

An Extension of the L^AT_EX-Theorem Environment*

Wolfgang May[†]
Institut für Informatik,
Universität Göttingen
Germany

Andreas Schedler[§]

2009/07/01

Abstract

`ntheorem.sty` is a package for handling theorem-like environments. Additionally to several features for defining the layout of theorem-like environments which can be regarded to be standard requirements for a theorem-package, it provides solutions for two related problems: placement of endmarks and generation of lists of theorem-like environments.

In contrast to former approaches, it solves the problem of setting endmarks of theorem-like environments (theorems, definitions, examples, and proofs) *automatically* at the right positions, even if the environment ends with a `displaymath` or (even nested) list environments, it also copes with the `amsmath` package. This is done in the same manner as the handling of labels by using the `.aux` file.

It also introduces the generation of lists of theorem-like environments in the same manner as `listoffigures`. Additionally, more comfortable referencing is supported.

After running L^AT_EX several times (depending on the complexity of references, in general, three runs are sufficient), the endmarks are set correctly, and theoremlists are generated.

Since `ntheorem.sty` uses the standard L^AT_EX `\newtheorem` command, existing documents can be switched to `ntheorem.sty` without having to change the `.tex` file. Also, it is compatible with L^AT_EX files using `theorem.sty` written by Frank Mittelbach.

*This file has version number 1.28, last revised 2009/07/01.

[†]`may@informatik.uni-goettingen.de`

[§]`ntheorem@andreas-schedler.de`

Contents

1	Introduction	3
2	The User-Interface	4
2.1	How to include the package	4
2.2	Defining New Theorem Sets	4
2.3	Defining the Layout of Theorem Sets	5
2.3.1	Common Parameters for all Theorem Sets	5
2.3.2	Parameters for Individual Sets	5
2.3.3	Font Selection	6
2.3.4	Predefined theorem styles	7
2.3.5	Default Setting	7
2.3.6	A Standard Set of Theorems	8
2.3.7	Framed and Boxed Theorems	8
2.3.8	Customization and Local Settings	9
2.4	Generating Theoremlists	9
2.4.1	Defining the List Layout	10
2.4.2	Writing Extra Stuff to the Theorem File	10
2.5	For Experts: Defining Layout Styles	11
2.5.1	Defining New Theorem Layouts	11
2.5.2	Defining New Theorem List Layouts	12
2.6	Setting End Marks	12
2.7	Extended Referencing Features	12
2.8	Miscellaneous	13
3	Possible Interferences	13
3.1	Interfering Document Options.	14
3.2	Combination with amslatex.	14
3.2.1	amsmath	14
3.2.2	amsthm	14
3.3	Babel	15
3.4	Hyperref	15
4	Examples	15
4.1	Extended Referencing Features	21
4.2	Framed and Shaded Theorems	22
4.3	Lists of Theorems and Friends	24
5	The End Mark Algorithm	26
5.1	The Idea	26
5.2	The Realization	26

6	Problems and Questions	28
6.1	Known Limitations	28
6.2	Known “Bugs” and Problems	29
6.3	Open Questions	30
7	Code Documentation	30
7.1	Documentation of the Macros	30
7.1.1	Thmmarks-Related Stuff	30
7.1.2	Option leqno to Thmmarks	37
7.1.3	Option fleqn to Thmmarks	38
7.1.4	Extended Referencing Facilities	39
7.1.5	Option amsmath to Thmmarks	42
7.1.6	Theorem-Layout Stuff	48
7.1.7	Theorem-Environment Handling Stuff	53
7.1.8	Framed and Boxed Theorems	62
7.1.9	Generation of Theorem Lists	63
7.1.10	Auxiliary macros	70
7.1.11	Other Things	71
7.2	The Standard Configuration	71
8	History and Acknowledgements	72
8.1	The endmark-Story (Wolfgang May)	72
8.2	Lists, Lists, Lists (Andreas Schedler)	73
8.3	Let’s come together	73
8.4	Acknowledgements	77

1 Introduction

For our purposes here, “theorems” are labelled enunciations, often set off from the main text by extra space and a font change. Theorems, corollaries, conjectures, definitions, examples, remarks, and proofs are all instances of “theorems”. The “header” of these structures is composed of the type of the structure (such as `THEOREM` or `REMARK`), a number which serializes the instances of the same type throughout the document, and an optional name (such as “Correctness Theorem”).

The layout of theorems can be changed by parameters as the fonts of the header and the body, the way how to arrange the headers, the indentation, and the way of numbering it. Confronted with these requirements, `theorem.sty`, a style for dealing with theorem layout was developed by Frank Mittelbach which was the standard theorem-environment for long time.

But then the desire for additional features like “endmarks” and “theorem-lists” arose. Two extensions of `theorem.sty` were developed: One for handling endmarks, `thmmarks.sty` and one for generating lists, `newthm.sty`.

Thus, Frank Mittelbach suggested to combine the new features into one “standard-to-be” package. And now, here it is.

2 The User-Interface

2.1 How to include the package

The package `ntheorem.sty` is included by

```
\usepackage[<options>]{ntheorem},
```

where the optional parameter *<options>* selects predefined configurations and special requirements.

The following *<options>* are available by now, concerning partially independent issues:

Predefined environments: (see Section 2.3.6) With `[standard]` and `[no-config]`, it can be chosen, if and what file is used for activating a (user-defined) standard set of theorem environments.

Fancy boxes around theorems: The `[framed]` option allows to use `framed.sty` that provides boxes even across pagebreaks.

Activation of endmarks: `[thmmarks]` enables the automatical placement of endmarks (see 2.3); when using the `amsmath`-package, `[thmmarks]` must be complemented by `[amsmath]` (see Section 3.2).

Activation of extended reference features: `[thref]` enables the extended reference features (see Section 4.1); when using the `amsmath`-package, `[thref]` must be complemented by `[amsmath]` (see Section 3.2).

Compatibility with `amsthm`: option `[amsthm]` provides compatibility with the theorem-layout commands of the `amsthm`-package (see Section 3.2).

Compatibility with `hyperref`: option `[hyperref]` provides compability with the `hyperref`-package (see section 3.4).

2.2 Defining New Theorem Sets

`\newtheorem` The syntax and semantics is exactly the same as in standard \LaTeX : the command `\newtheorem` defines a new “theorem set” or “theorem-like structure”. Two required arguments name the new environment set and give the text to be typeset with each instance of the new “set”, while an optional argument determines how the “set” is enumerated:

`\newtheorem{foo}{bar}` The theorem set `foo` (whose name is `bar`) uses its own counter.

`\newtheorem{foo2}[foo]{bar2}` The theorem set `foo2` (printed name `bar2`) uses the same counter as the theorem set `foo`.

`\newtheorem{foo3}{bar}[section]` The theorem set `foo3` (printed name `bar`) is enumerated within the counter `section`, i.e. with every new `\section` the enumeration begins again with 1, and the enumeration is composed from the section-number and the theorem counter itself.

For every environment $\langle name \rangle$ defined by `\newtheorem`, *two* environments $\langle name \rangle$ and $\langle name* \rangle$ are defined. In the main document, they have exactly the same effect, but the latter causes no entry in the respective list of theorems (cf. `\section` and `\section*`), see also Section 2.4.

`\renewtheorem` Theorem sets can be redefined by `\renewtheorem`, with the same arguments as explained for `\newtheorem`. When redefining a theorem set, the counter is not re-initialized.

2.3 Defining the Layout of Theorem Sets

For theorem-like environments, the user can set parameters by setting several switches and then calling `\newtheorem`. The layout of a theorem set is defined with the values of the switches at the time `\newtheorem` is called.

2.3.1 Common Parameters for all Theorem Sets

`\theorempreskipamount` `\theorempostskipamount` These additional parameters affect the vertical space around theorem environments: `\theorempreskipamount` and `\theorempostskipamount` define, respectively, the spacing before and after such an environment. These parameters apply for all theorem sets and can be manipulated with the ordinary length macros. They are rubber lengths, (`'skips'`), and therefore can contain `plus` and `minus` parts.

2.3.2 Parameters for Individual Sets

The layout of individual theorem sets can be further determined by switches controlling the appearance of the headers and the header-body-layout:

`\theoremstyle` • `\theoremstyle{\style}`: The general structure of the theorem layout is defined via its `\theoremstyle`. `\nttheorem` provides several predefined styles including those of Frank Mittelbach's `theorem.sty` (cf. Section 2.3.4. Additional styles can be defined by `\newtheoremstyle` (cf. Section 2.5.1).

`\theoremheaderfont` • `\theoremheaderfont{\fontcmds}`: The theorem header is set in the font specified by $\langle fontcmds \rangle$.
In contrast to `theorem.sty`, `\theoremheaderfont` can be set individually for each environment type.

<code>\theorembodyfont</code>	<ul style="list-style-type: none"> • <code>\theorembodyfont{⟨fontcmds⟩}</code>: The theorem body is set in the font specified by <code>⟨fontcmds⟩</code>.
<code>\theoremseparator</code>	<ul style="list-style-type: none"> • <code>\theoremseparator{⟨thing⟩}</code>: <code>⟨thing⟩</code> separates the header from the body of the theorem-environment. E.g., <code>⟨thing⟩</code> can be “:” or “.”.
<code>\theoremprework</code>	<ul style="list-style-type: none"> • <code>\theoremprework{⟨thing⟩}</code>: <code>⟨thing⟩</code> is performed before starting the theorem structure. E.g., <code>⟨thing⟩</code> can be <code>\hrule</code>.
<code>\theorempostwork</code>	<ul style="list-style-type: none"> • <code>\theorempostwork{⟨thing⟩}</code>: <code>⟨thing⟩</code> is performed after finishing the theorem structure. E.g., <code>⟨thing⟩</code> can be <code>\hrule</code>.
<code>\theoremindent</code>	<ul style="list-style-type: none"> • <code>\theoremindent{⟨dimen⟩}</code> can be used to indent the theorem wrt. the surrounding text. ! It’s a ‘(dimen)’, so the user shouldn’t try to specify a plus or minus part, because this leads to an error.
<code>\theoremnumbering</code>	<ul style="list-style-type: none"> • <code>\theoremnumbering{⟨style⟩}</code> specifies the appearance of the numbering of the theorem set. Possible <code>⟨styles⟩</code> are <code>arabic</code> (default), <code>alph</code>, <code>Alph</code>, <code>roman</code>, <code>Roman</code>, <code>greek</code>, <code>Greek</code>, and <code>fnsymbol</code>. Clearly, if a theorem-environment uses the counter of another environment type, also the numbering style of that environment is used.
<code>\theoremsymbol</code>	<ul style="list-style-type: none"> • <code>\theoremsymbol{⟨thing⟩}</code>: This is only active if <code>ntheorem.sty</code> is loaded with option <code>[thmmarks]</code>. <code>⟨thing⟩</code> is set as an endmark at the end of every instance of the environment. If no symbol should appear, say <code>\theoremsymbol{}</code>.

The flexibility provided by these command should relieve the users from the ugly hacking in `\newtheorem` to fit most of the requirements stated by publishers or supervisors.

`\theoremclass` With the command `\theoremclass{⟨theorem-type⟩}` (where `⟨theorem-type⟩` must be an already defined theorem type), these parameters can be set to the values which were used when `\newtheorem` was called for `⟨theorem-type⟩`. With `\theoremclass{LaTeX}`, the standard L^AT_EX layout can be chosen.

2.3.3 Font Selection

From the document structuring point of view, theorem environments are regarded as special parts inside a document. Furthermore, the theorem header is only a distinguished part of a theorem environment. Thus, `\theoremheaderfont` inherits characteristics of `\theorembodyfont` which also inherits in characteristics of the font of the surrounding environment. Thus, if for example `\theorembodyfont` is `\itshape` and `\theoremheaderfont` is `\bfseries` the font selected for the header will have

the characteristics ‘bold extended italic’. If this is not desired, the corresponding property has to be explicitly overwritten in `\theoremheaderfont`, e.g. by `\theoremheaderfont{\normalfont\bfseries}`

2.3.4 Predefined theorem styles

The following theorem styles are predefined, covering those from `theorem.sty`:

plain	This theorem style emulates the original L ^A T _E X definition, except that additionally the parameters <code>\theorem...skipamount</code> are used.
break	In this style, the theorem header is followed by a line break.
change	Header number and text are interchanged, without a line break.
changebreak	Like change , but with a line break after the header.
margin	The number is set in the left margin, without a line break.
marginbreak	Like margin , but with a line break after the header.
nonumberplain	Like plain , without number (e.g. for proofs).
nonumberbreak	Like break , without number.
empty	No number, no name. Only the optional argument is typeset.

2.3.5 Default Setting

If no option is given, i.e. `ntheorem.sty` is loaded by `\usepackage{ntheorem.sty}`, the following default is set up:

```
\theoremstyle{plain},
\theoremheaderfont{\normalfont\bfseries} and
\theorembodyfont{\itshape},
\theoremseparator{},
\theoremindent0cm,
\theoremnumbering{arabic},
\theoremsymbol{}
```

Thus, by only saying `\newtheorem{...}{...}`, the user gets the same layout as in standard L^AT_EX.

2.3.6 A Standard Set of Theorems

A standard configuration of theorem sets is provided within the file `ntheorem.std`, which will be included by the option `[standard]`. It uses the `amssymb` and `latexsym` (automatically loaded) packages and defines the following sets:

Theorems: `Theorem`, `Lemma`, `Proposition`, `Corollary`, `Satz`, `Korollar`,

Definitions: `Definition`,

Examples: `Example`, `Beispiel`,

Remarks: `Anmerkung`, `Bemerkung`, `Remark`,

Proofs: `Proof` and `Beweis`.

These theorem sets seem to be the most frequently used environments in english and german documents.

The layout is defined to be `theoremstyle plain`, `bodyfont \itshape`, `Headerfont \bfseries`, and `endmark (theoremsymbol) \ensuremath{_\Box}` for all theorem-like environments¹. For the definition-, remark- and example-like sets, the above setting is used, except `bodyfont \upshape`. The proof-like sets are handled a bit differently. There, the layout is defined as `theoremstyle nonumberplain`, `bodyfont \upshape`, `headerfont \scshape` and `endmark \ensuremath{_\blacksquare}`. For a more detailed information look at `ntheorem.std` or at the code-section.

2.3.7 Framed and Boxed Theorems

With the advent of the `framed` package (by Donald Arseneau) in 2001, a feature that has often been asked for for `ntheorem` could be implemented: theorems that are framed, or that are put into a colored box. It requires to load the `framed` package; shaded theorems also require the `pstricks` package. Frames and colored boxes are orthogonal to the existing theoremstyles – thus, they can be combined in arbitrary ways.

`\newframedtheorem` A theorem type can be framed by defining it by

$$\text{\texttt{\textbackslash newframedtheorem\{...\}\{...\}}}$$

with the same parameters as usually for `\newtheorem`. Note that the use of the `framed` package also allows to have longer theorems across a page break framed.

`\newshadedtheorem`

The same ideas hold for theorems in shaded boxes. The declaration

$$\text{\texttt{\textbackslash newshadedtheorem\{...\}\{...\}}}$$

¹Note, that `mathmode` is ensured for the symbol.

declares a theorem environment that is shaded. By default, the background color is `gray`. This can be changed by defining

```
\shadecolor{<color>}
```

before declaring the theorem type. Note that later declarations of other shaded theorem types can use another `shadecolor`.

By default, the box is given as a `\psframebox` (see `psstricks` package) with `shadecolor` as `linecolor` and `fillcolor`. All these parameters can be changed by setting

```
\def\theoremframecommand{<any box command>}
```

before declaring the theorem type (for examples, the user is referred to section 4).

2.3.8 Customization and Local Settings

Since the user should not change `ntheorem.std`, we've added the possibility to use an own configuration-file. If one places the file `ntheorem.cfg` in the path searched by `TeX`, this file is read automatically (if `[standard]` is not given). The usage of `ntheorem.cfg` can be prevented by the `[noconfig]` option. Thus, just a copy of `ntheorem.std` to `ntheorem.cfg` must be made which then can freely be modified by the user. Note, that if a configuration-file exists, this will always be used (I.e. with option `standard` and an existing configuration-file, the `.cfg` file will be used and the `.std` file won't).

2.4 Generating Theoremlists

`\listtheorems` Similar to the `LATEX` command `\listoffigures`, any theorem set defined with a `\newtheorem` statement may be listed at any place in your document by

```
\listtheorems{<list>}
```

The argument `<list>` is a comma-separated list of the theorem sets to be listed. For a theorem set `<name>`, only the instances are listed which are instantiated by `\begin{<name>}`. Those instantiated by `\begin{<name>}` are omitted (cf. `\section` and `\section*`).

For example, `\listtheorems{Corollary,Lemma}` leads to a list of all instances of one of the theorem sets “Corollary” or “Lemma”. Note, that the set name given to the command is the first argument which is specified by `\newtheorem` which is also the one to be used in `\begin{theorem} ... \end{theorem}`. If `\listtheorems` is called for a set name which is not defined via `\newtheorem`, the user is informed that a list is generated, but there will be no typeset output at all.

2.4.1 Defining the List Layout

`\theoremlisttype` Theoremlists can be formatted in different ways. Analogous to theorem layout, there are several predefined types which can be selected by

`\theoremlisttype{<type>}`

The following four *<type>*s are available (for examples, the user is referred to section 4).

all List any theorem of the specified set by number, (optional) name and pagenumber. This one is also the default value.

allname Like **all**, additionally with leading theoremname.

opt Analogous to **all**, but only the theorems which have an optional name are listed.

optname Like **opt**, with leading theoremname.

2.4.2 Writing Extra Stuff to the Theorem File

Similar to `\addcontentsline` and `\addtocontents`, additional entries to theoremlists are supported. Since entries to theoremlists are a bit more intricate than entries to the lists maintained by standard L^AT_EX `\addcontentsline` and `\addtocontents` cannot be used in a straightforward way².

`\addtheoremline` Analogous to `\addcontentsline`, an extra entry for a theorem list can be made by

`\addtheoremline{<name>}{<text>}`

where *<name>* is the name of a valid theorem set and *<text>* is the text, which should appear in the list. For example,

`\addtheoremline{Example}{Extra Entry with number}`

generates an entry with the following characteristics:

- The Label of the theorem “Example” is used.
- The current value of the counter for “Example” is used
- The current pagenumber is used.
- The specified text is the optional text for the theorem.

Thus, the above command has the same effect as it would be for

²for a theorem, its number has to be stored explicitly since different theorem sets can use the same counter. Also, it is optional to reset the counter for each section.

`\begin{Example}[Extra Entry with number] \end{Example}`

except, that there would be no output of the theorem, and the counter isn't advanced.

`\addtheoremline*` Alternatively you can use

`\addtheoremline*{Example}{Extra Entry}`

which is the same as above, except that the entry appears without number.

`\addtotheoremfile` Sometimes, e.g. for long lists, special control sequences (e.g. a pagebreak) or additional text should be inserted into a list. This is done by

`\addtotheoremfile[\langle name \rangle]{\langle text \rangle}`

where $\langle name \rangle$ is the name of a theorem set and $\langle text \rangle$ is the text to be written into the theorem file. If the optional argument $\langle name \rangle$ is omitted, the given text is inserted in every list, otherwise it is only inserted for the given theorem set.

2.5 For Experts: Defining Layout Styles

2.5.1 Defining New Theorem Layouts

`\newtheoremstyle` Additional layout styles for theorems can be defined by

`\newtheoremstyle{\langle name \rangle}{\langle head \rangle}{\langle opt-head \rangle}.`

After this, `\theoremstyle{\langle name \rangle}` is a valid `\theoremstyle`. Here, $\langle head \rangle$ has to be a statement using two arguments, **##1**, containing the keyword, and **##2**, containing the number. $\langle opt-head \rangle$ has to be a statement using three arguments where the additional argument **##3** contains the optional parameter.

Since L^AT_EX implements theorem-like environments by `\trivlists`, both header declarations must be of the form `\item[... \theorem@headerfont ...]`..., where the dotted parts can be formulated by the user. If there are some statements producing output after the `\item[...]`, you have to care about implicit spaces.

Because of the @, if `\newtheoremstyle` is used in a .tex file, it has to be put between `\makeatletter` and `\makeatother`.

For details, look at the code documentation or the definitions of the predefined theoremstyles.

`\renewtheoremstyle` Theorem styles can be redefined by `\renewtheoremstyle`, with the same arguments as explained for `\newtheoremstyle`.

2.5.2 Defining New Theorem List Layouts

`\newtheoremlisttype` Analogous, additional layouts for theorem lists can be defined by

`\newtheoremlisttype{<name>}{<start>}{<line> }{<end>}`.

The first argument, `<name>`, is the name of the listtype, which can be used as a valid `\theoremlisttype`. `<start>` is the sequence of commands to be executed at the very beginning of the list. Corresponding, `<end>` will be executed at the end of the list. These two are set to do nothing in the standard-types. `<line>` is the part to be called for every entry of the list. It has to be a statement using four arguments: `##1` will be replaced with the name of the theorem, `##2` with the number, `##3` with the theorem's optional text and `##4` with the pagenumber.

WARNING: Self-defined Layouts will break with the `hyperref`-package.

`\renewtheoremlisttype` Theorem list types can be redefined by `\renewtheoremlisttype`, with the same arguments as explained for `\newtheoremlisttype`.

2.6 Setting End Marks

The automatic placement of endmarks is activated by calling `ntheorem.sty` with the option `[thmmarks]`. Since then, the endmarks are set automatically, there are only a few commands for dealing with very special situations.

`\qed` If in a single environment, the user wants to replace the standard endmark by
`\qedsymbol` some other, this can be done by saying `\qed`, if `\qedsymbol` has been defined by `\qedsymbol{<something>}` (in option standard, `\qedsymbol` is defined to be the symbol used for proofs, since a potential use of this features is to close trivial corollaries without explicitly proving them).

Additionally, if in a single environment of a theorem set, that is defined without an endmark, the user wants to set an endmark, this is done with `\qedsymbol` and `\qed` as described above. `\qedsymbol` can be redefined everywhere in the document.

`\NoEndMark` On the other hand, if in some situation, the user decides to set the endmark
`\TheoremSymbol` manually (e.g. inside a figure or a minipage), the automatic handling can be turned off by `\NoEndMark` for the current environment. Then – assumed that the current environment is of type `<name>`, the endmark can manually be set by just saying `\<name>Symbol`.

Note that there must be no empty line in the input before the `\end{theorem}`, since then, the end mark is ignored (cf. Theorem 3 in Section 4).

2.7 Extended Referencing Features

The extended referencing features are activated by calling `ntheorem.sty` with the option `[thref]`.

Often, when writing a paper, one changes propositions into theorems, theorems into corollaries, lemmata into remarks and so on. Then, it is necessary to

adjust also the references, i.e., from “see Proposition~\ref{completeness}” to “see Theorem~\ref{completeness}”. For relieving the user from this burden, the type of the respective labeled entities can be associated with the label itself:

```
\label{<label>}[<type>]
```

associates the type $\langle type \rangle$ with $\langle label \rangle$.

This task is automated for theorem-like environments:

```
\begin{Theorem}[<name>]\label{<label>}
```

is equivalent to

```
\begin{Theorem}[<name>]\label{<label>}[Theorem]
```

`\thref` The additional information is used by

```
\thref{<label>}
```

which outputs the respective environment-type *and* the number, e.g., “Theorem 42”. Note that L^AT_EX has to be run twice after changing labels (similar to getting references OK; in the intermediate run, warnings about undefined reference types can occur).

The `[thref]` option interferes with the `babel` package, thus in this case, `ntheorem` has to be loaded *after* `babel`. It also interferes with `amsmath`; see Section 3.2.

2.8 Miscellaneous

Inside a theorem-like environment $\langle env \rangle$, the name given as optional argument is accessible by `\langle env \rangle name`.

3 Possible Interferences

Since `ntheorem` reimplements the handling of theorem-environments completely, it is incompatible with every package also concerning those macros. Additionally, the `thmmarks` algorithm for placing endmarks requires modifications of several environments (cf. Section 7). Thus, environments which are reimplemented or additionally defined by document options or styles are not covered by the endmark algorithm of `ntheorem.sty`.

The `[thref]` option changes the `\label` command and the treatment of labels when reading the `.aux` file. Thus it is potentially incompatible with all packages also changing `\label` (or `\newlabel`). Compatibility with `babel`’s `\newlabel` is achieved if `babel` is loaded before `ntheorem`.

3.1 Interfering Document Options.

`ntheorem.sty` also copes with the usual document options `leqno` and `fleqn`³. If one of those options is used in the `\documentclass` declaration, it is automatically recognized by the `thmmarks` part of `ntheorem.sty`. If one of those options is not used in `\documentclass`, but with `amsmath` (see next section), it must not be specified for `ntheorem`, since all `amsmath` environments detect this option by themselves.

3.2 Combination with `amslatex`.

`ntheorem.sty` interferes with `amsmath.sty` and `amsthm.sty`.

Note, that the LaTeX `amstex` package `amstex.sty` (L^AT_EX 2_ε) is obsolete and you should use `amsmath` and `amstext` for L^AT_EX 2_ε instead. Up to `ntheorem-1.18`, it is compatible with `amsmath-1.x`. Since `ntheorem-1.19`, it is (hopefully) compatible with `amsmath-2.x`.

We would be happy if someone knowing and using `amsmath` would join the development and maintenance of this style.

3.2.1 `amsmath`

Compatibility with `amsmath` (end marks for math environments, and handling of labels in math environments) is provided in the option `[amsmath]`, (i.e., if `\usepackage{amsmath}` is used then

- `\usepackage[thmmarks]{ntheorem}` must be completed to `\usepackage[amsmath,thmmarks]{ntheorem}`), and also
- `\usepackage[thref]{ntheorem}` must be completed to `\usepackage[amsmath,thref]{ntheorem}`).

Note, that `amsmath` has to be loaded *before* `ntheorem` since the definitions have to be overwritten.

3.2.2 `amsthm`

`amsthm.sty` conflicts with the definition of theorem layouts in `theorem.sty`, some features of `amsthm.sty` have been incorporated into option `[amsthm]` which has to be used *instead of* `\usepackage{amsthm}`.

The Option provides theoremstyles `plain`, `definition`, and `remark`, and a `proof` environment as in `amsthm.sty`.

The `\newtheorem*` command is defined even without this option. Note that `\newtheorem*` always switches to the nonnumbered version of the current theoremstyle which thus must be defined.

³although for `fleqn` and long formulas reaching to the right margin, equation numbers and endmarks can be smashed over the formula since `fleqn` does not use `\eqno` for controlling the setting of the equation number.

The command `\newtheoremstyle` is not taken over from `amsthm.sty`. Also, `\swapnumbers` is not implemented. Here, the user has to express his definitions by the `\newtheoremstyle` command provided by `ntheorem.sty`, including the use of `\theoremheaderfont` and `\theorembodyfont`. The options `[amsthm]` and `[standard]` are in conflict since they both define an environment `proof`.

Thus, we recommend not to use `amsthm`, since the features for defining theorem-like environments in `ntheorem.sty`—following `theorem.sty`—seem to be more intuitive and user-friendly.

3.3 Babel

The `[thref]` option interferes with the `babel` package, thus in case that `babel` is used, `ntheorem` has to be loaded *after* `babel`.

3.4 Hyperref

Since `hyperref` redefines the L^AT_EX `\contentsline`-command, it breaks with `ntheorem` below version 1.17. Since version 1.17, the option `[hyperref]` makes `ntheorem` work with `hyperref`. Theorem lists will then get linked list. **WARNING:** The definition and redefinition of Theorem List Layouts (see Section 2.5.2) isn't yet working with the `hyperref`-package.

4 Examples

The setting is as follows.

- For Theorems:

```
\theoremstyle{marginbreak}
\theoremheaderfont{\normalfont\bfseries}\theorembodyfont{\slshape}
\theoremsymbol{\ensuremath{\diamondsuit}}
\theoremseparator{:}
\newtheorem{Theorem}{Theorem}
```

- For Lemmas:

```
\theoremstyle{changebreak}
\theoremsymbol{\ensuremath{\heartsuit}}
\theoremindent0.5cm
\theoremnumbering{greek}
\newtheorem{Lemma}{Lemma}
```

- For Corollaries:

```
\theoremindent0cm
```

```

\theoremsymbol{\ensuremath{\spadesuit}}
\theoremnumbering{arabic}
\newtheorem{Corollary}[Theorem]{Corollary}

```

- For Examples:

```

\theoremstyle{change}
\theorembodyfont{\upshape}
\theoremsymbol{\ensuremath{\ast}}
\theoremseparator{}
\newtheorem{Example}{Example}

```

- For Definitions:

```

\theoremstyle{plain}
\theoremsymbol{\ensuremath{\clubsuit}}
\theoremseparator{.}
\theoremprework{\bigskip\hrule}
\theorempostwork{\hrule\bigskip}
\newtheorem{Definition}{Definition}

```

- For Proofs (note that `theoremprework` and `theorempostwork` are reset – proofs do not have lines above and below):

```

\theoremheaderfont{\sc}\theorembodyfont{\upshape}
\theoremstyle{nonumberplain}
\theoremseparator{}
\theoremsymbol{\rule{1ex}{1ex}}
\newtheorem{Proof}{Proof}

```

Note, that parts of the setting are inherited. For instance, the fonts are not reset before defining “Lemma”, so the font setting of “Theorem” is used.

1 Example (Simple one) The first example is just a text.

In the next examples, it is shown how an endmark is put at a displaymath, a single equation and both types of eqnarrays. *

1 Theorem (Long Theorem):

The examples are put into this theorem environment.

The next example will not appear in the list of examples since it is written as

```
\begin{Example*} ... \end{Example*}
```

2 Example (Ending with a displayed formula) Look, the endmark is really at the bottom of the line:

$$f^{(n)}(z) = \frac{n!}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{(\zeta - z)^{n+1}} d\zeta \quad *$$

At this point, we add an additional entry without number in the Example list:

$$\backslash\mathrm{addtheorem}\mathrm{line}\ast\{\mathrm{Example}\}\{\mathrm{Extra~Entry}\}$$

α Lemma (Display with array):

Lemmata are indented and numbered with greek symbols. Also for displayed arrays of this form, it looks good:

```
\[\begin{array}{l}
  a = \begin{array}{t}{l}
    first\ line \\\
    second\ line
  \end{array}%
  \mbox{try to put this text in the lowest line}\end{array}\]
```

Just try to get this with the presented array structure ... without using dirty tricks, you can position the outer array either $[t]$, $[c]$, or $[b]$, and you will not get the desired effect.

$a =$ *first line* try to put this text in the lowest line
second line ♡

β Lemma (Equation):

For equations, we decided to put the endmark after the equation number, which is vertically centered. Currently, we do not know, how to get the equation number centered and the endmark at the bottom (one has to know the internal height of the math material) ... If anyone knows, please inform us.

$$\int_{\gamma} f(z) dz := \int_a^b f(\gamma(t)) \gamma'(t) dt \quad (1) \quad \heartsuit$$

With the `break-theoremstyles`, if the environment is labeled and written as

 $\begin{array}{l} \text{\textit{Lemma}} \\ \text{\textit{Breakstyle}} \end{array}$

γ Lemma (Breakstyle):

you see, there is a leading space ...

If a percent (comment) (or an explicit `\ignorespaces`) is put directly after the label, e.g.

```
\begin{Lemma}[Breakstyle]\label{breakstyle}%,
```

the space disappears.

From the predefined styles, this is exactly the case for the break-styles. That's no bug, it's L^AT_EX-immanent.

The example goes on with an `eqnarray`:

$$f(z) = \frac{1}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{\zeta - z} d\zeta \tag{2}$$

$$= \frac{1}{2\pi} \int_0^{2\pi} f(z_0 + re^{it}) dt \tag{3}$$

♡

PROOF (OF NOTHING)

$$\begin{aligned} f(z) &= \frac{1}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{\zeta - z} d\zeta \\ &= \frac{1}{2\pi} \int_0^{2\pi} f(z_0 + re^{it}) dt \end{aligned} \quad \blacksquare$$

That's it (the end of the Theorem). ◇

If there are some environments in the same thm-environment, the last gets the endmark:

Definition 1 (With a list).

$$\int_{\gamma} f(z) dz := \int_a^b f(\gamma(t)) \gamma'(t) dt \tag{4}$$

- you've seen, how it works for text and
- math environments,
- and it works for lists. ♣

2 Corollary (Q.E.D.):

And here is a trivial corollary, which is ended by `\qedsymbol{\textrm{q.e.d}}` and `\qed`. q.e.d

3 Example

$$f^{(n)}(z) = \frac{n!}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{(\zeta - z)^{n+1}} d\zeta$$

If there is some text after an environment, the endmark is put after the text. *

The next one is done by the following sequence. Note, that `~\hfill~` is inserted to prevent \LaTeX from using its nested list management (a verbatim is also a trivlist), i.e. this causes \LaTeX to start the `verbatim`-Part in a new line.

```
\begin{Example}
~\hfill~
\begin{verbatim}
And, it also works for verbatim
... when the \end{verbatim} is in the
same line as the text ends. \end{verbatim}
~ this space is important !!
\end{Example}
```

4 Example (Using verbatim)

```
And, it also works for verbatim
... when the \end{verbatim} is in the
same line as the text ends. *
```

There must be no empty line in the input before the `\end{theorem}` (since then, the end mark is ignored)

```
\begin{Theorem}
some text ... but no end mark

\end{Theorem}
```

3 Theorem:

some text ... but no end mark

Now, there is a corollary which should appear with a different name in the list of corollaries:

```
\begin{Corollary*}[title in text]\label{otherlabel}
...
\end{Corollary*}\addtheoremline{Corollary}{title in list}
```

4 Corollary (title in text):

let's do something weird:

*It also works in the
center
environment.* ♠

5 Theorem (Quote):

In quote environments, the text is normally indented from left and right by the same space. The endmark is not indented from the right margin, i.e., it is typeset to the right margin of the surrounding text. ◇

Here is an example for turning off the endmark automatics and manual handling:

```
\begin{Theorem}[Manual End Mark]\label{somelabel}
a line of text with a manually set endmark \hfill\TheoremSymbol \\\
some more text, but no automatic endmark set. \NoEndMark
\end{Theorem}
```

6 Theorem (Manual End Mark):

a line of text with a manually set endmark ◇
some more text, but no automatic endmark set.

Also, one should note, that `\hfill` is inserted to set the endmark at the right margin.

5 Example (Quickie) It also works for short one's. *

If you are tired of the greek numbers and the indentation for lemmata ... you can redefine it:

```
\theoremstyle{changebreak}
\theoremheaderfont{\normalfont\bfseries}\theorembodyfont{\slshape}
\theoremsymbol{\ensuremath{\heartsuit}}
\theoremsymbol{\ensuremath{\diamondsuit}}
\theoremseparator{:}
\theoremindent0.5cm
\theoremnumbering{arabic}
\renewtheorem{Lemma}{Lemma}
```

4 Lemma:

another lemma, with arabic numbering ... note that the numbering continues. ◇

the optional argument (i.e. the ‘theorem’-name) can be accessed by `\(env)\name`.

```
\begin{Theorem}[somename]
Obviously, we are in Theorem~\Theoremname.
\end{Theorem}
```

7 Theorem (somename):

Obviously, we are in Theorem somename. ◇

This feature can e.g. be used for automatically generating executable code and a commented solution sheet:

```
\begin{exercise}[quicksort]
  \langle the exercise text \rangle
  \begin{verbatimwrite}{solutions/\exercisename.c}
  \langle C-code \rangle
  \end{verbatimwrite}
  \verbatiminput{solutions/\exercisename.c}
\end{exercise}
```

This will write the C-code to a file `solutions/quicksort.c` and type it also on the solution sheet.

Now, we define an environment `KappaTheorem` which uses the same style parameters as `Theorems` and is numbered together with Corollaries (`Theorems` are also numbered with Corollaries). Note that we define a complex header text and a complex end mark.

```
\theoremclass{Theorem}
\theoremsymbol{\ensuremath{a\atop b}}
\newtheorem{KappaTheorem}[Corollary]{\(\kappa\)-Theorem}
```

8 κ -Theorem (1st κ -Theorem):

That's the first Kappa-Theorem.

a
b

4.1 Extended Referencing Features

The standard `\label` command is extended by an optional argument which is intended to contain the “name” of the structure which is labeled, allowing more comfortable referencing; e.g., this section has been started with

```
\subsection*{Extended Referencing Features}%
\label{sec-ExtRef}[Section]
```

As already stated, for theorem-like environments the optional argument is filled in automatically, i.e.,

```
\begin{Theorem}[Manual End Mark]\label{somelabel}
```

(cf. page 20) is equivalent to

```
\begin{Theorem}[Manual End Mark]\label{somelabel}[Theorem]
```

`\thref{\langle label \rangle}` additionally outputs the contents of the optional argument which has been associated with `\langle label \rangle`:

```
This is \thref{sec-ExtRef}
A theorem end mark has been set manually in \thref{somelabel}.
A center environment has been shown in \thref{otherlabel}.
The first Kappa-Theorem has been given in \thref{kappatheorem1}.
```

generates

This is Section 4.1.

A theorem end mark has been set manually in Theorem 6. A center environment has been shown in Corollary 4. The first Kappa-Theorem has been given in κ -Theorem 8.

Here one must be careful that the handling of the optional argument is automated only for environments defined by `\newtheorem`, i.e., *not* for sectioning, equations, or enumerations.

Calling `\thref{<label>}` for a label which has been set without an optional argument can result in different unintended results: If `<label>` is not inside a theorem-like environment, an error message is obtained, otherwise the type of the surrounding theorem-like environment is output, e.g., calling `\thref{label}` then results in “Theorem *<number>*”! Additionally, currently there is no support for multiple references such as “see Theorems 5 and 7” (this would require plural-forms for different languages and handling of `\ref`-lists, probably splitting into different sublists for different environments)⁴.

4.2 Framed and Shaded Theorems

Framed theorem classes are defined as follows:

```
\theoremclass{Theorem}
\theoremstyle{break}
\newframedtheorem{importantTheorem}[Theorem]{Theorem}
```

defines important theorems to use the same design as for theorems (except that the break header style is used except the margin header style), number them with the same counter, and put a frame around them:

An instance is created by

```
\begin{importantTheorem}[Important Theorem]
This is an important theorem.
\end{importantTheorem}
```

Theorem 9 (Important Theorem):

This is an important theorem.

More important theorems are shaded – by default in grey:

⁴If someone is interested in programming this, please contact us; it seems to be algorithmically easy, but tedious.

```

\theoremclass{Theorem}
\theoremstyle{break}
\newshadedtheorem{moreImportantTheorem}[Theorem]{Theorem}
\begin{moreImportantTheorem}[More Important Theorem]
This is a more important theorem.
\end{moreImportantTheorem}

```

Theorem 10 (More Important Theorem):

This is a more important theorem.



Even more important theorems are shaded in red:

```

\theoremclass{Theorem}
\theoremstyle{break}
\shadecolor{red}
\newshadedtheorem{evenMoreImportantTheorem}[Theorem]{Theorem}
\begin{evenMoreImportantTheorem}[Even More Important Theorem]
This is an even more important theorem.
\end{evenMoreImportantTheorem}

```

Theorem 11 (Even More Important Theorem):

This is an even more important theorem.



Most important theorems get a framed, blue colored box with a shadow. Here, `\def\theoremframecommand` is used:

```

\theoremclass{Theorem}
\theoremstyle{break}
\shadecolor{red}
\def\theoremframecommand{%
    \psshadowbox[fillstyle=solid,fillcolor=blue,linecolor=black]}
\newshadedtheorem{MostImportantTheorem}[Theorem]{Theorem}
\begin{MostImportantTheorem}[Most Important Theorem]
This is a most important theorem.
\end{MostImportantTheorem}

```

Theorem 12 (Most Important Theorem):

This is a most important theorem.



4.3 Lists of Theorems and Friends

Note, that we put the following lists into the `quote`-environment to emphasize them from the surrounding text. So the lists are indented slightly at the margin.

With

```
\addtotheoremfile{Added into all theorem lists},
```

in every list, an additional line of text would be inserted. But it isn't actually done in this documentation since we want to use different list formats.

Only for the list of Examples, this one is added:

```
\addtotheoremfile[Example]{Only concerning Example lists}
```

With

```
\theoremlisttype{all}
\listtheorems{Lemma},
```

all lemmas are listed:

α	Display with array	17
β	Equation	17
γ	Breakstyle	17
4	20
5	28
6	29

From the examples, only those are listed which have an optional name:

```
\theoremlisttype{opt}
\listtheorems{Example}
```

leads to

0	Extra Entry with number	10
	Extra Entry	11
1	Simple one	16
	Extra Entry	17
4	Using <code>verbatim</code>	19
5	Quickie	20
	Only concerning Example lists	

One should note the line *Only concerning example lists*, which was added by the `\addtotheoremfile`-statement above.

For the next list, another layout, using the `tabular`-environment, is defined:

```
\newtheoremlisttype{tab}%
  {\begin{tabular*}{\linewidth}{@{}lrl@{\extracolsep{\fill}}r@{}}}%
  {##1&##2&##3&##4\\}%
  {\end{tabular*}}
```

Thus, by saying

```
\theoremlisttype{tab}
\listtheorems{Theorem,importantTheorem,moreImportantTheorem,
              evenMoreImportantTheorem,MostImportantTheorem,Lemma},
```

theorems (of all importance levels) and lemmata are listed:

Theorem	1	Long Theorem	16
Lemma	α	Display with array	17
Lemma	β	Equation	17
Lemma	γ	Breakstyle	17
Theorem	3		19
Theorem	5	Quote	20
Theorem	6	Manual End Mark	20
Lemma	4		20
Theorem	7	somename	20
Theorem	9	Important Theorem	22
Theorem	10	More Important Theorem	23
Theorem	11	Even More Important Theorem	23
Theorem	12	Most Important Theorem	23
Theorem	13	Correctness	27
Theorem	14	Completeness	27
Lemma	5		28
Lemma	6		29
Theorem	15		29

L^AT_EX-lists can also be used to format the theoremlist. The input

```
\newtheoremlisttype{list}%
  {\begin{trivlist}\item}
  {\item[##2 ##1:] \ ##3\dotfill ##4}%
  {\end{trivlist}}
\theoremlisttype{list}
\listtheorems{Corollary}
```

leads to

2 Corollary:	Q.E.D.	18
--------------	-------------	----

In this example, after the item, `_` is used instead of `_`, because in the latter case, `\dotfill` will produce an error if the optional argument (`##3`) is missing.

5 The End Mark Algorithm

5.1 The Idea

The handling of endmarks with `thmmarks.sty` is based on the same two-pass principle as the handling of labels: the necessary information about endmarks is contained in the `.aux` file.

With `thmmarks.sty`, \TeX is always aware whether it is in some theorem-like environment. There, potential positions for endmarks can be

1. at the end of simple text lines in open text,
2. at the end of displaymaths,
3. at the end of equations or equationarrays, or
4. at the end of text lines at the end of lists (or, more general, `trivlists`, such as `verbatim` or `center`).

The problem is, that in the cases (2)–(4), the endmarks has to be placed in a box which is already shipped out, when `\end{...}` is processed. Thus, in those situations, \TeX needs to know from the `.aux` file, whether it has to put an endmark.

When \TeX is in a theorem-like environment and comes to one of the points mentioned in (2)–(4), and the `.aux` file says that there is an endmark, then it is put there. Anyway, it maintains a counter of the potential positions of an end mark in the current theorem-like environment. When it comes to an `\end{theorem}`, it looks if it is in situation (1) (then the endmark is simply put at the end of the current line). Otherwise, the last horizontal box is already shipped out (thus it contains a situation (2)–(4)) and the endmark must be set in it. In this case, a note is written in the `.aux` file, where the endmark actually has to be set (ie, at the latest potential point for setting an endmark inside the theorem).

5.2 The Realization

Let $\langle env \rangle$ be a theorem-like environment. Then, additional to the counter $\langle env \rangle$, \TeX maintains two counters `curr $\langle env \rangle$ ctr` and `end $\langle env \rangle$ ctr`. In the i th environment of type $\langle env \rangle$, `curr $\langle env \rangle$ ctr` = i (the \LaTeX counter $\langle env \rangle$ cannot be used since a) environments can use the counter of other

environments, and b) often counters are reinitialized inside a document). `end⟨env⟩ctr` counts the potential situations for putting an endmark inside an environment. It is set to 1 when starting an environment. Each time, when a situation (2)–(4) is reached, the command

$$\backslash\mathrm{mark}<\backslash\mathrm{thm@romannum}{\mathrm{currentenvctr}}>\langle\mathrm{env}\rangle<\backslash\mathrm{thm@romannum}{\mathrm{end}\langle\mathrm{env}\rangle\mathrm{ctr}}>$$

is called (where `\thm@romannum` just writes the value of a counter as its roman numeral representation, e.g., 17 as xvii).

(`<\thm@romannum{curr⟨env⟩ctr}>\langle\mathrm{env}\rangle<\backslash\mathrm{thm@romannum}{\mathrm{end}\langle\mathrm{env}\rangle\mathrm{ctr}}>` uniquely identifies all situations (2)–(4) in a document).

If at this position an endmark has to be set,

$$\backslash\mathrm{mark}<\backslash\mathrm{thm@romannum}{\mathrm{curr}\langle\mathrm{env}\rangle\mathrm{ctr}}>\langle\mathrm{env}\rangle<\backslash\mathrm{thm@romannum}{\mathrm{end}\langle\mathrm{env}\rangle\mathrm{ctr}}>$$

is defined in the `.aux` file to be `\end⟨env⟩Symbol`, otherwise it is undefined and simply ignored.

When $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ comes to an `\end{⟨env⟩}`, it looks if it is in situation (1). If so, the endmark is simply put at the end of the current line. Otherwise,

$$\backslash\mathrm{def}\backslash\mathrm{mark}<\backslash\mathrm{thm@romannum}{\mathrm{currentenvctr}}>\langle\mathrm{env}\rangle\% \\ <\backslash\mathrm{thm@romannum}{\mathrm{end}\langle\mathrm{env}\rangle\mathrm{ctr}}>\{\langle\mathrm{env}\rangle\mathrm{Symbol}\}$$

is written to the `.aux` file for setting the endmark at the latest potential position inside the theorem in the next run.

13 Theorem (Correctness):

1. For a `.tex` file, which does not contain nested theorem-like environments of the same type, in the above situation, the following holds: When compiling, at the i th situation in the j th environment of type $\langle\mathrm{env}\rangle$, `mark j ⟨env⟩ i` is handled.

For `.tex` files which contain nested theorem-like environments of the same type, `mark k ⟨env⟩ l` is handled, where k is the number of the latest environment of type $\langle\mathrm{env}\rangle$ which has been called at this moment, and l is the number of situations (2)–(4) which have occurred in environments of type $\langle\mathrm{env}\rangle$ since the k th `\begin{⟨env⟩}`.

2. When finishing an environment, either an endmark is set directly (when in a text line) or an order to put the end symbol at the latest potential position is written to the `.aux` file. \diamond

14 Theorem (Completeness):

The handling of endmarks is complete wrt. plain text, `displaymath`, `equation`, `eqnarray`, `eqnarray*`, and all environments ended by `endtrivlist`, including `center` and `verbatim`. \diamond

So, where can be bugs ?

- in the plain \TeX handling of endmarks,
- in some special situations which have not been tested yet,
- in some special environments which have not been tested yet.
- in the `amsmath` environments. We seldom use them, so we do not know their pitfalls, and we ran only general test cases.

6 Problems and Questions

6.1 Known Limitations

- Since `ntheorem.sty` uses the `.aux` file for storing information about the positions of endmarks, \LaTeX must be run twice for correctly setting the endmarks.
- Since `ntheorem.sty` uses the `.aux` file for storing information about lists in the `.thm` file, a minimum of two runs is needed. If theorems move in any of these runs up to five runs can be needed to generate correct lists.
- Since we need to expand the optional argument of theorems in various ways for the lists, we decided to copy the text verbatim into the `.thm` file. Thus, if you use things like `\thesection` etc., the list won't show the correct text. Therefore you shouldn't use any command that needs to be expanded.
- In nested environments ending at the same time, only the endmark for the inner environment is set, as the following example shows:

```
\begin{Lemma}  
  Some text.  
  \begin{Proof} The Proof \end{Proof}  
\end{Lemma}
```

yields to

5 Lemma:
Some text.

PROOF The Proof

■

You can handle this by specifying something invisible after the end of the inner theorem. Then the endmark for the outer theorem is set in the next line:

```

\begin{Lemma}
  Some text.
  \begin{Proof} The Proof \end{Proof}~
\end{Lemma}

```

yields to

6 Lemma:
Some text.

PROOF The Proof

■

◇

- Document option `fleqn` is problematic: `fleqn` handles equations not by `$$` but by lists (check what happens for

```

\begin{theorem} \[ displaymath \] \end{theorem}

```

in standard \LaTeX : The `displaymath` is *not* set in an own line). Also, for long formulas, the equation number and the endmark are smashed into the formula at the right text margin.

- Naturally, `ntheorem.sty` will not work correctly in combination with other styles which change the handling of

1. theorem-like environments, or
2. environments concerned with the handling of endmarks, e.g. `\[...\]`, `eqnarray`, etc.

- `ntheorem.sty` is compatible with Frank Mittelbach's `theorem.sty`, which is the most widespread style for setting theorems.

It cannot be used *with* `theorem.sty`, but it can be used instead of it.

6.2 Known “Bugs” and Problems

- Ending a theorem *directly* after the text, e.g.

```

\begin{Theorem} text\end{Theorem}

```

suppresses the endmark:

15 Theorem:
text

Therefore a space or a newline should be inserted before `\end{...}`.

- With `theoremstyle break`, if the linebreak would cause ugly linebreaking in the following text, it is suppressed.

6.3 Open Questions

- For equations, we decided to put the endmark after the equation number, which is vertically centered. Currently, we do not know, how to get the equation number centered and the endmark at the bottom (one has to know the internal height of the math material).
- The placement of endmarks is mainly based on a check whether \LaTeX is in an ordinary text line when encountering an end-of-environment. This question is *partially* answered by `\ifhmode`: In a text line, \LaTeX is always in `\hmode`. But, after an `displaymath`, \LaTeX is also in `\hmode`. Thus, additionally `\lastskip` is checked: after a `displaymath`, `\lastskip=0` holds. In most situations, when text has been written into a line, `\lastskip` $\neq 0$. But, this does not hold, if the source code is of the following form: `...text\label{bla}`: then, `\lastskip=0`. In those situations, the endmark is suppressed.
?? How can it be detected whether \LaTeX has just ended a `displaymath`?
- The above problem with the label: The break style enforces a linebreak by `\hfill\penalty-8000` after the `\trivlist`-item. Thus, \TeX gets back into the horizontal mode. The label places a “whatsit” somewhere ... and, it seems that the “whatsit” makes \TeX think that there is a line of text.

If someone has a solution to one of those questions, please inform us. (You can be sure to be mentioned in the Acknowledgements.)

7 Code Documentation

7.1 Documentation of the Macros

```
1 \typeout{Style ‘\basename’, Version \fileversion\space <\filedate>}
2 \ProvidesPackage{ntheorem}[\filedate \space\fileversion]
3 \newif\if@thmmarks\@thmmarksfalse
4 \newif\if@thref\@threffalse
5 \newif\ifthm@tempif
```

general setup.

7.1.1 Thmmarks-Related Stuff

```
1 \DeclareOption{thmmarks}{%*****
2 \PackageInfo{\basename}{Option ‘thmmarks’ loaded}%
3 %
4 \@thmmarkstrue
5 \newcounter{endNonectr}
6 \newcounter{currNonectr}
7 \newif\ifsetendmark\setendmarktrue
```


can be put simply at the end of the line):

Line 16: case split: if in `hmode` and `\lastskip > 0`, then \TeX is in a text line, the endmark is set here.

Changes to Math Environments

Original: `ltmath.dtx`

```
\endequation For equations, end marks are placed behind the equation number:
27 \gdef\SetMark@endeqn{\quad}% as default, cf. option leqno
28 \gdef\endequation{\eqno \hbox{\@eqnnum \PotEndMark{\SetMark@endeqn}}}%
29 $$$\global\@ignoretrue}
```

Line 27: As default, work for equation numbers at the right: Then, a `\quad` is placed between equation number and endmark.

Line 28: In addition to the equation number (set by `\@eqnnum` at the right of the line) `\SetMark@endeqn` is carried out.

`\[` If an end mark is set, a `displaymath` is put into box such that the end marks appears at its bottom level at the right. Thus, also the definition of `\[` has to be changed:

```
30 \gdef\[{%
31   \relax\ifmmode
32     \@badmath
33   \else
34     \ifvmode
35       \nointerlineskip
36       \makebox[.6\linewidth]%
37     \fi
38     $$$\stepcounter{end\InTheoType ctr}%
39     \@ifundefined{mark\thm@romannum{curr\InTheoType ctr}%
40               \InTheoType\thm@romannum{end\InTheoType ctr}}{\relax}%
41     {\ifx\curname\InTheoType Symbol\endcurname\@empty\else
42       \boxmaxdepth=.5ex\begin{array}[b]{@{}l}%
43       \boxmaxdepth=\maxdimen\displaystyle\fi}%
44     \addtocounter{end\InTheoType ctr}{-1}%
45     %$$$ BRACE MATCH HACK
46   \fi}
```

Lines 31–37, 45, 46: the old definition.

Lines 38–41: The end position of a `displaymath` inside a theorem-environment corresponds to `end\InTheoType ctr+1`. An endmark has to be set there, if

`\mark<\thm@romannum{curr#1ctr}>#1 <\thm@romannum{end#1ctr}+1 >`

is defined and not the empty symbol.

Lines 42–43: If so, the whole displayed stuff is put in an array with maximal depth `0.5ex` and vertically adjusted with its bottom line (then, the endmarks will appear adjusted to its bottom line).

Line 44: The counter has to be re-decremented.

\] At the end of a displaymath, the end marks is set at its bottom level:

```

47 \gdef\]{%
48     \stepcounter{end\InTheoType ctr}%
49     \@ifundefined{mark\thm@romannum{curr\InTheoType ctr}%
50         \InTheoType\thm@romannum{end\InTheoType ctr}}{\relax}%
51         {\ifx\csname\InTheoType Symbol\endcsname\@empty\else
52             \end{array}\fi}%
53     \addtocounter{end\InTheoType ctr}{-1}%
54 \relax\ifmmode
55     \ifinner
56         \@badmath
57     \else
58         \PotEndMark{\eqno}\global\@ignoretrue$$$$ BRACE MATCH HACK
59     \fi
60 \else
61     \@badmath
62 \fi
63 \ignorespaces}

```

Lines 48–53: Look, if an endmark has to be set in this displaymath (analogous to lines 38–44 of `\def\[]`) If so, there is an inner array which has to be closed (line 52).

Lines 54–63: the old definition.

Line 58: changed to set an endmark at the right of the line if necessary (this is done by `\eqno`).

`\endeqnarray` For `\eqnarrays`, the end marks is set below the number of the last equation:

```

64 \gdef\SetMark@endeqnarray#1{\llap{\raisebox{-1.3em}{#1}}}
65 \gdef\endeqnarray{%
66     \global\let\Oldeqnnum=\@eqnnum
67     \gdef\@eqnnum{\Oldeqnnum\PotEndMark{\SetMark@endeqnarray}}%
68     \@eqnncr
69     \egroup
70     \global\advance\c@equation\m@ne
71     $$$\global\@ignoretrue
72     \global\let\@eqnnum\Oldeqnnum}

```

Line 64: As default work for equation numbers at the right: Then, the endmark is placed below the last equation number at the right margin.

New: Lines 66, 67, 72:

Line 66: save `\@eqnnum`.

Line 67: define `\@eqnnum` to carry out `\Oldeqnnum`, then a potential endmark position is handled: if an endmark is set, between the equation number and the endmark, the command sequence `\SetMark@endeqnarray` is

carried out – there, since `\SetMark@endeqnarray` is a function of one argument, the endmark will be this argument.

Lines 68–71: from `latex.ltx`. Line 68 sets the equation number.

Line 72: restore `\@eqnnum`.

`\endeqnarray*` In an `\eqnarray*`, the end mark is set at the right of the last equation:

```

73 \@namedef{endeqnarray*}{%
74     % from \@eqnocr:
75     \let\reserved@a\relax
76     \ifcase\@eqcnt \def\reserved@a{& & }\or \def\reserved@a{& & }%
77     \or \def\reserved@a{& }\else
78         \let\reserved@a\empty
79     \latex@error{Too many columns in eqnarray environment}\@ehc\fi
80     \reserved@a {\normalfont \normalcolor \PotEndMark{}}}%
81     \global\@eqnswtrue\global\@eqcnt\z@\cr
82     %
83     \egroup
84     \global\advance\c@equation\m@ne
85     $$\global\@ignoretrue}

```

This is just L^AT_EX's `\endeqnarray` where lines 75–81 are inserted from `\@eqnocr` and augmented (line 80) to set a potential endmark (with no additional commands) at the end of the current line.

Changes to Tabbing Environment

Original: `lftab.dtx`

`\endtabbing` Here, the `\endtrivlist` modification is not sufficient: L^AT_EX is not in hmode when it calls `\endtrivlist` from `\endtabbing`; additionally, `\@stopline` already outputs a linebreak. Thus, the end mark is inserted *before* `\@stopline` at the right margin (using `\'`).

```

86 \gdef\endtabbing{%
87     \PotEndMark{\'}\@stopline\ifnum\@tabpush >\z@ \badpoptabs
88     \fi\endtrivlist}

```

Changes to Center Environment

Original: `ltmiscen.dtx`

`\endcenter` In L^AT_EX, `\endcenter` just calls `\endtrivlist`. Here, the situation is more complex since the the endmark has to be put in the last line without affecting its centering: if in a text line (only then, here is a potential endmark position):

```

89 \gdef\endcenter{%
90     \@endtrivlist
91     {\PotEndMark{\rightskip0pt%
92         \settowidth{\leftskip}%

```

```

93      { \csname mark\thm@romannum{curr\InTheoType ctr}\InTheoType
94          \thm@romannum{end\InTheoType ctr}\endcsname}%
95      \advance\leftskip\@flushglue\hskip\@flushglue}}

```

The `\rightskip` of the line is set to 0, `\leftskip` is set to the width of one space (since on the right, one space is added after the text) plus the endmark and infinitely stretchable glue (`\@flushglue`), and also the line is continued with `\@flushglue` (the actual position is one space after the text), and then the endmark is placed (by `\PotEndMark`).

Handling of Endmarks

`\@endtheorem-thmmarks` `\@endtheorem` is called for every `\end{<env>}`, where `<env>` is a theorem-like environment. `\@endtheorem` is extended to organize the placement of the corresponding end mark (`\InTheoType` gives the innermost theorem-like environment, i.e. the one to be ended):

```

96 \gdef\@empty{}
97 \gdef\@endtheorem{%
98     \expandafter
99     \ifx\csname\InTheoType Symbol\endcsname\@empty\setendmarkfalse\fi
100 \@endtrivlist
101     {\ifsetendmark
102         \unskip\nobreak\hfill\nobreak\csname\InTheoType Symbol\endcsname
103         \setendmarkfalse \fi}%
104 \ifsetendmark\OrganizeTheoremSymbol\else\global\setendmarktrue\fi
105 \csname\InTheoType @postwork\endcsname
106 }

```

Lines 98, 99: if the end symbol of the environment `<env>` to be closed is empty, simply no end symbol has to be set (it makes a difference, if no end symbol is set, or if an empty end symbol is set).

Lines 100, 104: (originally, it calls `\endtrivlist`):

Lines 100, 102, 103: `\@endtrivlist` is called to put `<env>Symbol` at the end of the line and set `setendmark` to false if `TEX` is in a text line and `setendmark` is true.

At this point, `setendmark` is false iff the user has disabled it locally or the end symbol is empty.

Line 101: the endmark is not set, if `setendmark` is false.

Line 104: if `setendmark` is true, the correct placement of the end symbol is organized, else (ie either `setendmarkfalse` is set by the user, or the endmark is already set by `\@endtrivlist`) reset `setendmark` to true.

For further comments see `\@endtrivlist` and `\OrganizeTheoremSymbol`.

The construction in line 102 guarantees that the endmark is put at the end of the line, even if it is the only letter in this line.

`\NoEndMark` By `\NoEndMark`, the automatical setting of an end mark is blocked for the *current* environment.

```
107 \gdef\NoEndMark{\global\setendmarkfalse}
```

set `setendmark` to false. It is automatically reset to `true` after the end of the current environment.

`\qed` With `\qed`, the user can locally change the end symbol to appear:

```
108 \gdef\qed{\expandafter\def\csname \InTheoType Symbol\endcsname
109         {\the\qedsymbol}}%
```

When calling `\qed`, the end symbol of the innermost theorem-like environment at that time is set to the value stored in `\qedsymbol` at that time.

`\PotEndMark` Handling a potential endmark position:

```
110 \gdef\PotEndMark#1{\SetEndMark{\InTheoType}{#1}}%
```

Argument: `<cmd_seq>:=#1` is a command sequence to be executed when setting the endmark.

It adds the current theorem type `<env>` to the parameters, and calls `\PotEndMark{<env>}{<cmd_seq>}`.

`\SetEndMark` `\SetEndMark` sets an endmark for an environment. It is called by `\PotEndMark`.

```
111 \gdef\SetEndMark#1#2{%
112     \stepcounter{end#1ctr}%
113     \@ifundefined{mark\thm@romannum{curr#1ctr}#1\thm@romannum{end#1ctr}}{%
114         {\relax}%
115         {#2{\csname mark\thm@romannum{curr#1ctr}#1\thm@romannum{end#1ctr}\endcsname
116             \ifdim\rightmargin>\z@\hskip-\rightmargin\fi
117             \hbox to 0cm{}}}}%
```

Arguments:

`<env>:=#1`: current theorem-environment.

`<cmd_seq>:= #2`: is a command sequence to be executed when setting the endmark. Both arguments are transmitted from by `\PotEndMark`.

Line 112: increments `end<env>ctr` for preparing the next situation for setting a potential endmark.

Line 113, 114: if

$$\backslash\mathrm{mark}<\backslash\mathrm{thm@romannum}\{\mathrm{curr}\langle\mathrm{env}\rangle\mathrm{ctr}\}>\langle\mathrm{env}\rangle<\backslash\mathrm{thm@romannum}\{\mathrm{end}\langle\mathrm{env}\rangle\mathrm{ctr}\}>$$

is undefined – which is the case iff at this position no endmark has to be set –, nothing is done,

Line 115: otherwise, `<cmd_seq>` and then

$$\backslash\mathrm{mark}<\backslash\mathrm{thm@romannum}\{\mathrm{curr}\langle\mathrm{env}\rangle\mathrm{ctr}\}>\backslash\mathrm{env}<\backslash\mathrm{thm@romannum}\{\mathrm{end}\langle\mathrm{env}\rangle\mathrm{ctr}\}>,</math>$$

which is defined in the .aux file to be the end symbol are called.
The construction $\langle cmd_seq \rangle \{ \dots \}$ in line 115 allows the handling of the end symbol as an argument of $\langle cmd_seq \rangle$ as needed for `\endeqnarray`.

Line 116: By `\hskip-\rightmargin\hbox to 0cm{}`, a negative hspace of amount `\rightmargin` is added *after* the end symbol – thus, the symbol is set as there were no right margin (this concerns, e.g., `\quote` environments).

(applied only if `\rightmargin` is more than 0 – otherwise bug if preceding line ends with hyphenation.)

Writing to .aux file. (copied from `\def\label (ltxref.dtx)`)

```
118 \newskip\mysavskip
119 \gdef\@bbsphack{%
120   \ifvmode\else\mysavskip\lastskip
121   \unskip\fi}
122 %
123 \gdef\@eesphack{%
124   \ifdim\mysavskip>\z@
125     \vskip\mysavskip \else\fi}
```

Lines 119–121 and 122–124 are similar to `\@bsphack` and `\@bsphack` of `latex.ltx`. They undo resp. redo the last `skip`.

Note that `@bbsphack` and `@eesphack` are also part of the `thref` option. Change both if you change them.

`\OrganizeTheoremSymbol` The information for setting the end marks is written to the .aux file:

```
126 \gdef\OrganizeTheoremSymbol{%
127   \@bbsphack
128   \edef\thm@tmp{\expandafter\expandafter\expandafter\thm@meaning
129     \expandafter\meaning\csname\InTheoType Symbol\endcsname\relax}%
130   \protected@write\@auxout{%
131     {\string\global\string\def\string\mark%
132       \thm@romannum{curr\InTheoType ctr}\InTheoType \thm@romannum{end\InTheoType ctr}%
133       {\thm@tmp}}}%
134   \@eesphack}
```

Lines 130–132: Write

`\global\def\mark<\thm@romannum{curr}\langle env \rangle ctr> \langle env \rangle <\thm@romannum{end}\langle env \rangle ctr>{\langle env \rangle Symbol}>` to the .aux file.
 $\langle env \rangle := \text{\InTheoType}$ gives the innermost theorem-like environment, i.e. the one the end symbol has to be set for.

```
135 } % end of option [thmmarks]
```

7.1.2 Option `leqno` to `Thmmarks`

```
136 \DeclareOption{leqno}{% *****
```

```

137 \if@thmmarks
138 \PackageInfo{\basename}{Option 'leqno' loaded}%
139 \gdef\SetMark@endeqn#1{\hss\llap{#1}}
140 \gdef\SetMark@endeqnarray#1{\hss\llap{#1}}
141 \fi}%

```

`leqno` is only active if `thmmarks` is also active.

Line 139, 140: Since with `leqno`, the equation number is placed on the left, after infinitely stretchable glue, the endmark can be set straight at the right margin.

7.1.3 Option `fleqn` to `Thmmarks`

```

142 \DeclareOption{fleqn}{% *****
143 \if@thmmarks
144 \PackageInfo{\basename}{Option 'fleqn' loaded}%

```

`fleqn` is only active if `thmmarks` is also active.

\[Since `fleqn` treats displayed math as trivlists, it's quite another thing:

```

145 \renewcommand\[{ \relax
146 \ifmmode\@badmath
147 \else
148 \begin{trivlist}%
149 \@beginparpenalty\predisplaypenalty
150 \@endparpenalty\postdisplaypenalty
151 \item[]\leavevmode
152 \hb@xt@\linewidth\group $\m@th\displaystyle %$
153 \hskip\mathindent\group
154 \stepcounter{end\InTheoType ctr}%
155 \@ifundefined{mark\thm@romannum{curr\InTheoType ctr}}%
156 \InTheoType\thm@romannum{end\InTheoType ctr}}{\relax}%
157 {\ifx\cname\InTheoType Symbol\endcsname\@empty\else
158 \boxmaxdepth=.5ex\begin{array}[b]{@{}l}%
159 \boxmaxdepth=\maxdimen\displaystyle\fi}%
160 \addtocounter{end\InTheoType ctr}{-1}%
161 \fi}

```

Lines 145–153, 161: the old definition.

Line 154–160: if an endmark has to be set in this displaymath, it is put into an array with depth $\leq 0.5ex$, and vertically adjusted to the bottom line.

\] Here, the end mark is placed after a `\hfil` at the end of the line containing the displaymath:

```

162 \renewcommand\]{%
163 \stepcounter{end\InTheoType ctr}%
164 \@ifundefined{mark\thm@romannum{curr\InTheoType ctr}}%
165 \InTheoType\thm@romannum{end\InTheoType ctr}}{\relax}%

```

```

166      {\ifx\csname\InTheoType Symbol\endcsname\@empty\else
167        \end{array}\fi}%
168      \addtocounter{end\InTheoType ctr}{-1}%
169      \relax\ifmmode
170        \egroup $\hfil\PotEndMark{}\% $
171        \egroup
172        \end{trivlist}%
173      \else \@badmath
174      \fi}

```

Lines 163–167: Look, if an endmark has to be set in this displaymath. If so, close the inner array.

Lines 169–174: the old definition.

Line 170: Added `\PotEndMark`.

`\endequation` for equations, the end mark is also set with the equation number:

```

175 \gdef\endequation{%
176   $\hfil % $
177   \displaywidth\linewidth\hbox{\@eqnnum \PotEndMark{\SetMark@endeqn}}}%
178   \egroup
179   \endtrivlist}

```

Line 177: When the equation number is set, also the endmark is set with the same trick as for `\endequation` without `fleqn`.

`\endeqnarray` When the equation number is set, also the endmark is set with the same trick as for `\endeqnarray` without `fleqn` (see Lines 181, 182, 187):

```

180 \gdef\endeqnarray{%
181   \global\let\Oldeqnnum=\@eqnnum
182   \gdef\@eqnnum{\Oldeqnnum\PotEndMark{\SetMark@endeqnarray}}%
183   \@eqncr
184   \egroup
185   \global\advance\c@equation\m@ne$$% $$
186   \global\@ignoretrue
187   \global\let\@eqnnum\Oldeqnnum}
188 \fi}% end of option fleqn

```

7.1.4 Extended Referencing Facilities

```

189 \DeclareOption{thref}{%*****
190   \PackageInfo{\basename}{Option ‘thref’ loaded}%
191   \@threftrue

```

Option “thref” needs a special handling when combined with `amsmath`. This is also a reason why it is handled first.

`\bbsphack(2)`

```

192 \newskip\mysavskip
193 \gdef\bbsphack{%

```

```

194 \ifvmode\else\mysavskip\lastskip
195 \unskip\fi}
196 %
197 \gdef\@eesphack{%
198 \ifdim\mysavskip>\z@
199 \vskip\mysavskip \else\fi}

```

Note that `@bbsphack` and `@eesphack` are also part of the `thmmarks` option. Change both if you change them.

`\label` The original `\label` macro is extended (cf. `ltxref.dtx`) with an optional argument, containing the type of the labeled construct:

```

200 \def\label#1{%
201 \ifnextchar[%]
202 {\label@optarg{#1}}%
203 {\thm@makelabel{#1}}%
204 %
205 \def\thm@makelabel#1{%
206 \@bbsphack
207 \edef\thm@tmp{\expandafter\expandafter\expandafter\thm@meaning
208 \expandafter\meaning\csname\InTheoType Keyword\endcsname\relax}%
209 \protected@write\@auxout{%
210 {\string\newlabel{#1}{\@currentlabel}{\thepage}}[\thm@tmp]}%
211 \@eesphack}
212 %
213 \def\label@optarg#1[#2]{%
214 \@bsphack
215 \protected@write\@auxout{%
216 {\string\newlabel{#1}{\@currentlabel}{\thepage}}[#2]}%
217 \@esphack}

```

`thm@makelabel`: If no optional argument is given, the keyword of the current environment type is used instead.

`label@optarg`: The original definition, extended with the optional argument which is appended to the `\newlabel`-command to be written to the `.aux`-file.

`\newlabel` The original behavior of `\newlabel` (called when evaluating the `.aux`-file) is also adapted.

Original syntax: `\newlabel{<label>}{<section>}{<page>}`

Modified syntax: `\newlabel{<label>}{<section>}{<page>}[<type>]`

Definition of `\newlabel`: `\def\newlabel{\@newl@bel r}`.

Therefore, the modification is encoded into the `\@newl@bel` macro:

```

218 \def\@newl@bel#1#2#3{%
219 \@ifpackageloaded{babel}{\@safe@activestrue}\relax%
220 \@ifundefined{#1@#2}%
221 \relax
222 {\gdef \@multiplelabels {%

```



```

223 \latex@warning@no@line{There were multiply-defined labels}}%
224 \latex@warning@no@line{Label ‘#2’ multiply defined}}%
225 \global\@namedef{#1@#2}{#3}%
226 \ifnextchar[{\set@label@type{#1}{#2}}%]
227 \relax}%
228 \def\set@label@type#1#2[#3]{%
229 \global\@namedef{#1@#2@type}{#3}}

```

the macro is called with three arguments (same as originally):

#1=**r**,
 $\langle labelname \rangle := \mathbf{\#2}$ is the label name,
#3 is a pair (section, page-number) consisting of the values needed for **\ref**
and **\pageref**, respectively.

Line 219: adaptation to babel

Lines 220–225: The original definition (both standard L^AT_EX and babel).

Line 226: if an optional argument follows (containing the environment-type),
continue with **\set@label@type**, otherwise return (the original behavior).

Lines 228,229: set **\r@ $\langle labelname \rangle$ @type** to the type of the respective environment.

\testdef A problem occurred, when about 250 labels to theorem-like environments have been defined: after the end of a document, the .aux file is read once more (to check if references changed). Here, L^AT_EX redefines **\newl@bel** into **\@testdef** – and L^AT_EX does not know that ntheorem’s **\label** has an additional optional argument. Thus, the argument values are not processed, but are output as normal text. Normally, this did not matter since output has already been finished by a **\clearpage** in **\end{document}**. For so many labels, a page gets filled and the output routine is called.

```

230 \newcommand\org@testdef{}
231 \let\org@testdef\@testdef
232 \def\@testdef#1#2#3{%
233 \org@testdef{#1}{#2}{#3}%
234 \ifnextchar[{\@gobbleopt}{}%
235 }
236 \newcommand\@gobbleopt{}
237 \long\def\@gobbleopt[#1]{}

```

Line 233: process the optional argument.

\thref **\thref** is an adaptation of **\ref**:

```

238 \def\thref#1{%
239 \expandafter\ifx\csname r@#1@type\endcsname\None
240 \PackageWarning{\basename}{thref: Reference Type of ‘#1’ on page
241 \thepage \space undefined}\G@refundefinedtrue
242 \else\csname r@#1@type\endcsname~\fi%
243 \expandafter\@setref\csname r@#1\endcsname\@firstoftwo{#1}}

```

Lines 238,243: similar to `\ref`.

Line 219: if a legal theorem type is given, then output `\r@<labelname>@type` and avoid linebreaking between the type and the number.

```
244 }% end of option thref *****
```

7.1.5 Option `amsmath` to `Thmmarks`

Most of the commands are extensions of commands in `amsmath.sty`.

```
245 \DeclareOption{amsmath}{% *****
246 \if@thref
247 \PackageInfo{\basename}{option 'amsmath' handling for 'thref' loaded}%
```

if `thref` is active, the handling of labels in `amsmath` equations has also to be adapted.

`ams-thref`

```
248 \let\ltx@label\label
keep the handling of \label ... (the one defined above in the thref option).
amsmath implements a special handling of \label inside of displaymath
environments. It is extended to process the optional argument provided by
the thref option:

249 \global\let\thm@df@label@optarg\@empty
250 \def\label@in@display#1{%
251   \ifx\df@label\@empty\else
252     \@amsmath@err{Multiple \string\label's:
253       label '\df@label' will be lost}\@eha
254   \fi
255   \gdef\df@label{#1}%
256   \@ifnextchar[{\thm@label@in@display@optarg}{\thm@label@in@display@noarg}%]
257 }
258 \def\thm@label@in@display@noarg{%
259   \global\let\thm@df@label@optarg\@empty
260 }
261 \def\thm@label@in@display@optarg[#1]{%
262   \gdef\thm@df@label@optarg{#1}%
263 }
```

The contents of `\df@label` is handled when the equation is finished. (Currently) this happens in three macros. The modification consists of the check if `\thm@df@label@optarg` is non-empty (i.e., holds the optional argument), and to handle it.

```
264 \def\endmathdisplay@a{%
265   \if@eqnsw \gdef\df@tag{\tagform@theequation}\fi
266   \if@fleqn \exp\endmathdisplay@fleqn
267   \else \ifx\df@tag\@empty \else \veqno \alt@tag \df@tag \fi
268   \ifx\df@label\@empty \else
```

```

269     \ifx\thm@df@label@optarg\@empty \exp\ltx@label\exp{\df@label}
270         \else \exp\ltx@label\exp{\df@label}[\thm@df@label@optarg]\fi
271     \fi
272 \fi
273 \ifnum\dspbrk@lvl>\m@ne
274     \postdisplaypenalty -\@getpen\dspbrk@lvl
275     \global\dspbrk@lvl\m@ne
276 \fi
277 }
278 \def\make@display@tag{%
279     \if@eqnsw
280         \refstepcounter{equation}%
281         \tagform@\theequation
282     \else
283         \iftag@
284             \df@tag
285             \global\let\df@tag\@empty
286         \fi
287     \fi
288     \ifmeasuring@
289     \else
290         \ifx\df@label\@empty\else
291             \ifx\thm@df@label@optarg\@empty \exp\ltx@label\exp{\df@label}
292                 \else \exp\ltx@label\exp{\df@label}[\thm@df@label@optarg]\fi
293             \global\let\df@label\@empty
294         \fi
295     \fi
296 }
297 \def\endmathdisplay@fleqn{%
298     $\hfil\hskip\@mathmargin\egroup
299     \ifnum\badness<\inf@bad \let\too@wide\@ne \else \let\too@wide\z@ \fi
300     \ifx\@empty\df@tag
301     \else
302         \setbox4\hbox{\df@tag
303             \ifx\thm@df@label@optarg\@empty \exp\ltx@label\exp{\df@label}
304                 \else \exp\ltx@label\exp{\df@label}[\thm@df@label@optarg]\fi
305         }%
306     \fi
307     \csname emdf@%
308         \ifx\df@tag\@empty U\else \iftagsleft@ L\else R\fi\fi
309     \endcsname
310 }

311 \fi
312 % end of option amsmath/thref *****
313 \if@thmmarks
314 \PackageInfo{\basename}{option 'amsmath' handling for 'thmmarks' loaded}%
315 \newdimen\thm@amstmpdepth

```

A temporarily used register.

`\TagsPlusEndmarks` Since `amsmath` uses “tags” for setting end marks, some macros are defined which prepare tags which include endmarks:

```

316 \gdef\TagsPlusEndmarks{%
317     \global\let\Old@maketag@@@=\maketag@@@
318     \global\let\Old@df@tag=\df@tag
319     \if@eqnsw\SetTagPlusEndMark
320     \else
321         \iftag@\SetTagPlusEndMark
322         \else\SetOnlyEndMark
323     \fi
324 \fi}

```

Lines 317, 318: store the original macros.

Line 319: if equation numbers are set as default, call `\SetTagPlusEndMark` to set tag and end mark.

Lines 320, 321: if a tag is set manually, call `\SetTagPlusEndMark` to set tag and end mark.

Line 322: otherwise, call `\SetOnlyEndMark` to set only an end mark.

`\SetOnlyEndMark`

```

325 \gdef\SetOnlyEndMark{%
326     \global\tag@true
327     \iftagsleft@
328         \gdef\df@tag{\hbox
329             to \displaywidth{\hss\PotEndMark{\maketag@@@}}}%
330     \else
331         \gdef\df@tag{\PotEndMark{\maketag@@@}}%
332     \fi}

```

Set only an end mark:

Line 326: force setting the end mark as a tag:

Lines 328,329: if tags are set to the left, the tag consists of a `\hbox` over the whole displaywidth, with the (potential) endmark at its right.

Line 331: if tags are set to the right, the tag consists only of the (potential) endmark.

`\SetTagPlusEndMark`

```

333 \newdimen{\tagwidth}
334 \gdef\SetTagPlusEndMark{%
335     \iftagsleft@
336         \gdef\maketag@@@##1{%
337             \settowidth{\tagwidth}{$##1$}% % WM 17.10.2007
338             \hbox to \tagwidth{%
339                 \hbox to \displaywidth{\m@th\normalfont##1%
340                     \hss\PotEndMark{\hss}}\hss}%
341         \else
342             \gdef\maketag@@@##1{\hbox{\m@th\normalfont##1%

```

```

343                                     \llap{\hss\PotEndMark{\raisebox{-1.3em}{}}}}}%
344                                     \fi}

```

Set a tag *and* an end mark:

Lines 334–343: redefine the `\maketag@@@` macro:

Lines 335–339: if tags are set to the left, build a box of the whole displaywidth and put the original tag on the left, and the (potential) endmark at the right. Put this box with width 0 and continue.

Lines 340,341: if the tags are set to the right, the (potential) end mark is put below it.

`\RestoreTags`

```

345 \gdef\RestoreTags{%
346     \global\let\maketag@@@=\Old@maketag@@@
347     \global\let\df@tag=\Old@df@tag}

```

Lines 346,347: restore the original macros.

`\endgather` In the `gather` environment, just the augmented tag is used:

```

348 \gdef\endgather{%
349     \TagsPlusEndmarks % <<<<<<<<
350     \math@cr
351     \black@\totwidth@
352     \egroup
353     $$%
354     \RestoreTags % <<<<<<<<
355     \ignorespacesafterend}
356 %
357 \expandafter\let\csname endgather*\endcsname\endgather

```

New:

Line 349: the last tag contains the potential endmark.

Line 354: restore the original macros.

Line 357: Since `let` always takes the expansion of a macro when the `let` is executed, all `let`'s have to be adjusted (this is the same for all subsequent `let`-statements).

`\math@cr@@@align`

`\endalign` `\endalign` also uses the augmented tags:

```

358 \def\endalign{%
359     \ifingather@ \else % <<<<<<<<
360         \TagsPlusEndmarks\fi % <<<<<<<<
361     \math@cr
362     \black@\totwidth@
363     \egroup
364     \ifingather@
365     \restorealignstate@

```

```

366      \egroup
367      \nonumber
368      \ifnum0='{\fi\iffalse}\fi
369  \else
370      $$%
371      \RestoreTags          % <<<<<<<<
372  \fi
373  \ignorespacesafterend}

```

New:

Lines 359, 360: if the `align` is not inside another environment, its tags have to contain the endmarks.

Line 371: this case, the original macros have to be restored.

```

374 \expandafter\let\csname endalign*\endcsname\endalign
375 \let\endxalignat\endalign
376 \expandafter\let\csname endxalignat*\endcsname\endalign
377 \let\endxxalignat\endalign
378 \let\endalignat\endalign
379 \expandafter\let\csname endalignat*\endcsname\endalign
380 \let\endflalign\endalign
381 \expandafter\let\csname endflalign*\endcsname\endalign

```

Adjust let-statements.

`\lendmultline` The `multline` environment has two different `\end` commands, depending if the equation numbers are set on the left or on the right:

```

382 \def\lendmultline@{%
383     \global\@eqnswfalse\tag@false\tag@left@false
384     \rendmultline@}

```

End of `multline` environment if tags are set to the left: in this case, the last line of a `multline` does not contain a tag. Thus the situation of setting an endmark tag at the right is faked:

Lines 383, 384: display no equation number, don't set an equation tag (but use the tag mechanism for the end mark - see `\TagsPlusEndmarks` and `\SetOnlyEndMark`), set it at the right, and call `\rendmultline`.

`\rendmultline` `\rendmultline` also uses the augmented tags:

```

385 \def\rendmultline@{%
386     \TagsPlusEndmarks          % <<<<<<<<
387     \iftag@
388         $\let\endmultline@math\relax
389         \ifshifttag@
390             \hskip\multlinegap
391             \llap{\vtop{%
392                 \raise@tag
393                 \normalbaselines
394                 \setbox\@ne\null

```

```

395             \dp\@ne\lineht@
396             \box\@ne
397             \hbox{\strut@\make@display@tag}%
398         }}%
399     \else
400         \hskip\multlinetaggap
401         \make@display@tag
402     \fi
403 \else
404     \hskip\multlinegap
405 \fi
406 \hfilneg
407 \math@cr
408 \egroup$$%
409 \RestoreTags}          % <<<<<<<<

```

New:

Line 386: last tag contains the potential endmark.

Line 410: restore the original macros

\endmathdisplay

```

410 \def\endmathdisplay#1{%
411     \ifmode \else \@badmath \fi
412     \TagsPlusEndmarks % <<<<<<<<
413     \endmathdisplay@a
414     $$%
415     \RestoreTags          % <<<<<<<<
416     \global\let\df@label\@empty \global\let\df@tag\@empty
417     \global\tag@false \global\let\alt@tag\@empty
418     \global\@eqnswfalse
419 }

```

Added Line 413: set potential end mark at bottom niveau of displaymath.

equation

```

420 \renewenvironment{equation}{%
421     \edef\reset@equation{%
422         \@nx\setcounter{equation}{\number\c@equation}}%
423     \refstepcounter{equation}%
424     \st@rredfalse \global\@eqnswtrue
425     \mathdisplay{equation}%
426 }{%
427     \endmathdisplay{equation}%
428     \ignorespacesafterend
429 }
430 \renewenvironment{equation*}{%
431     \st@rredtrue \global\@eqnswfalse
432     \mathdisplay{equation*}%
433 }{%

```

```

434 \endmathdisplay{equation*}%
435 \ignorespacesafterend
436 }

unchanged from amsmath.sty.

437 \fi
438 }% end of option amsmath/thmmarks *****

```

7.1.6 Theorem-Layout Stuff

```

439 \let\thm@usestd\@undefined
440 \DeclareOption{standard}{\let\thm@usestd\relax}
441 \let\thm@noconfig\@undefined
442 \DeclareOption{noconfig}{\let\thm@noconfig\relax}

```

Options for selection of a configuration: if no such option is given `ntheorem.cfg` will be loaded (which has to be provided by the user), `[standard]` will load `ntheorem.std`, a predefined setting, and `[noconfig]` does not preload any configuration.

```

443 \gdef\InTheoType{None}
444 \gdef\NoneKeyword{None}
445 \gdef\NoneSymbol{None}
446 \gdef\None{None}

```

Set `\InTheoType` to `none` on the upper document level.

`\newtheoremstyle` With `\newtheoremstyle`, new theorem-layout styles are defined.

```

447 \gdef\newtheoremstyle#1#2#3{%
448 \expandafter\@ifundefined{th@#1}%
449 {\expandafter\gdef\csname th@#1\endcsname{%
450 \def\@begintheorem####1####2{#2}%
451 \def\@opargbegintheorem####1####2####3{#3}}}%
452 {\PackageError{\basename}{Theorem style #1 already defined}\@eha}}

```

Arguments:

$\langle style \rangle := \#1$: the name of the theoremstyle to be defined,

$\langle cmd_seq1 \rangle := \#2$: command sequence for setting the header for environment instances with no optional text,

$\langle cmd_seq2 \rangle := \#3$: command sequence for setting the header for environment instances with optional text.

Line 448: if this style is not yet defined, define it.

Line 449: define `\th@ $\langle style \rangle$` to be a macro which defines

Line 450: a) the two-argument macro `\@begintheorem#1#2` to be $\langle cmd_seq1 \rangle$,

Line 451: b) `\@opargbegintheorem#1#2#3` to be $\langle cmd_seq2 \rangle$.

The predefined theorem styles use this command.

`\renewtheoremstyle`

```

453 \gdef\renewtheoremstyle#1#2#3{%
454 \expandafter\@ifundefined{th@#1}%

```



```

455   {\PackageError{\basename}{Theorem style #1 undefined}\@ehc}%
456   {}%
457   \expandafter\let\csname th@#1\endcsname\relax
458   \newtheoremstyle{#1}{#2}{#3}}

```

Arguments:

$\langle style \rangle := \#1$: the name of the theoremstyle to be defined,

$\#2$, $\#3$ as for `\newtheoremstyle`.

Checks, if theoremstyle $\langle style \rangle$ is already defined. If so, `\th@ $\langle style \rangle$` is made undefined and `\newtheoremstyle` is called with the same arguments.

Predefined Theorem Styles

`theoremstyles` `th@plain`, `th@change`, and `th@margin` taken from `theorem.sty` by Frank Mittelbach; the break-styles have been changed.

```

459 \newtheoremstyle{plain}%
460   {\item[\hskip\labelsep \theorem@headerfont ##1\ ##2\theorem@separator]]}%
461   {\item[\hskip\labelsep \theorem@headerfont ##1\ ##2\ (##3)\theorem@separator]]}%
462   %
463 \newtheoremstyle{break}%
464   {\item[\rlap{\vbox{\hbox{\hskip\labelsep \theorem@headerfont
465     ##1\ ##2\theorem@separator}\hbox{\strut}}}}}%
466   {\item[\rlap{\vbox{\hbox{\hskip\labelsep \theorem@headerfont
467     ##1\ ##2\ (##3)\theorem@separator}\hbox{\strut}}}}}%
468   %
469 \newtheoremstyle{change}%
470   {\item[\hskip\labelsep \theorem@headerfont ##2\ ##1\theorem@separator]]}%
471   {\item[\hskip\labelsep \theorem@headerfont ##2\ ##1\ (##3)\theorem@separator]]}%
472   %
473 \newtheoremstyle{changebreak}%
474   {\item[\rlap{\vbox{\hbox{\hskip\labelsep \theorem@headerfont
475     ##2\ ##1\theorem@separator}\hbox{\strut}}}}}%
476   {\item[\rlap{\vbox{\hbox{\hskip\labelsep \theorem@headerfont
477     ##2\ ##1\ (##3)\theorem@separator}\hbox{\strut}}}}}%
478   %
479 \newtheoremstyle{margin}%
480   {\item[\theorem@headerfont \llap{##2}\hskip\labelsep ##1\theorem@separator]]}%
481   {\item[\theorem@headerfont \llap{##2}\hskip\labelsep ##1\ (##3)\theorem@separator]]}%
482   %
483 \newtheoremstyle{marginbreak}%
484   {\item[\rlap{\vbox{\hbox{\theorem@headerfont
485     \llap{##2}\hskip\labelsep\relax ##1\theorem@separator}\hbox{\strut}}}}}%
486   {\item[\rlap{\vbox{\hbox{\theorem@headerfont
487     \llap{##2}\hskip\labelsep\relax ##1\
488     (##3)\theorem@separator}\hbox{\strut}}}}}%
489   %
490 \newtheoremstyle{nonumberplain}%
491   {\item[\theorem@headerfont\hskip\labelsep ##1\theorem@separator]]}%

```

```

492 {\item[\theorem@headerfont\hskip \labelsep ##1\ (##3)\theorem@separator]}
493 %
494 \newtheoremstyle{nonumberbreak}%
495 {\item[\rlap{\vbox{\hbox{\hskip \labelsep \theorem@headerfont
496      ##1\theorem@separator}\hbox{\strut}}}}}%
497 {\item[\rlap{\vbox{\hbox{\hskip \labelsep \theorem@headerfont
498      ##1\ (##3)\theorem@separator}\hbox{\strut}}}}}%
499 %
500 \newtheoremstyle{empty}%
501 {\item[]}%
502 {\item[\theorem@headerfont \hskip \labelsep \relax ##3]}
503 \newtheoremstyle{emptybreak}%
504 {\item[]}%
505 {\item[\rlap{\vbox{\hbox{\hskip \labelsep \relax \theorem@headerfont
506      ##3\theorem@separator}\hbox{\strut}}}}}%
507 %
508 \@namedef{th@nonumbermargin}{\th@nonumberplain}
509 \@namedef{th@nonumberchange}{\th@nonumberplain}
510 \@namedef{th@nonumbermarginbreak}{\th@nonumberbreak}
511 \@namedef{th@nonumberchangebreak}{\th@nonumberbreak}
512 \@namedef{th@plainNo}{\th@nonumberplain}
513 \@namedef{th@breakNo}{\th@nonumberplain}
514 \@namedef{th@marginNo}{\th@nonumberplain}
515 \@namedef{th@changeNo}{\th@nonumberplain}
516 \@namedef{th@marginbreakNo}{\th@nonumberbreak}
517 \@namedef{th@changebreakNo}{\th@nonumberbreak}

```

For instance, `break` is commented:

`\newtheoremstyle{break}` results in

```

\gdef\th@break{%
  \def\@begintheorem##1##2{%
    \item[\hskip \labelsep \theorem@headerfont
      ##1\ ##2\theorem@separator]%
    \hfill\penalty-8000}%
  \def\@opargbegintheorem##1##2##3{%
    \item[\hskip \labelsep \theorem@headerfont
      ##1\ ##2\ (##3)\theorem@separator]%
    \hfill\penalty-8000}}

```

Then, calling `\th@break` sets `\@begintheorem` as follows:

Since each theorem environment is basically a trivial list, the header is set as the item contents: `\theorem@headerfont` holds the font commands for the header font, `##1` is the keyword to be displayed, and `##2` its environment number. The linebreak after the header is achieved by offering to fill the line with space and the distinct wish to put a linebreak after it. Thus, if plain text follows, the line break is executed, but if a list or a display follows, it is not executed.

Note: The `\hfill\penalty-8000` causes T_EX to leave vertical mode, setting the item contents (ie the header) and entering horizontal mode to perform the `\hfill`.

`\theoremstyle` The handling of `\theoremstyle`, `\theorembodyfont`, and `\theoremskipamounts` is taken from `theorem.sty` by Frank Mittelbach:

```
518 \gdef\theoremstyle#1{%
519   \ifundefined{th@#1}{\@warning
520     {Unknown theoremstyle '#1'. Using 'plain'}}%
521     \theorem@style{plain}}%
522   {\theorem@style{#1}}}%
523 \newtoks\theorem@style
524 \newtoks\theorem@@style
525 \global\theorem@style{plain}
```

If `\theoremstyle` is called, it is checked if the argument is a valid `theoremstyle`, and if so, it is stored in the token `\theorem@style`. It is initialized to `plain`.

`\theorembodyfont`

```
526 \newtoks\theorembodyfont
527 \global\theorembodyfont{\itshape}
```

`\theoremnumbering`

```
528 \newtoks\theoremnumbering
529 \global\theoremnumbering{arabic}
```

`\theorempreskipamount`

```
\theorempostskipamount 530 \newskip\theorempreskipamount
531 \newskip\theorempostskipamount
532 \global\theorempreskipamount\topsep
533 \global\theorempostskipamount\topsep
```

`\theoremindent`

```
534 \newdimen\theoremindent
535 \global\theoremindent0cm
536 \newdimen\theorem@indent
```

`\theoremheaderfont`

```
537 \newtoks\theoremheaderfont
538 \global\theoremheaderfont{\normalfont\bfseries}
539 \def\theorem@headerfont{\normalfont\bfseries}
```

`\theoremseparator`

```
540 \newtoks\theoremseparator
541 \global\theoremseparator{}
542 \def\theorem@separator{}
```

```

\theoremprework
\theorempostwork 543 \newtoks\theoremprework
                  544 \global\theoremprework{\relax}
                  545 \newtoks\theorempostwork
                  546 \global\theorempostwork{\relax}
                  547 \def\theorem@prework{}

\theoremsymbol
                  548 \newtoks\theoremsymbol
                  549 \global\theoremsymbol{}

\qedsymbol
                  550 \newtoks\qedsymbol
                  551 \global\qedsymbol{}

\theoremkeyword
                  552 \newtoks\theoremkeyword
                  553 \global\theoremkeyword{None}

\theoremclass
                  554 \gdef\theoremclass#1{%
                  555     \csname th@class@#1\endcsname}
                  556 \gdef\th@class@LaTeX{%
                  557     \theoremstyle{plain}
                  558     \theoremheaderfont{\normalfont\bfseries}
                  559     \theorembodyfont{\itshape}
                  560     \theoremseparator{}
                  561     \theoremprework{\relax}
                  562     \theorempostwork{\relax}
                  563     \theoremindent0cm
                  564     \theoremnumbering{arabic}
                  565     \theoremsymbol{}}

```

Calling `\theoremclass{<env>}` calls `\th@class@<env>` (which is defined in `\@newtheorem` in Lines –45667). `\th@class@<env>` restores all style parameters to their values given for `<env>`. Especially, `\th@class@LaTeX` restores the standard LaTeX parameters.

```

\qedsymbol
                  566 \newtoks\qedsymbol
                  567 \global\qedsymbol{}

```

Compatibility with amsthm.

```

amsthm
                  568 \DeclareOption{amsthm}{% *****
                  569     \PackageInfo{\basename}{Option ‘amsthm’ loaded}%
                  570 \def\swapnumbers{\PackageError{\basename}{swapnumbers not implemented.

```

```

571 Use theoremstyle change instead.}\@eha}
572
573 \gdef\th@plain{%
574   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
575   \def\@begintheorem##1##2{%
576     \item[\hskip\labelsep \theorem@headerfont ##1\ ##2.]}%
577   \def\@opargbegintheorem##1##2##3{%
578     \item[\hskip\labelsep \theorem@headerfont ##1\ ##2\ (##3).]}%
579 \gdef\th@nonumberplain{%
580   \def\theorem@headerfont{\normalfont\bfseries}\itshape%
581   \def\@begintheorem##1##2{%
582     \item[\hskip\labelsep \theorem@headerfont ##1.]}%
583   \def\@opargbegintheorem##1##2##3{%
584     \item[\hskip\labelsep \theorem@headerfont ##1\ (##3).]}%
585 \gdef\th@definition{%
586   \th@plain\def\theorem@headerfont{\normalfont\bfseries}\itshape}
587 \gdef\th@nonumberdefinition{%
588   \th@nonumberplain\def\theorem@headerfont{\normalfont\bfseries}\itshape}
589 \gdef\th@remark{%
590   \th@plain\def\theorem@headerfont{\itshape}\normalfont}
591 \gdef\th@nonumberremark{%
592   \th@nonumberplain\def\theorem@headerfont{\itshape}\normalfont}
593 \newcounter{proof}%
594 \if@thmmarks
595   \newcounter{currproofctr}%
596   \newcounter{endproofctr}%
597 \fi
598 \newcommand{\openbox}{\leavevmode
599   \hbox to.77778em{%
600     \hfil\vrule
601     \vbox to.675em{\hrule width.6em\vfil\hrule}%
602     \vrule\hfil}}
603 \gdef\proofSymbol{\openbox}
604 \newcommand{\proofname}{Proof}
605 \newenvironment{proof}[1][\proofname]{
606   \th@nonumberplain
607   \def\theorem@headerfont{\itshape}%
608   \normalfont
609   \theoremsymbol{\ensuremath{\_}\blacksquare}}
610   \@thm{proof}{proof}{#1}}%
611   {\@endtheorem}
612 }% end of option amsthm *****

```

Defines theorem styles plain, definition, and remark, and environment proof according to amsthm.sty.

7.1.7 Theorem-Environment Handling Stuff

Original: ltthm.dtx

```
613 \newskip\thm@topsepadd
```

An auxiliary variable.

Defining New Theorem-Environments.

`\newtheorem`

```
614 \gdef\newtheorem{%
615   \newtheorem@i%
616 }
```

`\newtheorem@i` The syntax of the original `\newtheorem` is retained. The macro is extended to deal with the additional requirements:

```
617 \gdef\newtheorem@i{%
618   \@ifstar
619     {\expandafter\@ifundefined{th@nonumber\the\theorem@style}%
620     {\PackageError{\basename}{Theorem style {nonumber\the\theorem@style}
621       undefined (you need it here for newtheorem*) }\@ehc}%
622     {}}%
623   \edef\@tempa{{nonumber\the\theorem@style}}%
624   \expandafter\theorem@@style\@tempa\newtheorem}%
625   {\edef\@tempa{{\the\theorem@style}}%
626   \expandafter\theorem@@style\@tempa\newtheorem}}
```

Defines `\theorem@@style` to be the current `\theoremstyle` or – in case of `\newtheorem*` – to be its non-numbered equivalent (which has to be defined!), and then calls `\@newtheorem`.

`\renewtheorem`

```
627 \gdef\renewtheorem{%
628   \@ifstar
629     {\expandafter\@ifundefined{th@nonumber\the\theorem@style}%
630     {\PackageError{\basename}{Theorem style {nonumber\the\theorem@style}
631       undefined (you need it here for newtheorem*) }\@ehc}%
632     {}}%
633   \edef\@tempa{{nonumber\the\theorem@style}}%
634   \expandafter\theorem@@style\@tempa\renewtheorem}%
635   {\edef\@tempa{{\the\theorem@style}}%
636   \expandafter\theorem@@style\@tempa\renewtheorem}}
```

Analogous to `\newtheorem`.

`\@newtheorem` `\@newtheorem` does the main job for initializing a new theorem environment type. It is called by `\newtheorem`.

```
637 \gdef\@newtheorem#1{%
638   \thm@tempiffalse
639   \expandafter\@ifdefinable\csname #1\endcsname
640   {\expandafter\@ifdefinable\csname #1*\endcsname
641   {\thm@tempiftrue
```

```

642 \thm@definethm{#1}% for lists
643 \if@thmmarks
644 \expandafter\ifundefined{c@curr#1ctr}%
645 {\newcounter{curr#1ctr}}{}%
646 \expandafter\ifundefined{c@end#1ctr}%
647 {\newcounter{end#1ctr}}{}%
648 \fi
649 \expandafter\protected@xdef\csname #1Symbol\endcsname{\the\theoremsymbol}%
650 \expandafter\protected@xdef\csname #1postwork\endcsname{%
651 \the\theorempostwork}%
652 \expandafter\gdef\csname#1\endcsname{%
653 \let\thm@starredenv\@undefined
654 \csname mkheader@#1\endcsname}%
655 \expandafter\gdef\csname#1*\endcsname{%
656 \let\thm@starredenv\relax
657 \csname mkheader@#1\endcsname}%
658 \def\@tempa{\expandafter\noexpand\csname end#1\endcsname}%
659 \expandafter\xdef\csname end#1*\endcsname{\@tempa}%
660 \expandafter\xdef\csname setparms@#1\endcsname
661 {\noexpand \def \noexpand \theorem@headerfont
662 {\the\theoremheaderfont\noexpand\theorem@checkbold}%
663 \noexpand \def \noexpand \theorem@separator
664 {\the\theoremseparator}%
665 \noexpand \def \noexpand \theorem@prework
666 {\the\theorem@prework}%
667 \noexpand \def \noexpand \theorem@indent
668 {\the\theoremindent}%
669 \the \theorembodyfont
670 \noexpand\csname th@\the \theorem@@style \endcsname}%
671 \expandafter\xdef\csname th@class@#1\endcsname
672 {\noexpand\theoremstyle{\the\theorem@style}%
673 \noexpand\theoremheaderfont{\the\theoremheaderfont}%
674 \noexpand\theorembodyfont{\the \theorembodyfont}%
675 \noexpand\theoremseparator{\the\theoremseparator}%
676 \noexpand\theorem@prework{\the\theorem@prework}%
677 \noexpand\theorem@postwork{\the\theorem@postwork}%
678 \noexpand\theorem@indent\the\theorem@indent%
679 \noexpand\theorem@numbering{\the\theorem@numbering}%
680 \noexpand\theorem@symbol{\the\theorem@symbol}}%
681 }%
682 \theorem@prework{\relax}%
683 \theorem@postwork{\relax}%
684 \@ifnextchar[{\@othm{#1}}{\@nthm{#1}}}% MUST NOT BE IN ANY IF !!!

```

Argument: $\langle env \rangle := \#1$ is the (internal) environment name to be defined, which is read from the L^AT_EX source.

Line 639: check if $\langle env \rangle$ is not yet defined (or is redefined).

Lines 641–666 are executed exactly if $\langle env \rangle$ and $\langle env \rangle^*$ are not yet defined.

Line 641: $\backslash\thm@tempif=true$ iff $\langle env \rangle$ and $\langle env \rangle^*$ are not yet defined.

Line 642: Initialize theorem list handling for $\langle env \rangle$.

Lines 644–647: if `thmmarks` is active and the counters are not yet defined, for every theorem-like, define
`curr $\langle env \rangle$ ctr`: in the i th environment of type $\langle env \rangle$, `curr $\langle env \rangle$ ctr` = i ,
and
`end $\langle env \rangle$ ctr`: when the innermost environment is of type $\langle env \rangle$, in the j th potential position for an end mark in this environment, `end $\langle env \rangle$ ctr` = j .
(if the counters are already defined, $\langle env \rangle$ is redefined, and these internal counters have to be continued).

Lines 649–666: define several commands: (`\xdef` expands the definition at the time it is called and makes it global):

Line 649: store the current value of `\theoremsymbol` (`\edef`: expand `\the\theoremsymbol` now) as $\langle env \rangle$ `Symbol`.

Line 650: store the current value of `\theorempostwork` (`\edef`: expand `\the\theorempostwork` now) as $\langle env \rangle$ `postwork`.

Lines 651–653, 654–656: Define the commands `\env` and `\env*` to set the header of `\env`. (using a switch `\thm@starredenv`: `\relax` iff starred).

Lines 657, 658: Set `\end $\langle env \rangle$ *` to `\end $\langle env \rangle$` .

Lines 659–669: define `\setparms@ $\langle env \rangle$` to set the style parameters of the header for every $\langle env \rangle$ environment (in the sequel, *current* means, at the moment `\@newtheorem` is called):

Lines 660, 661: setting `\theorem@headerfont` to the *current* value of `\theoremheaderfont`, followed by a check if it is a bold style,

Lines 662, 663: setting `\theorem@separator` to the *current* value of `\theoremseparator`,

Lines 664, 665: setting `\theorem@prework` to the *current* value of `\theoremprework`,

Lines 666, 667: setting `\theorem@indent` to the *current* value of `\theoremindent`,

Line 668: executing the command sequence currently stored in `\theorembodyfont`,
and

Line 669: calling `th@the\theorem@@style` (which initializes `\@begintheorem` and `\@opargbegintheorem` according to the *current* value of `\theoremstyle` by calling `th@the\theorem@@style`).

Line 670–684: define `\th@class@ $\langle env \rangle$` to initialize all style parameters as they are set for the $\langle env \rangle$ environment.

Note, that the `\@ifdefinable` from line 639 ends after line 684.

Line 685: According to the next character, call `\@othm{ $\langle env \rangle$ }` (if another counter is used) or `\@nthm{ $\langle env \rangle$ }`.

Thus, when calling `\@newthm` with `#1= $\langle env \rangle$` , for current values `\theoremstyle=plain`,
`\theorembodyfont=\upshape`, `\theoremheaderfont=\bf`, `\theoremseparator=:`,
`\theoremindent=1cm`, `\theoremnumbering=arabic`, and `\theoremsymbol=\Box`,
the macro `\setparms@ $\langle env \rangle$` is defined as


```

\setparms@⟨env⟩ == \def\theorem@headerfont{\bf\theorem@checkbold}
                  \def\theorem@separator{:}
                  \def\theorem@indent{0cm}
                  \upshape
                  \th@plain

```

and the macro `\th@class@⟨env⟩` is defined as

```

\th@class@⟨env⟩ == \def\theoremstyle{plain}
                  \def\theoremheaderfont{\bf}
                  \def\theorembodyfont{\upshape}
                  \def\theoremseparator{:}
                  \def\theoremindent{0cm}
                  \def\theoremnumbering{arabic}
                  \def\theoremsymbol{\Box}

```

Note, that line 668 must not be inside *any* `\if... \fi` construct.

`\@renewtheorem`

```

685 \gdef\@renewtheorem#1{%
686   \expandafter\@ifundefined{#1}%
687   {\PackageError{\basename}{Theorem keyword #1 undefined}\@ehc}%
688   {}%
689   \expandafter\let\csname #1\endcsname\relax
690   \expandafter\let\csname #1*\endcsname\relax
691   \@newtheorem{#1}}

```

Argument: `⟨env⟩:=#1` is the (internal) environment name to be redefined, which is read from the L^AT_EX source.

If `⟨env⟩` is already defined, make it (and `⟨env⟩*`, too) undefined and call `\@newtheorem{⟨env⟩}`.

`\@nthm` `\@nthm` is called by `\@newtheorem` if the environment to be defined has a counter of its own.

```

692 \gdef\@nthm#1#2{%
693   \expandafter\protected@xdef\csname num@addtheorem#1\endcsname{%
694     \noexpand\@num@addtheorem#1{#2}}%
695   \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
696     \noexpand\@nonum@addtheorem#1{#2}}%
697   \theoremkeyword{#2}%
698   \expandafter\protected@xdef\csname #1Keyword\endcsname
699   {\the\theoremkeyword}%
700   \ifnextchar[{\@xnthm{#1}{#2}}{\@ynthm{#1}{#2}}}

```

Arguments:

`⟨env⟩:=#1` is the (internal) environment name to be defined (transmitted from `\@newtheorem`).

`⟨output_name⟩:=#2` is its keyword to be used in the output (read from the L^AT_EX source).

Lines 693–696: Define `\(no)num@addtheorem\line{env}` to call
`\@{(no)num@addtheorem\line{<env>}}{<output_name>}`.
For comments on `\@num@addtheorem\line` and `\@nonum@addtheorem\line`
see Section 7.1.9.

Lines 697–699: Define `\(env)Keyword{env}` to typeset/output `<output_name>`.
(note the similarity with the handling of `\theoremsymbol` for handling
complex keywords)

Line 700: According to the next character, call `\@xnthm{<env>}{<output_name>}`
(if `<env>`-environments should be numbered relative to some structuring
level) or `\@ynthm{<env>}{<output_name>}`.

`\@othm` `\@othm` is called by `\@newtheorem` if the environment to be defined uses
another counter.

```

701 \gdef\@othm#1[#2]#3{%
702   \ifundefined{c@#2}{\@nocounterr{#2}}%
703   {\ifthen@tempif
704     \global\@namedef{the#1}{\@nameuse{the#2}}%
705     \expandafter\protected@xdef\csname num@addtheorem#1\endcsname{%
706       \noexpand\@num@addtheorem{#1}{#3}}%
707     \expandafter\protected@xdef\csname nonum@addtheorem#1\endcsname{%
708       \noexpand\@nonum@addtheorem{#1}{#3}}%
709     \theoremkeyword{#3}%
710     \expandafter\protected@xdef\csname #1Keyword\endcsname
711       {\the\theoremkeyword}%
712     \expandafter\gdef\csname mkheader@#1\endcsname
713       {\csname setparams@#1\endcsname
714         \@thm{#1}{#2}{#3}}%
715     \global\@namedef{end#1}{\@endtheorem}\fi}
```

Arguments:

`<env>:=#1` is the (internal) environment name to be defined (transmitted
from `\@newtheorem`).

`<use_ctr>:=#2` is the internal name of the theorem which counter is used,
and

`<output_name>:=#3` is its “name” to be used in the output (both read from
the L^AT_EX source).

Line 702: if the counter to be used is undefined, goto error, else set `\the<env>`
to use `\the<use_ctr>` and do the following:

Lines 704–712 happen only if `<env>` is not yet defined or gets redefined:

Line 704: (from latex.ltx) make `<env>` use the counter `<use_ctr>`.

Lines 705–711 similar to lines 693–699 of `\@nthm`.

Lines 712–714 define `\mkheader@<env>` to set the style parameters of the
header and set the header (by `\@thm`):

`\mkheader@<env> == \setparams@<env>\@thm{<env>}{<use_ctr>}{<output_name>}`.

(`\setparms@<env>` is defined when `\@newtheorem{<env>}` is carried out).

Line 715: (from latex.ltx): `\end<env>` calls `\@endtheorem`.

`\@xnthm` `\@xnthm` is called by `\@nthm` if the numbering is relative to some structuring level.

```

716 \gdef\@xnthm#1#2[#3]{%
717   \ifthm@tempif
718     \expandafter\@ifundefined{c@#1}%
719       {\@definecounter{#1}}{}%
720     \@newctr{#1}[#3]%
721     \expandafter\xdef\csname the#1\endcsname{%
722       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
723       {\noexpand\csname\the\theoremnumbering\endcsname{#1}}}%
724     \expandafter\gdef\csname mkheader@#1\endcsname
725       {\csname setparms@#1\endcsname
726        \@thm{#1}{#1}{#2}}%
727     \global\@namedef{end#1}{\@endtheorem}\fi}

```

Arguments:

`<env>:=#1` is the (internal) environment name to be defined (transmitted from `\@newtheorem`).

`<output_name>:=#2` is its keyword to be used in the output,

`<level>:=#3` is the structuring level relative to which `<env>` has to be numbered (both read from the L^AT_EX source).

Lines 718–727 happen only if `<env>` is not yet defined or gets redefined:

Lines 718,719: in not yet defined, define `<env>`- counter (otherwise, `<env>` is redefined).

Line 721: (from latex.ltx): define the counter for `<env>` and add `<level>` to its reset-triggers.

Lines 722, 723: define `\the<env>` to be the command sequence

`\the<level>\@thmcountersep<numbering>\{<env>\}` ,

where `<numbering>` is the value of `\theoremnumbering` when `\@xnthm` (and thus, `\newtheorem{<env>}`) is called.

Lines 724–726: define `\mkheader@<env>` to set the style parameters of the header and set the header (by `\@thm`):

`\mkheader@<env> == \setparms@<env>\@thm{<env>\{<env>\}\{<output_name>\}`.

(`\setparms@<env>` is defined when `\@newtheorem{<env>}` is carried out).

Line 727: (from latex.ltx): `\end<env>` calls `\@endtheorem`.

`\@ynthm` `\@ynthm` is called by `\@nthm` if the counter is not relative to any structuring level.

```

728 \gdef\@ynthm#1#2{%

```

```

729 \ifthm@tempif
730   \expandafter\ifundefined{c@#1}%
731     {\@definecounter{#1}}{}%
732   \expandafter\xdef\csname the#1\endcsname
733     {\noexpand\csname\the\theoremnumbering\endcsname{#1}}%
734   \expandafter\gdef\csname mkheader@#1\endcsname
735     {\csname setparms@#1\endcsname
736       \@thm{#1}{#1}{#2}}%
737   \global\@namedef{end#1}{\@endtheorem}\fi}

```

Arguments:

$\langle env \rangle := \#1$ is the (internal) environment name to be defined (transmitted from $\backslash\@newtheorem$).

$\langle output_name \rangle := \#2$ is its keyword to be used in the output.

$\backslash\@ynthm$ works analogous to $\backslash\@xnthm$.

Handling Instances of Theorem-Environments.

$\backslash\@thm$ $\backslash\@thm$ is called by $\backslash\@(\langle env \rangle)$ (which is defined by $\backslash\@othm/\backslash\@xnthm/\backslash\@ynthm$).

```

738 \gdef\@thm#1#2#3{%
739   \if@thmmarks
740     \stepcounter{end\InTheoType ctr}%
741   \fi
742   \renewcommand{\InTheoType}{#1}%
743   \if@thmmarks
744     \stepcounter{curr#1ctr}%
745     \setcounter{end#1ctr}{0}%
746   \fi
747   \refstepcounter{#2}%
748   \theorem@prework
749   \thm@topsepadd \theorempostskipamount % cf. latex.ltx: \@trivlist
750   \ifvmode \advance\thm@topsepadd\partopsep\fi
751   \trivlist
752   \@topsep \theorempreskipamount
753   \@topsepadd \thm@topsepadd % used by \@endparenv
754   \advance\linewidth -\theorem@indent
755   \advance\@totalleftmargin \theorem@indent
756   \parshape \@ne \@totalleftmargin \linewidth
757   \@ifnextchar[{\@ythm{#1}{#2}{#3}}{\@xthm{#1}{#2}{#3}}

```

Changed to three instead of two parameters (the first one is new):

$\langle env \rangle := \#1$: (added) internal name of the theorem environment,

$\langle use_ctr \rangle := \#2$: internal name of the theorem which counter is used,

$\langle output_name \rangle := \#3$: keyword to be displayed in the output; all arguments are transmitted from $\backslash\@othm/\backslash\@xnthm/\backslash\@ynthm$.

Lines 739–741: if `thmmarks` is active, the counter for the current environment $\langle env' \rangle$ is incremented, since the last endmark in environment $\langle env' \rangle$ is

definitely not the position for its endmark (necessary for nested environments ending at the same time).

Line 742: set `\InTheoType` to `\env`.

Lines 743–746: if `thmmarks` is active, increment `curr\env`ctr and set `end\env`ctr to 0.

Line 747: adapted from `latex.ltx`: increment the corresponding counter.

Line 748: perform `prework` (before theorem structure is generated).

Lines 749–753: handle `\theorempreskipamount` and `\theorempostskipamount` (if in `vmode`, there is additional space, cf. `\trivlist` and `\@trivlist` in `latex.ltx`).

Lines 754–756: handle `\theoremindent`.

Line 757: if there is an optional argument, call `\@ythm\env\use_ctr\output_name`, otherwise call `\@xthm\env\use_ctr\output_name`.

`\@xthm` `\@xthm` is called by `\@thm` if there is no optional text in the theorem header.

```
758 \def\@xthm#1#2#3{%
759   \@begintheorem{#3}{\csname the#2\endcsname}%
760   \ifx\thm@starredenv\@undefined
761     \thm@thmcaption{#1}{#3}{\csname the#2\endcsname}{}}\fi
762   \ignorespaces}
```

Changed to three instead of two parameters (the first one is new):
`\env`:=#1: (added) internal name of the theorem environment,
`\use_ctr`:=#2: internal name of the theorem which counter is used,
`\output_name`:=#3: keyword to be displayed in the output.
All arguments are transmitted from `\@thm`.
For comments, see `\@ythm`.

`\@ythm` `\@ythm` is called by `\@thm` if there is an optional text in the theorem header.

```
763 \def\@ythm#1#2#3[#4]{%
764   \expandafter\global\expandafter\def\csname#1name\endcsname{#4}%
765   \@opargbegintheorem{#3}{\csname the#2\endcsname}{#4}%
766   \ifx\thm@starredenv\@undefined
767     \thm@thmcaption{#1}{#3}{\csname the#2\endcsname}{#4}}\fi%
768   \ignorespaces}
```

Changed to four instead of three parameters (the first one is new):
`\env`:=#1: (added) internal name of the theorem environment,
`\use_ctr`:=#2: internal name of the theorem which counter is used,
`\output_name`:=#3: keyword to be displayed in the output.
`\opt_text`:=#4: optional text to appear in the header.
#1–#3 are transmitted from `\@thm`, #4 is read from the L^AT_EX source.
Line 764: define `\env`name to be the optional argument.

Line 765: call

```
\@opargbegintheorem{\<output_name>}{\the\<use_ctr>}{\<opt_text>}
```

which outputs the header.

Line 766, 767: if $\langle env \rangle$ is not the starred version, call

```
\thm@thmcaption{\<env>}{\<output_name>}{\the\<use_ctr>}{\<opt_text>}}
```

which makes an entry into the theorem list.

`\@endtheorem` `\@endtheorem` is called for every `\end{\<env>}`, where $\langle env \rangle$ is a theorem-like environment. (note that `\@endtheorem` it is also changed by option [thmmarks] to organize the placement of the corresponding end mark). `\InTheoType` gives the innermost theorem-like environment, i.e. the one to be ended:

```
769 \gdef\@endtheorem{%
770   \endtrivlist
771   \csname\InTheoType @postwork\endcsname
772 }
```

7.1.8 Framed and Boxed Theorems

The option ‘framed’ activates framed and boxed layouts. It requires to load the `framed` package and the `pstricks` package.

`framed`

```
773 \DeclareOption{framed}{%*****
774 \newtoks\shadecolor
775 \shadecolor{gray}
776 \let\theoremframecommand\relax
```

`newshadedtheorem`

```
777 \def\newshadedtheorem#1{%
778   \expandafter\global\expandafter\xdef\csname#1@shadecolor\endcsname{%
779     \the\shadecolor}%
780   \ifx\theoremframecommand\relax
781     \expandafter\global\expandafter\xdef\csname#1@framecommand\endcsname{%
782       \noexpand\psframebox[fillstyle=solid,
783         fillcolor=\csname#1@shadecolor\endcsname,
784         linecolor=\csname#1@shadecolor\endcsname]}%
785   \else
786     \expandafter\global\expandafter\let\csname#1@framecommand\endcsname%
787     \theoremframecommand%
788   \fi
789   \theoremprework{%
790     \def\FrameCommand{\csname#1@framecommand\endcsname}\framed}%
791   \theorempostwork{\endframed}%
792   \newtheorem@i{#1}%
793 }
```

newframedtheorem

```

794 \def\newframedtheorem#1{%
795   \theoremprework{\framed}%
796   \theorempostwork{\endframed}%
797   \newtheorem@i{#1}%
798 }
799}% end of option framed *****

```

7.1.9 Generation of Theorem Lists

The following macros are needed for the generation of theorem-lists. We will document it for the theorem `\begin{definition}[optional]`, which we assume to be the first definition at all and which is placed on page 5.

`\thm@thmcaption` This macro, used internally, strips of the outer brackets from the second argument and calls `\thm@@thmcaption`. It's typically called like this

```
\thm@thmcaption{definition}{{Definition}{1}{optional}}
```

(internal name of the environment, output keyword, running number, optional text)

```
800 \def\thm@thmcaption#1#2{\thm@@thmcaption{#1}#2}
```

`\thm@@thmcaption` `\thm@caption` is called from `\thm@caption`; it writes an appropriate entry to the .thm-file.

```

801 \def\thm@@thmcaption#1#2#3#4{%
802   \thm@parseforwriting{#2}%
803   \let\thm@tmpii\thm@tmp%
804   \thm@parseforwriting{#4}%
805   \edef\thm@t{\thm@tmpii}{#3}{\thm@tmp}}%
806   \addcontentsline{thm}{#1}{\thm@t}}

```

Arguments: `<env>:=#1` is the internal environment name, `<output_name>:=#2` is its keyword to be used in the output, `#3` is the running number, and `#4` is the optional text argument in the header.

Lines 801,802: the command sequence for the output keyword is prepared by `\thm@parseforwriting` (which returns `\thm@tmpii`) and then stored in `\thm@tmpii`.

Line 803: the optional text is also prepared by `\thm@parseforwriting`

Lines 804,805: The output is collected and written into the .aux file, which will forward it to the theorem-file.

The following two macros are just shortcuts, often needed for the output of one single line in the theorem-lists. The first one is used in unnamed lists, the second one in named. Warning: Don't remove the leading `\let`, since you will get wrong `\if-fi`-nesting without it, if you don't use `hyperref`.

`\thm@@thmline@noname`

```
807 \def\thm@@thmline@noname#1#2#3#4{%
808     \@dottedtocline{-2}{0em}{2.3em}%
809     {\protect\numberline{#2}#3}%
810     {#4}}
```

`\thm@@thmline@name`

```
811 \def\thm@@thmline@name#1#2#3#4{%
812     \@dottedtocline{-2}{0em}{2.3em}%
813     {#1 \protect\numberline{#2}#3}%
814     {#4}}
```

`\thm@thmline` This is another short one, which only discards the outer brackets from the first argument and calls `\thm@@thmline`. It's normally called like this:

`\thm@@thmline{{Definition}}{1}{optional}}{5}`

```
815 \def\thm@thmline#1#2{\thm@@thmline#1{#2}}
```

`\thm@lgobble` The next macro is used to ignore all theorem-sets, which should not be listed.

```
816 \long\def\thm@lgobble#1#2{\ignorespaces}
```

The following four macros set up the predefined list-types. To do so, they define the internal macros `\thm@@thmlstart` (containing the code to be executed at the beginning of the list), `\thm@@thmlend` (code to be executed at the end of the list) and `\thm@@thmline` (code to be executed for every line). In order to gain compatibility with `newthm.sty`, we decided not to make these commands inaccessible to the user. But we recommend not using these commands, because they may disappear in later distributions.

`\theoremlistall` This one implements the type `all`.

```
817 \def\theoremlistall{%
818     \let\thm@@thmlstart=\relax
819     \let\thm@@thmlend=\relax
820     \let\thm@@thmline=\thm@@thmline@noname}
```

`\theoremlistallname` And here's the type `allname`.

```
821 \def\theoremlistallname{%
822     \let\thm@@thmlstart=\relax
823     \let\thm@@thmlend=\relax
824     \let\thm@@thmline=\thm@@thmline@name}
```

`\theoremlistoptional` This one is the list-type `opt`. In case of `[hyperref]`, the fifth argument, which is provided by `hyperref.sty` is automatically given to `\thm@@thmline@noname`.

```
825 \def\theoremlistoptional{%
826     \let\thm@@thmlstart=\relax
```



```

827 \let\thm@thmlend=\relax
828 \def\thm@thmline##1##2##3##4{%
829     \ifx\empty ##3%
830     \else
831         \thm@thmline@noname{##1}{##2}{##3}{##4}%
832     \fi}}

```

`\theoremlistoptname` And the last type, `optname`. In case of `[hyperref]`, the fifth argument, which is provided by `hyperref.sty` is automatically given to `\thm@thmline@name`.

```

833 \def\theoremlistoptname{%
834     \let\thm@thmlstart=\relax
835     \let\thm@thmlend=\relax
836     \def\thm@thmline##1##2##3##4{%
837         \ifx\empty ##3%
838         \else%
839             \thm@thmline@name{##1}{##2}{##3}{##4}%
840         \fi}}

```

Theoremlists and Hyperref Since the `hyperref`-package redefines `\contentsline`, we must redefine some commands to handle this:

1. Let the different versions of `\thm@thmline@..` take a 5th argument, the one provided by `hyperref`.
2. Don't use `hyperref`'s `contentsline`: restore the normal definition at the beginning of `\thm@processlist` (see there).
3. Let `\thm@lgobble` take one more argument, the one provided by `hyperref`.
4. Do the hyperlinks manually in the different versions of `\thm@thmline` as defined by the theorem types.

`hyperref`

```

841 \DeclareOption{hyperref}{% *****
842     \def\thm@thmline@noname#1#2#3#4#5{%
843         \ifx\#5\%
844             \@dottedtocline{-2}{0em}{2.3em}%
845                 {\protect\numberline{#2}#3}%
846                 {#4}%
847         \else
848             \ifHy@linktopage\relax\relax
849                 \@dottedtocline{-2}{0em}{2.3em}%
850                 {\protect\numberline{#2}#3}%
851                 {\hyper@linkstart{link}{#5}{#4}\hyper@linkend}
852             \else
853                 \@dottedtocline{-2}{0em}{2.3em}%

```

```

854             {\hyper@linkstart{link}{#5}{\protect\numberline{#2}#3}%
855             \hyper@linkend}%
856             {#4}%
857         \fi
858     \fi}%
859 \def\thm@thmline@name#1#2#3#4#5{%
860     \ifx\#5\%
861         \@dottedtocline{-2}{0em}{2.3em}%
862         {#1 \protect\numberline{#2}#3}%
863         {#4}
864     \else
865         \ifHy@linktocpage\relax\relax
866             \@dottedtocline{-2}{0em}{2.3em}%
867             {#1 \protect\numberline{#2}#3}%
868             {\hyper@linkstart{link}{#5}{#4}\hyper@linkend}%
869         \else
870             \@dottedtocline{-2}{0em}{2.3em}%
871             {\hyper@linkstart{link}{#5}%
872             {#1 \protect\numberline{#2}#3}\hyper@linkend}%
873             {#4}%
874         \fi
875     \fi}
876 \def\thm@thmline#1#2#3{\thm@thmline#1{#2}{#3}}
877 \long\def\thm@lgobble#1#2#3{\ignorespaces}
878 \def\theoremlistoptional{%
879     \let\thm@thmlstart=\relax
880     \let\thm@thmlend=\relax
881     \def\thm@thmline##1##2##3##4##5{%
882         \ifx\empty ##3%
883         \else%
884             \thm@thmline@noname{##1}{##2}{##3}{##4}{##5}%
885         \fi}}
886 \def\theoremlistoptname{%
887     \let\thm@thmlstart=\relax
888     \let\thm@thmlend=\relax
889     \def\thm@thmline##1##2##3##4##5{%
890         \ifx\empty ##3%
891         \else%
892             \thm@thmline@name{##1}{##2}{##3}{##4}{##5}%
893         \fi}}
894 }% end of option hyperref *****

```

`\theoremlisttype` The next one is the user-interface for selecting the list-type. It simply calls `\thm@thml@<type>`, if the given *<type>* is defined.

```

895 \def\theoremlisttype#1{%
896     \@ifundefined{thm@thml@#1}%
897     {\PackageError{\basename}{Listtype #1 not defined}\@eha}%
898     {\csname thm@thml@#1\endcsname}}

```

Now, here is the code, which maps the types – selected by `\theoremlisttype` – to the defined macros.

```
899 \def\thm@thml@all{\theoremlistall}
900 \def\thm@thml@opt{\theoremlistoptional}
901 \def\thm@thml@optname{\theoremlistoptname}
902 \def\thm@thml@allname{\theoremlistallname}
```

`\newtheoremlisttype` According to the given documentation, this one can be used to define new list-types. It's done by defining the macro `\thm@thml@<type>`, which *locally* redefines the commands `\thm@thmlstart`, `\thm@thmlline` and `\thm@thmlend`.

```
903 \def\newtheoremlisttype#1#2#3#4{%
904   \ifundefined{thm@thml@#1}%
905   {\expandafter\gdef\csname thm@thml@#1\endcsname{%
906     \def\thm@thmlstart{#2}%
907     \def\thm@thmlline####1####2####3####4{#3}%
908     \def\thm@thmlend{#4}}%
909   }{\PackageError{\basename}{list type #1 already defined}\@eha}}
```

`\renewtheoremlisttype`

```
910 \def\renewtheoremlisttype#1#2#3#4{%
911   \ifundefined{thm@thml@#1}%
912   {\PackageError{\basename}{List type #1 not defined}\@ehc}{}%
913   \expandafter\let\csname thm@thml@#1\endcsname\relax
914   \newtheoremlisttype{#1}{#2}{#3}{#4}}
```

if the list type to be redefined is already defined, make it undefined and define it.

`\thm@definethm` For each theorem-set, we need to setup two commands. Thus, the next macro is called by `\newtheorem`. It defines the command `\l@<theorem-set>` and `\thm@listdo<theorem-set>`, which initially discard the input by calling `\thm@lgobble`. The first one is called for each theorem when you are generating lists. The second one is called, if you've added something with `\addtotheoremfile`.

```
915 \def\thm@definethm#1{%
916   \expandafter\gdef\csname l@#1\endcsname{\thm@lgobble}%
917   \expandafter\gdef\csname thm@listdo#1\endcsname{\thm@lgobble}}
```

`\thm@inlistdo` If in some case, you've written additional text into the theorem-file by `\addtotheoremfile`, this one is called internally. It simply discards the first argument and strips of the outer brackets from the second one.

```
918 \long\def\thm@inlistdo#1#2{#2}%
```

`\listtheorems` Now we need the user-interface for generating lists. This is done by the next macro. We set the `tocdepth` to `-2` to assure that the predefined

list-types work. After storing the names of the theorem-sets, we call `\thm@processlist`, which actually generates the list.

```

919 \def\listtheorems#1{\begingroup
920   \c@tocdepth=-2%
921   \def\thm@list{#1}\thm@processlist
922   \endgroup}

```

`\thm@processlist` The file `\jobname.thm` contains commands of the form

`\contentsline{<theoremset>}{<header>}{<number>}{<page>}`.

Thus, dependent on which theorems should be listed, `\contentsline` must be defined to evaluate the first argument and then to output all arguments, or to discard the second and third one.

This is done the following way: The commands `\l@<theorem-set>` and `\thm@listdo<theorem-set>` (which initially were set to ignore everything by `\newtheorem`) are redefined for the theorem sets which should be listed to generate output. `\contentsline` is defined to call `\l@<theorem-set>`, adding a line to the list or ignoring the entry. Since for theorem sets which are not yet known (i.e., if the list is created at the beginning of the document, and the theoremset is only defined later), `\l@<theorem-set>` is not yet defined, `\contentsline` has to check if the command is defined, otherwise ignore the arguments.

Then, the `.thm` file is processed, evaluating the `\contentsline` commands. After processing the theorem-file, the mentioned commands are again re-defined to discard everything. We need to define the macros globally for dealing with complex, user-defined, list-types.

```

923 \def\thm@processlist{%
924   \begingroup
925   \typeout{** Generating table of \thm@list}%
926   \def\contentsline##1{%
927     \expandafter\ifundefined{l@##1}{\thm@lgobble}{\csname l@##1\endcsname}}%
928   \thm@thmlstart
929   \@for\thm@currentlist:=\thm@list\do{%
930     \ifx\thm@currentlist\@empty\else
931       \expandafter\gdef\csname l@\thm@currentlist\endcsname{\thm@thmline}%
932       \expandafter\gdef\csname thm@listdo\thm@currentlist\endcsname{\thm@inlistdo}%
933     \fi
934   }%
935   \@input{\jobname .thm}%
936   \thm@thmlend
937   \@for\thm@currentlist:=\thm@list\do{%
938     \ifx\thm@currentlist\@empty\else
939       \expandafter\gdef\csname l@\thm@currentlist\endcsname{\thm@lgobble}%
940       \expandafter\gdef\csname thm@listdo\thm@currentlist\endcsname{\thm@lgobble}%
941     \fi
942   }%
943   \endgroup}

```

`\thm@enablelistoftheorems` Up to now, we've set up various macros for writing and reading the theorem-file. Thus, it's time to set up the file itself. This is done by the next macro. We simply took the lines for `\@starttoc` from the L^AT_EX-base and changed some things. The main intention to copy `\@starttoc` is that we don't want the file to be input when it is set up – like it's done by `\@starttoc`.

```

944 \def\thm@enablelistoftheorems{%
945   \begingroup
946   \makeatletter
947   \if@files
948     \expandafter\newwrite\csname tf@thm\endcsname%
949     \immediate\openout \csname tf@thm\endcsname \jobname.thm\relax%
950   \fi
951   \@nobreakfalse
952   \endgroup}

```

`\addtheoremline` By `\addtheoremline{<theorem-set>}{<entry>}`, the user can insert an extra entry into the theorem-file. `\addtheoremline*` calls the internal macro `\nonum@addtheoremline`, otherwise `\num@addtheoremline` is called. `\num/nonum@addtheoremline{<theorem-set>}{<entry>}` calls `\num/nonum@addtheoremline{<theorem-set>}{<entry>}{<keyword>}` which are defined when `<theorem-set>` is declared (cf. `\@nthm`). These in turn call `\@num/nonum@addtheoremline{<theorem-set> }{<keyword>}{<entry>}` which write information to the theorem file.

```

953 \def\addtheoremline{\ifstar{\nonum@addtheoremline}{\num@addtheoremline}}
954 \def\nonum@addtheoremline#1{\csname nonum@addtheoremline#1\endcsname}%
955 \def\num@addtheoremline#1{\csname num@addtheoremline#1\endcsname}%

```

`\@nonum@addtheoremline` `\@num@addtheoremline` and `\@nonum@addtheoremline` write the actual entries to the .thm file.

Syntax: `\@num/nonum@addtheoremline{ <theorem-set> }{<keyword>}{<entry>}`

```

956 \def\@nonum@addtheoremline#1#2#3{%
957   \thm@parseforwriting{#3}%
958   \edef\thm@t{{#2}}{\thm@tmp}}%
959   \addcontentsline{thm}{#1}{\thm@t}}

```

`\@num@addtheoremline`

```

960 \def\@num@addtheoremline#1#2#3{%
961   \thm@parseforwriting{#3}%
962   \edef\thm@t{{#2}}{\csname the#1\endcsname}{\thm@tmp}}%
963   \addcontentsline{thm}{#1}{\thm@t}}

```

`\addtotheoremfile` To write any additional stuff into the theorem-file, the next macro is used. It first checks, if the optional name of a theorem-set is given. In that case, the macro `\@@addtotheoremfile`, otherwise `\@addtotheoremfile` is used to write the stuff into the file.

```

964 \long\def\addtotheoremfile{%
965   \ifnextchar[{\@@addtotheoremfile}{\@addtotheoremfile}}

```

`\@addtotheoremfile` Write additional stuff for all theorems.

```
966 \long\def\@addtotheoremfile#1{%
967   \thm@parseforwriting{#1}%
968   \protected@write\@auxout%
969     {}{\string\@writefile{thm}{\thm@tmp}}}
```

`\@@addtotheoremfile` Write additional stuff for a given theorem-set.

```
970 \long\def\@@addtotheoremfile[#1]#2{%
971   \thm@parseforwriting{#2}%
972   \protected@write\@auxout%
973     {}{\string\@writefile{thm}{\string\theoremlistdo{#1}{\thm@tmp}}}}
```

`\theoremlistdo` This one is called from the theorem-file to insert the additional stuff for a theorem-set.

```
974 \long\def\theoremlistdo#1#2{\expandafter\@ifundefined{thm@listdo#1}%
975   \relax{\csname thm@listdo#1\endcsname{#1}{#2}}}
```

Now we assure, that the theorem-file is activated. This is done by inserting a hook at the end of the document.

```
976 \AtEndDocument{\thm@enablelistoftheorems}
```

7.1.10 Auxiliary macros

For generating theorem-lists, we need to write informations into a separate file. Beause we don't want to expand this information, we parse it specially for writing.

```
977 \def\thm@meaning#1->#2\relax{#2}% remove "macro: ->"
978 \long\def\thm@parseforwriting#1{%
979   \def\thm@tmp{#1}%
980   \edef\thm@tmp{\expandafter\thm@meaning\meaning\thm@tmp\relax}}
```

In some countries it's usual to number theorems with greek letters:

`\theorem@checkbold` For correctness, we need to check if a bold font is active. This is done by the following macro:

```
981 \def\theorem@checkbold{\if b\expandafter\@car\f@series\@nil\boldmath\fi}
```

`\@greek` According to L^AT_EX-base, this is the internal command for generating lower-case greek numberings.

```
982 \def\@greek#1{\theorem@checkbold%
983   \ifcase#1\or$\alpha$\or$\beta$\or$\gamma$\or$\delta$\or$\varepsilon$%
984   \or$\zeta$\or$\eta$\or$\vartheta$\or$\iota$\or$\kappa$\or$\lambda$\or$%
985   \mu$\or$\nu$\or$\xi$\or$ o$\or$\varpi$\or$\varrho$\or$\varsigma$\or$\tau$%
986   \or$\upsilon$\or$\varphi$\or$\chi$\or$\psi$\or$\omega$\else\@ctrerr\fi}
```

`\@Greek` According to L^AT_EX-base, this is the internal command for generating up-percase greek numberings.

```

987 \def\@Greek#1{\theorem@checkbold%
988 \ifcase#1\or A\or B\or$\Gamma$\or$\Delta$\or E%
989 \or Z\or H\or$\Theta$\or I\or K\or$\Lambda$\or M%
990 \or N\or$\Xi$\or O\or$\Pi$\or P\or$\Sigma$\or T%
991 \or$\Upsilon$\or$\Phi$\or X\or$\Psi$\or$\Omega$\else\@ctrerr\fi}

```

`\greek` According to L^AT_EX-base, this is the user interface for lowercase greek numberings.

```

992 \def\greek#1{\@greek{\csname c@#1\endcsname}}

```

`\Greek` According to L^AT_EX-base, this is the user interface for uppercase greek numberings.

```

993 \def\Greek#1{\@Greek{\csname c@#1\endcsname}}

```

7.1.11 Other Things

After declaring several package-options, we need to process the specified ones. The additional `\relax` was mentioned by Rainer Schöpf at DANTE'97.

```

994 \ProcessOptions\relax

```

Now we set up the default theorem listtype. Make sure this is called after processing the options. Otherwise, `ntheorem` will break with `hyperref`.

```

995 \theoremlistall

```

If automatical configuration is not disabled by `[noconfig]`, it is checked if the file `ntheorem.cfg` exists and in this case the definitions in this file are read. If it does not exist and the option `standard` was specified, the file `ntheorem.std` is used.

```

996 \ifx\thm@noconfig\@undefined
997 \InputIfFileExists{ntheorem.cfg}%
998 {\PackageInfo{\basename}{Local config file ntheorem.cfg used}}%
999 {\ifx\thm@usestd\@undefined%
1000 \else%
1001 \InputIfFileExists{ntheorem.std}%
1002 {\PackageInfo{\basename}{Standard config file ntheorem.std used}}{-}
1003 \fi}
1004 \fi

```

7.2 The Standard Configuration

```

1 \theoremnumbering{arabic}
2 \theoremstyle{plain}
3 \RequirePackage{latexsym}
4 \theoremsymbol{\ensuremath{\_ \Box}}
5 \theorembodyfont{\itshape}
6 \theoremheaderfont{\normalfont\bfseries}
7 \theoremseparator{}

```

```

8 \newtheorem{Theorem}{Theorem}
9 \newtheorem{theorem}{Theorem}
10 \newtheorem{Satz}{Satz}
11 \newtheorem{satz}{Satz}
12 \newtheorem{Proposition}{Proposition}
13 \newtheorem{proposition}{Proposition}
14 \newtheorem{Lemma}{Lemma}
15 \newtheorem{lemma}{Lemma}
16 \newtheorem{Korollar}{Korollar}
17 \newtheorem{korollar}{Korollar}
18 \newtheorem{Corollary}{Corollary}
19 \newtheorem{corollary}{Corollary}
20
21 \theorembodyfont{\upshape}
22 \newtheorem{Example}{Example}
23 \newtheorem{example}{Example}
24 \newtheorem{Beispiel}{Beispiel}
25 \newtheorem{beispiel}{Beispiel}
26 \newtheorem{Bemerkung}{Bemerkung}
27 \newtheorem{bemerkung}{Bemerkung}
28 \newtheorem{Anmerkung}{Anmerkung}
29 \newtheorem{anmerkung}{Anmerkung}
30 \newtheorem{Remark}{Remark}
31 \newtheorem{remark}{Remark}
32 \newtheorem{Definition}{Definition}
33 \newtheorem{definition}{Definition}
34
35 \theoremstyle{nonumberplain}
36 \theoremheaderfont{\scshape}
37 \theorembodyfont{\normalfont}
38 \theoremsymbol{\ensuremath{\_ \blacksquare}}
39 \RequirePackage{amssymb}
40 \newtheorem{Proof}{Proof}
41 \newtheorem{proof}{Proof}
42 \newtheorem{Beweis}{Beweis}
43 \newtheorem{beweis}{Beweis}
44 \qedsymbol{\ensuremath{\_ \blacksquare}}
45 \theoremclass{LaTeX}

```

8 History and Acknowledgements

8.1 The endmark-Story (Wolfgang May)

In 1995, I started a hack for setting endmarks semiautomatically at the end of displayed formulas. The work on `thmmarks.sty` begun in October 1996 by a thread asking for a routine for setting endmarks in *de.comp.tex* initiated by Boris Piwinger. Version 0.1 incorporated the main features for setting endmarks automagically by using the `.aux` file. Version 0.2 included some bugfixes and was the first one accessible on the internet. Boris suggested

to include `fleqn` and `leqno` which has been done in version 0.3 (which was never made public). Since at this point, `thmmarks.sty` was incompatible to the widely used `theorem.sty` written by Frank Mittelbach, in Version 0.4, the features of `theorem.sty` have been integrated.

With version 0.5, the case of “empty” end symbols has been handled, `\qed` has been added (also suggested by Boris), and the handling of `theoremstyles` by `\newtheoremstyle` has been included.

For version 0.6, the handling of endmarks in `displaymaths` has been changed in order to adjust them with the bottom of the displayed math.

Version 0.6 was the first one announced in *comp.text.tex*. For version 0.7, I added the handling of `amsmath` features, suggested by my colleague Peter Neuhaus.

Versions 0.71 and 0.72 incorporated minor bugfixes.

8.2 Lists, Lists, Lists (Andreas Schedler)

I often saw questions on `theoremlists` in the german newsgroup *de.comp.text.tex*, but I never spent any attention on those postings. This changed in summer 1996, when I needed those lists for myself. Thus, I asked the holy question. But none of the given answers satisfied my wish for a simple, easy to use and short solution.

I decided to take a look at Frank Mittelbachs `theorem.sty`. First I didn’t understand much of the code, but Bernd Raichle helped me a lot by answering my boring questions and I finally understood it.

I started the coding and within a few days, a first experimental version was born. Not only that I had implemented the lists, I also inserted a separator and a flexible numbering of the theorems.

After a long period of testing, I wanted to share the new features with other \TeX -Freaks and wrote an article for the “Die \TeX nische Komödie” (Journal of german tug, DANTE e.V.). As soon as I had sent the article to DANTE, I got first reactions on the style. Gerd Neugebauer gave me many hints. I hid several cryptical notations in easy definitions and improved the user interface.

In January 1997, I released “newthm” to the world and it was uploaded to the CTAN-Archives. Few days later I sent my files to Frank Mittelbach in order to show him my extensions. He told me, that already other extensions were made, and that it would be good to combine alltogether.

8.3 Let’s come together

With version 0.8, in February 1997, the combination of `thmmarks.sty` with `newthm.sty` to `ntheorem.sty` has been started. On April 21, 1997, version 0.94 beta has been made public as version 1.0.

In course of the development, the following changes were made:

v0.80	General: Started integration of ‘thmmarks.sty’ with ‘newthm.sty’: 1	v0.90	General: changed ‘addtheorem-line, @nthm, @othm’ (WM) . 1
	\theoremstyle: ‘theoremseparator’ added (WM) 51		counter only active if ‘if@thmmarks’ (WM) 1
v0.81	theoremstyles: ‘theoremnumbering’ and styles ...No added (WM) 50		\endeqnarray: fixed endmark for ‘eqnarrays’ (WM) 33
v0.82	General: included handling of theoremlists from newthm.sty (WM) 1	v0.91	\@endtheorem-thmmarks: ‘@empty’ fixed (WM) 35
v0.83	\@newtheorem: fixed, for bold math in headers (AS) 54		\@thm: ‘theorem...skip’ fixed (WM) 60
	General: added ‘AtEndDocument’-Hook for lists (AS) 70		General: added name* (no entry in list) (WM) 1
	\theorem@checkbold: fixed greek numbering for bold headers (AS) 70		fixed ‘OrganizeTheoremSymbol’ (WM) 37
v0.84	General: added ‘ntheorem.cfg’ feature (AS) 71		included .sty in .dtx file (WM) 1
	moved standard-theorems to extra file (AS) 71		moved things from @othm, @xnthm, @ynthm to @newtheorem, introduced @@name (WM) 1
v0.85	General: replaced ‘bf’ by corresponding L ^A T _E X 2 _ε -commands (AS) 1		\endeqnarray: fixed ‘endeqnarray’ (WM) 33
v0.86	\newtheoremlisttype: added (AS) 67	v0.92	\listtheorems: made commands global in order to handle tabular-lists (AS) 67
v0.87	General: option ‘thmmarks’ added (WM) 30		\newtheoremlisttype: added error-handling (AS) 67
	Renamed style to ‘ntheorem.sty’ (WM) 1		\theoremlisttype: added error-handling (AS) 66
v0.88	General: fixed some package-infos (AS) 1		\thm@enablelistoftheorems: renamed (AS) 69
v0.89	\addtheoremline: added (AS) 69	v0.93	\@newtheorem: check ifdefinable star-env. added (WM) . . . 54
	\addtotheoremfile: added (AS) 69		newcounters only if not yet defined (WM) 54
	\listtheorems: fixed a bug for lists (AS) 67		\@renewtheorem: introduced ‘renewtheorem’ (WM) 57
			\@xnthm: @definecounter only if not yet defined (WM) 59
			\@ynthm: ‘@definecounter’ only if not yet defined (WM) 59
			\newtheoremstyle: ‘newtheoremstyle’ only if not yet defined (WM) 48
			\renewtheorem: introduced ‘renewtheorem’ (WM) 54

<code>\renewtheoremlisttype:</code> introduced <code>newtheoremlisttype</code> (WM)	67	<code>theoremstyles:</code> <code>theoremstyle</code> empty added (WM)	50
<code>\renewtheoremstyle:</code> introduced <code>'renewtheoremstyle'</code> (WM)	48	<code>\@nthm:</code> added <code>'output@'</code> to <code>'@nthm'</code> and <code>'@othm'</code> . (WM)	57
v0.94		<code>\@othm:</code> added <code>'output@'</code> to <code>'@nthm'</code> and <code>'@othm'</code> . (WM)	58
<code>\@endtheorem-thmmarks:</code> <code>'setendmarktrue'</code> globalized (WM)	35	<code>\@ythm:</code> added definition of <code>'envname'</code> (WM)	61
<code>\endmathdisplay:</code> end mark with <code>raisebox</code> (WM)	47	<code>\label:</code> added optional argument to <code>'label'</code> . (WM)	40
<code>\NoEndMark:</code> <code>'NoEndMark'</code> introduced (WM)	36	<code>\newlabel:</code> added modified macro <code>'@newl@bel'</code> . (WM)	40
<code>\theorempreskipamount:</code> <code>'theorem...skip'</code> fixed (WM) . . .	51	<code>\thref:</code> added macro <code>'thref'</code> . (WM)	41
v1.00		v1.11	
General: First official version, not changed against 0.94 (WM) .	1	General: added <code>'noconfig'</code> option (AS/WM)	48, 71
v1.01		v1.12	
General: changed some <code>'def'</code> to <code>'gdef'</code> and <code>'edef'</code> to <code>'xdef'</code> in <code>'newtheorem'</code> and related macros (WM)	1	<code>\@othm:</code> fixed a bug in <code>'@output'</code> (WM, reported by David Epstein)	58
v1.02		<code>\math@cr@@@align:</code> dropped redefinition of <code>'math@cr@@@align'</code> (WM, reported by Frank-Christian Otto)	45
<code>\@newtheorem:</code> fixed collision at <code>'@thm'</code> , introduced <code>'setparms'</code> and <code>'mkheader'</code> (WM)	54	v1.13	
<code>\@othm:</code> fixed collision at <code>'@thm'</code> , introduced <code>'setparms'</code> and <code>'mkheader'</code> (WM)	58	<code>\thref:</code> made <code>'thref'</code> an option. (WM)	41
<code>\@xnthm:</code> fixed collision at <code>'@thm'</code> , introduced <code>'setparms'</code> and <code>'mkheader'</code> (WM)	59	v1.15	
<code>\@ynthm:</code> fixed collision at <code>'@thm'</code> , introduced <code>'setparms'</code> and <code>'mkheader'</code> (WM)	59	<code>theoremstyles:</code> fixed nonumber-break (WM)	49
<code>amsthm:</code> proof-environment fixed (WM)	52	v1.16	
v1.03		<code>\@newtheorem:</code> introduced <code>'th@class'</code> (WM)	54
<code>theoremstyles:</code> break styles changed (WM)	49	<code>\SetEndMark:</code> extended for handling right indents (quote) (WM)	36
<code>\TagsPlusEndmarks:</code> Fixed <code>'TagsPlusEndMarks'</code> , introduced <code>'SetOnlyEndMark'</code> and <code>'SetTagPlusEndMark'</code> (WM)	44	removed tilde in <code>hbox</code> (WM)	36
v1.04		<code>\theoremclass:</code> introduced <code>'theoremclass'</code> and defined <code>'th@class@LaTeX'</code> (WM) .	52
<code>\endtabbing:</code> added <code>'endtabbing'</code> (WM)	34	v1.17	
		General: added <code>'hyperref'</code> option (AS, based on a proposal by Didier Verna)	65
		included option <code>noconfig</code> in driver (AS)	1

Y2K for changes in documentation (AS)	1	General: adapted to amsmath-2.0 (WM, several solutions by Giovanni Dore)	1
<code>\thm@thmline@noname:</code> hyperref adjustment (AS)	64	added handling of amsmath-labels with thref; moved option thref before amsmath (amsmath needs redefinition of 'label') (WM)	1
intruduced shortcuts for single lines in lists (AS)	64	<code>\bbsphack(2)</code> : added to thref option (error if thref was used without thmmarks; OK/WM)	39
<code>\thm@processlist:</code> Switch to normal contentsline-behaviour for lists (hyperref) (AS)	68	<code>\endalign</code> : adapted to amsmath-2.0 (GD/WM)	46
v1.18		<code>\endgather</code> : adapted to amsmath-2.0 (GD/WM)	45
<code>\@newtheorem:</code> 'protected@xdef' for Symbol (WM)	54	<code>\endmathdisplay</code> : completely new for amsmath-2.0, begin[deleted (GD/WM)	47
<code>\@nthm:</code> 'output' changed and modified into 'Keyword' (WM)	57	<code>\endtrivlist</code> : changed hbox into unskip (OK/WM)	31
'protected@xdef' instead of 'xdef' (WM)	57	<code>theoremstyles</code> : break styles and empty style changed, now it is analogous to theorem.sty (OK/WM)	49
<code>\@othm:</code> 'output' changed and modified into 'Keyword' (WM)	58	<code>ams-thref</code> : added handling of thref in ams-equations (reported by Lars Relund).	42
'protected@xdef' instead of 'xdef' (WM)	58	<code>equation</code> : adapted to amsmath-2.0 (WM)	47
<code>\@renewtheorem:</code> calls '@newtheorem' instead of 'newtheorem' (WM)	57	<code>\rendmultline</code> : adapted to amsmath-2.0 (GD/WM)	47
General: adapted for complex theorem keywords (WM, reported by Jonathan King)	1	<code>\testdef</code> : added (HO/WM, reported by Hans-Christoph Wirth)	41
debugged starred version of 'newtheorem' (WM, reported by Jonathan King)	1	<code>\theoremlistdo</code> : check for yet undefined theoremsets (WM)	70
<code>\label:</code> Adapted for complex keywords (WM)	40	<code>\thm@processlist</code> : check for yet undefined theoremsets (WM)	68
<code>\newlabel:</code> adapted to babel-3.6 (WM)	40	v1.20	
<code>\newtheorem:</code> debugged starred version of 'newtheorem' (WM)	54	General: check that not in vmode in bbsphack (WM)	37
<code>\renewtheorem:</code> debugged starred version of 'renewtheorem' (WM)	54	<code>\bbsphack(2)</code> : check that not in vmode in bbsphack; (WM)	39
<code>\thm@thmcaption:</code> 'thm@parseforwriting' also for #2 (WM)	63	<code>\lendmultline</code> : debugged (GD/WM)	46
v1.19		<code>\SetEndMark</code> : apply negative hskip only if more than 0 (WM)	37
<code>\@endtheorem-thmmarks:</code> changed hbox into unskip (OK/WM)	35		

<code>\SetTagPlusEndMark</code> : added hss	<code>orem@i</code> ' (WM, reported by	
if tags left (GD/WM)	Christoph Kluss)	45 54
<code>\theoremstyle</code> : 'theorempre-	<code>\newtheorem@i</code> : moved normal	
work' and 'theorempostwork'	'newtheorem' into 'ntheo-	
added (WM)	rem@i' (WM)	51 54
<code>hyperref</code> : replaced ifhy by ifHy	v1.24	
(AS, reported by J.J.Bataille) 66	<code>\@newtheorem</code> : debugged: reset	
v1.21	theorempre/postwork (WM) 54	
<code>\@endtheorem</code> : added changes	<code>\newtheorem</code> : debugged: reset	
to '@endtheorem' in case	after call, moved downwards	
that [thmmarks] is not active	(WM)	54
(WM)	v1.25	
62	<code>\@renewtheorem</code> : error message:	
<code>\@endtheorem-thmmarks</code> : added	'keyword' (WM)	57
handling of 'theorempost-	v1.26	
work' (WM)	<code>\SetTagPlusEndMark</code> : fixed: box	
35	to tagwidth (problem with	
<code>\@newtheorem</code> : added handling of	leqno; WM)	44
'theoremprework' and 'theo-	v1.27	
rempostwork' (WM)	<code>\[</code> : fixed: start array with no ad-	
54	ditional space (WM, reported	
<code>\@thm</code> : added handling of 'theo-	by Tillmann Berg)	32, 38
remprework' (WM)	v1.28	
60	<code>\[</code> : replaced roman by	
General: added theoremprework	thm@romannum (WM)	32, 38
and postwork (WM)	<code>\]</code> : replaced roman by	
1	thm@romannum (WM)	33, 38
included option 'framed'	<code>\endcenter</code> : replaced roman by	
(WM)	thm@romannum (WM)	34
62	<code>\OrganizeTheoremSymbol</code> :	
<code>newshadedtheorem</code> : added	replaced roman by	
(WM)	thm@romannum (WM)	37
62	<code>\SetEndMark</code> : replaced roman by	
v1.22	thm@romannum (WM)	36
<code>ams-thref</code> : fixed thm@df@label@optarg	<code>\thm@romannum</code> : duplicate latex's	
(WM, reported by Marija Ku-	definition (WM, reported by	
las)	Ch. Garcia Duarte)	31
42		
v1.23		
<code>\newtheorem</code> : moved normal		
'newtheorem' into 'ntheo-		
rem@i' (WM)		
54		
normal 'newtheorem': reset		
theoremprework and theo-		
rempostwork, then call 'nthe-		

8.4 Acknowledgements

This place is dedicated to all those, who helped us developing our separate styles and this combined package. Thanks to (listed in alphabetical order):

Donald Arseneau, Giovanni Dore, Oliver Karch, Frank Mittelbach, Gerd Neugebauer, Heiko Oberdiek, Boris Piwinger, Bernd Raichle, Rainer Schöpf, Didier Verna.

Index

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

Symbols	<code>\@endparpenalty</code> . 150	<code>\@mathmargin</code> ... 298
<code>\@addtotheoremfile</code>	<code>\@endtheorem</code>	<code>\@multiplelabels</code> 222
..... 965, <u>970</u> 97, 611,	<code>\@namedef</code> . 73, 225,
<code>\@eqncr</code> .. 68, 74, 183	715, 727, 737, <u>769</u>	229, 508–517,
<code>\@Greek</code> 987, 993	<code>\@endtheorem-thmmarks</code>	704, 715, 727, 737
<code>\@addtotheoremfile</code> <u>96</u>	<code>\@nameuse</code> 704
..... 965, <u>966</u>	<code>\@endtrivlist</code>	<code>\@ne</code> 299, 394–396, 756
<code>\@amsmath@err</code> ... 252	.. 11, <u>12</u> , 90, 100	<code>\@newctr</code> 720
<code>\@auxout</code> 130,	<code>\@eqcnt</code> 76, 81	<code>\@newl@bel</code> 218
209, 215, 968, 972	<code>\@eqnum</code>	<code>\@newtheorem</code>
<code>\@badmath</code> .. 32, 56,	28, 66, 67, 72,	624, 626, <u>637</u> , 691
61, 146, 173, 411	177, 181, 182, 187	<code>\@nil</code> 981
<code>\@badpoptabs</code> 87	<code>\@eqnswfalse</code>	<code>\@nobreakfalse</code> .. 951
<code>\@bbsphack</code> 383, 418, 431	<code>\@nocounterr</code> ... 702
119, 127, 193, 206	<code>\@eqnswtrue</code> .. 81, 424	<code>\@noitemerr</code> 14
<code>\@beginparpenalty</code> 149	<code>\@esphack</code> 217	<code>\@nonum@addtheoremline</code>
<code>\@begintheorem</code> ...	<code>\@firstoftwo</code> ... 243	... 696, 708, <u>956</u>
450, 575, 581, 759	<code>\@flushglue</code> 95	<code>\@nthm</code> 684, <u>692</u>
<code>\@bsphack</code> 214	<code>\@for</code> 929, 937	<code>\@num@addtheoremline</code>
<code>\@car</code> 981	<code>\@getpen</code> 274	... 694, 706, <u>960</u>
<code>\@ctrerr</code> 986, 991	<code>\@gobbleopt</code>	<code>\@nx</code> 422
<code>\@currentlabel</code> 210, 216	... 234, 236, 237	<code>\@opargbegintheorem</code>
<code>\@definecounter</code> ..	<code>\@greek</code> <u>982</u> , 992	451, 577, 583, 765
..... 719, 731	<code>\@ifdefinable</code> 639, 640	<code>\@othm</code> 684, <u>701</u>
<code>\@dottedtocline</code> ..	<code>\@ifnextchar</code> 201,	<code>\@outerparskip</code> ... 22
..... 808,	226, 234, 256,	<code>\@renewtheorem</code> ...
812, 844, 849,	684, 700, 757, 965	... 634, 636, <u>685</u>
853, 861, 866, 870	<code>\@ifpackageloaded</code> 219	<code>\@safe@activestru</code>
<code>\@eesphack</code>	<code>\@ifstar</code> 618, 628, 953 219
123, 134, 197, 211	<code>\@ifundefined</code>	<code>\@setref</code> 243
<code>\@eha</code> 253,	.. 39, 49, 113,	<code>\@stopline</code> 87
452, 571, 897, 909	155, 164, 220,	<code>\@tabpush</code> 87
<code>\@ehc</code> 79, 455,	448, 454, 519,	<code>\@tempa</code> .. 623–626,
621, 631, 687, 912	619, 629, 644,	633–636, 658, 659
<code>\@empty</code> 41,	646, 686, 702,	<code>\@tempskipa</code> ... 21–23
51, 78, 96,	718, 730, 896,	<code>\@testdef</code> ... 231, 232
99, 157, 166,	904, 911, 927, 974	<code>\@thm</code> 610,
249, 251, 259,	<code>\@ignoretrue</code>	714, 726, 736, <u>738</u>
267–269, 285,	29, 58, 71, 85, 186	<code>\@thmcountersep</code> . 722
290, 291, 293,	<code>\@input</code> 935	<code>\@thmmarksfalse</code> ... 3
300, 303, 308,	<code>\@latex@error</code> 79	<code>\@thmmarkstrue</code> 4
416, 417, 930, 938	<code>\@latex@warning@no@line</code>	<code>\@threfalse</code> 4
<code>\@endpary</code> .. 25, 753 223, 224	<code>\@threftrue</code> 191

<code>\df@tag</code> .. 265, 267,	376, 379, 381,	448, 449, 454,
284, 285, 300,	449, 457, 555,	457, 619, 624,
302, 308, 318,	639, 640, 649,	626, 629, 634,
328, 331, 347, 416	650, 652, 654,	636, 639, 640,
<code>\displaystyle</code>	655, 657–660,	644, 646, 649,
.... 43, 152, 159	670, 671, 689,	650, 652, 655,
<code>\displaywidth</code>	690, 693, 695,	658–660, 671,
... 177, 329, 339	698, 705, 707,	686, 689, 690,
<code>\do</code> 929, 937	710, 712, 713,	693, 695, 698,
<code>\dp</code> 395	721–725, 732–	705, 707, 710,
<code>\dspbrk@lvl</code> . 273–275	735, 759, 761,	712, 718, 721,
	764, 765, 767,	722, 724, 730,
	771, 778, 781,	732, 734, 764,
	783, 784, 786,	778, 781, 786,
	790, 898, 905,	905, 913, 916,
	913, 916, 917,	917, 927, 931,
	927, 931, 932,	932, 939, 940,
	939, 940, 948,	948, 974, 980, 981
	949, 954, 955,	
	962, 975, 992, 993	
E		F
<code>\edef</code> 128,	<code>\endeqnarray</code> . <u>64</u> , <u>180</u>	<code>\f@series</code> 981
207, 421, 623,	<code>\endeqnarray*</code> <u>73</u>	<code>\fi</code> 13, 14,
625, 633, 635,	<code>\endequation</code> . <u>27</u> , <u>175</u>	17, 18, 24, 26,
805, 958, 962, 980	<code>\endflalign</code> 380	37, 43, 46, 52,
<code>\egroup</code> 69,	<code>\endframed</code> .. 791, 796	59, 62, 79, 88,
83, 170, 171,	<code>\endgather</code> <u>348</u>	99, 103, 104,
178, 184, 298,	<code>\endgroup</code> 922, 943, 952	116, 121, 125,
352, 363, 366, 408	<code>\endmathdisplay</code> ..	141, 159, 161,
<code>\else</code> ... 17, 19, 33,	... <u>410</u> , 427, 434	167, 174, 188,
41, 51, 57, 60,	<code>\endmathdisplay@a</code>	195, 199, 242,
77, 104, 120, 264, 413	254, 265, 267,
125, 147, 157,	<code>\endmathdisplay@fleqn</code>	270–272, 276,
166, 173, 194, 266, 297	286, 287, 292,
199, 242, 251,	<code>\endmultiline@math</code> 388	294, 295, 299,
267, 268, 270,	<code>\endtabbing</code> <u>86</u>	304, 306, 308,
282, 289, 290,	<code>\endtrivlist</code>	311, 323, 324,
292, 299, 301,	<u>10</u> , 12, 88, 179, 770	332, 344, 360,
304, 308, 320,	<code>\endxalignat</code> ... 375	368, 372, 402,
322, 330, 341,	<code>\endxxalignat</code> ... 377	405, 411, 437,
359, 369, 399,	<code>\ensuremath</code>	597, 648, 715,
403, 411, 785,	... 4, 38, 44, 609	727, 737, 741,
830, 838, 847,	<code>\eqno</code> 28, 58	746, 750, 761,
852, 864, 869,	<code>\equation</code> <u>420</u>	767, 788, 832,
883, 891, 930,	<code>\eta</code> 984	840, 857, 858,
938, 986, 991, 1000	<code>\expandafter</code> 8, 98,	874, 875, 885,
<code>\empty</code> 829, 837, 882, 890	108, 128, 129,	893, 933, 941,
<code>\end</code> 52, 167, 172	207, 208, 239,	950, 981, 986,
<code>\endalign</code> <u>358</u> , 374–381	243, 357, 374,	991, 1003, 1004
<code>\endalignat</code> 378	376, 379, 381,	<code>\filedate</code> 1, 2
<code>\endcenter</code> <u>89</u>		<code>\fileversion</code> ... 1, 2
<code>\endcsname</code> 8,		
41, 51, 94,		
99, 102, 105,		
108, 115, 129,		
157, 166, 208,		
239, 242, 243,		
309, 357, 374,		

\FrameCommand . . .	790	\Greek	993	\iffalse	368
\framed	773, 790, 795	\greek	992	\ifhmode	15
G		H		\ifHy@linktocpage 848, 865	
\G@refundefinedtrue	241	\hb@xt@	152	\ifingather@	359, 364
\Gamma	988	\hbox	28, 117, 177, 302, 328, 338, 339, 342, 397, 464–467, 474–477, 484–486, 488, 495–498, 505, 506, 599	\ifinner	55
\gamma	983	\hfil	170, 176, 298, 600, 602	\ifmeasuring@	288
\gdef	8–10, 12, 27, 28, 30, 47, 64, 65, 67, 86, 89, 96, 97, 107, 108, 110, 111, 119, 123, 126, 139, 140, 175, 180, 182, 193, 197, 222, 255, 262, 265, 316, 325, 328, 331, 334, 336, 342, 345, 348, 443–447, 449, 453, 518, 554, 556, 573, 579, 585, 587, 589, 591, 603, 614, 617, 627, 637, 652, 655, 685, 692, 701, 712, 716, 724, 728, 734, 738, 769, 905, 916, 917, 931, 932, 939, 940	\hfill	11, 102	\ifmmode	31, 54, 146, 169, 411
		\hfilneg	406	\ifnum	87, 273, 299, 368
		\hrule	601	\ifsetendmark	7, 101, 104
		\hskip	95, 116, 153, 298, 390, 400, 404, 460, 461, 464, 466, 470, 471, 474, 476, 480, 481, 485, 487, 491, 492, 495, 497, 502, 505, 576, 578, 582, 584	\iftag@	283, 321, 387
		\hss	139, 140, 329, 340, 343	\iftagsleft@	308, 327, 335
		\hyper@linkend	851, 855, 868, 872	\ifthm@tempif	5, 703, 717, 729
		\hyper@linkstart . . .	851, 854, 868, 871	\ifvmode	34, 120, 194, 750
\global	29, 58, 66, 70–72, 81, 84, 85, 104, 107, 131, 181, 185–187, 225, 229, 249, 259, 275, 285, 293, 317, 318, 326, 346, 347, 383, 416–418, 424, 431, 525, 527, 529, 532, 533, 535, 538, 541, 544, 546, 549, 551, 553, 567, 704, 715, 727, 737, 764, 778, 781, 786	\hyperref	841	\ifx	41, 51, 99, 157, 166, 239, 251, 267–269, 290, 291, 300, 303, 308, 760, 766, 780, 829, 837, 843, 860, 882, 890, 930, 938, 996, 999
		I		\ignorespaces	63, 762, 768, 816, 877
		\if	981	\ignorespacesafterend	355, 373, 428, 435
		\if@eqnsw	265, 279, 319	\immediate	949
		\if@files	947	\indent	13
		\if@fleqn	266	\inf@bad	299
		\if@inlabel	13	\InputIfFileExists	997, 1001
		\if@newlist	14	\InTheoType	38–41, 44, 48–51, 53, 93, 94, 99, 102, 105, 108, 110, 129, 132, 154–157, 160, 163–166, 168, 208, 443, 740, 742, 771
		\if@nopalrlist	19	\iota	984
		\if@thmmarks	3, 137, 143, 313, 594, 643, 739, 743	\item	151, 460, 461, 464, 466, 470,
		\if@thref	4, 246		
		\ifcase	76, 983, 988		
		\ifdim	16, 20, 116, 124, 198		

471, 474, 476,	834, 835, 879,	<code>\newframedtheorem</code>
480, 481, 484,	880, 887, 888, 913 8, 794, 794
486, 491, 492,	<code>\lineht@</code> 395	<code>\newif</code> 3–5, 7
495, 497, 501,	<code>\linewidth</code> ... 36,	<code>\newlabel</code> 210, 216, 218
502, 504, 505,	152, 177, 754, 756	<code>\newshadedtheorem</code>
576, 578, 582, 584	<code>\listtheorems</code> . 9, 919 8, 777, 777
<code>\itshape</code> 5, 527, 559,	<code>\llap</code> 64, 139,	<code>\newskip</code> 118,
574, 580, 586,	140, 343, 391,	192, 530, 531, 613
588, 590, 592, 607	480, 481, 485, 487	<code>\newtheorem</code> 4, 8–19,
	<code>\long</code> 237, 816,	22–33, 40–43, 614
J	877, 918, 964,	<code>\newtheorem<i>i</i></code>
<code>\jobname</code> 935, 949	966, 970, 974, 978	615, 617, 792, 797
	<code>\ltx@label</code>	<code>\newtheoremlisttype</code>
K	248, 269, 270, 12, 903, 914
<code>\kappa</code> 984	291, 292, 303, 304	<code>\newtheoremstyle</code> .
		. 11, 447, 458,
L	M	459, 463, 469,
<code>\label</code> .. 200, 248, 252	<code>\m@ne</code> 70,	473, 479, 483,
<code>\label@in@display</code> 250	84, 185, 273, 275	490, 494, 500, 503
<code>\label@optarg</code> 202, 213	<code>\m@th</code> ... 152, 339, 342	<code>\newtoks</code>
<code>\labelsep</code>	<code>\make@display@tag</code>	523, 524, 526,
460, 461, 464,	... 278, 397, 401	528, 537, 540,
466, 470, 471,	<code>\makeatletter</code> ... 946	543, 545, 548,
474, 476, 480,	<code>\makebox</code> 36	550, 552, 566, 774
481, 485, 487,	<code>\maketag@@@</code>	<code>\newwrite</code> 948
491, 492, 495, 317, 329,	<code>\nobreak</code> 11, 102
497, 502, 505,	331, 336, 342, 346	<code>\NoEndMark</code> ... 12, 107
576, 578, 582, 584	<code>\mark</code> 131	<code>\noexpand</code> ... 658,
<code>\Lambda</code> 989	<code>\math@cr</code> 350, 361, 407	661–663, 665,
<code>\lambda</code> 984	<code>\math@cr@@@align</code> 358	667, 670, 672–
<code>\lastskip</code>	<code>\mathdisplay</code> 425, 432	680, 694, 696,
16, 20, 21, 120, 194	<code>\mathindent</code> 153	706, 708, 722,
<code>\leavevmode</code> . 151, 598	<code>\maxdimen</code> 43, 159	723, 733, 782
<code>\leftskip</code> 92, 95	<code>\meaning</code> 129, 208, 980	<code>\nointerlineskip</code> . 35
<code>\lendmultline</code> ... 382	<code>\mu</code> 985	<code>\None</code> 239, 446
<code>\lendmultline@</code> .. 382	<code>\multlinegap</code> 390, 404	<code>\NoneKeyword</code> ... 444
<code>\let</code> ... 66, 72, 75,	<code>\multlinetaggap</code> . 400	<code>\NoneSymbol</code> 445
78, 181, 187,	<code>\mysavskip</code> .. 118,	<code>\nonum@addtheoremline</code>
231, 248, 249,	120, 124, 125, 953, 954
259, 285, 293,	192, 194, 198, 199	<code>\nonumber</code> 367
299, 317, 318,	N	<code>\normalbaselines</code> 393
346, 347, 357,	<code>\newcommand</code>	<code>\normalcolor</code> 80
374–381, 388,	230, 236, 598, 604	<code>\normalfont</code> 6, 37,
416, 417, 439–	<code>\newcounter</code>	80, 339, 342,
442, 457, 653, 5, 6, 593,	538, 539, 558,
656, 689, 690,	595, 596, 645, 647	574, 580, 586,
776, 786, 803,	<code>\newdimen</code>	588, 590, 592, 608
818–820, 822–	315, 333, 534, 536	<code>\nu</code> 985
824, 826, 827,	<code>\newenvironment</code> . 605	<code>\null</code> 394

<code>\num@addtheoremline</code>	<code>\proofSymbol</code> ... 603	<code>\renewenvironment</code>
..... 953, 955	<code>\protect</code> 809, 420, 430
<code>\number</code> 422	813, 845, 850,	<code>\renewtheorem</code> . 5, <u>627</u>
<code>\numberline</code> . 809,	854, 862, 867, 872	<code>\renewtheoremlisttype</code>
813, 845, 850,	<code>\protected@write</code> 12, <u>910</u>
854, 862, 867, 872 130,	<code>\renewtheoremstyle</code>
	209, 215, 968, 972 11, <u>453</u>
O	<code>\protected@xdef</code> ..	<code>\RequirePackage</code> 3, 39
<code>\Old@df@tag</code> . 318, 347 649,	<code>\reserved@a</code> 75–78, 80
<code>\Old@maketag@@@</code> ..	650, 693, 695,	<code>\reset@equation</code> . 421
..... 317, 346	698, 705, 707, 710	<code>\restorealignstate@</code>
<code>\Oldeqnnum</code> . 66, 67,	<code>\ProvidesPackage</code> .. 2 365
72, 181, 182, 187	<code>\psframebox</code> 782	<code>\RestoreTags</code> <u>345</u> ,
<code>\Omega</code> 991	<code>\Psi</code> 991	354, 371, 409, 415
<code>\omega</code> 986	<code>\psi</code> 986	<code>\rightmargin</code> ... 116
<code>\openbox</code> 598, 603		<code>\rightskip</code> 91
<code>\openout</code> 949	Q	<code>\rlap</code> 464, 466,
<code>\or</code> 76, 77,	<code>\qed</code> 12, <u>108</u>	474, 476, 484,
983–986, 988–991	<code>\qedsymbol</code> ... 12,	486, 495, 497, 505
<code>\org@testdef</code>	44, 109, <u>550</u> , <u>566</u>	<code>\romannumeral</code> 9
... 230, 231, 233	<code>\quad</code> 27	
<code>\OrganizeTheoremSymbol</code>		S
..... 104, <u>126</u>		<code>\scshape</code> 36
P	R	<code>\set@label@type</code> ..
<code>\PackageError</code>	<code>\raise@tag</code> 392 226, 228
.... 452, 455,	<code>\raisebox</code> 64, 343	<code>\setbox</code> 302, 394
570, 620, 630,	<code>\refstepcounter</code> ..	<code>\setcounter</code> . 422, 745
687, 897, 909, 912	... 280, 423, 747	<code>\SetEndMark</code> . 110, <u>111</u>
<code>\PackageInfo</code> 2, 138,	<code>\relax</code> 31,	<code>\setendmarkfalse</code> .
144, 190, 247,	40, 50, 54, 75, 99, 103, 107
314, 569, 998, 1002	114, 129, 145,	<code>\setendmarktrue</code> 7, 104
<code>\PackageWarning</code> . 240	156, 165, 169,	<code>\SetMark@endeqn</code> ..
<code>\par</code> 16, 17	208, 219, 221,	. 27, 28, 139, 177
<code>\parshape</code> 756	227, 388, 440,	<code>\SetMark@endeqnarray</code>
<code>\parskip</code> 22	442, 457, 485,	. 64, 67, 140, 182
<code>\partopsep</code> 750	487, 502, 505,	<code>\SetOnlyEndMark</code> ..
<code>\Phi</code> 991	544, 546, 561, 322, <u>325</u>
<code>\Pi</code> 990	562, 656, 682,	<code>\SetTagPlusEndMark</code>
<code>\postdisplaypenalty</code>	683, 689, 690,	... 319, 321, <u>333</u>
..... 150, 274	776, 780, 818,	<code>\settowidth</code> .. 92, 337
<code>\PotEndMark</code>	819, 822, 823,	<code>\shadecolor</code>
11, 28, 58, 67,	826, 827, 834,	... 774, 775, 779
80, 87, 91, <u>110</u> ,	835, 848, 865,	<code>\Sigma</code> 990
170, 177, 182,	879, 880, 887,	<code>\space</code> 1, 2, 241
329, 331, 340, 343	888, 913, 949,	<code>\st@rredfalse</code> ... 424
<code>\predisdisplaypenalty</code>	975, 977, 980, 994	<code>\st@rredtrue</code> ... 431
..... 149	<code>\rendmultline</code> ... <u>385</u>	<code>\stepcounter</code>
<code>\ProcessOptions</code> . 994	<code>\rendmultline@</code> 384, 385	.. 38, 48, 112,
<code>\proofname</code> .. 604, 605	<code>\renewcommand</code>	154, 163, 740, 744
	... 145, 162, 742	

<code>\string</code> .. 131, 210, 216, 252, 969, 973	474, 476, 480, 481, 484, 486, 491, 492, 495, 497, 502, 505, 539, 574, 576, 578, 580, 582, 584, 586, 588, 590, 592, 607, 661	<code>\theoremnumbering</code> 1, 6, <u>528</u> , 564, 679, 723, 733
<code>\strut</code> ... 465, 467, 475, 477, 485, 488, 496, 498, 506		<code>\theorempostskipamount</code> 5, <u>530</u> , 749
<code>\strut@</code> 397		<code>\theorempostwork</code> 6, <u>543</u> , 562, 651, 677, 683, 791, 796
<code>\swapnumbers</code> ... 570		<code>\theoremreskipamount</code> 5, <u>530</u> , 752
T		
<code>\tag@false</code> .. 383, 417	<code>\theorem@indent</code> .. 536, 667, 754, 755	<code>\theoremprerwork</code> 6, <u>543</u> , 561, 666, 676, 682, 789, 795
<code>\tag@true</code> 326	<code>\theorem@prework</code> 547, 665, 748	<code>\theoremseparator</code> 6, 7, <u>540</u> , 560, 664, 675
<code>\tagform@</code> ... 265, 281	<code>\theorem@separator</code> 460, 461, 465, 467, 470, 471, 475, 477, 480, 481, 485, 488, 491, 492, 496, 498, 506, 542, 663	<code>\theoremstyle</code> .. 2, 5, 35, <u>518</u> , 557, 672
<code>\tagsleft@false</code> . 383	<code>\theorem@style</code> 521–523, 525, 619, 620, 623, 625, 629, 630, 633, 635, 672	<code>\theoremstyles</code> .. <u>459</u> <code>\TheoremSymbol</code> ... 12 <code>\theoremsymbol</code> 4, 6, 38, <u>548</u> , 565, 609, 649, 680
<code>\TagsPlusEndmarks</code> <u>316</u> , 349, 360, 386, 412	<code>\theorembodyfont</code> . .. 5, 6, 21, 37, <u>526</u> , 559, 669, 674	<code>\thepage</code> 210, 216, 241 <code>\Theta</code> 989 <code>\thm@@thmcaption</code> 800, <u>801</u>
<code>\tagwidth</code> 333, 337, 338	<code>\theoremclass</code> 6, 45, <u>554</u>	<code>\thm@@thmlend</code> 819, 823, 827, 835, 880, 888, 908, 936
<code>\tau</code> 985	<code>\theoremframecommand</code> ... 776, 780, 787	<code>\thm@@thmline</code> 815, 820, 824, 828, 836, 876, 881, 889, 907
<code>\testdef</code> <u>230</u>	<code>\theoremheaderfont</code> 5, 6, 36, <u>537</u> , 558, 662, 673	<code>\thm@@thmline@name</code> <u>811</u> , 824, 839, 859, 892
<code>\th@class@LaTeX</code> . 556	<code>\theoremindent</code> . 6, <u>534</u> , 563, 668, 678	<code>\thm@@thmline@noname</code> <u>807</u> , 820, 831, 842, 884
<code>\th@definition</code> .. 585	<code>\theoremkeyword</code> <u>552</u> , 697, 699, 709, 711	<code>\thm@@thmlstart</code> 818, 822, 826, 834, 879, 887, 906, 928
<code>\th@nonumberbreak</code> 510, 511, 516, 517	<code>\theoremlistall</code> <u>817</u> , 899, 995	<code>\thm@amstmpdepth</code> 315
<code>\th@nonumberdefinition</code> 587	<code>\theoremlistallname</code> <u>821</u> , 902	<code>\thm@currentlist</code> . 929–932, 937–940
<code>\th@nonumberplain</code> 508, 509, 512–515, 579, 588, 592, 606	<code>\theoremlistdo</code> 973, <u>974</u>	<code>\thm@definethm</code> 642, <u>915</u>
<code>\th@nonumberremark</code> 591	<code>\theoremlistoptional</code> ... 825, 878, 900	
<code>\th@plain</code> 573, 586, 590	<code>\theoremlistoptname</code> ... 833, 886, 901	
<code>\th@remark</code> 589	<code>\theoremlisttype</code> 10, <u>895</u>	
<code>\the</code> 109, 619, 620, 623, 625, 629, 630, 633, 635, 649, 651, 662, 664, 666, 668–670, 672–680, 699, 711, 723, 733, 779		
<code>\theequation</code> 265, 281		
<code>\theorem@style</code> 524, 624, 626, 634, 636, 670		
<code>\theorem@checkbold</code> 662, <u>981</u> , 982, 987		
<code>\theorem@headerfont</code> 460, 461, 464, 466, 470, 471,		

<code>\thm@df@label@optarg</code>	<code>\thm@starredenv</code> ..	17, 102, 121, 195
.... 249, 259,	653, 656, 760, 766	<code>\upshape</code> 21
262, 269, 270,	<code>\thm@t</code> ... 805, 806,	<code>\Upsilon</code> 991
291, 292, 303, 304	958, 959, 962, 963	<code>\upsilon</code> 986
<code>\thm@enablelistoftheorems</code>	<code>\thm@tempiffalse</code> 638	
..... 944, 976	<code>\thm@tempiftrue</code> . 641	V
<code>\thm@inlistdo</code> 918, 932	<code>\thm@thmcaption</code> ..	<code>\varepsilon</code> 983
<code>\thm@label@in@display@noarg</code> ... 761, 767, 800		<code>\varphi</code> 986
..... 256, 258	<code>\thm@thml@call</code> ... 899	<code>\varpi</code> 985
<code>\thm@label@in@display@optarg</code>	<code>\thm@thml@callname</code> 902	<code>\varrho</code> 985
..... 256, 261	<code>\thm@thml@opt</code> ... 900	<code>\varsigma</code> 985
<code>\thm@lgobble</code>	<code>\thm@thml@optname</code> 901	<code>\vartheta</code> 984
816, 877, 916,	<code>\thm@thmline</code>	<code>\vbox</code> 464, 466, 474,
917, 927, 939, 940	... 815, 876, 931	476, 484, 486,
<code>\thm@list</code>	<code>\thm@tmp</code> . 128, 133,	495, 497, 505, 601
921, 925, 929, 937	207, 210, 803,	<code>\veqno</code> 267
<code>\thm@makelabel</code> 203, 205	805, 958, 962,	<code>\vfil</code> 601
<code>\thm@meaning</code>	969, 973, 979, 980	<code>\vrule</code> 600, 602
128, 207, 977, 980	<code>\thm@tmpii</code> .. 803, 805	<code>\vskip</code> 21, 23, 125, 199
<code>\thm@noconfig</code>	<code>\thm@topsepadd</code> ...	<code>\vtop</code> 391
... 441, 442, 996	613, 749, 750, 753	
<code>\thm@parseforwriting</code>	<code>\thm@usestd</code>	X
802, 804, 957,	... 439, 440, 999	<code>\xdef</code> 659, 660, 671,
961, 967, 971, 978	<code>\thref</code> 13, 238	721, 732, 778, 781
<code>\thm@processlist</code> .	<code>\too@wide</code> 299	<code>\Xi</code> 990
..... 921, 923	<code>\topsep</code> 532, 533	<code>\xi</code> 985
<code>\thm@roman@num</code> .. 8, 9	<code>\totwidth@</code> .. 351, 362	
<code>\thm@romannum</code>	<code>\trivlist</code> 751	Z
.... 8, 39, 40,	<code>\typeout</code> 1, 925	<code>\z@</code> . 16, 20, 81, 87,
49, 50, 93, 94,		116, 124, 198, 299
113, 115, 132,	U	<code>\zeta</code> 984
155, 156, 164, 165	<code>\unskip</code> 11, 16,	