#2*}[##1]%
93       \begingroup%
94         \csname phfthm@hook@start@#2*\endcsname{##1}%
95       }{%
96         \csname phfthm@hook@end@#2*\endcsname%
97       \endgroup%
98     \end{phfthm@#2*}%
99   }%
100  \fi%

```

Finally, define the default hooks specific to the starred version of the theorem (see [subsection 3.3](#)).

```

101  \csdef{phfthm@hook@start@#2*}##1{\phfthm@hook@startcommonstar{#2}{##1}}%
102  \csdef{phfthm@hook@end@#2*}{\phfthm@hook@endcommonstar{#2}}%
103 }

```

7.2.2 Default hooks for theorems

`\phfthm@hook@startcommonnostar` Common default hooks definitions for start of the theorems.

`\phfthm@hook@startcommonstar`
`\phfthm@hook@startcommon` For all three of these hooks, we have #1 = theorem name, e.g. proposition and #2 = full (optional) title of proposition, if given, or empty.

Make sure to invoke the `\label` re-definition hack only for non-starred theorems/propositions; indeed, if no theorem label is set we don't want to interfere with labels set to inner equations, itemizes etc. Hence, call `\phfthm@def@label@thmlabel` only in the "nostar" hook.

```
104 \def\phfthm@hook@startcommonnostar#1#2{%
105   \phfthm@hook@startcommon{#1}{#2}%
106 }
107 \def\phfthm@hook@startcommonstar#1#2{%
108   \phfthm@hook@startcommon{#1}{#2}%
109 }
110 \def\phfthm@hook@startcommon#1#2{%
```

Furthermore, in any case, set the `\postdisplaypenalty` to avoid an orphan line on a new page after display equation.

```
111   \postdisplaypenalty=10000\relax%
112 }
```

`\phfthm@hook@afterlabelcommon` Further hooks, for after the theorem main `\label` command
`\phfthm@hook@endcommonnostar` (`\phfthm@hook@afterlabelcommon`) and for the end of the theorem.

```
\phfthm@hook@endcommonstar
\phfthm@hook@endcommon
113 \def\phfthm@hook@afterlabelcommon#1{}
114 \def\phfthm@hook@endcommonnostar#1{\phfthm@hook@endcommon{#1}}
115 \def\phfthm@hook@endcommonstar#1{\phfthm@hook@endcommon{#1}}
116 \def\phfthm@hook@endcommon#1{}
```

7.2.3 Proof-ref mechanism (on the theorem side)

These macros enable the proof-ref mechanism (section 5). The theorem's label is stored upon calling `\label`, because we (locally) hack into the definition of `\label`. (After the first usage of `\label` its meaning is restored.)

`\phfthm@def@label@thmlabel` Main macro to invoke at the beginning of the theorem environment, so that the theorem label is stored in a local macro once `\label` is invoked. This hacks the `\label` macro locally. Here, #1 = the theorem environment name, e.g. proposition.

```
117 \def\phfthm@def@label@thmlabel#1{%
118   \ifdefined\phfthm@old@label
119     \PackageWarning{phfthm}{Internal inconsistency: \string\phfthm@def@label@thmlabel
120       called twice for the same theorem environment!}
121   \else
122     \let\phfthm@old@label\label%
123     \edef\label{\noexpand\phfthm@thmlabel{#1}}%
```



```
124 \fi
125 }
```

`\phfthm@thmlabel` The first call to `\label` within the theorem redirects to the macro `\phfthm@thmlabel`. (Applies to theorem environments for which `\phfthm@def@label@thmlabel` was called, which is the default).

Here #1 = theorem environment name, e.g. `proposition`; and #2 = the label value (argument to the `\label` macro).

```
126 \def\phfthm@thmlabel#1#2{%
```

First, store the label value into a macro called `\phfthm@val@thmlabel`.

```
127 \def\phfthm@val@thmlabel{#2}%
```

Then, call the original `\label` macro to do what \TeX would normally do for a `\label{...}` call.

```
128 \phfthm@old@label{#2}%
```

Restore the old `\label` definition, in case there are other items in the theorem environment such as equations, itemizes etc. which may themselves have `\label`'s.

```
129 \let\label\phfthm@old@label%
```

Invoke the `\phfthm@hook@afterlabel@thmname` hook for this theorem environment.

```
130 \csname phfthm@hook@afterlabel@#1\endcsname%
```

Finally, ignore any spaces following the `\label` command. (Maybe we should have done something with `\@bsphack` and `\@esphack` but oh well...

```
131 \ignorespaces%
132 }
```

7.3 Definitions for proof environments

Improved, smarter proof environments.

```
\phfthm@old@proof Save old proof environment provided by amsthm.
\endphfthm@old@proof
133 \let\phfthm@old@proof\proof
134 \let\endphfthm@old@proof\endproof
```

`\proofname` And provide a default name for proofs (this should normally already be provided by `amsthm`).

```
135 \providecommand\proofname{Proof}
```

`\proofofname` Default text to display when we want to say e.g. “Proof of Theorem 3.”

```
136 \def\proofofname#1{\proofname\space of #1}
```

The default counter for proofs. The value of this counter is typically not displayed, we just use it to pin down anchors for labels for cross references.

```
137 \newcounter{phfthmproofcnt}
```

Utility: to see if an argument was specified (possibly empty) to the proof environment.

```
138 \def\phfthm@NOPROOFARG{}
```

```
139 \def\phfthm@test@NOPROOFARG{\phfthm@NOPROOFARG}
```

7.3.1 Define a proof environment: `\phfMakeProofEnv`

Declare some key-value options accepted by `\phfMakeProofEnv`. See [subsection 4.1](#) for the documentation of these options.

```
140 \define@cmdkey{phfmkprf}{displayenv}{}
141 \define@cmdkey{phfmkprf}{defaultproofname}{}
142 \define@boolkey{phfmkprf}{override}[true]{}
143 \define@cmdkey{phfmkprf}{internalcounter}{}
144 \define@cmdkey{phfmkprf}{proofofname}{}
145 \define@boolkey{phfmkprf}{parselabel}[true]{}
146 \define@cmdkey{phfmkprf}{parselabelcmd}{}

```

`\phfMakeProofEnv` Make a proof environment. Syntax: `\phfMakeProofEnv [(options)] {<proof environment name>}`.

```
147 \newcommand\phfMakeProofEnv[2][{}]{%
```

Parse the key-value options. First, make sure that all the defaults are set, then parse the options.

```
148 \KV@phfmkprf@overridefalse%
149 \def\cmdKV@phfmkprf@displayenv{*}%
150 \def\cmdKV@phfmkprf@defaultproofname{\proofname}%
151 \def\cmdKV@phfmkprf@internalcounter{phfthmproofcnt}%
152 \def\cmdKV@phfmkprf@proofofname{\proofofname}%
153 \KV@phfmkprf@parselabeltrue
154 \def\cmdKV@phfmkprf@parselabelcmd{\phfthm@proof@parselabel}%
155 \setkeys{phfmkprf}{#1}%

```

The meaning of the options are detailed in [subsection 4.1](#).

The general idea here is first to pre-process all the options, and save all the useful information in macros named `\phfthm@prfenv@<proof-environment-name>@val@<something>`.

Then, we can define the begin/end environment macros which will recall the saved information.

Take care of the display environment to use. Recall that if `displayenv=*`, we use our own default; if `displayenv=` (empty), there is no display environment. Here, we set `\phfthm@prfenv@<proof-environment-name>@val@displayenv` to the name of the environment to use (possibly empty), for later reference.

```

156 \def\phfmkprf@tmp@star{*}%
157 \ifx\cmdKV@phfmkprf@displayenv\phfmkprf@tmp@star\relax%
158   \def\cmdKV@phfmkprf@displayenv{phfthm@proof@defaultdisplayenv}%
159   \fi
160 \cslet{phfthm@prfenv@#2@val@displayenv}\cmdKV@phfmkprf@displayenv%

```

Process the default proof name. If none is given, use `\proofname` and pass no option to the underlying display environment whenever the proof environment is called with no option. Here, we set `\phfthm@prfenv@<proof-environment-name>@val@defaultproofnameargs` and `\phfthm@prfenv@<proof-environment-name>@val@setdefaultprooftitle`; the former is the tokens to put in front of the proof environment invocation in case no explicit proof title is given to the proof environment while the latter contains the command to set `\phfthm@val@prooftitle` to the default proof name.

```

161 \if\relax\detokenize\expandafter{\cmdKV@phfmkprf@defaultproofname}\relax%
162   \csdef{phfthm@prfenv@#2@val@defaultproofnameargs}{}%
163   \csdef{phfthm@prfenv@#2@val@setdefaultprooftitle}{%
164     \def\phfthm@val@prooftitle{\proofname}}%
165   \else
166     \csedef{phfthm@prfenv@#2@val@defaultproofnameargs}{%
167       [\expandonce{\cmdKV@phfmkprf@defaultproofname}]}%
168     \csedef{phfthm@prfenv@#2@val@setdefaultprooftitle}{%
169       \noexpand\def\noexpand\phfthm@val@prooftitle{%
170         \expandonce{\cmdKV@phfmkprf@defaultproofname}}}%
171     \fi
172   \csedef{phfthm@prfenv@#2@val@parselabelandmkdisplayargs}##1{%
173     \ifKV@phfmkprf@parselabel
174       \expandonce\cmdKV@phfmkprf@parselabelcmd{##1}%
175     \else
176       \noexpand\phfthm@proof@noparselabel{##1}%
177     \fi
178     \noexpand\if\noexpand\relax\noexpand\detokenize%
179       \noexpand\expandafter{\noexpand\phfthm@val@prooftitle}\noexpand\relax
180     \noexpand\def\noexpand\phfthm@val@displayargs{
181     \noexpand\else
182     \noexpand\def\noexpand\phfthm@val@displayargs{[%
183       \expandafter\noexpand\csname phfthm@prfenv@#2@val@proofofname\endcsname
184       {\noexpand\phfthm@val@prooftitle}%
185     ]}%
186     \noexpand\fi
187   }

```

Store the macro which creates the “Proof of ...” text (proofofname option).

```
188 \cslet{phfthm@prfenv@#2@val@proofofname}\cmdKV@phfmkprf@proofofname%
```

Create the macro which will take care of pinning down the label for the proof-ref (see [section 5](#)). This macro first ref-steps the internal counter and then pins down a label, if appropriate.

```
189 \csdef{phfthm@prfenv@#2@val@pinproofanchor}{%
190   \csname phfthm@prfenv@#2@val@refstepinternalcounter\endcsname%
191   \if\relax\detokenize\expandafter{phfthm@val@proofoflabel}\relax\else%
192     \edef\phfthm@tmp@larg{{proof:\phfthm@val@proofoflabel}}%
193     \expandafter\label\phfthm@tmp@larg%
194   \fi
195 }%
196 %
```

The command to ref-step the internal proof counter. Use the value of the `internalcounter` command option.

```
197 \csedef{phfthm@prfenv@#2@val@refstepinternalcounter}{%
198   \noexpand\refstepcounter{\cmdKV@phfmkprf@internalcounter}}%
```

Make macros `\phfthm@prfenv@<proof-environment-name>@val@displayenvbegincmd` and `\phfthm@prfenv@<proof-environment-name>@val@displayenvendcmd`, which essentially expand to `\begin{<the-display-env>}` and `\end{<the-display-env>}` for the display environment given in the option `displayenv`.

```
199 \if\relax\detokenize\expandafter{\cmdKV@phfmkprf@displayenv}\relax%
200   \csdef{phfthm@prfenv@#2@val@displayenvbegincmd}##1{%
201   \csdef{phfthm@prfenv@#2@val@displayenvendcmd}##1{%
202   \else
203   \csedef{phfthm@prfenv@#2@val@displayenvbegincmd}##1{%
204     \noexpand\begin{\csname phfthm@prfenv@#2@val@displayenv\endcsname}##1}%
205   \csedef{phfthm@prfenv@#2@val@displayenvendcmd}##1{%
206     \noexpand\end{\csname phfthm@prfenv@#2@val@displayenv\endcsname}##1}%
207   \fi
```

See if we need to call `\newenvironment` or `\renewenvironment`, depending on the value of the `override` option.

```
208 \def\phfthm@tmp@defcmd{\newenvironment}%
209 \ifKV@phfmkprf@override\def\phfthm@tmp@defcmd{\renewenvironment}\fi%
```

Finally, (re-)define the environment. The default value of the optional argument is the token `\phfthm@NOPROOFARG`, which indicates that no argument was provided.

Start by storing the value of the argument into a macro, and then call the “start” hook (see proof hooks in [subsection 4.2](#)).

```

210 \phfthm@tmp@defcmd{#2}[1] [\phfthm@NOPROOFARG]{%
211 \def\phfthm@val@proofarg{##1}%
212 \csname phfthm@hookproof@#2@start\endcsname%

```

First, parse the optional argument into proof label (maybe) and proof title. If no optional argument was given, don't give any argument to the underlying display environment. If an empty argument was given, set some defaults; otherwise, use the necessary command to potentially parse the label and create the proper arguments for the underlying display environment.

```

213 \ifx\phfthm@val@proofarg\phfthm@test@NOPROOFARG\relax%
214 \def\phfthm@val@proofoflabel{}%
215 \csname phfthm@prfenv@#2@val@setdefaultprooftitle\endcsname%
216 \letcs\phfthm@val@displayargs{phfthm@prfenv@#2@val@defaultproofnameargs}%
217 \else%
218 \if\relax\detokenize{##1}\relax%
219 \def\phfthm@val@proofoflabel{}%
220 \csname phfthm@prfenv@#2@val@setdefaultprooftitle\endcsname%
221 \def\phfthm@val@displayargs{[%
222 \csname phfthm@prfenv@#2@val@proofofname\endcsname
223 {\phfthm@val@prooftitle}%
224 ]}%
225 \else
226 \csname phfthm@prfenv@#2@val@parselabelandmkdisplayargs\endcsname{##1}%
227 \fi
228 \fi%

```

Define the `\phfPinProofAnchor` command (locally) in case the display formatting environment takes care of where to place the anchor already.

```

229 \def\phfPinProofAnchor{%
230 \csname phfthm@prfenv@#2@val@pinproofanchor\endcsname%
231 \global\let\phfPinProofAnchor\relax}%

```

[Also provide the obsolete `\phfthmPinProofAnchor` which I previously had in older versions of this package:]

```

232 \def\phfthmPinProofAnchor{\phfPinProofAnchor}%

```

Start the proof's display environment. Don't be fooled here by the curly braces after `\x`, it only protects the argument to the `\phfthm@prfenv@#2@val@displayenvbegincmd` command itself: the `\phfthm@val@displayargs` are still just tokens which will be expanded in front of the `\begin{<proof-display-env>}` command.

```

233 \def\x{\csname phfthm@prfenv@#2@val@displayenvbegincmd\endcsname}%
234 \expandafter\x\expandafter{\phfthm@val@displayargs}%

```

And call the corresponding hook:

```

235 \csname phfthm@hookproof@#2@startafterdisplay\endcsname%

```

If required, `pin anchor` after the `proof-display-environment`. (`\phfPinProofAnchor` auto-destructs after first use, so it's safe to potentially call it a second time here). Then, call the corresponding hook.

```
236 \phfPinProofAnchor%
237 \expandafter\noexpand\csname phfthm@hookproof@#2@startlast\endcsname%
238 }%
```

Now, the definitions for the “end” part of the environment. Just call the relevant hooks and close the display environment.

```
239 {%
240 \expandafter\noexpand\csname phfthm@hookproof@#2@end\endcsname%
241 \csname phfthm@prfenv@#2@val@displayenvendcmd\endcsname
242 \expandafter\noexpand\csname phfthm@hookproof@#2@final\endcsname%
243 }%
```

Finally, define the default values of the proof-environment-specific hooks. These just call the corresponding global hooks (see [subsection 4.2](#)).

```
244 \csdef{phfthm@hookproof@#2@start}{\phfthm@hookproof@startcommon{#2}}%
245 \csdef{phfthm@hookproof@#2@startafterdisplay}{%
246 \phfthm@hookproof@startafterdisplaycommon{#2}}%
247 \csdef{phfthm@hookproof@#2@startlast}{\phfthm@hookproof@startlastcommon{#2}}%
248 \csdef{phfthm@hookproof@#2@end}{\phfthm@hookproof@endcommon{#2}}%
249 \csdef{phfthm@hookproof@#2@final}{\phfthm@hookproof@finalcommon{#2}}%
250 }
```

7.3.2 Common hooks for proofs

The hooks are documented in [subsection 4.2](#).

```
251 \def\phfthm@hookproof@startcommon#1{}
252 \def\phfthm@hookproof@startafterdisplaycommon#1{}
253 \def\phfthm@hookproof@startlastcommon#1{}
254 \def\phfthm@hookproof@endcommon#1{}
255 \def\phfthm@hookproof@finalcommon#1{}

```

7.3.3 Default display environment for proofs

`phfthm@proof@defaultdisplayenv` Provide an environment which displays a proof in a similar fashion as *AMS*, but with some small additional features.

```
256 \newenvironment{phfthm@proof@defaultdisplayenv}[1][\proofname]{%
257 \par
258 \pushQED{\qed}%
259 \normalfont \topsep6\p@\@plus6\p@\relax
260 \trivlist\item\relax
261 \phfPinProofAnchor
```

```

262 \phfthm@ProofTitleFmt{#1}%
263 \phfthm@ProofTitleHspace
264 \ignorespaces
265 }{%
266 \popQED\endtrivlist\@endpefalse
267 }

```

`\phfthm@ProofTitleFmt` These macros may be overridden to change the proof title appearance.
`\phfthm@ProofTitleHspace`

```

268 \def\phfthm@ProofTitleFmt#1{%
269   {\itshape #1.}%
270 }
271 \def\phfthm@ProofTitleHspace{%
272   \hspace{1.5ex plus 0.5ex minus 0.2ex}%
273 }

```

7.3.4 Parsing the proof argument

These macros parse the argument of the proof environment to see if it is of the form `*<some-label>` (see [section 5](#)).

`\phfthm@proof@parselabel` Call `\phfthm@proof@parselabel{<proof environment argument>}` to parse the argument string. This macro will set `\phfthm@val@proofoflabel` and `\phfthm@val@prooftitle` to appropriate values (respectively, the label name of the corresponding theorem and a representative title such as “Theorem 6”).

```

274 \def\phfthm@proof@parselabel#1{%
275   \phfthm@proof@parselabel@maybelabel#1\phfthm@proof@parselabel@END%
276 }

```

If the argument is `[*thm-label]` we want “Proof of *<ref to thm-label>*.”. If the argument is `[**thm-label]` we want “Proof.” but we’re informing the proofref-system that this is the proof of *<thm-label>* (e.g., to avoid undefined references to proofs). Otherwise if we get `[Some Custom Theorem]` then we want “Proof of Some Custom Theorem.”.

```

277 \def\phfthm@proof@parselabel@maybelabel{%
278   \@ifnextchar*\phfthm@proof@parselabel@label\phfthm@proof@parselabel@title%
279 }
280 \def\phfthm@proof@parselabel@label*{%
281   \@ifnextchar*\phfthm@proof@parselabel@labelsilent\phfthm@proof@parselabel@labelnormal%
282 }
283 \def\phfthm@proof@parselabel@labelsilent*#1\phfthm@proof@parselabel@END{%

```

The use of `\detokenize` here is a trick to make sure that all chars in the label text have a non-active category (e.g. we would have problems, e.g., if in the label “`thm:gauss`” the “`:`” is an active char—such as in French):

```

284 \edef\phfthm@val@proofoflabel{\detokenize{#1}}%

```

```

285 \def\phfthm@val@prooftitle{}%
286 }
287 \def\phfthm@proof@parselabel@labelnormal#1\phfthm@proof@parselabel@END{%
288 \edef\phfthm@val@proofoflabel{\detokenize{#1}}%
289 \def\phfthm@val@prooftitle{\phfthm@autoref{#1}}%
290 }
291 \def\phfthm@proof@parselabel@title#1\phfthm@proof@parselabel@END{%
292 \def\phfthm@val@proofoflabel{}%
293 \def\phfthm@val@prooftitle{#1}%
294 }

```

`\phfthm@proof@noparselabel` Enjoys the same syntax as `\phfthm@proof@parselabel`, i.e., it is a drop-in replacement for the latter, except that it invariably sets `\phfthm@val@proofoflabel` to an empty value and `\phfthm@val@prooftitle` to the argument itself. You could use this as a `parselabelcmd` macro if you didn't want to parse the label.

```

295 \def\phfthm@proof@noparselabel#1{%
296 \def\phfthm@val@proofoflabel{}%
297 \def\phfthm@val@prooftitle{#1}%
298 }

```

In order to look up what we are a proof of, we use `\autoref` provided by the `hyperref` package. Crucially, it is NOT the job of `phfthm` to load this package, and we should only use them if they are available (we don't want to depend on it). If it is not available, then we fall back to the regular `\ref` command.

```

299 \def\phfthm@autoref{\ref}
300 \AtBeginDocument{%
301 \@ifpackageloaded{hyperref}{\def\phfthm@autoref{\autoref}}{}%
302 }

```

7.4 Implementation of the proof-ref machinery

7.4.1 Small general stuff

`\proofonname` The macro `\proofonname` displays “Proof on ...” Here, #2 is the full page reference and #1 is the label name of the referenced theorem.

```

303 \providecommand\proofonname[2]{Proof on #2.}

```

`\proofrefsize` Format the proof reference “Proof on page ...”. This macro is meant to set the font size (or other font properties), but it may also be defined to take one argument, the proof reference text.

```

304 \def\proofrefsize{\footnotesize}

```


`\noproofref` Use `\noproofref` inside a theorem to signify that no proof reference should be attempted.

The implementation just defines `\phfthm@val@noproofref`. If this macro is defined, then no proof ref should be generated for the current thmlabel. Also, restore `\label` to its original definition in case it was overridden.

```
305 \def\noproofref{%
306   \def\phfthm@val@noproofref{1}%
307   \ifdefined\phfthm@old@label \let\label\phfthm@old@label \fi%
308 }
```

`\phfthm@proofref@warnnolabel` Produce a warning that no label was provided in order to infer the proof reference.

```
309 \def\phfthm@proofref@warnnolabel{%
310   \PackageWarning{phfthm}{No label provided for proof reference!}%
311 }
```

`\phfthm@proofref@expandthmlabeltoarg` Utility to expand the value of `\phfthm@val@thmlabel` as an argument to a callback command. #1 = the macro to relay the call to.

```
312 \def\phfthm@proofref@expandthmlabeltoarg#1{%
```

First, check if the proof-ref mechanism was explicitly temporarily disabled, and do nothing if that is the case.

```
313   \ifdefined\phfthm@val@noproofref\relax%
314   \else%
```

Then make sure `\phfthm@val@thmlabel` is defined (maybe empty), and then either call the callback macro #1 with the value of `\phfthm@val@thmlabel` as argument, or generate a warning if that value is empty.

```
315   \providecommand\phfthm@val@thmlabel{}%
316   \edef\phfthm@tmpa{\phfthm@val@thmlabel}%
317   \expandafter\notblank\phfthm@tmpa{%
318     \expandafter#1\phfthm@tmpa%
319   }{%
320     \phfthm@proofref@warnnolabel% no label provided
321   }%
322 \fi%
323 }
```

7.4.2 Utilities for interacting with `\autoref` labels

In this context, we also need some generic utilities for interacting with `\autoref` labels.

`\phfthm@autorefnameof` The macro `\phfthm@autorefnameof` extracts the name of the counter which generated this reference (e.g. “section” or “theorem”).

```
324 \def\phfthm@autorefnameof#1{%
```

Extract the counter part of the reference `section.NN`, which is 4th element in the `\r@label` macro. (Code extracted from `hyperref.sty`.)

```
325 \expandafter\ifx\csname r@#1\endcsname\relax%
326 \textbf{??}%
327 \else%
328 \expandafter\expandafter\expandafter\phfthm@HyPsd@autorefname%
329 \csname r@#1\endcsname{}{}{}{} \@nil%
330 \fi%
331 }
332 \def\phfthm@HyPsd@autorefname#1#2#3#4#5\@nil{%
333 \ifx\#4\%
334 \else%
335 \phfthm@HyPsd@@autorefname#4.\@nil%
336 \fi%
337 }
338 \def\phfthm@HyPsd@@autorefname#1.#2\@nil{%
339 \ltx@ifundefined{#1autorefname}{%
340 \ltx@ifundefined{#1name}{%
341 }{%
342 \csname#1name\endcsname%
343 }%
344 }{%
345 \csname#1autorefname\endcsname%
346 }%
347 }
```

`\phfthm@min@pageref` A minimal `pageref` macro, which just extracts the page number on which the given label is located.

The dark magic going on here is beyond me. The code was copied from `hyperref.sty`, in “`\def\HyPsd@@@pageref...`” and seems to work.

```
348 \def\phfthm@min@pageref#1{%
349 \ifcsname r@#1\endcsname%
350 \expandafter\expandafter\expandafter\expandafter
351 \expandafter\expandafter\expandafter\@car
352 \expandafter\expandafter\expandafter\@gobble
353 \csname r@#1\endcsname{} \@nil
354 \else%
355 0%
356 \fi%
357 }
```

7.4.3 Default proof-ref style, with basic machinery

Now we define the relevant callbacks for the default style. See documentation in subsection 5.3. Recall a proof-ref style just needs to define `\phfthm@proofrefstyle@<stylename>@setup`, which in turn should just define the callbacks `\phfthm@proofref@impl@start`, `\phfthm@proofref@impl@afterlabel` and `\phfthm@proofref@impl@end`. For our default style, these callbacks further call other callbacks of the form `\phfthm@proofref@impl@...`, such that these definitions can be re-used to create new styles. The main proof-ref generation routine is `\phfthm@proofrefstyle@default@main`, which can be used for either the `...@afterlabel` or the `...@end` callback.

```
\phfthm@proofrefstyle@default@-
fmt Format and display the proof reference. #1 = the theorem's label (e.g. prop:1);
#2 = the full reference (e.g. "page XYZ").
```

This macro is the default value of the callback `\phfthm@proofref@impl@fmt`, which is called by the default style itself.

Use correct spacing for right-aligning the reference.² If there is room on the current line, just right-align the proof-ref text; if not, add it on a separate line. [We can achieve this with the sequences `\hfil\null\hfil`: if there is space, it all fits on the same line, if not, the line breaks at the `\null` point.]

```
358 \def\phfthm@proofrefstyle@default@fmt#1#2{%
359   {\parfillskip=Opt\relax%
360    \hfil\null\hfil\null\hfil%
361    \hbox{\proofrefsize{\proofonname{#1}{#2}}}\par}%
362 }
```

```
\phfthm@proofrefstyle@default@-
fmtfarback These macros are the default values of the callbacks
\phfthm@proofrefstyle@default@- \phfthm@proofref@impl@fmtfarback, \phfthm@proofref@impl@fmtfarahead,
\phfthm@proofrefstyle@default@- and \phfthm@proofref@impl@fmtcloseby, which are called by the default
fmtcloseby style itself. These callbacks define how to format and (possibly not) display
the proof reference depending on whether the proof is "far behind" (several
pages back), "far ahead" (several pages ahead) or "close by" (neither far
back nor far ahead), as defined by \phfProofrefPageBackTolerance and
\phfProofrefPageAheadTolerance.
```

```
363 \def\phfthm@proofrefstyle@default@fmtfarback#1#2{%
364   \phfthm@proofref@impl@fmt{#1}{#2}}
365 \def\phfthm@proofrefstyle@default@fmtfarahead#1#2{%
366   \phfthm@proofref@impl@fmt{#1}{#2}}
367 \def\phfthm@proofrefstyle@default@fmtcloseby#1#2{}
```

```
\phfProofrefPageBackTolerance The macros \phfProofrefPageBackTolerance and
\phfProofrefPageAheadTolerance \phfProofrefPageAheadTolerance define how many pages back or
ahead the proof should be in order to consider it "far back" or "far ahead."
```

²Thanks <http://tex.stackexchange.com/a/43239/32188>!

Either value may be set to -1 to force the proof to be considered “far back” or “far ahead.”

```
368 \newcommand\phfProofrefPageBackTolerance{1}
369 \newcommand\phfProofrefPageAheadTolerance{1}
```

Define the internal counter which allows to check on which page we are at the place of the proof reference. This is used by `\phfthm@proofrefstyle@default@main`.

```
370 \newcounter{phfthmInternalProofrefCounter}
```

`\phfthm@proofrefstyle@default@main`

The main proof-ref generation routine. The argument #1 is the current label of the theorem; the referenced label is `proof:#1`.

```
371 \def\phfthm@proofrefstyle@default@main#1{%
```

Check to see if the proof is far away ahead or back (as defined by the tolerance macros above). Depending on each case, call the corresponding callbacks.³

```
372 \refstepcounter{phfthmInternalProofrefCounter}%
373 \label{internalproofref\thephfthmInternalProofrefCounter}%
374 \edef\phfthm@proofref@tmp@proofpage{\phfthm@min@pageref{proof:#1}}%
375 \edef\phfthm@proofref@tmp@thispage{%
376   \phfthm@min@pageref{internalproofref\thephfthmInternalProofrefCounter}}%
377 \edef\phfthm@proofref@tmp@pagediff{%
378   \the\numexpr\phfthm@proofref@tmp@proofpage-\phfthm@proofref@tmp@thispage\relax}%
```

If the proof is “far back,” call the corresponding callback.

```
379 \ifnum\numexpr\phfthm@proofref@tmp@pagediff\relax%
380   <\numexpr-\phfProofrefPageBackTolerance\relax%
381   \phfthm@proofref@impl@fmtfarback{#1}{\autopageref{proof:#1}}%
382 \else%
```

If the proof is “far ahead,” call the corresponding callback.

```
383 \ifnum\numexpr\phfthm@proofref@tmp@pagediff\relax%
384   >\numexpr\phfProofrefPageAheadTolerance\relax%
385   \phfthm@proofref@impl@fmtfarahead{#1}{\autopageref{proof:#1}}%
```

Otherwise, it is close by.

```
386 \else%
387   \phfthm@proofref@impl@fmtcloseby{#1}{\autopageref{proof:#1}}%
388 \fi%
389 \fi%
390 %% [\number\numexpr\phfthm@proofref@tmp@proofpage\relax} vs % DEBUG
```

³See <http://tex.stackexchange.com/a/2526> to test whether ref is on same page. Note that was problematic, probably due to hyperref. I needed to use my own `\phfthm@min@pageref` without any hyper linking mechanism in place.

```

391 %% \number\numexpr\phfthm@proofref@tmp@thispage\relax or % DEBUG
392 %% \number\numexpr1+\phfthm@proofref@tmp@thispage\relax] % DEBUG
393 }

```

`\phfthm@proofrefstyle@default@setup` The main set-up macro for the default proof-ref style. It sets all the call-backs to the default ones.

```

394 \def\phfthm@proofrefstyle@default@setup{%
395   \let\phfthm@proofref@impl@start\relax
396   \let\phfthm@proofref@impl@afterlabel\@gobble
397   \let\phfthm@proofref@impl@end\phfthm@proofrefstyle@default@main
398   \let\phfthm@proofref@impl@fmtfarback\phfthm@proofrefstyle@default@fmtfarback
399   \let\phfthm@proofref@impl@fmtfarahead\phfthm@proofrefstyle@default@fmtfarahead
400   \let\phfthm@proofref@impl@fmtcloseby\phfthm@proofrefstyle@default@fmtcloseby
401   \let\phfthm@proofref@impl@fmt\phfthm@proofrefstyle@default@fmt
402 }

```

7.4.4 Other proof-ref styles: `margin` and `marginbottom`

These styles simply use the same mechanism as the default style, but plug in different sub-callbacks.

Here’s a command that actually places the `marginpar`:

```

\phfthmproofref@placemarginpar
403 \newcommand\phfthmproofref@placemarginpar [2] {%
404   \marginpar [%
405     \raggedleft #1%
406   ] {%
407     \raggedright #2%
408   }%
409 }

```

`\phfthmproofref@placephantommarginpar` In the `marginbottom` proof-ref style, we place an invisible `\marginpar` at the top of the proof so that the proof-ref margin note will not be shifted too high when we try to align it to the bottom of the theorem statement. To work around a weird bug in Rev \TeX , where I get the error message ‘command sequence undefined: `\@capttype`’, I have found that a fix is to override this local invisible `\marginpar` to locally make sure the `\@capttype` is defined. The wild fix I’ve found—no guarantees, there are dark magic forces at work here—is the following, which you can include in your document preamble if you’re banging your head against this problem and you’re desperate:

```

\makeatletter
\renewcommand\phfthmproofref@placephantommarginpar{%
  \begingroup\csdef{@capttype}{figure}\marginpar{%
    \vspace{-\baselineskip}\rule{0pt}{0pt}}\endgroup%

```

```

}
\makeatother

```

```

410 \newcommand\phfthmproofref@placephantommarginpar{%
411   \leavevmode\marginpar{\vspace{-\baselineskip}\rule{\z@}{\z@}}%
412 }%

```

The margin style

`\phfthm@proofrefstyle@margin@setup` Set-up macro for the “margin” proof-ref style (displays the proof reference in the margin of the page).

```

413 \def\phfthm@proofrefstyle@margin@setup{%
414   \phfthm@proofrefstyle@default@setup

```

The proof reference should be displayed directly at the top, not at the end of the theorem, so plug in `\phfthm@proofref@default@main` onto `...@afterlabel` and not onto `...@end`. Don't forget that these macros accept one argument, the theorem label.

```

415   \let\phfthm@proofref@impl@afterlabel\phfthm@proofrefstyle@default@main
416   \let\phfthm@proofref@impl@end\@gobble

```

Define the formatting callback to put the note in the margin of the page using a `\marginpar`. We need `\leavevmode` to make sure it's aligned properly vertically with the paragraph.⁴

```

417   \def\phfthm@proofref@impl@fmt##1##2{%
418     \leavevmode\phfthm@proofref@placemarginpar{%
419       \proofrefsize{\proofonname{##1}{##2}}%
420     }{%
421       \proofrefsize{\proofonname{##1}{##2}}%
422     }%
423   }%
424 }

```

The marginbottom style

`\phfthm@proofrefstyle@marginbottom@setup` Set-up macro for the “marginbottom” proof-ref style (displays the proof reference in the margin of the page).

```

425 \def\phfthm@proofrefstyle@marginbottom@setup{%
426   \phfthm@proofrefstyle@default@setup

```

⁴See <http://tex.stackexchange.com/a/16161/32188>

The proof reference should be displayed directly at the bottom, not at the end of the theorem, so plug in `\phfthm@proofref@default@main` onto `...@end` and not onto `...@afterlabel`. Don't forget that these macros accept one argument, the theorem label. In `...@afterlabel` we place a dummy marginpar with an invisible rule one `\baselineskip` upwards, so that we force any proof-ref marginpar not to extend above the top of the theorem statement.

```
427 \def\phfthm@proofref@impl@afterlabel##1{%
428   \phfthmproofref@placephantommarginpar
429 }%
430 \let\phfthm@proofref@impl@end\phfthm@proofrefstyle@default@main
```

Define the formatting callback to put the note in the margin of the page using a `\marginpar`. We use a trick to align the margin note with the bottom of the paragraph.⁵

```
431 \def\phfthm@proofref@impl@fmt##1##2{%
432   \phfthmproofref@marginbottom@domarginpar{##1}{##2}%
433 }%
434 }
435 \def\phfthmproofref@marginbottom@domarginpar#1#2{%
436   \setbox\phfthm@tmp@boxa=\hbox to \marginparwidth{%
437     \parbox[b]{\marginparwidth}{\raggedleft\sloppy
438       \proofrefsize{\strut\proofonname{#1}{#2}\strut}}}%
439   \setbox\phfthm@tmp@box=\hbox to \marginparwidth{%
440     \parbox[b]{\marginparwidth}{\raggedright\sloppy
441       \proofrefsize{\strut\proofonname{#1}{#2}\strut}}}%
442   \phfthmproofref@placemarginpar{%
443     \box\phfthm@tmp@boxa%
444   }{%
445     \box\phfthm@tmp@box%
446   }%
447 }
448 \newbox\phfthm@tmp@box
449 \newbox\phfthm@tmp@boxa
```

7.5 Thmheading definition-like environments

7.5.1 Manually define a thmheading environment

Define the key-value options accepted by `\phfMakeThmheadingEnvironment`.

```
450 \define@cmdkey{phfthmmkthmheading}{thmstyle}{}
451 \define@cmdkey{phfthmmkthmheading}{internalcounter}{}

452 \newcounter{phfthmheadingcounter}%
```

⁵<https://tex.stackexchange.com/a/388018/32188>

`\phfMakeThmheadingEnvironment` Creates a new environment `\begin{thmheading}{Title}... \end{thmheading}` for customizing the heading on-the-fly (see documentation in [section 6](#)). Useful for an alternative formatting of definitions. The syntax is:

```
\phfMakeThmheadingEnvironment [<key-value options>] {<environment name>}
```

You can also use `\label` and `\ref` (the latter simply displays the given title).

```
453 \newcommand\phfMakeThmheadingEnvironment[2] [] {% }
```

Parse the options. First set defaults, and then parse the input string.

```
454 \def\cmdKV@phfthmmkthmheading@thmstyle{plain}%
455 \def\cmdKV@phfthmmkthmheading@internalcounter{phfthmheadingcounter}%
456 \setkeys{phfthmmkthmheading}{#1}%
```

And now, produce the relevant definitions:

```
457 \csdef{phfthm@thmheading@#2@val@title}{$\langle$No Title Given$\rangle$}%
458 \theoremstyle{\cmdKV@phfthmmkthmheading@thmstyle}%
```

We use `\newtheorem*` to create an unnumbered theorem. The fixed title is just a single token, the macro which will be set to the relevant title at the last moment.

```
459 \newtheorem*{phfthm@internal@thmheading@#2}{%
460 \csname phfthm@thmheading@#2@val@title\endcsname}%
```

Define the actual environment.

```
461 \newenvironment{#2}[1]{%
462 \csdef{phfthm@thmheading@#2@val@title}{##1}%
463 \letcs\thephfthmheadingcounter{phfthm@thmheading@#2@val@title}%
```

Relay call to the internal AMS-defined “theorem:”

```
464 \csname phfthm@internal@thmheading@#2\endcsname%
```

Pin down an anchor. The use of `\hspace*{0pt}` is explained at <http://tex.stackexchange.com/a/88493/32188> (see especially the first comment).

```
465 \hspace*{0pt}\refstepcounter{\cmdKV@phfthmmkthmheading@internalcounter}%
466 \csname phfthm@hook@thmheading@#2@start\endcsname{##1}%
```

Also, let’s add some flexibility in the hspace:

```
467 \hskip 0em plus 0.5em minus 0em%
468 \ignorespaces%
469 }%
```

Now, the END part of the environment: just call the callback and close the internal AMS-defined theorem.


```

470 {%
471   \csname phfthm@hook@thmheading@#2@end\endcsname%
472   \csname endphfthm@internal@thmheading@#2\endcsname%
473 }%

```

Also define the relevant callbacks, which just relay their calls to the default callbacks.

```

474 \csdef{phfthm@hook@thmheading@#2@start}##1{%
475   \phfthm@hook@thmheading@start{##1}}%
476 \csdef{phfthm@hook@thmheading@#2@end}{\phfthm@hook@thmheading@end}%
477 }

```

Provide as well the obsolete command `\phfthmMakeThmheadingEnvironment` which was provided in earlier versions of this package:

```

478 \def\phfthmMakeThmheadingEnvironment{\phfMakeThmheadingEnvironment}

```

`\phfthm@hook@thmheading@start` Global callbacks which are called for all thmheading-type environments defined with `\phfMakeThmheadingEnvironment` (unless their hooks have been changed in order for them not to call these global hooks).

```

479 \def\phfthm@hook@thmheading@start#1{}
480 \def\phfthm@hook@thmheading@end{}

```

7.6 Theorem sets

Here, we define the theorem sets proposed by the package for quick loading.

We first define the names. These are defined in any case regardless of whether we are loading a theorem set or of which theorem set we are loading.

```

481 \def\theoremname{Theorem}
482 \def\propositionname{Proposition}
483 \def\lemmaname{Lemma}
484 \def\corollaryname{Corollary}
485 \def\conjecturename{Conjecture}
486 \def\remarkname{Remark}
487 \def\definitionname{Definition}
488 \def\ideaname{Idea}
489 \def\questionname{Question}
490 \def\claimname{Claim}
491 \def\observationname{Observation}
492 \def\problemlistname{Problem}

```

As we define the theorem sets, remember the names in a comma-separated list which we can display in help text. The `\phfthm@def@thmset` replaces the `\def` command and expects the definitions to follow immediately.

```

493 \def\phfthm@def@thmset@optlist{}

```

```

494 \def\phfthm@def@thmset#1{%
495   \appto\phfthm@def@thmset@optlist{#1,}\csdef{phfthm@thmset@#1}}

```

`\phfthm@def@thmset@mktheorem` In definitions of theorem sets, use these macros to define a new theorem-like environment (theorem, proposition, corollary, etc.) or definition-like environment (definition, remark). The macros `\phfthm@val@mkthmoptarg@theorem` and `\phfthm@val@mkthmoptarg@defn` are defined by `\phfLoadThmSet`.

```

496 \def\phfthm@def@thmset@mktheorem{%
497   \expandafter\phfMakeTheorem\phfthm@val@mkthmoptarg@theorem}
498 \def\phfthm@def@thmset@mkdefn{%
499   \expandafter\phfMakeTheorem\phfthm@val@mkthmoptarg@defn}

```

The default set (empty name, or name “empty”) provides no theorem. (The first line uses `\def` directly so that we don’t include an empty item in the list of available choices.)

```

500 \def\phfthm@thmset@{}
501 \phfthm@def@thmset{empty}{}

```

Theorem set simple:

```

502 \phfthm@def@thmset{simple}{
503   \phfthm@def@thmset@mktheorem{theorem}{\theoremname}
504   \phfthm@def@thmset@mktheorem{proposition}{\propositionname}
505   \phfthm@def@thmset@mktheorem{lemma}{\lemmaname}
506   \phfthm@def@thmset@mktheorem{corollary}{\corollaryname}
507   \phfthm@def@thmset@mkdefn{definition}{\definitionname}
508 }

```

Theorem set default:

```

509 \phfthm@def@thmset{default}{
510   \phfthm@def@thmset@mktheorem{theorem}{\theoremname}
511   \phfthm@def@thmset@mktheorem{proposition}{\propositionname}
512   \phfthm@def@thmset@mktheorem{lemma}{\lemmaname}
513   \phfthm@def@thmset@mktheorem{corollary}{\corollaryname}
514   \phfthm@def@thmset@mktheorem{conjecture}{\conjecturename}
515   \phfthm@def@thmset@mktheorem{remark}{\remarkname}
516   \phfthm@def@thmset@mkdefn{definition}{\definitionname}
517 }

```

Theorem set shortnames:

```

518 \phfthm@def@thmset{shortnames}{
519   \phfthm@def@thmset@mktheorem{thm}{\theoremname}
520   \phfthm@def@thmset@mktheorem{prop}{\propositionname}
521   \phfthm@def@thmset@mktheorem{lem}{\lemmaname}
522   \phfthm@def@thmset@mktheorem{cor}{\corollaryname}
523   \phfthm@def@thmset@mktheorem{conj}{\conjecturename}
524   \phfthm@def@thmset@mktheorem{rem}{\remarkname}

```

```
525 \phfthm@def@thmset@mkdefn{defn}{\definitionname}
526 }
```

Theorem set `rich`. Add definitions to the default set:

```
527 \phfthm@def@thmset{rich}{
528   \phfthm@thmset@default
529   \phfthm@def@thmset@mktheorem{idea}{\ideaname}
530   \phfthm@def@thmset@mktheorem{question}{\questionname}
531   \phfthm@def@thmset@mktheorem{claim}{\claimname}
532   \phfthm@def@thmset@mktheorem{observation}{\observationname}
533   \phfthm@def@thmset@mktheorem{problem}{\problemname}
534 }
```

`\phfLoadThmSet` The macro `\phfLoadThmSet` loads a theorem set. See documentation at [subsection 3.2](#).

`#1` = options to `\phfMakeTheorem` for theorem-like environments

`#2` = options to `\phfMakeTheorem` for definition-like environments

`#3` = name of the theorem set to load

```
535 \newcommand\phfLoadThmSet[3]{%
536   \ifcsname phfthm@thmset@#3\endcsname%
537   \edef\phfthm@val@mkthmoptarg@theorem{#1}%
538   \edef\phfthm@val@mkthmoptarg@defn{#2}%
539   \csname phfthm@thmset@#3\endcsname%
540   \else%
541     \PackageWarning{phfthm}{Unknown theorem set: '#3'!}%
542   \fi%
543 }
```

For compatibility with my earlier versions of `phfthm`, also provide the obsolete `\phfthmLoadThmSet`:

```
544 \def\phfthmLoadThmSet{\phfLoadThmSet}
```

7.7 Package option handling

The machinery is in place, now define and parse the package options.

7.7.1 Declaring the package options

The package options all use the `keyval` parsing mechanism using the `xkeyval` package.

Recall when using `\define@XXXkey` that the optional argument after the second mandatory argument is the value which is assumed if the key is given with no explicit value; it is not the initial default value.

The `resetstyle` package option

An option to reset all options so that the package provides only stand-alone definitions and is not invasive (see [section 2](#)).

This option does not expect any argument (i.e., you should specify `\usepackage[resetstyle,<other options>]{phfthm}`, and not `\usepackage[resetstyle=true,<other options>]{phfthm}`).

```
545 \define@key{phfthmpkg}{resetstyle}[]{}%
546 \KV@phfthmpkg@smallproofsfalse%
547 \KV@phfthmpkg@qedsymbolblacksquarefalse%
548 \KV@phfthmpkg@prooftitleitbfbfalse%
549 \KV@phfthmpkg@sepcountersfalse%
550 \KV@phfthmpkg@countpersectionfalse%
551 \KV@phfthmpkg@proofreffalse%
552 \if\relax\detokenize{#1}\relax\else%
553   \PackageError{phfthm}{'resetstyle' does not take any argument.}{You
554     specified the 'resetstyle' argument and provided a value to it
555     ('resetstyle=...'). However the 'resetstyle' option does not accept
556     any value argument.}
557 \fi%
558 }
```

Options for loading theorem sets

Define the various package options for the loading of predefined theorem sets ([subsection 2.1](#)).

The `sepcounters` option, off by default.

```
559 \define@boolkey{phfthmpkg}{sepcounters}[true]{}
560 \KV@phfthmpkg@sepcountersfalse
```

The `countpersection` option, off by default.

```
561 \define@boolkey{phfthmpkg}{countpersection}[true]{}
562 \KV@phfthmpkg@countpersectionfalse
```

The `proofref` option. The proof-ref is on initially by default.

```
563 \newif\ifKV@phfthmpkg@proofref
564 \KV@phfthmpkg@proofreftrue
565 \def\cmdKV@phfthmpkg@proofref@style{default}
```

Actually define the option itself. Here we do some customized parsing of the value of the `proofref=...` option, to treat the cases `proofref=` (empty argument) and `proofref=false` separately.

```
566 \define@key{phfthmpkg}{proofref}[]{}%
567 \ifblank{#1}{}%
```

If a blank argument provided, set some sensible defaults with `proofref` on:

```
568 \KV@phfthmpkg@proofreftrue%
569 \def\cmdKV@phfthmpkg@proofref@style{default}%
570 }{%
```

Otherwise, check to see if the value is false, in which case deactivate the proof-ref mechanism, or else, activate it and set the given style value as documented in [subsection 2.1](#).

```
571 \ifstrequal{#1}{false}{%
572   \KV@phfthmpkg@proofreffalse%
573 }{%
574   \KV@phfthmpkg@proofreftrue%
575   \def\cmdKV@phfthmpkg@proofref@style{#1}%
576 }%
577 }%
578 }
```

The `thmset` option. We subtly construct the command `\define@choicakey{phfthmpkg}{thmset}[\val]{\phfthm@def@thmset@optlist}`, but with the last macro (option list) expanded.

```
579 \def\@tmpa{\define@choicakey{phfthmpkg}{thmset}[\val]}
580 \edef\@tmpb{\phfthm@def@thmset@optlist}
581 \expandafter\@tmpa\@tmpb{%
582   \xdef\cmdKV@phfthmpkg@thmset{\val}%
583 }
```

By default we should load the default set.

```
584 \def\cmdKV@phfthmpkg@thmset{default}
```

The options `theoremstyle` and `definitionstyle` set which theorem style to use for theorems and definitions, when loading the given `thmset`.

```
585 \define@cmdkey{phfthmpkg}{theoremstyle}{}
586 \def\cmdKV@phfthmpkg@theoremstyle{plain}
587 \define@cmdkey{phfthmpkg}{definitionstyle}{}
588 \def\cmdKV@phfthmpkg@definitionstyle{definition}
```

Proof environment options

Define the package options `proofenv`, `smallproofs`, `qedsymbolblacksquare`, and `prooftitleitbf` ([subsection 2.2](#)).

```
589 \define@boolkey{phfthmpkg}{proofenv}[true]{}
590 \define@boolkey{phfthmpkg}{smallproofs}[true]{}
591 \define@boolkey{phfthmpkg}{qedsymbolblacksquare}[true]{}
592 \define@boolkey{phfthmpkg}{prooftitleitbf}[true]{}

```

Set the initial default values for these options.

```
593 \KV@phfthmpkg@smallproofstrue
594 \KV@phfthmpkg@qedsymbolblacksquaretrue
595 \KV@phfthmpkg@proofenvtrue
596 \KV@phfthmpkg@prooftitleitbfalse
```

Options for a theorem-like heading environment

Define the `thmheading` and `thmheadingstyle` package options, documented in [section 6](#).

```
597 \define@boolkey{phfthmpkg}{thmheading}[true]{}
598 \define@cmdkey{phfthmpkg}{thmheadingstyle}{}
```

The `thmheading` environment is provided by default; it's a stand-alone definition anyway. The style defaults to the plain style.

```
599 \KV@phfthmpkg@thmheadingtrue
600 \def\cmdKV@phfthmpkg@thmheadingstyle{plain}
```

7.7.2 Parsing the package options

The usual stuff (xkeyval-flavored).

```
601 \DeclareOptionX*{%
602   \PackageWarning{phfthm}{Invalid option: '\CurrentOption'}%
603 }
604 \ProcessOptionsX<phfthmpkg>
```

7.7.3 Execute package options-controlled actions

Loading a theorem set

First, we need to take into account the options which alter the way the theorem sets will be loaded (separate counters, proof-ref, etc.).

Take care of the proof-ref stuff. First, define the possible styles (note that these are not the same as the values to the `proofrefstyle` argument to the `\phfMakeTheorem` command).

```
605 \def\phfthm@val@mkthmopt@proofrefstyle{}
606 \ifKV@phfthmpkg@proofref
607   \def\phfthm@proofref@style@default{}
608   \def\phfthm@proofref@style@{}
```

Note that `proofref=always` and `proofref=onlyifveryfar` have a global effect, because they set `\phfProofrefPageBackTolerance` and `\phfProofrefPageAheadTolerance` (see documentation in [subsection 2.1](#)).

```

609 \def\phfthm@proofref@style@always{
610   \def\phfProofrefPageBackTolerance{-1}
611   \def\phfProofrefPageAheadTolerance{-1}
612 }
613 \def\phfthm@proofref@style@onlyifveryfar{
614   \def\phfProofrefPageBackTolerance{2}
615   \def\phfProofrefPageAheadTolerance{4}
616 }
617 \def\phfthm@proofref@style@margin{
618   \def\phfthm@val@mkthmopt@proofrefstyle{proofrefstyle=margin}
619 }
620 \def\phfthm@proofref@style@marginbottom{
621   \def\phfthm@val@mkthmopt@proofrefstyle{proofrefstyle=marginbottom}
622 }
623 \def\phfthm@proofref@style@longref{

```

For longref: by setting `\proofonname` globally, this option can be combined with other styles. But then we also change the default style formatting to avoid ugly line breaks.

```

624   \def\proofonname##1##2{The proof of this \phfthm@autorefnameof{##1} can
625     be found on ##2.}
626   \def\phfthm@proofrefstyle@default@fmt##1##2{%
627     \par{\raggedleft\proofrefsize{(\proofonname{##1}{##2})}\par}%
628   }
629 }
630 \def\phfthm@proofref@style@off{
631   \def\phfthm@val@mkthmopt@proofrefstyle{proofref=false}
632 }

```

Now execute the given styles. Construct the command `\phfthm@internal@execattribs{phfthm@proofref@style@}{ProofRef Style}``\cmdKV@phfthmpkg@proofref@style}`, but with the last macro expanded.

```

633 \def\x{%
634   \phfthm@internal@execattribs{phfthm@proofref@style@}{ProofRef Style}}
635 \expandafter\x\expandafter{\cmdKV@phfthmpkg@proofref@style}

```

Prepare the proof-ref option in case we don't want any proof reference mechanism on.

Changed in v1.1 [2019/03/12]: Bug fix: package option `proofref=false` didn't have any effect.

```

636 \else
637   \def\phfthm@val@mkthmopt@proofrefstyle{proofref=false}
638 \fi

```

Take care of counters. In any case, define a common counter, in case we use a common counter for all theorem types. (The counter is defined in any case, to

avoid breaking other code which might use it if suddenly the user decides to use `sepcounters=true` for their document.)

```
639 \newcounter{phfthmcounter}
640 \setcounter{phfthmcounter}{0}
```

Prepare an argument to `\phfMakeTheorem` according to the `countpersection` and `sepcounters` options. First determine the parent counter in case `countpersection` is set (chapter if chapter counter exists, else section).

```
641 \ifKV@phfthmpkg@countpersection
642   \ifcsname c@chapter\endcsname
643   \def\phfthm@tmp@parentcountername{chapter}
644   \else
645   \def\phfthm@tmp@parentcountername{section}
646   \fi
647 \fi
648 \ifKV@phfthmpkg@sepcounters
649   \ifKV@phfthmpkg@countpersection
650   \edef\phfthm@val@mkthmopt@counteropts{parentcounter=\phfthm@tmp@parentcountername}
651   \else
652   \def\phfthm@val@mkthmopt@counteropts{}
653   \fi
654 \else
655   \def\phfthm@val@mkthmopt@counteropts{counter=phfthmcounter}
656   \ifKV@phfthmpkg@countpersection
657   \numberwithin{phfthmcounter}{\phfthm@tmp@parentcountername}
658   \fi
659 \fi
```

Finally, load the theorem set defined by the options. The first argument regroups the options for theorem environments (Theorem, Proposition, Corollary, ...); the second argument regroups the options for definition environments (Definition); the third argument is the theorem set name itself.

```
660 \phfLoadThmSet%
661 {[\phfthm@val@mkthmopt@counteropts,\phfthm@val@mkthmopt@proofrefstyle,
662  thmstyle=\cmdKV@phfthmpkg@theoremstyle]}%
663 {[\phfthm@val@mkthmopt@counteropts,proofref=false,
664  thmstyle=\cmdKV@phfthmpkg@definitionstyle]}%
665 {\cmdKV@phfthmpkg@thmset}
```

(Note the absence of the proof-ref for definitions.)

Define the proof environment

If requested, define the proof environment (subsection 2.2). First, make sure we take into account the options `smallproofs`, `qedsymbolblacksquare` and


```
prooftitleitbf.
```

```
666 \def\phfthm@pkgopterr@require@proofenv#1{%
667   \ifKV@phfthmpkg@proofenv\else%
668     \PackageError{phfthm}{Option '#1' depends on 'proofenv=true'}%
669   \fi
670 }
671 \ifKV@phfthmpkg@smallproofs
672   %\phfthm@pkgopterr@require@proofenv{smallproofs}
673   \apptocmd\phfthm@hookproof@startcommon{%
674     \def\baselinestretch{1.2}\footnotesize}{\%
675     Failed to change command \string\phfthm@hook@start@proof}
676 \fi
677 \ifKV@phfthmpkg@qedsymbolblacksquare
678   %\phfthm@pkgopterr@require@proofenv{qedsymbolblacksquare}
679   \RequirePackage{amssymb}
680   \providecommand\filledsquare{\ensuremath{\blacksquare}}
681   \renewcommand\qedsymbol{\text{\tiny\ensuremath{\filledsquare}}}
682 \fi
683 \ifKV@phfthmpkg@prooftitleitbf
684   %\phfthm@pkgopterr@require@proofenv{prooftitleitbf}
685   \def\phfthm@ProofTitleFmt#1{{\itshape\bfseries#1.}}
686 \fi
```

Go ahead and define the proof environment. Because we have already loaded `amsthm`, we need to override the existing proof environment.

```
687 \ifKV@phfthmpkg@proofenv
688   \phfMakeProofEnv[override=true]{proof}
689 \fi
```

Define the theorem-heading environment

Define the `thmheading` environment, if requested.

```
690 \ifKV@phfthmpkg@thmheading
691   \phfMakeThmheadingEnvironment%
692     [thmstyle=\cmdKV@phfthmpkg@thmheadingstyle]{thmheading}
693 \fi
```

Change History

v1.0	General: Initial version	1
v1.1	General: Bug fix: package option <code>proofref=false</code> didn't have any effect	47

Index

Numbers written in *italics* 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	B
* (cmd. opt.)	12
\@car	351
\@endpefalse	266
\@for	7
\@gobble	352, 396, 416
\@ifnextchar	278, 281
\@ifpackageloaded	301
\@nil	329, 332, 335, 338, 353
\@plus	259
\@tmpa	579, 581
\@tmpb	580, 581
\@	333
{ } (pkg. opt.)	5
A	C
aliascnt	8, 20
\aliascntresetthe	50
aliascounter (cmd. opt.)	8
always (pkg. opt.)	5, 6
amsmath	20
amsthm	2, 4, 6, 7, 10, 12, 20, 25, 49
\appto	495
\apptocmd	673
\AtBeginDocument	300
\autopageref	381, 385, 387
\autoref	301
babel	10, 13
\baselineskip	411
\baselinestretch	674
\begin	57, 92, 204
\begingroup	58, 93
\bfseries	685
\blacksquare	680
\box	443, 445
claim (environment)	4
\claimname	10, 490, 531
\cmdKV@phfmkprf@defaultproofname	150, 161, 167, 170
\cmdKV@phfmkprf@displayenv	149, 157, 158, 160, 199
\cmdKV@phfmkprf@internalcounter	151, 198
\cmdKV@phfmkprf@parselabelcmd	154, 174
\cmdKV@phfmkprf@proofofname	152, 188
\cmdKV@phfmkthm@counter	25, 38, 48, 53
\cmdKV@phfmkthm@parentcounter	30, 39, 42
\cmdKV@phfmkthm@proofrefstyle	31, 68

<code>\cmdKV@phfmkthm@thmstyle</code>	26, 33, 35	
<code>\cmdKV@phfthmmkthmheading@internalcounter</code>	455, 465	
<code>\cmdKV@phfthmmkthmheading@thmstyle</code>	454, 458	
<code>\cmdKV@phfthmpkg@definitionstyle</code>	588, 664	
<code>\cmdKV@phfthmpkg@proofref@style</code>	565, 569, 575, 635	
<code>\cmdKV@phfthmpkg@theoremstyle</code>	586, 662	
<code>\cmdKV@phfthmpkg@thmheadingstyle</code>	600, 692	
<code>\cmdKV@phfthmpkg@thmset</code>	582, 584, 665	
command options:		
<code>*</code>	12	
<code>aliascounter</code>	8	
<code>counter</code>	8, 22	
<code>default</code>	9	
<code>defaultproofname</code>	12	
<code>defnostar</code>	8	
<code>defstar</code>	9	
<code>displayenv</code>	12, 13, 28	
<code>internalcounter</code>	13, 19, 28	
<code>longref</code>	9	
<code>margin</code>	9	
<code>marginbottom</code>	9	
<code>override</code>	13, 28	
<code>parselabel</code>	12	
<code>parselabelcmd</code>	12, 32	
<code>proofenvstyle</code>	18	
<code>proofofname</code>	13	
<code>proofref</code>	9, 11, 18	
<code>proofrefstyle</code>	9	
<code>thmstyle</code>	8, 19	
<code>true</code>	8, 11, 18	
<code>conjecture</code> (environment)	4	
<code>\conjecturename</code>	10, 485, 514, 523	
<code>corollary</code> (environment)	4	
<code>\corollaryname</code>	10, 484, 506, 513, 522	
<code>counter</code> (cmd. opt.)	8, 22	
<code>countpersection</code> (pkg. opt.)	5, 44, 48	
<code>\csdef</code>	45, 51, 101, 102, 162, 163, 189, 200, 201, 244, 245, 247, 248, 249, 457, 462, 474, 476, 495	
<code>\csedef</code>	65, 74, 81, 166, 168, 172, 197, 203, 205	
<code>\cslet</code>	160, 188	
<code>\CurrentOption</code>	602	
		D
<code>\DeclareOptionX</code>	601	
<code>default</code> (cmd. opt.)	9	
<code>default</code> (pkg. opt.)	4, 5	
<code>defaultproofname</code> (cmd. opt.)	12	
<code>\define@boolkey</code>	16, 18, 19, 20, 142, 145, 559, 561, 589, 590, 591, 592, 597	
<code>\define@choicekey</code>	579	
<code>\define@cmdkey</code>	15, 17, 21, 22, 140, 141, 143, 144, 146, 450, 451, 585, 587, 598	
<code>\define@key</code>	545, 566	
<code>definition</code> (environment)	4	
<code>\definitionname</code>	10, 487, 507, 516, 525	
<code>definitionstyle</code> (pkg. opt.)	4, 45	
<code>defnostar</code> (cmd. opt.)	8	
<code>defstar</code> (cmd. opt.)	9	
<code>\detokenize</code>	33, 38, 39, 161, 178, 191, 199, 218, 284, 288, 552	
<code>displayenv</code> (cmd. opt.)	12, 13, 28	
<code>\do</code>	7	
		E
<code>empty</code> (pkg. opt.)	4	
<code>\end</code>	63, 98, 206	
<code>\endgroup</code>	62, 97	
<code>\endphfthm@old@proof</code>	133	
<code>\endproof</code>	134	
<code>\endtrivlist</code>	266	
<code>\ensuremath</code>	680, 681	
environments:		
<code>claim</code>	4	
<code>conjecture</code>	4	
<code>corollary</code>	4	
<code>definition</code>	4	
<code>idea</code>	4	
<code>lemma</code>	4	
<code>observation</code>	4	
<code>phfthm@proof@defaultdisplayenv</code>	256	
<code>problem</code>	4	
<code>proof</code>	6	
<code>proposition</code>	4	
<code>question</code>	4	
<code>remark</code>	4	
<code>theorem</code>	4	
<code>thmheading</code>	7	
<code>\expandonce</code>	42, 167, 170, 174	

F	
false (pkg. opt.)	5, 6
\filledsquare	680, 681
\footnotesize	304, 674
G	
\global	231
H	
\hbox	361, 436, 439
\hfil	360
\hskip	467
\hspace	272, 465
hyperref	8, 32
I	
idea (environment)	4
\ideaname	10, 488, 529
IEEEtran	12
\ifblank	567
\ifcsname	8, 349, 536, 642
\ifdefined	118, 307, 313
\ifKV@phfmkprf@override	209
\ifKV@phfmkprf@parselabel	173
\ifKV@phfmkthm@aliascounter	47
\ifKV@phfmkthm@defnostar	37
\ifKV@phfmkthm@defstar	89
\ifKV@phfmkthm@proofref	66, 75, 82
\ifKV@phfthmpkg@countpersection	641, 649, 656
\ifKV@phfthmpkg@proofenv	667, 687
\ifKV@phfthmpkg@proofref	563, 606
\ifKV@phfthmpkg@prooftitleitbf	683
\ifKV@phfthmpkg@qedsymbolblacksquare	677
\ifKV@phfthmpkg@sepcounters	648
\ifKV@phfthmpkg@smallproofs	671
\ifKV@phfthmpkg@thmheading	690
\ifnum	379, 383
\ifstrequal	571
\ignorespaces	131, 264, 468
internalcounter (cmd. opt.)	13, 19, 28
\item	260
\itshape	269, 685
K	
\KV@phfmkprf@overridefalse	148
\KV@phfmkprf@parselabeltrue	153
\KV@phfmkthm@aliascountertrue	24
\KV@phfmkthm@defnostartrue	27
\KV@phfmkthm@defstartrue	28
\KV@phfmkthm@proofreftrue	29
\KV@phfthmpkg@countpersectionfalse	550, 562
\KV@phfthmpkg@proofenvtrue	595
\KV@phfthmpkg@proofreffalse	551, 572
\KV@phfthmpkg@proofreftrue	564, 568, 574
\KV@phfthmpkg@prooftitleitbffalse	548, 596
\KV@phfthmpkg@qedsymbolblacksquarefalse	547
\KV@phfthmpkg@qedsymbolblacksquaretrue	594
\KV@phfthmpkg@sepcountersfalse	549, 560
\KV@phfthmpkg@smallproofsfalse	546
\KV@phfthmpkg@smallproofstrue	593
\KV@phfthmpkg@thmheadingtrue	599
L	
\label	122, 123, 129, 193, 307, 373
\langle	457
\leavevmode	411, 418
lemma (environment)	4
\lemmaname	10, 483, 505, 512, 521
\letcs	216, 463
longref (cmd. opt.)	9
longref (pkg. opt.)	5
\ltx@ifundefined	339, 340
M	
margin (cmd. opt.)	9
margin (pkg. opt.)	5
marginbottom (cmd. opt.)	9
marginbottom (pkg. opt.)	5
\marginpar	404, 411
\marginparwidth	436, 437, 439, 440
N	
\newaliascnt	48
\newbox	448, 449
\newcounter	137, 370, 452, 639
\newif	563
\newtheorem	40, 42, 49, 53, 90, 459
\next	7, 8, 9, 11
\noexpand	40, 42, 67, 69, 70, 72, 76, 77, 79, 83, 84, 86, 123, 169, 176, 178,

179, 180, 181, 182, 183, 184,	
186, 198, 204, 206, 237, 240, 242	
<code>\noproofref</code>	17, 305
<code>\normalfont</code>	259
<code>\notblank</code>	317
<code>\null</code>	360
<code>\number</code>	390, 391, 392
<code>\numberwithin</code>	657
<code>\numexpr</code>	378,
379, 380, 383, 384, 390, 391, 392	
O	
observation (environment)	4
<code>\observationname</code>	491, 532
off (pkg. opt.)	5
onlyifveryfar (pkg. opt.)	6
onyifveryfar (pkg. opt.)	5
override (cmd. opt.)	13, 28
P	
<code>\p@</code>	259
package options:	
<code>{}</code>	5
always	5, 6
countpersection	5, 44, 48
default	4, 5
definitionstyle	4, 45
empty	4
false	5, 6
longref	5
margin	5
marginbottom	5
off	5
onlyifveryfar	6
onyifveryfar	5
parentcounter	5
proofenv	6, 45
proofref	5, 6, 44
prooftitleitbf	6, 45, 49
qedsymbolblacksquare ..	6, 45, 48
resetstyle	3, 44
rich	4
sepcounters	5, 44, 48
shortnames	4
simple	4
smallproofs	6, 45, 48
theoremstyle	4, 45
thmheading	7, 46
thmheadingstyle	7, 46
thmset	3–6, 45
true	6, 7
<code>\PackageError</code>	553, 668
packages:	
aliascnt	8, 20
amsmath	20
amsthm ...	2, 4, 6, 7, 10, 12, 20, 25, 49
babel	10, 13
hyperref	8, 32
IEEEtran	12
phfnote	20
phfqitlx	1
phfthm 1–4, 6, 7, 11, 12, 15, 19, 32, 43	
xkeyval	43, 46
<code>\PackageWarning</code> 11, 119, 310, 541, 602	
<code>\par</code>	257, 361, 627
<code>\parbox</code>	437, 440
parentcounter (pkg. opt.)	5
<code>\parfillskip</code>	359
parselabel (cmd. opt.)	12
parselabelcmd (cmd. opt.)	12, 32
<code>\phfLoadThmSet</code>	9, 535, 660
<code>\phfMakeProofEnv</code>	12, 147, 688
<code>\phfMakeTheorem</code>	7, 23, 497, 499
<code>\phfMakeThmheadingEnvironment</code>	19, 453, 691
<code>\phfmkprf@tmp@star</code>	156, 157
phfnote	20
<code>\phfPinProofAnchor</code>	
.....	14, 229, 231, 232, 236, 261
<code>\phfProofrefPageAheadTolerance</code>	368, 384, 611, 615
<code>\phfProofrefPageBackTolerance</code>	368, 380, 610, 614
phfqitlx	1
phfthm	1–4, 6, 7, 11, 12, 15, 19, 32, 43
<code>\phfthm@autoref</code>	289, 299, 301
<code>\phfthm@autorefnameof</code>	324, 624
<code>\phfthm@def@label@thmlabel</code> 69, 117	
<code>\phfthm@def@thmset</code>	
.....	494, 501, 502, 509, 518, 527
<code>\phfthm@def@thmset@mkdefn</code>	
.....	496, 507, 516, 525
<code>\phfthm@def@thmset@mktheorem</code> .	
.....	496, 503, 504,
505, 506, 510, 511, 512, 513,	
514, 515, 519, 520, 521, 522,	
523, 524, 529, 530, 531, 532, 533	
<code>\phfthm@def@thmset@optlist</code> ...	
.....	493, 495, 580
<code>\phfthm@hook@afterlabel@thmname</code>	11

<code>\phfthm@hook@afterlabelcommon</code>	<code>\phfthm@proof@noparselabel</code>	176, 295
.....	<code>\phfthm@proof@parselabel</code>	154, 274
<code>\phfthm@hook@end@thmname</code>	<code>\phfthm@proof@parselabel@END</code>	.
.....	275, 283, 287, 291
<code>\phfthm@hook@end@thmname*</code>	<code>\phfthm@proof@parselabel@label</code>	
.....	278, 280
<code>\phfthm@hook@endcommon</code>	<code>\phfthm@proof@parselabel@labelnormal</code>	
.....	281, 287
<code>\phfthm@hook@endcommonnostar</code>	<code>\phfthm@proof@parselabel@labelsilent</code>	
.....	281, 283
<code>\phfthm@hook@endcommonstar</code>	<code>\phfthm@proof@parselabel@maybelabel</code>	
.....	275, 277
<code>\phfthm@hook@start@proof</code>	<code>\phfthm@proof@parselabel@title</code>	
.....	278, 291
<code>\phfthm@hook@start@thmname</code>	<code>\phfthm@proofref@expandthmlabeltoarg</code>	
.....	76, 83, 312
<code>\phfthm@hook@start@thmname*</code>	<code>\phfthm@proofref@impl@afterlabel</code>	
.....	77, 396, 415, 427
<code>\phfthm@hook@startcommon</code>	<code>\phfthm@proofref@impl@end</code>	
.....	84, 397, 416, 430
<code>\phfthm@hook@startcommonnostar</code>	<code>\phfthm@proofref@impl@fmt</code>	
.....	364, 366, 401, 417, 431
<code>\phfthm@hook@startcommonstar</code>	<code>\phfthm@proofref@impl@fmtcloseby</code>	
.....	387, 400
<code>\phfthm@hook@thmheading@end</code>	<code>\phfthm@proofref@impl@fmtfarahead</code>	
.....	385, 399
<code>\phfthm@hook@thmheading@start</code>	<code>\phfthm@proofref@impl@fmtfarback</code>	
.....	381, 398
<code>\phfthm@hookproof@...@end</code>	<code>\phfthm@proofref@impl@start</code>	
.....	70, 395
<code>\phfthm@hookproof@...@final</code>	<code>\phfthm@proofref@style@</code>	
.....	608
<code>\phfthm@hookproof@...@start</code>	<code>\phfthm@proofref@style@always</code>	
.....	609
<code>\phfthm@hookproof@...@startafterdisplay</code>	<code>\phfthm@proofref@style@default</code>	
.....	607
<code>\phfthm@hookproof@...@startlast</code>	<code>\phfthm@proofref@style@longref</code>	
.....	623
<code>\phfthm@hookproof@endcommon</code>	<code>\phfthm@proofref@style@margin</code>	
.....	617
<code>\phfthm@hookproof@finalcommon</code>	<code>\phfthm@proofref@style@marginbottom</code>	
.....	620
<code>\phfthm@hookproof@startafterdisplaycommon</code>	<code>\phfthm@proofref@style@off</code>	
.....	630
<code>\phfthm@hookproof@startcommon</code>	<code>\phfthm@proofref@style@onlyifveryfar</code>	
.....	613
<code>\phfthm@hookproof@startlastcommon</code>	<code>\phfthm@proofref@tmp@pagediff</code>	
.....	377, 379, 383
<code>\phfthm@HyPsd@autorefname</code>	<code>\phfthm@proofref@tmp@proofpage</code>	
335, 338	374, 378, 390
<code>\phfthm@HyPsd@autorefname</code>	<code>\phfthm@proofref@tmp@thispage</code>	
328, 332	375, 378, 391, 392
<code>\phfthm@internal@execattribs</code>	<code>\phfthm@proofref@warnnolabel</code>	
6, 634	309, 320
<code>\phfthm@min@pageref</code>		
348, 374, 376		
<code>\phfthm@NOPROOFARG</code>		
138, 139, 210		
<code>\phfthm@old@label</code>		
.....		
118, 122, 128, 129, 307		
<code>\phfthm@old@proof</code>		
.....		
133		
<code>\phfthm@pkgopterr@require@proofenv</code>		
.....		
666, 672, 678, 684		
<code>phfthm@proof@defaultdisplayenv</code>		
(environment)		
.....		
256		

<code>\phfthm@proofrefstyle@default@fmt</code>	<code>\phfthmproofref@marginbottom@domarginpar</code>
..... 358, 401, 626 432, 435
<code>\phfthm@proofrefstyle@default@fmtclose</code>	<code>\phfthmproofref@placemarginpar</code>
..... 363, 400 403, 418, 442
<code>\phfthm@proofrefstyle@default@fmtfarleft</code>	<code>\phfthmproofref@placephantommarginpar</code>
..... 363, 399 410, 428
<code>\phfthm@proofrefstyle@default@fmtfarright</code>	<code>\pushQED</code>
..... 363, 398 266
<code>\phfthm@proofrefstyle@default@main</code>	<code>\postdisplaypenalty</code>
..... 371, 397, 415, 430 111
<code>\phfthm@proofrefstyle@default@setup</code>	<code>problem (environment)</code>
..... 394, 414, 426 4
<code>\phfthm@proofrefstyle@margin@setup</code>	<code>\problemname</code>
..... 413 10, 492, 533
<code>\phfthm@proofrefstyle@marginbottom@setup</code>	<code>\ProcessOptionsX</code>
..... 425 604
<code>\phfthm@ProofTitleFmt</code>	<code>\proof</code>
262, 268, 685 133
<code>\phfthm@ProofTitleHspace</code>	<code>proof (environment)</code>
.. 263, 268 6
<code>\phfthm@test@NOPROOFARG</code>	<code>proofenv (pkg. opt.)</code>
.. 139, 213 6, 45
<code>\phfthm@thmlabel</code>	<code>proofenvstyle (cmd. opt.)</code>
..... 123, 126 18
<code>\phfthm@thmset@</code>	<code>\proofname</code>
..... 500	.. 13, 135, 136, 150, 164, 256
<code>\phfthm@thmset@default</code>	<code>\proofofname</code>
..... 528 13, 136, 152
<code>\phfthm@tmp@box</code>	<code>proofofname (cmd. opt.)</code>
..... 439, 445, 448 13
<code>\phfthm@tmp@boxa</code>	<code>\proofonname</code>
..... 436, 443, 449 303,
<code>\phfthm@tmp@defcmd</code>	361, 419, 421, 438, 441, 624, 627
... 208, 209, 210	<code>proofref (cmd. opt.)</code>
<code>\phfthm@tmp@larg</code> 9, 11, 18
..... 192, 193	<code>proofref (pkg. opt.)</code>
<code>\phfthm@tmp@parentcountername</code> 5, 6, 44
..... 643, 645, 650, 657	<code>\proofrefsize</code>
<code>\phfthm@tmpa</code>	.. 304, 361, 419, 421, 438, 441, 627
..... 316, 317, 318	<code>proofrefstyle (cmd. opt.)</code>
<code>\phfthm@val@displayargs</code> 9
..... 180, 182, 216, 221, 234	<code>prooftitleitbf (pkg. opt.)</code>
<code>\phfthm@val@mkthmopt@counteropts</code>	... 6, 45, 49
..... 650, 652, 655, 661, 663	<code>proposition (environment)</code>
<code>\phfthm@val@mkthmopt@proofrefstyle</code> 4
..... 605, 618, 621, 631, 637, 661	<code>\propositionname</code>
<code>\phfthm@val@mkthmoptarg@defn</code>	10, 482, 504, 511, 520
..... 499, 538	<code>\providecommand</code>
<code>\phfthm@val@mkthmoptarg@theorem</code>	... 135, 303, 315, 680
..... 497, 537	<code>\pushQED</code>
<code>\phfthm@val@noproofref</code> 258
... 306, 313	<code>\qed</code>
<code>\phfthm@val@proofarg</code> 258
..... 211, 213	<code>\qedsymbol</code>
<code>\phfthm@val@proofoflabel</code> 681
.. 191,	<code>qedsymbolblacksquare (pkg. opt.)</code>
192, 214, 219, 284, 288, 292, 296 6, 45, 48
<code>\phfthm@val@prooftitle</code>	<code>question (environment)</code>
164, 169, 4
179, 184, 223, 285, 289, 293, 297	<code>\questionname</code>
<code>\phfthm@val@thmlabel</code>	10, 489, 530
.. 127, 315, 316	
<code>\phfthmLoadThmSet</code>	
..... 544	
<code>\phfthmMakeThmheadingEnvironment</code>	
..... 478	
<code>\phfthmPinProofAnchor</code>	
..... 232	

Q

<code>\qed</code> 258
<code>\qedsymbol</code> 681
<code>qedsymbolblacksquare (pkg. opt.)</code> 6, 45, 48
<code>question (environment)</code> 4
<code>\questionname</code>	10, 489, 530

R

<code>\raggedleft</code> 405, 437, 627
<code>\raggedright</code> 407, 440
<code>\rangle</code> 457
<code>\ref</code> 299
<code>\refstepcounter</code> 198, 372, 465
<code>\relax</code> 33, 38,
	39, 111, 157, 161, 178, 179, 191,
	199, 213, 218, 231, 259, 260,
	313, 325, 359, 378, 379, 380,
	383, 384, 390, 391, 392, 395, 552
<code>remark (environment)</code> 4

<code>\remarkname</code>	10, 486, 515, 524	<code>\theoremstyle</code>	35, 458
<code>\renewcommand</code>	681	<code>theoremstyle</code> (pkg. opt.)	4, 45
<code>\renewenvironment</code>	209	<code>\thephfthmheadingcounter</code>	463
<code>\RequirePackage</code>	1, 2, 3, 4, 5, 679	<code>\thephfthmInternalProofrefCounter</code>	373, 376
<code>resetstyle</code> (pkg. opt.)	3, 44	<code>thmheading</code> (pkg. opt.)	7, 46
<code>rich</code> (pkg. opt.)	4	<code>thmheading</code> (environment)	7
<code>\rule</code>	411	<code>thmheadingstyle</code> (pkg. opt.)	7, 46
S			
<code>sepcounters</code> (pkg. opt.)	5, 44, 48	<code>thmset</code> (pkg. opt.)	3–6, 45
<code>\setbox</code>	436, 439	<code>thmstyle</code> (cmd. opt.)	8, 19
<code>\setcounter</code>	640	<code>\tiny</code>	681
<code>\setkeys</code>	32, 155, 456	<code>\topsep</code>	259
<code>shortnames</code> (pkg. opt.)	4	<code>\trivlist</code>	260
<code>simple</code> (pkg. opt.)	4	<code>true</code> (cmd. opt.)	8, 11, 18
<code>\sloppy</code>	437, 440	<code>true</code> (pkg. opt.)	6, 7
<code>smallproofs</code> (pkg. opt.)	6, 45, 48	V	
<code>\space</code>	136	<code>\val</code>	579, 582
<code>\string</code>	119, 675	<code>\vspace</code>	411
<code>\strut</code>	438, 441	X	
T			
<code>\text</code>	681	<code>\x</code>	40, 42, 44, 233, 234, 633, 635
<code>\textbf</code>	326	<code>xkeyval</code>	43, 46
<code>\the</code>	378	Z	
<code>theorem</code> (environment)	4	<code>\z@</code>	411
<code>\theoremname</code> ...	10, 481, 503, 510, 519		