The amsart, amsproc, and amsbook document classes

American Mathematical Society
Michael Downes
updated by Barbara Beeton

Version 2.20.6, 2020/05/29

1 Introduction

This file (amsclass.dtx) is the master file for three LaTeX document classes, amsart, amsproc, and amsbook, which are intended for articles and books containing mathematical research. They produce output that follows the style conventions of American Mathematical Society publications. The theorem setup features of these document classes are also available in a separate package, amsthm.

2 Implementation

Three document class files and one package file (amsthm.sty) are produced from this source. Most of the code of the amsthm package is used in all four derived files. Most of the remaining code is used in all three document class files. Fine tuning is done with additional docstrip guards.

The usual name, date, and version information. (Note: the reason each \ProvidesClass command is placed on a line by itself, with separate begin and end guards for docstripping, is to make automatic update of file date and version slightly easier and more robust.)

```
\NeedsTeXFormat{LaTeX2e}% LaTeX 2.09 can't be used (nor non-LaTeX)
[1995/06/01]% LaTeX date must be June 1995 or later
⟨*amsart⟩
\ProvidesClass{amsart}[2020/05/29 v2.20.6]
⟨/amsart⟩
⟨*amsproc⟩
\ProvidesClass{amsproc}[2020/05/29 v2.20.6]
⟨/amsproc⟩
⟨*amsbook⟩
\ProvidesClass{amsbook}[2020/05/29 v2.20.6]
⟨/amsbook⟩
```

For amsthm we need to guard against redundant loading via

```
\documentclass{amsart}
\usepackage{amsthm}
```

because in that case the usual `\RequirePackage` mechanism for avoiding redundant loading will not apply. We need to simulate the loading of the `amsthm` package.

```
⟨*classes⟩
\global\expandafter\let\csname ver@amsthm.sty\expandafter\endcsname
  \csname ver@\@currname.\@currext\endcsname
⟨/classes⟩
⟨*amsthm⟩
\ProvidesPackage{amsthm}[2020/05/29 v2.20.6]
⟨/amsthm⟩
```

The following code is shared by the classes and the `amsthm` package. Cf. `amsgen.sty`.

```
\let\@xp=\expandafter
\let\@nx=\noexpand
\def\@oparg#1[#2]{\@ifnextchar[{#1}{#1[#2]}}
\long\def\@ifempty#1{\@xifempty#1@@..\@nil}
\long\def\@xifempty#1#2@#3#4#5\@nil{%
  \ifx#3#4\@xp\@firstoftwo\else\@xp\@secondoftwo\fi}
\long\def\@ifnotempty#1{\@ifempty{#1}{}}
\def\setboxz@h{\setbox\z@\hbox}
\def\@addpunct#1{%
  \relax\ifhmode
    \ifnum\spacefactor>\@m \else#1\fi
  \fi}
```

`\nopunct` should have a value for `\spacefactor` that is not used for `\frenchspacing`.

```
\def\nopunct{\spacefactor 1007 }
\def\frenchspacing{\sfcode`\.1006\sfcode`\?1005\sfcode`\!1004%
  \sfcode`\:1003\sfcode`\;1002\sfcode`\,1001 }
```

If this class is loaded by a parent document class, then we want to use the name of the parent class. Otherwise the name of the current class file.

```
⟨*classes⟩
\def\@tempa#1#2\@nil{\edef\@classname{#1}}
\expandafter\@tempa\@currnamestack{}{}{}\@nil
\ifx\@classname\@empty \edef\@classname{\@currname}\fi
```

2.1 Support for conditional text

[This needs to be documented in the users' guide, including the idea of using `\for{5ed}{\linebreak}` to mark edition-specific line and page breaks. [mjd,1999/12/27]]

We would sometimes like to be able to mark fragments of text to be conditionally discarded or typeset. For example in the title of a section if we want to add a linebreak but prevent this linebreak from also taking effect in the table of contents.

Certain kinds of switches need to be built into the low-level structure of our document class in order to be useful. For example, inside the toc we need to arrange for an "in-toc?" test to yield true.

```
\def\@True{00}
\def\@False{01}
\newcommand\newswitch[2][False]{%
  \expandafter\@ifdefinable\csname ?@#2\endcsname{%
    \global\expandafter\let\csname ?@#2\expandafter\endcsname
      \csname @#1\endcsname
  }%
}
\newcommand{\setFalse}[1]{%
  \expandafter\let\csname ?@#1\endcsname\@False
}
\newcommand{\setTrue}[1]{%
  \expandafter\let\csname ?@#1\endcsname\@True
}
```

The empty switch is by default false; i.e., if you write

```
\for{}{...}
```

the material will be discarded.

```
\newswitch{}
```

To get a line break in a section title but not in the table of contents line for that section, use `\except{toc}{\linebreak}`. (Presumably you are already giving a shortened running head version separately, if applicable.)

```
\DeclareRobustCommand{\except}[1]{%
  \if\csname ?@#1\endcsname \expandafter\@gobble
  \else \expandafter\@firstofone
  \fi
}
\DeclareRobustCommand{\for}[1]{%
  \if\csname ?@#1\endcsname \expandafter\@firstofone
  \else \expandafter\@gobble
  \fi
}
```

The `\forany` command needs to run through a comma-separated list of switch names and print its second argument if any of the switches are true.

```
\DeclareRobustCommand{\forany}[1]{%
  \csname for@any@01\endcsname#1,?,\@nil
}
\@namedef{for@any@\@False}#1,{%
  \csname for@any@%
    \csname ?@\zap@space#1 \@empty\endcsname
  \endcsname
}
\@namedef{?@?}{x}
\@namedef{for@any@\@True}#1\@nil#2{#2}
\def\for@any@x{\@car\@gobble}
```

2.2 Options

Notes

Options will be processed in the order they are declared; cf. `\ProcessOptions`.

Paper size

The option `letterpaper` (default) sets the target paper width and height to U.S. letter size, 8.5 in x 11 in. An option `a4paper` is also supported, but we don't include some of the more unusual paper options (`legalpaper`, `a5paper`, `executivepaper`) of the generic `article` class. For A4 paper we not only change the paper size but also add 4pc to the normal textheight of 50.5pc (the difference between 297mm and 11in is 50pt).

```
\DeclareOption{a4paper}{\paperheight 297mm\paperwidth 210mm
  \textheight 54.5pc }
\DeclareOption{letterpaper}{\paperheight 11in\paperwidth 8.5in }
```

The options `landscape` and `portrait` swap paper height and width.

```
\DeclareOption{landscape}{\@tempdima\paperheight
  \paperheight\paperwidth \paperwidth\@tempdima}
\DeclareOption{portrait}{}
```

Two-sided or one-sided printing

For two-sided printing we set the switch `\if@twoside` which will cause the margins to be adjusted so that the type blocks of back-to-back pages will line up. The `\if@mparswitch` makes margin paragraphs print in the outside margin.

```
\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
```

Draft or final version

The `draft` option causes overfull lines to be marked with a black slug in the right margin.

```
\DeclareOption{draft}{\overfullrule5\p@
  \ClassWarningNoLine{\@classname}{%
    When the draft option is used, the
      \protect\includegraphics\MessageBreak
    command will print blank placeholder boxes\MessageBreak
    for the graphics}%
}
\DeclareOption{final}{\overfullrule\z@ }
```

Posting date

The date when an article is officially posted to the WWW is recorded in a variable `\@dateposted` with the `\dateposted` command.

```
\def\dateposted#1{\def\@dateposted{#1}}%
\let\@dateposted\@empty
```

```
⟨*amsart⟩
\def\@setdateposted{%
  \newline Article electronically published on \@dateposted}
⟨/amsart⟩
```

Logos

The following logo is used for regular journal articles. The one for proceedings articles and the one for e-only journals are given separately.

```
⟨*amsart⟩
\def\article@logo{%
  \set@logo{%
    \publname
```

Current volume might be empty when an article is first posted to the WWW. In that case leave out the issue-specific info.

```
    \ifx\@empty\currentvolume
    \else \newline\volinfo, \pageinfo
    \fi
    \newline \@PII
    \ifx\@empty\@dateposted \else \@setdateposted\fi
  }%
}
\def\eonly@logo{%
  \set@logo{%
    \publname
    \newline\volinfo, \pageinfo
    \ifx\@empty\@dateposted \else \@setdateposted\fi
    \newline \@PII
  }%
}
⟨/amsart⟩
```

```
⟨*amsart | amsproc⟩
\def\@logofont{\fontsize{6}{7\p@}\selectfont}
\long\def\set@logo#1{%
  \vbox to\headheight{%
    \@parboxrestore \@logofont
    \noindent#1\par\vss
  }%
}
⟨/amsart | amsproc⟩
```

```
⟨*amsproc⟩
\def\procart@logo{%
  \set@logo{\publname
    \ifx\@empty\volinfo \else\newline\volinfo\fi}%
}
⟨/amsproc⟩
```

E-only journal

Electronic-only journals (for `amsart` only) have different information in the series logo than paper-only or dual journals. Only the volume number is reported (no issue or year), and the posting date is added following the page numbers. [bnb, 1996/10/31]

This option will be invoked only from publication-specific `.cls` files.

```
⟨*amsart⟩
\DeclareOption{e-only}{%
  \def\volinfo{Volume \currentvolume}%
  \dateposted{Xxxx XX, XXXX}%
  \def\@setdateposted{\ (\@dateposted)}%
  \let\article@logo\eonly@logo
}
⟨/amsart⟩
```

Title page

The title and related information can optionally be printed on a separate page.

```
\newif\if@titlepage
\DeclareOption{titlepage}{\@titlepagetrue}
\DeclareOption{notitlepage}{\@titlepagefalse}
```

Start on right- or left-hand page

For some book series, it's permissible to start chapters on a left-hand page. Default to 'openright', the usual AMS book style.

```
⟨*amsbook⟩
\newif\if@openright
\DeclareOption{openright}{\@openrighttrue}
\DeclareOption{openany}{\@openrightfalse}
\@openrighttrue
⟨/amsbook⟩
```

Two-column printing

Two-column layout is handled through a predefined internal switch.

```
\DeclareOption{onecolumn}{\@twocolumnfalse}
\DeclareOption{twocolumn}{\@twocolumntrue}
```

The nomath option

The `nomath` option causes most of the extra math features to be omitted. Some utility functions will be defined below if this option is specified.

```
\DeclareOption{nomath}{}
```

Some font options

The `noamsfonts` option means to avoid declaring math alphabets or symbol fonts for the extra math fonts in the AMSFonts set. If these fonts are declared, it means that the corresponding `.tfm` files are required even for documents that do not use any symbols from those fonts. So we allow optionally to not declare them, for convenience of users who don't have those fonts on their system and don't want the hassle of getting them.

```
\DeclareOption{noamsfonts}{}
```

The `psamsfonts` option, passed on to the `amsfonts` package, means that alternative `.fd` files should be used that do not refer to `.tfm` files for sizes 6,8,9 (which are not present in the PostScript (Type 1) AMS fonts set from Y&Y/Blue Sky Research). This should also trigger the `cmex10` option of `amsmath`, to avoid trying to load sizes 7–9 of `cmex`.

```
\DeclareOption{psamsfonts}{%
  \PassOptionsToPackage{psamsfonts}{amsfonts}%
  \PassOptionsToPackage{cmex10}{amsmath}}
```

Equation numbering on the left or right

The option `leqno`—equation numbers on the left—is the default in AMS styles. Therefore we provide also a `reqno` option.

```
\newif\iftagsleft@
\DeclareOption{leqno}{%
  \tagsleft@true \PassOptionsToPackage{leqno}{amsmath}}
\DeclareOption{reqno}{%
  \tagsleft@false \PassOptionsToPackage{reqno}{amsmath}}
```

Vertical centering of equation numbers

For multiline equations the equation number is by default centered vertically on the total height of the equation. To make the equation number print on the first line (for left-hand numbers) or the last line (right-hand numbers), there is a `tbtags` option 'top/bottom tags'.

```
\newif\ifctagsplit@
\DeclareOption{centertags}{%
  \ctagsplit@true \PassOptionsToPackage{centertags}{amsmath}}
\DeclareOption{tbtags}{%
  \ctagsplit@false \PassOptionsToPackage{tbtags}{amsmath}}
```

Flush left displays

The option `fleqn` causes displayed equations to print aligned on the left instead of centered, with an indentation of `\mathindent` from the prevailing left margin. If the `amsmath` package is loaded, most of this code will be overridden, but it seems we need it anyway because of the possibility of the `nomath` class option.

```
\DeclareOption{fleqn}{}%
```

Dealing with font sizes

`\@mainsize` `\@ptsize` Instead of the miserly `\@ptsize` variable from LaTeX's ancient history that contains only the last digit of the main typesize, we set up a proper variable `\@mainsize` that contains all the digits of the main typesize. Just in case it is needed for someone using an old package, we will keep `\@ptsize` also.

```
\newcommand{\@mainsize}{10}
\newcommand{\@ptsize}{0}
```

`\larger` `\smaller` This function is an easy byproduct of the work done to fold typesize-specific code into the main class file. The range of font sizes is `\normalsize`, `\small`, `\Small`, `\SMALL`, `\tiny`, `\Tiny`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`. Spaces are left at either end of the case statement to accommodate adding `\TINY` and `\HUGE` in the future but it's not clear that they're really needed.

```
\newcommand{\larger}[1][1]{%
  \count@\@currsizeindex \advance\count@#1\relax
  \ifnum\count@<\z@ \count@\z@ \else\ifnum\count@>12 \count@12 \fi\fi
```

The various size-changing commands \normalsize, etc., will take care of updating \@currsizeindex.

```
  \ifcase\count@
    \Tiny\or\Tiny\or\tiny\or\SMALL\or\Small\or\small
    \or\normalsize
    \or\large\or\Large\or\LARGE\or\huge\or\Huge\else\Huge
  \fi
}
\newcommand{\smaller}[1][1]{\larger[-#1]}
```

The \@adjustvertspacing function adapts some vertical spacing amounts to the current type size. We don't expect large sections of vertical text to occur in the extraordinarily small or large type sizes, so the \@adjustvertspacing function is only called in the range between 'footnote' size and 'Large' size. Notice that no shrinkability is used.

```
\def\@adjustvertspacing{%
```

\big/med/smallskipamount are generic space values that will be used by the commands \bigskip, \medskip, \smallskip. We also link the spacing around displayed equations to these amounts.

```
  \bigskipamount.7\baselineskip plus.7\baselineskip
  \medskipamount\bigskipamount \divide\medskipamount\tw@
  \smallskipamount\medskipamount \divide\smallskipamount\tw@
  \abovedisplayskip\medskipamount
  \belowdisplayskip \abovedisplayskip
```

The above-display short space is zero but with the same stretchability as the above-display normal space. And the below-display short space is similar, but has a base value equal to \smallskipamount. Use of the multiplier 1 is an arcane TEX trick that coerces the skip value to a dimen value, i.e., gives us the base value of the skip register without the stretch or shrink values.

```
  \abovedisplayshortskip\abovedisplayskip
  \advance\abovedisplayshortskip-1\abovedisplayskip
  \belowdisplayshortskip\abovedisplayshortskip
  \advance\belowdisplayshortskip 1\smallskipamount
```

The traditional value for \jot is 3pt, which we generalize to \baselineskip/4. This is used to adjust interline spacing in multiline displayed equations.

```
  \jot\baselineskip \divide\jot 4 \relax
}
```

We fill out the range of typesize changing commands to a full eleven: five large/huge commands and five small/tiny commands. (The capitalization of the command names suggests that there should actually be thirteen—add \TINY and \HUGE—but let's be conservative and leave those out until a real need for them is known to exist.) An unavoidable side effect is that \tiny now selects 6pt instead of 5pt by default.

In version 1.1 of amsart and amsbook \small was the same as \footnotesize (amsproc didn't exist in v. 1.1).

The only size-changing command that is predefined by LaTeX is `\normalsize`; that's why it's the only one for which we use `\renewcommand` below.

```
\renewcommand\normalsize{\@xsetfontsize\normalsize 6%
  \@adjustvertspacing \let\@listi\@listI}
\DeclareRobustCommand{\Tiny}{\@xsetfontsize\Tiny 1}
\DeclareRobustCommand{\tiny}{\@xsetfontsize\tiny 2}
\DeclareRobustCommand{\SMALL}{\@xsetfontsize\SMALL 3}
\DeclareRobustCommand{\Small}{\@xsetfontsize\Small 4%
  \@adjustvertspacing
  \def\@listi{\topsep\smallskipamount \parsep\z@skip \itemsep\z@skip
    \leftmargin=\leftmargini
    \labelwidth=\leftmargini \advance\labelwidth-\labelsep
  }%
}
\DeclareRobustCommand{\small}{\@xsetfontsize\small 5\@adjustvertspacing}
```

For backward compatibility we had better define `\footnotesize` and `\scriptsize`. Also there is the small discrepancy with `\tiny` to worry about.

```
\def\footnotesize{\Small}
\def\scriptsize{\SMALL}
```

The sizes above 10pt use magstep values, stored in the functions `\@xipt`, `\@xiipt`, etc.

```
\DeclareRobustCommand{\large}{\@xsetfontsize\large 7\@adjustvertspacing}
\DeclareRobustCommand{\Large}{\@xsetfontsize\Large 8\@adjustvertspacing}
\DeclareRobustCommand{\LARGE}{\@xsetfontsize\LARGE 9}
\DeclareRobustCommand{\huge}{\@xsetfontsize\huge{10}}
\DeclareRobustCommand{\Huge}{\@xsetfontsize\Huge{11}}
%\DeclareRobustCommand\HUGE{\@xsetfontsize\HUGE{12}}
```

So now we had better define the `\@xsetfontsize` function. The size-changing commands use `\@setfontsize` instead of `\fontsize` to (a) give an error message if used in math mode and (b) set the `\@currsize` variable.

```
\def\@xsetfontsize#1#2{%
  \chardef\@currsizeindex#2\relax
  \edef\@tempa{\@nx\@setfontsize\@nx#1%
    \@xp\ifcase\@xp\@currsizeindex\@typesizes
```

Add nonsense values 99/99 at the end just in case some extreme error turns up.

```
      \else{99}{99}\fi}%
  \@tempa
}
```

For the record let's initialize `\@currsizeindex`.

```
\chardef\@currsizeindex=6
```

Set page-breaking penalties to prevent all widows, orphans, and hyphens at the end of a page.

```
\widowpenalty=10000
\clubpenalty=10000
\brokenpenalty=10000
```

Set some default linespacing values. The variable \linespacing is usually the normal interline space in the main text. It is used to specify vertical space for elements such as section heads and theorems in proportion to the normal interline space.

```
\newdimen\linespacing
\lineskip=1pt \lineskiplimit=1pt
\normallineskip=1pt \normallineskiplimit=1pt
\let\baselinestretch=\@empty
```

Settings for \textheight and \textwidth. We start with the value 50.5pc specified in AMS journal specifications as the total height of the type block and then subtract the running head height and adjust for \topskip to get the proper value for the text block.

```
\headheight=8pt \headsep=14pt
⟨amsbook⟩\footskip=18pt
⟨amsart | amsproc⟩\footskip=12pt
\textheight=50.5pc \topskip=10pt
\textwidth=30pc
\columnsep=10pt \columnseprule=0pt
```

Some settings for marginpars.

```
\marginparwidth=90pt
\marginparsep=11pt
\marginparpush=5pt
```

To avoid setting text before begin-document, we postpone the setting of \footnotesep using \AtBeginDocument.

```
\AtBeginDocument{\settoheight{\footnotesep}{\footnotesize M$^1$}}

\skip\footins=7pt plus11pt
\skip\@mpfootins=\skip\footins

\fboxsep=3pt \fboxrule=.4pt

\arrayrulewidth=.4pt    \doublerulesep=2pt
\labelsep=5pt           \arraycolsep=\labelsep
\tabcolsep=\labelsep    \tabbingsep=\labelsep

\floatsep=15pt plus 12pt    \dblfloatsep=15pt plus 12pt
\textfloatsep=\floatsep  \dbltextfloatsep=15pt plus 12pt
\intextsep=\floatsep

\@fptop=0pt plus1fil     \@dblfptop=0pt plus1fil
\@fpbot=0pt plus1fil     \@dblfpbot=0pt plus1fil
\@fpsep=8pt plus2fil     \@dblfpsep=8pt plus2fil\relax
```

Note that \parskip gets no stretch; this is at variance with the generic LaTeX classes.

```
\parskip=0pt \relax
```

\@parboxrestore, used by \@footnotetext, sets \parindent to 0pt; since this is not what we want, we make a new dimen \normalparindent and after calling \@parboxrestore, \@footnotetext resets \parindent back to normal.

```
\newdimen\normalparindent
⟨amsart⟩\normalparindent=12pt
⟨amsproc | amsbook⟩\normalparindent=18pt
\parindent=\normalparindent

\partopsep=0pt \relax \parsep=0pt \relax \itemsep=0pt \relax

\@lowpenalty=51          \@medpenalty=151        \@highpenalty=301
\@beginparpenalty=-\@lowpenalty
\@endparpenalty=-\@lowpenalty
\@itempenalty=-\@lowpenalty
```

Typesize-specific code

The class option `12pt` sets the main typesize to 12 pt and makes various adaptations, primarily sliding the size-changing commands up the scale of magsteps. This makes it more likely that someone with bitmapped fonts will have all the fonts and sizes that they need. The `8pt` option is for those who like to conserve paper.

By parameterizing some aspects it is possible to make a great deal of the typesize-specific code automatically adapt to the selected size. Then there is so little typesize-specific code remaining that it no longer makes sense to put the code in separate `.clo` files. So instead of analogs for the generic `size10,11,12.clo` files we have the code for those options entirely contained in the `.cls` file in the form of declared options.

Some miscellaneous remarks.

—If PostScript fonts are used, it may seem a little strange to use fonts following the magstep'd point sizes 10.95, 14.4, 17.28, 20.74, 24.88 instead of simply 11, 14, 17, 21, 25. But it is not easy for us to make that distinction here in the document class definitions of the fontsize changing commands. So we don't try.

```
\DeclareOption{10pt}{\def\@mainsize{10}\def\@ptsize{0}%
  \def\@typesizes{%
```

There should be 11 typesize/baselineskip pairs: five below `\normalsize` and five above.

```
    \or{5}{6}\or{6}{7}\or{7}{8}\or{8}{10}\or{9}{11}%
    \or{10}{12}% normalsize
    \or{\@xipt}{13}\or{\@xiipt}{14}\or{\@xivpt}{17}%
    \or{\@xviipt}{20}\or{\@xxpt}{24}}%
  \normalsize \linespacing=\baselineskip
}
%
\DeclareOption{11pt}{\def\@mainsize{11}\def\@ptsize{1}%
  \def\@typesizes{%
    \or{6}{7}\or{7}{8}\or{8}{10}\or{9}{11}\or{10}{12}%
    \or{\@xipt}{13}% normalsize
    \or{\@xiipt}{14}\or{\@xivpt}{17}\or{\@xviipt}{20}%
    \or{\@xxpt}{24}\or{\@xxvpt}{30}}%
  \normalsize \linespacing=\baselineskip
```

```
}
%
\DeclareOption{12pt}{\def\@mainsize{12}\def\@ptsize{2}%
  \def\@typesizes{%
    \or{7}{8}\or{8}{10}\or{9}{11}\or{10}{12}\or{\@xipt}{13}%
    \or{\@xiipt}{14}% normalsize
    \or{\@xivpt}{17}\or{\@xviipt}{20}\or{\@xxpt}{24}%
    \or{\@xxvpt}{30}\or{\@xxvpt}{30}}%
  \normalsize \linespacing=\baselineskip
}
%
\DeclareOption{8pt}{\def\@mainsize{8}\def\@ptsize{8}%
  \def\@typesizes{%
    \or{5}{6}\or{5}{6}\or{5}{6}\or{6}{7}\or{7}{8}%
    \or{8}{10}% normalsize
    \or{9}{11}\or{10}{12}\or{\@xipt}{13}%
    \or{\@xiipt}{14}\or{\@xivpt}{17}}%
  \normalsize \linespacing=\baselineskip
}
%
\DeclareOption{9pt}{\def\@mainsize{9}\def\@ptsize{9}%
  \def\@typesizes{%
    \or{5}{6}\or{5}{6}\or{6}{7}\or{7}{8}\or{8}{10}%
    \or{9}{11}% normalsize
    \or{10}{12}\or{\@xipt}{13}\or{\@xiipt}{14}%
    \or{\@xivpt}{17}\or{\@xviipt}{20}}%
  \normalsize \linespacing=\baselineskip
}
```

Running heads

The normal application of pagestyle functions `\ps@xxx` is to determine the contents of running heads and feet. The function `\@mkboth` is used internally by commands `\chapter`, `\section`, and the like to set the running heads.

```
\def\ps@empty{\let\@mkboth\@gobbletwo
  \let\@oddhead\@empty \let\@evenhead\@empty
  \let\@oddfoot\@empty \let\@evenfoot\@empty
```

The current implementation in `amsart`/`amsproc`/`amsbook` of the vertical space at the top of an opening page uses `\topskip`, which means that we need to do some resetting here.

```
  \global\topskip\normaltopskip}
```

Pagestyle 'plain' has the page numbers in the running feet.

```
\def\ps@plain{\ps@empty
  \def\@oddfoot{\normalfont\scriptsize \hfil\thepage\hfil}%
  \let\@evenfoot\@oddfoot}
```

Pagestyle 'headings' uses text from sectioning commands for running heads. Empty running feet.

```
\newswitch{runhead}
```

```
\def\ps@headings{\ps@empty
  \def\@evenhead{%
    \setTrue{runhead}%
    \normalfont\scriptsize
    \rlap{\thepage}\hfil
    \def\thanks{\protect\thanks@warning}%
    \leftmark{}{}\hfil}%
  \def\@oddhead{%
    \setTrue{runhead}%
    \normalfont\scriptsize \hfil
    \def\thanks{\protect\thanks@warning}%
    \rightmark{}{}\hfil \llap{\thepage}}%
  \let\@mkboth\markboth
⟨*amsbook⟩
  \def\partmark{\@secmark\markboth\partrunhead\partname}%
  \def\chaptermark{%
    \@secmark\markboth\chapterrunhead{}}%
  \def\sectionmark{%
    \@secmark\markright\sectionrunhead\sectionname}%
⟨/amsbook⟩
}
```

ctionname ctionname ctionname graphname graphname

Initialize section headings.

```
\let\sectionname\@empty
\let\subsectionname\@empty
\let\subsubsectionname\@empty
\let\paragraphname\@empty
\let\subparagraphname\@empty
```

The default definitions of `\leftmark`, `\rightmark` are not what we want: the section title (or whatever) reported in the right-hand running head should report the section that is current at the bottom of the right-hand page. And the left-hand running head should report the status at the top of the page. Cf. `amsppt.sty`.

```
\def\leftmark{\expandafter\@firstoftwo\topmark{}{}}
\def\rightmark{\expandafter\@secondoftwo\botmark{}{}}
```

Journal and Proceedings articles require an indication of the first page so the logo and copyright line can appear.

```
⟨*amsart | amsproc⟩
\def\ps@firstpage{\ps@plain
  \def\@oddfoot{\normalfont\scriptsize \hfil\thepage\hfil
```

Stick in the reset of `\topskip` here so it only gets executed after the first page is completed.

```
    \global\topskip\normaltopskip}%
  \let\@evenfoot\@oddfoot
  \def\@oddhead{\@serieslogo\hss}%
  \let\@evenhead\@oddhead % in case an article starts on a left-hand page
}
```

⟨/amsart | amsproc⟩

\@nilgobble Something that apparently doesn't exist in the kernel?

```
\long\def\@nilgobble#1\@nil{}
```

A general section-marking function. Arg 1 is either \markright or \markboth indicating which kind of marking action is desired (this gives us some string pool/hash table savings by allowing the \@secmark function to serve for both cases). Arg 2 is the function that should be called in the running head to process the remaining three args. Arg 3 is normally \xxxname (but could be empty). Arg 4 is the section-title text. Assumption: whenever \@secmark is called, the section-number variable \@secnumber has been set to the value of the current section number (possibly empty, in the case of a * section for example).

⟨*amsbook⟩

```
\def\@secmark#1#2#3#4{%
```

We want to apply expansion to \xxxname and \thexxx but not to the other elements.

```
  \begingroup \let\protect\@unexpandable@protect
  \edef\@tempa{\endgroup \toks@{\protect#2{#3}{\@secnumber}}}%
  \@tempa
  \toks@\@xp{\the\toks@{#4}}%
```

If a \markright operation is called for, use the current left-mark via \@temptokena.

```
  \afterassignment\@nilgobble\@temptokena\@themark{}\@nil
  \edef\@tempa{\@nx\@mkboth{%
    \ifx\markright#1\the\@temptokena\else\the\toks@\fi}{\the\toks@}}%
  \@tempa}
```

Init \@secnumber.

```
\let\@secnumber\@empty
```

⟨/amsbook⟩

Fix \markboth so that \@secmark can work without too much thrashing.

```
\def\markboth#1#2{%
  \begingroup
    \@temptokena{{#1}{#2}}\xdef\@themark{\the\@temptokena}%
    \mark{\the\@temptokena}%
  \endgroup
  \if@nobreak\ifvmode\nobreak\fi\fi}
```

With the myheadings pagestyle, no automatic running heads will be provided by the document class; only running heads specified by the user through explicit \markboth or \markright commands will be used.

```
\def\ps@myheadings{\ps@headings \let\@mkboth\@gobbletwo}
```

Save normal topskip value in a skip register.

```
\newskip\normaltopskip
\normaltopskip=10pt \relax
```

We also want to turn off all section marks. First-level section heads will be defined in `\ps@headings`.

```
\let\sectionmark\@gobble
\let\subsectionmark\@gobble
\let\subsubsectionmark\@gobble
\let\paragraphmark\@gobble
```

Unrecognized options

The `makeidx` option is redundant; everything that it does in the generic LaTeX classes is already done anyway in this class.

```
\DeclareOption{makeidx}{}
⟨/classes⟩
```

Unrecognized options for `amsthm` are treated as references to auxiliary theorem setup (`.thm`) files. This allows a user to create theorem styles using internal commands (with `@` signs) without having to be concerned about category coding.

Here is an example from the file `thmtest.tex` which is part of this collection. See that file for further information.

```
\begin{filecontents}{exercise.thm}
\def\th@exercise{%
  \normalfont % body font
  \thm@headpunct{:}%
}
\end{filecontents}
```

This facility is available only when `amsthm` is used as an independent package, not as part of an AMS document class.

```
⟨*amsthm⟩
\DeclareOption*{\input{\CurrentOption .thm}}
\ProcessOptions
⟨/amsthm⟩
```

2.3 Process options

Black boxes for overfull lines are turned off by default (the `final` option). This can be overridden with the `draft` option.

```
⟨*classes⟩
\ExecuteOptions{leqno,centertags,letterpaper,portrait,%
  10pt,twoside,onecolumn,final}
```

Options will be processed in the order of the associated `\DeclareOption` commands.

```
\ProcessOptions\relax
```

In compatibility mode, we want to load the frozen version of `amstex.sty` instead of the `amsmath` package. This is rather a horrible kluge but I can't see anything better at the moment. [mjd,1995/01/27]

```
\if@compatibility
  \def\@tempa{\RequirePackage{amstex}\relax}%
\else
  \@ifclasswith{\@classname}{nomath}{%
```

```
    \let\@tempa\relax
  }{%
    \def\@tempa{\RequirePackage{amsmath}\relax}%
  }%
\fi
\@tempa % load amstex.sty or amsmath.sty
```

If the `nomath` option was specified, then `\numberwithin` and `\@emptytoks` remain to be defined.

```
\@ifundefined{numberwithin}{%
  \newcommand{\numberwithin}[3][\arabic]{%
    \@ifundefined{c@#2}{\@nocounterr{#2}}{%
      \@ifundefined{c@#3}{\@nocnterr{#3}}{%
        \@addtoreset{#2}{#3}%
        \@xp\xdef\csname the#2\endcsname{%
          \@xp\@nx\csname the#3\endcsname .\@nx#1{#2}}}}%
  }
  \csname newtoks\endcsname\@emptytoks
}{}
```

If the `noamsfonts` option was called for, skip the `amsfonts` package load.

```
\if@compatibility
\else
  \@ifclasswith{\@classname}{noamsfonts}{%
    % amsfonts package is not wanted
  }{%
    % amsfonts package IS wanted; test whether a recent enough version
    % seems to be installed
    \begingroup \fontencoding{U}\fontfamily{msa}\try@load@fontshape\endgroup
    \global\@xp\let\csname U+msa\endcsname\relax % reset
    \@ifundefined{U/msa/m/n}{%
      \ClassError{\@classname}{%
  Package `amsfonts' not installed, or version too old?\MessageBreak
  Unable to get font info for the `msam' fonts in the expected form%
      }{%
  The amsfonts package will not be loaded, to avoid probable\MessageBreak
  incompatibility problems. You can (a) use the `noamsfonts'
      documentclass\MessageBreak
  option next time, or (b) check that the amsfonts package is
      installed\MessageBreak
  correctly, and is not too old to be compatible.%
      }%
    }{%
      \RequirePackage{amsfonts}[1995/01/01]\relax
    }%
  }
\fi % end yesamsfonts branch
```

2.4 Basic AMS style features

AMS style requires that blank pages between chapters be *really* blank: no running heads, no page numbers. To accomplish this, redefine `\cleardoublepage`

to do the right thing. [bnb, 1999/07/17]

```
\let\cleardouble@page\cleardoublepage
```

Postpone the redefinition of `\cleardoublepage` to begin-document to work around difficulties with old versions of `gsm-l.cls`.

```
\AtBeginDocument{%
  \ifx\cleardouble@page\cleardoublepage
    \def\cleardoublepage{\clearpage{\pagestyle{empty}\cleardouble@page}}
  \fi
}
```

Now a utility macro to do `\uppercase` but sidestep any math, to prevent uppercasing math variables. In order to be handled properly the `$...$` or `\(...\)` must be on the outer level (not enclosed in braces). We did not try to handle the possibility `\begin{math} ... \end{math}` in a title at the present time (too complicated). Also we increase inter-word space in the uppercase text.

One other little problem: uppercasing of a few special characters like the German ß (`\ss`) and the undotted i and j (`\i` and `\j`), used sometimes with accents. We redefine them to be uppercase equivalents. (Undotted `\i` and `\j` in math would be typed as `\imath` and `\jmath`.)

Spaceskip is changed in accordance with recommendations for increased interword spacing in all-caps text by e.g. 'Words into Type'.

```
\newcommand{\uppercasenonmath}[1]{\toks@\@emptytoks
%    Insert an extra \@empty to avoid removing braces around arg \arg{1}.
  \@xp\@skipmath\@xp\@empty#1$$%
```

The `\protect` here is in case the shorttitle gets used for shortauthors and we get redundant application of `\MakeUppercase`. Double braces limit the scope so that later elements in title block aren't uppercased, e.g., `\i` in an address. [bnb, 2004/04/01]

```
  \edef#1{{\@nx\protect\@nx\@upprep\the\toks@}}%
}
```

`\@upprep` Preparations for printing all-caps text.

```
\newcommand{\@upprep}{%
  \spaceskip1.3\fontdimen2\font plus1.3\fontdimen3\font
  \upchars@}
```

`\upchars@` In all-caps text, esszet should print as SS, dotless i should print as normal cap I, Mc should print with a small-caps (not lowercase) c, and so forth.

```
\newcommand{\upchars@}{%
  \def\ss{SS}\def\i{I}\def\j{J}\def\ae{\AE}\def\oe{\OE}%
  \def\o{\O}\def\aa{\AA}\def\l{\L}\def\Mc{M{\scshape c}}}
```

`\Mc` The use of `\Mc` makes it possible for 'Mc' to get special treatment when uppercasing is applied.

```
\providecommand{\Mc}{Mc}
```

\@skipmath
\@xskipmath

\@skipmath searches for $...$ in order to keep from applying \uppercase to the contents. Then it calls \@xskipmath to search for \(...\).

```
\newcommand{\@skipmath}{}
\long\def\@skipmath#1$#2$%
  \@xskipmath#1\(\)%
  \@ifnotempty{#2}{\toks@\@xp{\the\toks@$#2$}\@skipmath\@empty}}%
%
\newcommand{\@xskipmath}{}
\long\def\@xskipmath#1\(#2\){%
%    Expand away the added \@empty
  \uppercase{\toks@\@xp\@xp\@xp{\@xp\the\@xp\toks@#1}}%
  \@ifnotempty{#2}{\toks@\@xp{\the\toks@\(#2\)}\@xskipmath\@empty}}%
```

\altucnm

```
\def\altucnm#1{%
  \MakeTextUppercase{\toks@{#1}}%
  \edef#1{\the\toks@}%
}
\AtBeginDocument{%
  \@ifundefined{MakeTextUppercase}{}{\let\uppercasenonmath\altucnm}%
}
```

For older versions of LaTeX this might be needed too:

```
\@ifundefined{MakeUppercase}{\let\MakeUppercase\uppercase}{}%
```

\today

The command \today produces today's date in the form most commonly used in the U.S.

```
\newcommand{\today}{%
  \relax\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
```

2.5 Old font commands

The \em command is not redefined here (let's say, to give an 'obsolete' warning and recommend instead \emph) because there is no alternative internal command \emshape.

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
```

This warning might have been a good idea back in 1995 but I don't think we can add it now [mjd,2000/03/10].

```
%\if@compatibility
```

```
%\else
%  \def\@obsolete@fontswitch#1#2#3{%
%    \@latex@warning@no@line{%
%      Command {\string#1...}\on@line\space is obsolete;\MessageBreak
%      the LaTeX2e equivalent is \string#3{...}}%
%    \gdef#1{\@fontswitch\relax#3}%
%  }
%  \DeclareRobustCommand*\cal{%
%    \@xp\@obsolete@fontswitch\csname cal \endcsname\relax\mathcal}
%  \DeclareRobustCommand*\mit{%
%    \@xp\@obsolete@fontswitch\csname mit \endcsname\relax\mathnormal}
%\fi
```

There's too great a chance that some people out there have documents that begin with

```
\documentclass{amsart}
...
\newcommand{\cal}{\mathcal}
```

and such documents would now get error messages.

Since `\cal` is not documented as a valid command for LaTeX 2_ε, I think it is OK to leave the status as is for AMS document classes. If you use `\documentstyle`, `\cal` will work. If you use `\documentclass`, well, it's a good idea to update to `\mathcal` instead of leaving old instances of `\cal`.

2.6 Top matter

For the `\title` command, we support an optional argument to give a shortened version of the title for running heads.

```
\renewcommand*{\title}[2][]{\gdef\shorttitle{#1}\gdef\@title{#2}}
```

The default value for the optional argument is 'same as the mandatory arg' but there doesn't seem to be an easy way to get that effect with `\[re]newcommand`. Here is how to use `\@dblarg` in conjunction with the preceding `\newcommand`:

```
\edef\title{\@nx\@dblarg
  \@xp\@nx\csname\string\title\endcsname}
```

The `\author` command accepts an optional argument similar to that of the `\title` command. Moved update of `\addresses` within scope of `\else` to avoid adding anything if no authors, and thus avoiding output of "Author address" on monograph titlepage. [bnb, 1996/11/03]

```
\renewcommand{\author}[2][]{%
  \ifx\@empty\authors
    \gdef\authors{#2}%
  \else
    \g@addto@macro\authors{\and#2}%
    \g@addto@macro\addresses{\author{}}%
  \fi
  \@ifnotempty{#1}{%
    \ifx\@empty\shortauthors
      \gdef\shortauthors{#1}%
```

```
    \else
      \g@addto@macro\shortauthors{\and#1}%
    \fi
  }%
}
\edef\author{\@nx\@dblarg
  \@xp\@nx\csname\string\author\endcsname}
```

Initialize some variables.

```
\let\shortauthors\@empty   \let\authors\@empty
```

`\contrib` Contributors are similar to authors except that they are responsible for only part of a work, e.g., an appendix. The optional argument for the first contributor of a group identifies what has been contributed. There can be more than one group of contributors; each group is treated separately, using the same "and" conventions within the group as for authors. Contributor groups are strung together separated by a comma; if the word "and" is desired before the final group of contributors, it must be included in the optional argument for that group.

```
⟨*amsart | amsproc⟩
\newif\ifresetcontrib  \resetcontribfalse
\newcommand\contrib[2][]{%
  \def\@tempa{#1}%
  \ifx\@empty\@tempa
  \else
    \ifresetcontrib \@xcontribs
    \else \global\resetcontribtrue
    \fi
  \fi
  \ifx\@empty\contribs
    \gdef\contribs{#1 #2}%
  \else
    \g@addto@macro\contribs{\and#1 #2}%
  \fi
```

Accumulate contribs separately for the table of contents. Here, this is just a dummy; it is fully defined for in-house processing.

```
  \@wraptoccontribs{#1}{#2}%
}
\def\wraptoccontribs#1#2{}
\def\@xcontribs{%
  \author@andify\contribs
  \ifx\@empty\xcontribs
    \xdef\xcontribs{\contribs}%
  \else
    \xdef\xcontribs{\xcontribs, \contribs}%
  \fi
  \let\contribs\@empty
}
```

Initialize some more variables.

```
\let\contribs\@empty  \let\xcontribs\@empty  \let\toccontribs\@empty
⟨/amsart | amsproc⟩
\let\addresses\@empty      \let\thankses\@empty
```

The optional arguments of `\address`, `\curraddr`, `\email` are to indicate which author the address applies to, if a document has multiple authors and there is not a normal one-to-one correspondence between authors and addresses.

```
\newcommand{\address}[2][]{\g@addto@macro\addresses{\address{#1}{#2}}}
\newcommand{\curraddr}[2][]{\g@addto@macro\addresses{\curraddr{#1}{#2}}}
\newcommand{\email}[2][]{\g@addto@macro\addresses{\email{#1}{#2}}}
\newcommand{\urladdr}[2][]{\g@addto@macro\addresses{\urladdr{#1}{#2}}}
```

Someone who does not look closely at the `amsart` documentation is likely to put the `\thanks` command inside that argument of `\author`.

```
\long\def\thanks@warning#1{%
  \ClassError{\@classname}{%
    \protect\thanks\space should be given separately, not inside author name.%
  }\@ehb
}
\renewcommand{\thanks}[1]{%
  \@ifnotempty{#1}{\g@addto@macro\thankses{\thanks{#1}}}%
}
```

The following example of addresses for three authors of a tri-author paper illustrates the kind of complications that need to be handled.

```
\author{Roland Campbell}
\address{Department of Mathematics\\
  Pennsylvania State University\\
  Pittsburgh, Pennsylvania 13593}
\email[R.~Campbell]{campr@@galois.psu.edu}

\author{Mark M. Dane}
% Same address as R. Campbell
\curraddr[M.~Dane]{Atmospheric Research Station\\
  Pala Lundi, Fiji}
\email[M.~Dane]{DaneMark@@ffr.choice}

\author{Jeremiah Jones}
\address[J.~Jones]{Department of Philosophy\\
Freedman College\\
Periwinkle, Colorado 84320}
\email[J.~Jones]{id739e@@oseoi44 (Bitnet)}
```

In an article, typesetting of the address information is done at the end of the document, by calling `\@setaddresses`. This is done through a parent function `\enddoc@text`, because some AMS journals also print the abstract there instead of at the beginning, and it's easier to redefine `\enddoc@text` than to try undoing material already added to the `\AtEndDocument` hook.

```
⟨*amsart | amsproc⟩
```

```
\def\enddoc@text{\ifx\@empty\@translators \else\@settranslators\fi
  \ifx\@empty\addresses \else\@setaddresses\fi}
\AtEndDocument{\enddoc@text}
⟨/amsart | amsproc⟩
\def\curraddrname{{\itshape Current address}}
\def\emailaddrname{{\itshape Email address}}
\def\urladdrname{{\itshape URL}}
\def\@setaddresses{\par
  \nobreak \begingroup
⟨amsart | amsproc⟩\footnotesize
  \def\author##1{\nobreak\addvspace\bigskipamount}%
```

Address is supposed to go all on one line, so we redefine `\\` to just insert a comma instead of doing a line break.

```
  \def\\{\unskip, \ignorespaces}%
```

No page breaks in the address section is accomplished by `\interlinepenalty\@M` and by the `\nobreak` before the `\bigskip`.

```
  \interlinepenalty\@M
  \def\address##1##2{\begingroup
```

If there are two addresses for the same author, add a `\bigskip` between them.

```
    \par\addvspace\bigskipamount\indent
```

If the name of the author to whom this address applies was given, typeset it.

```
    \@ifnotempty{##1}{(\ignorespaces##1\unskip) }%
```

Now the main part of the address:

```
    {\scshape\ignorespaces##2}\par\endgroup}%
```

Current address:

```
  \def\curraddr##1##2{\begingroup
    \@ifnotempty{##2}{\nobreak\indent\curraddrname
      \@ifnotempty{##1}{, \ignorespaces##1\unskip}\/:\space
      ##2\par}\endgroup}%
```

And then email. In versions 1.0 and 1.1 `@@` was required to print a single `@` character; for bulletproofing we convert doubled `@` characters if found.

```
  \def\email##1##2{\begingroup
    \@ifnotempty{##2}{\nobreak\indent\emailaddrname
      \@ifnotempty{##1}{, \ignorespaces##1\unskip}\/:\space
      \ttfamily##2\par}\endgroup}%
```

URLaddr is simply a replica of the email address, with the addition of a feature to enable `~` to print.

```
  \def\urladdr##1##2{\begingroup
    \def~{\char`\~}%
    \@ifnotempty{##2}{\nobreak\indent\urladdrname
      \@ifnotempty{##1}{, \ignorespaces##1\unskip}\/:\space
      \ttfamily##2\par}\endgroup}%
  \addresses
  \endgroup
}
```

Some other administrative info. For \date we can just use the default definition provided by LaTeX, except that we initialize the date to empty instead of to \today.

```
\let\@date\@empty
\def\dedicatory#1{\def\@dedicatory{#1}}
\let\@dedicatory=\@empty
\def\keywords#1{\def\@keywords{#1}}
\let\@keywords=\@empty
```

To allow various versions of the subject classification, accept an optional value to identify the version, provide text for the two currently in use, and give a warning if the version specified is unknown. Default to 2020 version.

```
\newcommand*\subjclass[2][2020]{%
  \def\@subjclass{#2}%
  \@ifundefined{subjclassname@#1}{%
    \ClassWarning{\@classname}{Unknown edition (#1) of Mathematics
      Subject Classification; using '2020'.}%
  }{%
    \@xp\let\@xp\subjclassname\csname subjclassname@#1\endcsname
  }%
}
\let\@subjclass=\@empty
⟨amsart⟩\def\commby#1{\def\@commby{(Communicated by #1)}}
⟨amsart⟩\let\@commby=\@empty
```

We handle translator names like author names, just in case there is more than one translator. [mjd,1994/10/19]

```
\def\translname{Translated by}
\def\translator#1{%
  \ifx\@empty\@translators \def\@translators{#1}%
  \else\g@addto@macro\@translators{\and#1}\fi}
\let\@translators=\@empty
```

```
⟨*amsart | amsproc⟩
\def\@settranslators{\par\begingroup
  \addvspace{6\p@\@plus9\p@}%
  \hbox to\columnwidth{\hss\normalfont\normalsize
    \translname{ }%
    \andify\@translators \uppercasenonmath\@translators
    \@translators}
  \endgroup
}
⟨/amsart | amsproc⟩
```

The general function to convert a list of items in the form

```
A\and B\and C\and D
```

to the form 'A, B, C, and D' is \xandlist:

```
\xandlist{, }{ and }{, and }{A\and B\and C\and D}
```

This is a completely expandable macro, with the return value being the converted list. There is also a 'no-execute' version whose fourth argument should

be a macro; the text to be converted will be taken from that macro and after conversion will be put back as the macro's new replacement text.

```
\nxandlist{, }{ and }{, and }\result
```

I don't think I want to explain this except by recommending that you watch it in operation with `\tracingmacros` if you're interested. [mjd,1994/10/19]

```
\newcommand{\xandlist}[4]{\@andlista{{#1}{#2}{#3}}#4\and\and}
\def\@andlista#1#2\and#3\and{\@andlistc{#2}\@ifnotempty{#3}{%
  \@andlistb#1{#3}}}
\def\@andlistb#1#2#3#4#5\and{%
  \@ifempty{#5}{%
    \@andlistc{#2#4}%
  }{%
    \@andlistc{#1#4}\@andlistb{#1}{#3}{#3}{#5}%
  }}
\let\@andlistc\@iden
\newcommand{\nxandlist}[4]{%
  \def\@andlistc##1{\toks@\@xp{\the\toks@##1}}%
  \toks@{\toks@\@emptytoks \@andlista{{#1}{#2}{#3}}}%
  \the\@xp\toks@#4\and\and
  \edef#4{\the\toks@}%
  \let\@andlistc\@iden}
```

andify The `\andify` function is provided as a convenient abbreviation for the most common case. See also `\author@andify` (for `amsart` and `amsproc` only), which gives better results in cases with a large number of authors. Provide a substitutable text string to simplify language-specific modifications.

```
\def\@@and{and}
\newcommand{\andify}{%
  \nxandlist{\unskip, }{\unskip{} \@@and~}{\unskip, \@@and~}}
```

Override the funny default definition of `\and` from LaTeX. This is not actually used by AMS classes, however.

```
\def\and{\unskip{ }\@@and{ }\ignorespaces}
```

maketitle Set up the style of an article opening page. For books, see below. For articles, we must add the copyright info footnote.

```
⟨*amsart | amsproc⟩
\def\maketitle{\par
  \@topnum\z@ % this prevents figures from falling at the top of page 1
  \@setcopyright
  \thispagestyle{firstpage}% this sets first page specifications
```

Do some setup for the running heads here. If there are no author names, we set the left-hand running head to the value of the right-hand running head.

```
  \uppercasenonmath\shorttitle
  \ifx\@empty\shortauthors \let\shortauthors\shorttitle
  \else \andify\shortauthors
  \fi
```

The following hook is used to activate the writing of author and title information to an 'issue table of contents' when multiple articles are being processed for a journal issue or a proceedings volume.

```
\@maketitle@hook
\begingroup
\@maketitle
\toks@\@xp{\shortauthors}\@temptokena\@xp{\shorttitle}%
\toks4{\def\\{ \ignorespaces}}% defend against questionable usage
\edef\@tempa{%
  \@nx\markboth{\the\toks4
    \@nx\MakeUppercase{\the\toks@}}{\the\@temptokena}}%
\@tempa
\endgroup
\c@footnote\z@
\@cleartopmattertags
}
\def\@cleartopmattertags{%
  \def\do##1{\let##1\relax}%
  \do\maketitle \do\@maketitle \do\title \do\@xtitle \do\@title
  \do\author \do\@xauthor \do\address \do\@xaddress
  \do\contrib \do\contribs \do\xcontribs \do\toccontribs
  \do\email \do\@xemail \do\curraddr \do\@xcurraddr
⟨amsart⟩  \do\commby \do\@commby
  \do\dedicatory \do\@dedicatory \do\thanks \do\thankses
  \do\keywords \do\@keywords \do\subjclass \do\@subjclass
}
⟨/amsart | amsproc⟩
```

The hook `\@maketitle@hook` is placed into `\maketitle` rather than `\@maketitle` because the latter tends to get redefined by derived classes using this one as a base. The initial motivation for this hook is to extract title and author information to an external file, so we can't do it with `\AtBeginDocument`: `\title` and `\author` commands might occur between `\begin{document}` and `\maketitle`.

```
⟨*amsart | amsproc⟩
\def\@maketitle@hook{\global\let\@maketitle@hook\@empty}
⟨/amsart | amsproc⟩
```

Set up the style of an article opening page.

```
⟨*amsart | amsproc⟩
\def\@maketitle{%
```

Set font to normal, just in case.

```
  \normalfont\normalsize
```

Special footnotes are put here to ensure that they come first at the bottom of the page.

```
  \@adminfootnotes
```

If `\pagestyle{myheadings}` was specified, `\@mkboth` will be a no-op.

```
  \@mkboth{\@nx\shortauthors}{\@nx\shorttitle}%
```

```
⟨amsproc⟩  \global\topskip8pc\relax % 10pc to base of first title line
⟨amsart⟩  \global\topskip42\p@\relax % 5.5pc   "   "   "     "     "
  \@settitle
  \ifx\@empty\authors \else \@setauthors \fi
```

Likewise with `\@dedicatory` and `\@date`.

```
  \ifx\@empty\@dedicatory
  \else
⟨amsproc⟩    \baselineskip26\p@
⟨amsart⟩    \baselineskip18\p@
    \vtop{\centering{\footnotesize\itshape\@dedicatory\@@par}%
      \global\dimen@i\prevdepth}\prevdepth\dimen@i
  \fi
  \@setabstract
```

Space before the main text should be 32 + 14 base-to-base; we accomplish this by doing a vskip of that amount with `\baselineskip` subtracted.

```
  \normalsize
  \if@titlepage
    \newpage
  \else
    \dimen@34\p@ \advance\dimen@-\baselineskip
    \vskip\dimen@\relax
  \fi
} % end \@maketitle
```

Segregate the definitions of administrative footnotes to permit easier customization, especially for translation journals.

```
\def\@adminfootnotes{%
  \let\@makefnmark\relax  \let\@thefnmark\relax
⟨amsart⟩  \ifx\@empty\@date\else \@footnotetext{\@setdate}\fi
  \ifx\@empty\@subjclass\else \@footnotetext{\@setsubjclass}\fi
  \ifx\@empty\@keywords\else \@footnotetext{\@setkeywords}\fi
```

In order to make multiple thanks footnotes work inside a single `\@footnotetext` argument we need to make the first `\par` be ignored. Cf. `\@setthanks`.

```
  \ifx\@empty\thankses\else \@footnotetext{%
    \def\par{\let\par\@par}\@setthanks}%
  \fi
}
⟨/amsart | amsproc⟩
```

2.7 Journal/series logo for articles

`\publname` either will be defined by a parent class that is calling `amsart` or `amsproc` as a base class, or will be absent, in which case this can be used as a signal to omit the `\@serieslogo`. A typical value for `\publname` would be

```
\def\publname{JOURNAL OF THE\newline
  AMERICAN MATHEMATICAL SOCIETY}
```

Initialize `\publname` and `\@serieslogo` to no-op if `\publname` is not already defined.

```
⟨*amsart | amsproc⟩
\AtBeginDocument{%
  \@ifundefined{publname}{%
    \let\publname\@empty
    \let\@serieslogo\@empty
  }{%
⟨amsart⟩    \def\@serieslogo{\article@logo}%
⟨amsproc⟩    \def\@serieslogo{\procart@logo}%
  }%
}
⟨/amsart | amsproc⟩
```

The `\number` prefix on current issue is to work around inconsistencies in the form of issue numbers as passed in from system level. Sometimes they will get passed in with a leading zero, which we don't want to print if it happens to occur. The `0` prevents an error if `\currentissue` happens to be empty.

```
⟨*amsart⟩
\AtBeginDocument{%
  \@ifundefined{volinfo}{%
    \def\volinfo{%
      Volume \currentvolume, Number \number0\currentissue
```

Month/year is not included initially when a journal article is posted on the WWW prior to print publication.

```
      \if\@printyear , \currentmonth\ \currentyear\fi
    }%
  }{}%
}
\def\@printyear{TF}% boolean false
⟨/amsart⟩

⟨*amsproc⟩
\AtBeginDocument{%
  \@ifundefined{volinfo}{\let\volinfo\@empty}{}
}
⟨/amsproc⟩
```

Default values for information such as volume, year, and so on are provided as follows.

```
⟨*amsart | amsproc⟩
\def\issueinfo#1#2#3#4{\def\currentvolume{#1}\def\currentissue{#2}%
  \def\currentmonth{#3}\def\currentyear{#4}}
\issueinfo{00}% volume number
  {0}%        % issue number
  {Xxxx}%     % month
  {XXXX}%     % year
⟨/amsart | amsproc⟩
```

Copyright year may be different from issue year. Allow it to be specified separately. It is probably more natural anyway, from the user's perspective,

to give the copyright year in the same command when giving the name of the copyright holder.

```
\newcommand{\copyrightinfo}[2]{%
  \def\copyrightyear{#1}%
  \@ifnotempty{#2}{\def\copyrightholder{#2}}%
}
\copyrightinfo{0000}{(copyright holder)}
```

Provide page span information. If negative number is given, convert to roman numeral form.

```
⟨*amsart | amsproc⟩
\def\pagespan#1#2{\setcounter{page}{#1}%
  \ifnum\c@page<\z@ \pagenumbering{roman}\setcounter{page}{-#1}\fi
  \def\start@page{#1}\def\end@page{#2}}
\pagespan{000}{000}
⟨/amsart | amsproc⟩
```

Formatting for journal page numbers. [bnb, 1996/09/11]

```
⟨*amsart⟩
\AtBeginDocument{%
  \@ifundefined{pageinfo}{%
    \def\pageinfo{%
      \ifnum\start@page=\z@
        Pages 000--000
      \else
        \ifx\start@page\end@page
          Page \start@page
        \else
          Pages \start@page--\end@page
        \fi
      \fi}%
  }{}%
}
⟨/amsart⟩
```

Publisher Item Identifier (we started using them in journal logos as of January 1997).

```
⟨amsart⟩\@ifundefined{ISSN}{\def\ISSN{0000-0000}}{}
⟨amsart⟩\newcommand\PII[1]{\def\@PII{#1}}
⟨amsart⟩\PII{S \ISSN(XX)0000-0}
```

2.8 Copyright block

Doing the copyright info on the first page is a little tricky. We want it to come at the bottom, after any footnotes and floating inserts, but before the page number. If we simply put it into `\@oddfoot` (in `\ps@plain`) its height will not be subtracted from the height of the text and then the page number will be lower than we want. So we do it as an insert.

Through version 1.2, this code was included only for `amsart` and `amsproc`; the formatting of some book series requires an insert at the bottom of the text

block, so this code has been extended to all AMS document classes. [bnb, 1999/07/14]

```
\newinsert\copyins
```

We set the skip register associated with this insert to the *base-to-base* distance from the bottom of the page contents to the base of the first line in the copyright info. See the definition of `\@setcopyright`.

```
\skip\copyins=1.5pc
\count\copyins=1000 % magnification factor, 1000 = 100%
\dimen\copyins=.5\textheight % maximum allowed per page
```

`\copyins` is ignored if a float is input on the first page; adding it to `\@reinserts` will make the output routine behave. [bnb; 2004/06/09; B-365]

```
\g@addto@macro\@reinserts{%
  \ifvoid\copyins\else\insert\copyins{\unvbox\copyins}\fi
}
```

Put the contents into a TEX insert. This information is omitted unless `\@serieslogo` is non-null. In other words it will normally not print except when an AMS publication-specific document class such as `tran-l` is used. And even if the series logo is printed, omit the copyright line if requested by `\coprightinfo{}{}`. [bnb, 1996/10/17]

```
\def\@copyinsfontsize{\fontsize{6}{7\p@}\normalfont\upshape}
\newif\if@extracrline  \@extracrlinefalse
\let\@extracrline\@empty
\relax
\def\@setcopyright{%
  \ifx\@empty\@serieslogo
  \else\ifx\@empty\copyrightyear
  \else
    \insert\copyins{\hsize\textwidth
      \parfillskip\z@\relax
      \leftskip\z@\@plus.9\textwidth\relax \rightskip\z@\relax
      \@copyinsfontsize
```

The spacing between the preceding text and the copyright info is done with a strut of height `\skip\copyins`. (Note that `\lineskip` and `\baselineskip` are 0 in the LATEX output routine.) The negative vskip gives an effective distance of 0 from the top of the box to the base of the first line (assuming `\skip\copyins` is greater than the height of that line). Then the apparent total height of the box will work well with TEX's calculations involving `\skip\copyins` for how much room to leave for this object. An extra 6pt is allowed when an additional line is present; this adjustment was found adequate in some borderline cases where tight pages reset with the additional line had text lines moved to the next page, causing the paper length to expand by a page. [bnb, 2004/05/07-06/24]

```
      \everypar{}%
      \vskip-\skip\copyins
      \if@extracrline
        \vskip-6pt
      \fi
```

```
      \nointerlineskip
      \leavevmode\hfill\vrule\@width\z@\@height\skip\copyins
      \copyright\copyrightyear\ \copyrightholder\ignorespaces
      \if@extracrline \@extracrline \fi
      \par
```

This kern of 0pt forces the depth of the last line (if any) to be added to the height of the box.

```
      \kern\z@}%
  \fi\fi
}
```

When `\@combinefloats` is called, the box `\@outputbox` already contains the main text of the page and any footnotes. Then LaTeX adds top and bottom figures. We want to add our copyright info at the very bottom, but still inside of the vbox.

```
\def\@combinefloats{%
  \ifx \@toplist\@empty \else \@cflt \fi
  \ifx \@botlist\@empty \else \@cflb \fi
  \ifvoid\copyins \else \@cflci \fi
}
```

combine-floats-copyright-insert

In the twocolumn/firstcolumn case, postpone adding the drop folio. Put an empty box of the same height at the bottom of the left-hand column to make the columns balance. Allow a smidge of stretch in case the first page of a chapter has no internal stretch, so that the drop folio will be flush to the bottom of the text block; don't do this for a twocolumn page, to avoid an uneven bottom.

```
\def\@cflci{%
  \setbox\@outputbox\vbox{%
    \unvbox\@outputbox
    \vskip\skip\copyins
    \if@twocolumn \else \vskip\z@ plus\p@ \fi
    \hbox to\columnwidth{%
      \hss\vbox to\z@{\vss
```

In two-column layout, put an empty box in the first column instead of the drop folio.

```
        \if@twocolumn
          \if@firstcolumn \else \unvbox\copyins \fi
        \else
          \unvbox\copyins
        \fi
        }}}%
```

Now redo the insert to make sure we get the right amount of space reserved for it in the second column.

```
  \if@twocolumn \if@firstcolumn
    \insert\copyins{\unvbox\copyins}%
  \fi\fi
```

Reset the `\copyins` flag so that a subsequent insert (e.g. `\@dropfolio` in some book series) will work.

```
  \global\count\copyins=999 \relax
}
```

[End of code to support inserts at end of text block.]

For journals only, provide a switch that indicates the author has agreed to revert copyright to the public domain; this results in an addition to the copyright block on the article.

```
⟨*amsart⟩
\newif\if@revertcopyright  \@revertcopyrightfalse
\newcommand{\revertcopyright}{%
  \global\@revertcopyrighttrue
  \global\@extracrlinetrue}
```

Add notation regarding reversion of copyright to public domain if author has agreed to it. Permit this to be set in a different size than the copyright line (required for some author packages).

```
\def\@revertcrfontsize{\fontsize{6}{7\p@}\normalfont\upshape}
\def\@extracrline{%
  \if@revertcopyright
    \unskip\\
    \@revertcrfontsize
    Reverts to public domain 28 years from publication
  \fi
}
⟨/amsart⟩
```

2.9 Other top matter info

Some name setup.

```
\newcommand{\abstractname}{Abstract}
\newcommand{\keywordsname}{Key words and phrases}
```

Support the most recent versions. Earlier, unsupported versions were:

- 1980 Mathematics Subject Classification
- 1980 Mathematics Subject Classification (1985 Revision)

```
\@namedef{subjclassname@1991}{%
  \textup{1991} Mathematics Subject Classification}
\@namedef{subjclassname@2000}{%
  \textup{2000} Mathematics Subject Classification}
\@namedef{subjclassname@2010}{%
  \textup{2010} Mathematics Subject Classification}
\@namedef{subjclassname@2020}{%
  \textup{2020} Mathematics Subject Classification}
```

Default to the 2020 edition.

```
\@xp\let\@xp\subjclassname\csname subjclassname@2020\endcsname
```

For the date we have a special little problem: We only want to add the 'Received by the editors' text for publication-specific document classes such as `tran-l`.

```
⟨amsbook⟩\def\@tempb{amsbook}
⟨amsproc⟩\def\@tempb{amsproc}
⟨amsart⟩\def\@tempb{amsart}
\ifx\@classname\@tempb
  \newcommand{\datename}{\textit{Date}:}
\else
  \newcommand{\datename}{Received by the editors}
\fi
```

```
⟨*amsart | amsproc⟩
\def\@settitle{\begin{center}%
⟨amsart⟩  \baselineskip14\p@\relax
⟨amsproc⟩ \Large
    \bfseries
⟨amsart⟩\uppercasenonmath\@title
  \@title
  \end{center}%
}
⟨/amsart | amsproc⟩
```

For multiple authors we need to combine the author names into a list of the form Author One, Author Two, ..., and Author Last. Change line breaking penalties to avoid a line break in the middle of an author name if there are a lot of authors. This should probably better be done by changing spaces within author names to `\nolinebreak[3]\space`, but that would take more work. [mjd,2000/12/27]

```
⟨*amsart | amsproc⟩
\def\author@andify{%
  \nxandlist {\unskip ,\penalty-1 \space\ignorespaces}%
    {\unskip {} \@@and~}%
    {\unskip ,\penalty-2 \space \@@and~}%
}
```

```
\def\@setauthors{%
  \begingroup
  \def\thanks{\protect\thanks@warning}%
  \trivlist
⟨amsart⟩  \centering\footnotesize \@topsep30\p@\relax
⟨amsart⟩  \advance\@topsep by -\baselineskip
⟨amsproc⟩  \centering \fontsize{11}{13\p@}\selectfont\@topsep13\p@\relax
  \item\relax
  \author@andify\authors
```

Make dbl-backslash robust to prevent problems if/when `\MakeTextUppercase` expansion is applied.

```
  \def\\{\protect\linebreak}%
⟨amsart⟩  \MakeUppercase{\authors}%
⟨amsproc⟩  \authors
  \ifx\@empty\contribs
  \else
    ,\penalty-3 \space \@setcontribs
```

```
    \@closetoccontribs
  \fi
  \endtrivlist
  \endgroup
}
\def\@closetoccontribs{}
\def\@setcontribs{%
  \@xcontribs
⟨amsart⟩  \MakeUppercase{\xcontribs}%
⟨amsproc⟩  \xcontribs
}
\def\@setdate{\datename\ \@date\@addpunct.}
\def\@setsubjclass{%
  {\itshape\subjclassname.}\enspace\@subjclass\@addpunct.}
\def\@setkeywords{%
  {\itshape \keywordsname.}\enspace \@keywords\@addpunct.}
⟨/amsart | amsproc⟩
```

The following definitions suffice for all the AMS document classes.

```
\def\@setthanks{\def\thanks##1{\par##1\@addpunct.}\thankses}
```

'Abstract' can be changed to, say, 'Résumé' (French) by redefining `\abstractname`. This and other control sequence names (`\refname`, `\contentsname`, `\appendixname`, and so on) are compatible with the `babel` package (the AMS sometimes publishes articles in languages other than English.)

In AMS document classes, the abstract should be placed before `\maketitle` (otherwise the desired ordering of frontmatter elements cannot be ensured in all cases).

```
\newbox\abstractbox
```

We start by checking whether `\maketitle` has already been used (in which case it was reset to `\relax`); if so, we give a warning that the abstract should be placed before `\maketitle`.

```
\newenvironment{abstract}{%
  \ifx\maketitle\relax
    \ClassWarning{\@classname}{Abstract should precede
      \protect\maketitle\space in AMS document classes; reported}%
  \fi
  \global\setbox\abstractbox=\vtop \bgroup
    \normalfont\Small
    \list{}{\labelwidth\z@
      \leftmargin3pc \rightmargin\leftmargin
      \listparindent\normalparindent \itemindent\z@
      \parsep\z@ \@plus\p@
```

In order to get equation numbers indented with the rest of the abstract, we have to do this:

```
      \let\fullwidthdisplay\relax
    }%
    \item[\hskip\labelsep\scshape\abstractname.]%
}{%
```

```
915   \endlist\egroup
```

If the abstract was supposed to be typeset earlier, then `\@setabstract` is now equal to `\relax`, and we had better drop the contents of the abstract box onto the page immediately, to salvage the situation as best we can.

```
916   \ifx\@setabstract\relax \@setabstracta \fi
917 }
```

Because the abstract might be postponed until the end of an article, we cannot simply use the fact of a preceding `\maketitle` to tell whether `\endabstract` should immediately put the abstract box contents on the page. So we use an auxiliary function that will be reset to no-op once we have passed the point where the abstract should normally be typeset.

```
918 \def\@setabstract{\@setabstracta \global\let\@setabstract\relax}
919 \def\@setabstracta{%
920   \ifvoid\abstractbox
921   \else
922     \skip@20\p@ \advance\skip@-\lastskip
923     \advance\skip@-\baselineskip \vskip\skip@
924     \box\abstractbox
925     \prevdepth\z@ % because \abstractbox is a vtop
926   \fi
927 }
```

Title page environment does nothing much; information and formatting to be provided by the user.

```
928 \def\titlepage{%
929 ⟨amsbook⟩   \cleardoublepage
930 ⟨amsart | amsproc⟩   \clearpage
931   \thispagestyle{empty}\setcounter{page}{0}}
932 \def\endtitlepage{\newpage}
```

2.10 Macros for list labels

Through version 1.2, first-level enumerated item labels were formatted with a following period, which is not AMS style. Effective with version 2.0 these labels are formatted with parentheses. Anyone requiring the period style will have to redefine `\labelenumi`.

```
933 \def\labelenumi{(\theenumi)}
934 \def\theenumi{\@arabic\c@enumi}
935 \def\labelenumii{(\theenumii)}
936 \def\theenumii{\@alph\c@enumii}
937 \def\p@enumii{\theenumi}
938 \def\labelenumiii{(\theenumiii)}
939 \def\theenumiii{\@roman\c@enumiii}
940 \def\p@enumiii{\theenumi(\theenumii)}
941 \def\labelenumiv{(\theenumiv)}
942 \def\theenumiv{\@Alph\c@enumiv}
943 \def\p@enumiv{\p@enumiii\theenumiii}

944 \def\labelitemi{$\m@th\bullet$}
```

```
\def\labelitemii{\bfseries --}% \upshape already done by \itemize
\def\labelitemiii{$\m@th\ast$}
\def\labelitemiv{$\m@th\cdot$}
```

2.11 Verse and quotation environments

```
\newenvironment{verse}{\let\\\@centercr
  \list{}{\itemsep\z@ \itemindent -1.5em\listparindent\itemindent
  \rightmargin\leftmargin \advance\leftmargin 1.5em}\item[]%
}{%
  \endlist
}
\let\endverse=\endlist % for efficiency
```

The left/right margins of the quotation environment are supposed to be the same as for the abstract environment.

```
\newenvironment{quotation}{\list{}{%
    \leftmargin3pc \listparindent\normalparindent
    \itemindent\z@
    \rightmargin\leftmargin \parsep\z@ \@plus\p@}%
  \item[]%
}{%
  \endlist
}
\let\endquotation=\endlist % for efficiency
\newenvironment{quote}{%
  \list{}{\rightmargin\leftmargin}\item[]%
}{%
  \endlist
}
\let\endquote=\endlist % for efficiency
```

2.12 List environments

Changed definition of `\trivlist`, `enumerate`, and `itemize` in order to have `\makelabel` apply `\upshape`.

```
\def\trivlist{\parsep\parskip\@nmbrlistfalse
  \@trivlist \labelwidth\z@ \leftmargin\z@
  \itemindent\z@
  \let\@itemlabel\@empty
  \def\makelabel##1{\upshape##1}}

\renewenvironment{enumerate}{%
  \ifnum \@enumdepth >3 \@toodeep\else
      \advance\@enumdepth \@ne
      \edef\@enumctr{enum\romannumeral\the\@enumdepth}\list
      {\csname label\@enumctr\endcsname}{\usecounter
        {\@enumctr}\def\makelabel##1{\hss\llap{\upshape##1}}}\fi
}{%
  \endlist
}
\let\endenumerate=\endlist % for efficiency
```

```
\renewenvironment{itemize}{%
  \ifnum\@itemdepth>3 \@toodeep
  \else \advance\@itemdepth\@ne
    \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
    \list{\csname\@itemitem\endcsname}%
      {\def\makelabel##1{\hss\llap{\upshape##1}}}%
  \fi
}{%
  \endlist
}
\let\enditemize=\endlist % for efficiency
\newcommand{\descriptionlabel}[1]{\hspace\labelsep \upshape\bfseries #1:}
\newenvironment{description}{\list{}{%
```

Adjust the indent of the first line to the desired value:

```
  \advance\leftmargini6\p@ \itemindent-12\p@
  \labelwidth\z@ \let\makelabel\descriptionlabel}%
}{
  \endlist
}
\let\enddescription=\endlist % for efficiency
```

\upn The command \upn can be used to force upright font for punctuation or digits in italic text. For example

```
... as numbered by \upn{``A''} or \upn{``1''} hereafter\upn{)}
```

```
\let\upn=\textup
```

Since these margin settings are dependent on the fonts used, we postpone them until begin-document. (This means that we cannot use the values directly for calculations before begin-document.) Allow for a reasonable maximum value; 13 = xiii = m should be adequate.

```
\AtBeginDocument{%
  \labelsep=5pt\relax
  \setcounter{enumi}{13}\setcounter{enumii}{13}%
  \setcounter{enumiii}{13}\setcounter{enumiv}{13}%
  \settowidth\leftmargini{\labelenumi\hskip\labelsep}%
  \advance\leftmargini by \normalparindent
  \settowidth\leftmarginii{\labelenumii\hskip\labelsep}%
  \settowidth\leftmarginiii{\labelenumiii\hskip\labelsep}%
  \settowidth\leftmarginiv{\labelenumiv\hskip\labelsep}%
  \setcounter{enumi}{0}\setcounter{enumii}{0}%
  \setcounter{enumiii}{0}\setcounter{enumiv}{0}%
  \leftmarginv=10pt  \leftmarginvi=\leftmarginv
  \leftmargin=\leftmargini
  \labelwidth=\leftmargini \advance\labelwidth-\labelsep
  \@listi}
```

In some contexts the space above/below lists needs to be suppressed. So we put it into a variable \listisep.

```
\newskip\listisep
\listisep\smallskipamount
\def\@listI{\leftmargin\leftmargini \parsep\z@skip
  \topsep\listisep \itemsep\z@skip
  \listparindent\normalparindent}
\let\@listi\@listI
```

Is it necessary to reset `\parsep`, `\partopsep`, `\itemsep` to their default values (0) in each of the subordinate list setup functions? I don't believe so, but I leave the settings in the listii function just in case some unusual nesting of environments might cause trouble. [mjd,1994/09/22]

```
\def\@listii{\leftmargin\leftmarginii
  \labelwidth\leftmarginii \advance\labelwidth-\labelsep
  \topsep\z@skip \parsep\z@skip \partopsep\z@skip \itemsep\z@skip}
\def\@listiii{\leftmargin\leftmarginiii
  \labelwidth\leftmarginiii \advance\labelwidth-\labelsep}
\def\@listiv{\leftmargin\leftmarginiv
  \labelwidth\leftmarginiv \advance\labelwidth-\labelsep}
\def\@listv{\leftmargin\leftmarginv
  \labelwidth\leftmarginv \advance\labelwidth-\labelsep}
\def\@listvi{\leftmargin\leftmarginvi
  \labelwidth\leftmarginvi \advance\labelwidth-\labelsep}
```

2.13 Fleqn option

```
\@ifclasswith{\@classname}{fleqn}{%
  \let\@tempa\@iden
  \AtBeginDocument{\mathindent\leftmargini}%
}{\let\@tempa\@gobble}%
\@ifpackageloaded{amsmath}{\let\@tempa\@gobble}{%
  \@ifpackageloaded{amstex}{\let\@tempa\@gobble}{}%
}
\@tempa{%
  \def\[{\relax
    \ifmmode\@badmath
    \else
      \begin{trivlist}%
      \@beginparpenalty\predisplaypenalty
      \@endparpenalty\postdisplaypenalty
      \item[]\leavevmode
      \hbox to\linewidth\bgroup$\displaystyle
```

Note that the `\m@th` should go at the end in `\]` just in case an embedded small math formula inside `\text` occurs in the display.

Why the extra bgroup here? I think it's not needed. [mjd,3-Feb-1994]

```
      \hskip\mathindent\bgroup
    \fi}%
  \def\]{\relax
    \ifmmode
      \egroup \m@th$\hfil \egroup
      \end{trivlist}%
```

```
    \else \@badmath
    \fi}%
  \renewenvironment{equation}{%
    \@beginparpenalty\predisplaypenalty
    \@endparpenalty\postdisplaypenalty
    \refstepcounter{equation}%
    \@topsep\abovedisplayskip \trivlist
    \item[]\leavevmode
    \hbox to\linewidth\bgroup\hskip\mathindent$\displaystyle
  }{%
    \m@th$\hfil \displaywidth\linewidth \hbox{\@eqnnum}\egroup
    \endtrivlist
  }%
  \renewenvironment{eqnarray}{%
    \stepcounter{equation}\let\@currentlabel\theequation
    \global\@eqnswtrue \global\@eqcnt\z@ \tabskip\mathindent
    \let\\=\@eqncr \abovedisplayskip\topsep
    \ifvmode \advance\abovedisplayskip\partopsep \fi
    \belowdisplayskip\abovedisplayskip
    \belowdisplayshortskip\abovedisplayskip
    \abovedisplayshortskip\abovedisplayskip
    $$\everycr{}\halign to\linewidth\bgroup
      \hskip\@centering
      $\displaystyle\tabskip\z@skip####\m@th$&%
      \@eqnsel \global\@eqcnt\@ne
      \hfil${}####{}\m@th$\hfil&%
      \global\@eqcnt\tw@
      $\displaystyle ####\m@th$\hfil\tabskip\@centering&%
      \global\@eqcnt\thr@@
      \hbox to \z@\bgroup\hss####\egroup\tabskip\z@skip\cr
    }{%
      \@@eqncr \egroup \global\advance\c@equation\m@ne$$%
      \global\@ignoretrue
    }%
  \newdimen\mathindent
  \mathindent\leftmargini
}
```

2.14 Redefined internal sectioning commands

In amsart.sty `\@startsection`, `\@sect`, and a couple of other things are redefined to fix a few hard-coded things that would interfere with the desired style. The most noteworthy difference is that all section headings will go into the table of contents (governed by secnumdepth as usual), EVEN WHEN THE * FORM IS USED. The only section heading not listed in the table of contents is the heading for the toc itself.

A second major departure from standard LaTeX is that when a short form of a section title is given, it is used only for the running heads; the table of contents still gets the full version of the title. This is correct for AMS editorial practice. However if one wants to get a line break into the table of contents it

means that the standard LaTeX method cannot be used. See the `\except` and `\for` commands.

```
\def\@startsection#1#2#3#4#5#6{%
```

Section titles, if they are run-in with the following text, are stored in a box instead of being typeset right away. They will be typeset by `\everypar`, but if one section heading follows right after another, this won't happen. So by doing `\leavevmode` we force this to happen. (`\if@noskipsec` is true if the previous section title has not yet been typeset.)

```
 \if@noskipsec \leavevmode \fi
 \par \@tempskipa #4\relax
 \@afterindenttrue
 \ifdim \@tempskipa <\z@ \@tempskipa -\@tempskipa \@afterindentfalse\fi
 \if@nobreak \everypar{}\else
     \addpenalty\@secpenalty\addvspace\@tempskipa\fi
%    Don't call \@ssect in the ifstar case:
 \@ifstar{\@dblarg{\@sect{#1}{\@m}{#3}{#4}{#5}{#6}}}%
         {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%
}
```

We add `\textup` here in order to make section numbers always roman, even if the rest of the section head is italic.

```
\def\@seccntformat#1{%
  \protect\textup{\protect\@secnumfont
    \csname the#1\endcsname
    \protect\@secnumpunct
  }%
}
```

Some journals require a different font for section numbers. (As coded here, this option permits only a change of weight, to `\bfseries`.) The `amsart` font is checked again later, when processing `\section`, and if there is no section title, it is made bold.

```
⟨amsbook | amsproc⟩\let\@secnumfont\@empty
⟨amsart⟩\def\@secnumfont{\mdseries}
```

For reference, here is the argument list for `\@sect`.

```
% #1    #2     #3      #4          #5          #6     #7      #8
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}[SHORTT]{TITLE}
```

```
\def\@sect#1#2#3#4#5#6[#7]#8{%
```

Define `\@toclevel` for for `\@tocwrite` (to `\@tochangmeasure`).

```
  \edef\@toclevel{\ifnum#2=\@m 0\else\number#2\fi}%
  \ifnum #2>\c@secnumdepth \let\@secnumber\@empty
  \else \@xp\let\@xp\@secnumber\csname the#1\endcsname\fi
```

If the value of afterskip > 0, then this is not a run-in section heading, and we want to suppress final punctuation.

```
  \@tempskipa #5\relax
```

\@svsec will be the section number plus some formatting if the star form was not used and if the depth of numbering extends to the current level. The user can change secnumdepth to control how many levels of sectioning will be numbered.

```
\ifnum #2>\c@secnumdepth
  \let\@svsec\@empty
\else
  \refstepcounter{#1}%
```

If the section heading is not run-in and there is no title, omit final punctuation and space. If it is run-in, omit extra space.

```
  \edef\@secnumpunct{%
    \ifdim\@tempskipa>\z@ % not a run-in section heading
      \@ifnotempty{#8}{.\@nx\enspace}%
    \else
      \@ifempty{#8}{.}{.\@nx\enspace}%
    \fi
  }%
```

For amsart, if a subsection has no title, make the section number bold; otherwise leave it alone. [bnb; 2004/06/08; per vwa, B-442]

```
⟨amsart⟩    \@ifempty{#8}{%
⟨amsart⟩      \ifnum #2=\tw@ \def\@secnumfont{\bfseries}\fi}{}%
```

If the *-form was not used (#2 less than 1000), we add \sectionname or whatever as a prefix, separated by a space. We need the ifundefined test in order to know whether the space should be added or not. There must be a better way to do this but I haven't thought of it yet.

```
  \protected@edef\@svsec{%
    \ifnum#2<\@m
      \@ifundefined{#1name}{}{%
        \ignorespaces\csname #1name\endcsname\space
      }%
    \fi
    \@seccntformat{#1}%
  }%
\fi
\ifdim \@tempskipa>\z@ % then this is not a run-in section heading
  \begingroup #6\relax
  \@hangfrom{\hskip #3\relax\@svsec}{\interlinepenalty\@M #8\par}%
  \endgroup
```

Section headings don't set marks for the running heads in the article style, only in the amsbook style. Assumption: \sectionmark is defined to call \@secnumber as its penultimate argument.

```
⟨amsbook⟩  \csname #1mark\endcsname{#7}%
```

If #2 (level) is greater than 1000 then we don't do a table of contents entry. This happens only for the section heading above the table of contents itself.

```
  \ifnum#2>\@m \else \@tocwrite{#1}{#8}\fi
\else
```

Otherwise we're doing a run-in heading; it is stored as `\@svsechd`, which will be typeset by `\everypar` as soon as some text is encountered.

```
\def\@svsechd{#6\hskip #3\@svsec
```

To allow for the possibility that the user wants an empty section title, leaving just the section number, we check whether `#8` is nonempty before adding the period.

```
    \@ifnotempty{#8}{\ignorespaces#8\unskip
```

The following test is to prevent a period being added if the section title ended in a question mark or other punctuation.

```
        \@addpunct.}%
    \ifnum#2>\@m \else \@tocwrite{#1}{#8}\fi
  }%
  \fi
```

In a previous version of `amsart` `\@nobreaktrue` was added to `\@xsect` for some reason. Let's keep that just in case it was done to prevent a certain kind of bug. [mjd,17-Aug-1994]

```
\global\@nobreaktrue
```

`\@xsect` does some more stuff based on whether this is a run-in heading or not.

```
\@xsect{#5}}
```

Undefine `\@ssect` to save memory; it's not needed in `amsart`.

```
\let\@ssect\relax
```

2.15 Chapters and sections

Allocate counters for sectioning commands. Paragraph and subparagraph counters are allocated but normally not used.

```
\newcounter{part}
⟨amsbook⟩\newcounter{chapter}
⟨amsproc | amsart⟩\newcounter{section}
⟨amsbook⟩\newcounter{section}[chapter]
⟨amsbook⟩\def\thesection{\arabic{section}}
\newcounter{subsection}[section]
\newcounter{subsubsection}[subsection]
\newcounter{paragraph}[subsubsection]
\newcounter{subparagraph}[paragraph]
```

Set numbering style for sectioning commands. In a couple of cases resetting is unnecessary but we include the full list here for completeness.

```
\renewcommand\thepart           {\arabic{part}}
⟨amsbook⟩\renewcommand\thechapter       {\arabic{chapter}}
\renewcommand\thesection        {\arabic{section}}
\renewcommand\thesubsection     {\thesection.\arabic{subsection}}
\renewcommand\thesubsubsection {\thesubsection .\arabic{subsubsection}}
\renewcommand\theparagraph      {\thesubsubsection.\arabic{paragraph}}
\renewcommand\thesubparagraph   {\theparagraph.\arabic{subparagraph}}
```

Depth of section numbering; if `secnumdepth` were 2 instead of 3, `\subsubsection` would not be numbered.

```
\setcounter{secnumdepth}{3}
```

The arguments of `\@startsection` are given for reference:

```
% #1    #2     #3      #4          #5          #6
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}
```

```
\def\partname{Part}
⟨∗amsart | amsproc⟩
\def\part{\@startsection{part}{0}%
  \z@{\linespacing\@plus\linespacing}{.5\linespacing}%
  {\normalfont\bfseries\raggedright}}
⟨/amsart | amsproc⟩
```

Specialsection correlates to our inhouse Z-head.

```
\def\specialsection{\@startsection{section}{1}%
⟨amsart⟩  \z@{\linespacing\@plus\linespacing}{.5\linespacing}%
⟨amsproc | amsbook⟩  \z@{2\linespacing\@plus\linespacing}{.5\linespacing}%
⟨amsart⟩  {\normalfont\centering}}
⟨amsproc | amsbook⟩  {\large\scshape\centering}}
```

In the book class `\part` puts the part title on a separate page.

```
⟨∗amsbook⟩
\def\part{\cleardoublepage \thispagestyle{empty}%
  \null\vfil  \markboth{}{}\secdef\@part\@spart}
%
\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >-2\relax \refstepcounter{part}%
    \addcontentsline{toc}{part}{\partname\ \thepart.
        \protect\enspace\protect\noindent#1}%
  \else
    \addcontentsline{toc}{part}{#1}\fi
  \begingroup\centering
  \ifnum \c@secnumdepth >-2\relax
       {\fontsize{\@xviipt}{22}\bfseries
         \partname\ \thepart} \vskip 20\p@ \fi
  \fontsize{\@xxpt}{25}\bfseries
      #1\vfil\vfil\endgroup \newpage\thispagestyle{empty}}

\def\@spart#1{\addcontentsline{toc}{part}{\protect\noindent#1}%
  \begingroup\centering
  \fontsize{\@xxpt}{25}\bfseries
     #1\vfil\vfil\endgroup \newpage\thispagestyle{empty}}
⟨/amsbook⟩
```

The arguments of `\partrunhead` are `\partname`, `\thepart`, and the text of the part title. The first two were fully expanded during the marking process. Use of a mere interword space between the first two args makes it possible to apply `\ignorespaces` and `\unskip` as shown here to produce the desired results if one or the other is empty.

```
⟨*amsbook⟩
\def\partrunhead#1#2#3{%
  \@ifnotempty{#2}{\uppercase{\ignorespaces#1 #2\unskip}\@ifnotempty{#3}{. }}%
  \def\@tempa{#3}%
  \ifx\@empty\@tempa\else
    \begingroup \def\\{ \ignorespaces}% defend against questionable usage
    \uppercasenonmath\@tempa\@tempa
    \endgroup
  \fi
}
\let\chapterrunhead\partrunhead
\let\sectionrunhead\partrunhead
⟨/amsbook⟩
```

Section headings in the amsbook style differ from the amsart style in a couple of ways: The ones that aren't centered are indented on the left, instead of flush left; and the first level, `\section`, is not small caps but bold. Cf. the definition of `\appendix`.

```
\def\section{\@startsection{section}{1}%
  \z@{.7\linespacing\@plus\linespacing}{.5\linespacing}%
⟨amsart⟩  {\normalfont\scshape\centering}}
⟨amsbook | amsproc⟩  {\normalfont\bfseries\centering}}
```

Negative value for `#5` is a signal to make a run-in heading instead of doing a vskip after the heading.

```
\def\subsection{\@startsection{subsection}{2}%
⟨amsart⟩  \z@{.5\linespacing\@plus.7\linespacing}{-.5em}%
⟨amsbook | amsproc⟩  \normalparindent{.5\linespacing\@plus.7\linespacing}{-.5em}%
  {\normalfont\bfseries}}
\def\subsubsection{\@startsection{subsubsection}{3}%
⟨amsart⟩  \z@{.5\linespacing\@plus.7\linespacing}{-.5em}%
⟨amsbook | amsproc⟩  \normalparindent\z@{-.5em}%
  {\normalfont\itshape}}
```

Fontdimen 2 of the current font is the ideal interword space of the font. Thus the following spec says that the space after the paragraph heading should be a normal interword space (but nonstretching and nonshrinking).

```
\def\paragraph{\@startsection{paragraph}{4}%
⟨amsart⟩  \z@\z@{-\fontdimen2\font}%
⟨amsbook | amsproc⟩  \normalparindent\z@{-\fontdimen2\font}%
  \normalfont}
\def\subparagraph{\@startsection{subparagraph}{5}%
  \z@\z@{-\fontdimen2\font}%
  \normalfont}
⟨*amsart | amsproc⟩
\def\appendix{\par\c@section\z@ \c@subsection\z@
   \let\sectionname\appendixname
   \def\thesection{\@Alph\c@section}}
\def\appendixname{Appendix}
⟨/amsart | amsproc⟩
```

A slower, but fully expandable definition of \@Roman to avoid the nonexpandable \uppercase which is undesirable in certain circumstances.

```
\def\@Roman#1{\@xp\@slowromancap
  \romannumeral#1@}%
%
\def\@slowromancap#1{\ifx @#1% then terminate
  \else
```

Note: \if is required here, not \ifx, because \romannumeral returns category 12 letters!

```
    \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
    c#1C\else\if m#1M\else#1\fi\fi\fi\fi\fi\fi
    \@xp\@slowromancap
  \fi
}
```

2.16 Book features

Books (monographs) comprise three distinct sections, \frontmatter, \mainmatter, and \backmatter. The \frontmatter would consist of the title page, copyright page, table of contents, preface, etc. The \mainmatter would be the body of the book. The \backmatter would include the appendix, bibliography, glossary, and index.

```
⟨*amsbook⟩
\def\frontmatter{\cleardoublepage\pagenumbering{roman}}
\def\mainmatter{\cleardoublepage\pagenumbering{arabic}}
\def\backmatter{%
  \if@openright\cleardoublepage\else\clearpage\fi
  \let\chaptername\relax}
⟨/amsbook⟩
```

Book proceedings and monographs allow a signature to print at the end of a preface.

```
⟨*!amsart⟩
\def\aufm#1{\par\vspace*{12pt}{\flushright #1\par}}
⟨/!amsart⟩
```

Monographs can have a special exercise environment.

```
⟨*amsbook⟩
\newenvironment{xcb}{%
  \setcounter{enumi}{0}%
  \settowidth{\leftmargini}{\labelenumi\hskip\labelsep}%
  \setcounter{enumii}{4}% letter d
  \settowidth{\leftmarginii}{\labelenumii\hskip\labelsep}%
  \@startsection{section}% counter name; ignored because of the
                                % * below
  {1}% sectioning level
  {\z@}% indent to the left of the section title
  {18\p@\@plus2\p@}% vertical space above
  {1sp}% Space below of 13pt base-to-base, so none needs to be added
      % here; but \z@ would cause the following text to be run-in, so we
```

```
      % use 1sp instead.
  {\bfseries}% The font of the subsection title
  *% always unnumbered
}{%
  \par
}
```
⟨/amsbook⟩

2.17 Book chapters

The `\chapter` command is provided only in the `amsbook` class, not in `amsart` or `amsproc`.

```
⟨*amsbook⟩
\def\chapter{%
  \if@openright\cleardoublepage\else\clearpage\fi
  \thispagestyle{plain}\global\@topnum\z@
  \@afterindenttrue \secdef\@chapter\@schapter}
```

`\@chapter` for numbered chapters.

```
\def\@chapter[#1]#2{\refstepcounter{chapter}%
  \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
  \else \let\@secnumber\thechapter \fi
  \typeout{\chaptername\space\@secnumber}%
  \def\@toclevel{0}%
  \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{#2}%
  \else \@tocwriteb\tocchapter{chapter}{#2}\fi
  \chaptermark{#1}%
  \addtocontents{lof}{\protect\addvspace{10\p@}}%
  \addtocontents{lot}{\protect\addvspace{10\p@}}%
  \@makechapterhead{#2}\@afterheading}
```

`\@schapter` for unnumbered chapters.

```
\def\@schapter#1{\typeout{#1}%
  \let\@secnumber\@empty
  \def\@toclevel{0}%
  \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{#1}%
  \else \@tocwriteb\tocchapter{chapter}{#1}\fi
  \chaptermark{#1}%
  \addtocontents{lof}{\protect\addvspace{10\p@}}%
  \addtocontents{lot}{\protect\addvspace{10\p@}}%
  \@makeschapterhead{#1}\@afterheading}

\newcommand\chaptername{Chapter}
\newcommand\appendixname{Appendix}

\def\@makechapterhead#1{\global\topskip 7.5pc\relax
  \begingroup
  \fontsize{\@xivpt}{18}\bfseries\centering
```

In order to keep the chapter number "CHAPTER III" from getting in the way of the `\topskip` we put it inside the paragraph containing the main title. Then we have to do some laborious `\rlap`ing and `\hskip`ing to position it correctly.

```
    \ifnum\c@secnumdepth>\m@ne
      \leavevmode \hskip-\leftskip
      \rlap{\vbox to\z@{\vss
          \centerline{\normalsize\mdseries
              \uppercase\@xp{\chaptername}\enspace\thechapter}
          \vskip 3pc}}\hskip\leftskip\fi
     #1\par \endgroup
  \skip@34\p@ \advance\skip@-\normalbaselineskip
  \vskip\skip@ }
\def\@makeschapterhead#1{\global\topskip 7.5pc\relax
  \begingroup
  \fontsize{\@xivpt}{18}\bfseries\centering
  #1\par \endgroup
  \skip@34\p@ \advance\skip@-\normalbaselineskip
  \vskip\skip@ }
⟨/amsbook⟩
```

The `\appendix` command, following the LaTeX book, marks the start of a division after `\mainmatter` and before `\backmatter` that consists of appendixes.

```
⟨*amsbook⟩
\def\appendix{\par
  \c@chapter\z@ \c@section\z@
  \let\chaptername\appendixname
  \def\thechapter{\@Alph\c@chapter}}
⟨/amsbook⟩
```

2.18 Table of contents macros

`\tableofcontents` is like `\chapter` or `\section` except for no number and no table of contents entry.

`\@pnumwidth` is the maximum width for page numbers in a table of contents. `1.6em` allows enough room for three digits.

```
\newcommand{\@pnumwidth}{1.6em}
```

`\@tocrmarg` is `\@pnumwidth` plus the desired minimum space (`1em`) between page numbers and the preceding text.

```
\newcommand{\@tocrmarg}{2.6em}
⟨amsart⟩\setcounter{tocdepth}{2}
⟨amsbook | amsproc⟩\setcounter{tocdepth}{1}

\newswitch{toc}
\newswitch{lof}
\newswitch{lot}
```

Since table of contents, list of figures and list of figures are identical in design as far as the chapter heading and other preliminaries go, we redefine `\@starttoc` to do the necessary work, rather than defining a new macro (which would use up another control sequence name).

We define first the article form of `\@starttoc`, then the book form.

Owing to confusion about what font to use for `\contentsname`, a symbolic name has been assigned to provide flexibility.

```
⟨*amsart | amsproc⟩
\newcommand\contentsnamefont{\scshape}
\def\@starttoc#1#2{\begingroup
  \setTrue{#1}%
```

Remove the skip after the abstract so that we can substitute another.

```
  \par\removelastskip\vskip\z@skip
```

The first two arguments of `\@startsection` here are special values that cause different internal branches to be taken.

```
  Arguments: {} = name = empty
  \@M = no number should be used and no table of contents entry
  \z@ = indent amount
  12pt + 12pt = vskip before
  6pt = vskip after
  \centering\contentsnamefont = format
  \@startsection{}\@M\z@{\linespacing\@plus\linespacing}%
    {.5\linespacing}{\centering\contentsnamefont}{#2}%
```

If we have a list of figures or list of tables we want to put them in the main table of contents, but we don't want to put an entry there for the main table of contents itself. So we check to see if argument 2 is `\contentsname` and if it is then we refrain from doing `\addcontentsline`.

```
  \ifx\contentsname#2%
  \else \addcontentsline{toc}{section}{#2}\fi
  \makeatletter
  \@input{\jobname.#1}%
  \if@filesw
    \@xp\newwrite\csname tf@#1\endcsname
    \immediate\@xp\openout\csname tf@#1\endcsname \jobname.#1\relax
  \fi
  \global\@nobreakfalse \endgroup
  \addvspace{32\p@\@plus14\p@}%
  \let\tableofcontents\relax
}
⟨/amsart | amsproc⟩
```

And here is the book form of `\@starttoc`.

```
⟨*amsbook⟩
\def\@starttoc#1#2{%
  \begingroup
  \setTrue{#1}%
```

Inside this group we change `\secdef` so that we can call `\chapter` and only get the preliminary part of its definition that we need.

```
  \let\secdef\@gobbletwo \chapter
```

If we have a list of figures or list of tables we want to put them in the main table of contents, but we don't want to put an entry there for the main table of contents itself. So we check to see if argument 2 is `\contentsname` and if it is then we refrain from doing `\addcontentsline`.

```
  \let\@secnumber\@empty % for \@tocwrite and \chaptermark
  \ifx\contentsname#2%
  \else \@tocwrite{chapter}{#2}\fi
```

Now we do the equivalent of `\@schapter`. Expand `#2` so that it will be easy to apply uppercasing to it. (For `\@starttoc` we assume that `#2` is always a control such as `\contentsname`.)

```
  \typeout{#2}\@xp\chaptermark\@xp{#2}%
  \@makeschapterhead{#2}\@afterheading
```

Protect against document classes that have nonzero `\parskip`.

```
  \parskip\z@skip
```

And finally we read in the `.toc` (or whatever) file.

```
  \makeatletter
  \@input{\jobname.#1}%
  \if@filesw
    \@xp\newwrite\csname tf@#1\endcsname
    \immediate\@xp\openout\csname tf@#1\endcsname \jobname.#1\relax
  \fi
  \global\@nobreakfalse \endgroup
  \newpage
}
```

⟨/amsbook⟩

Now it is easy to define `\tableofcontents` and its relatives.

```
\def\contentsname{Contents}
\def\listfigurename{List of Figures}
\def\listtablename{List of Tables}
\def\tableofcontents{%
  \@starttoc{toc}\contentsname
}
\def\listoffigures{\@starttoc{lof}\listfigurename}
\def\listoftables{\@starttoc{lot}\listtablename}
```

In order to automatically leave enough space for the 'number' part of toc entries, we compute the maximum width of the 'number' part for each sectioning level and pass that information to `\@tocline` through the `.aux` file.

Init the tocindents if they are not yet set (first run).

```
\AtBeginDocument{%
  \@for\@tempa:=-1,0,1,2,3\do{%
    \@ifundefined{r@tocindent\@tempa}{%
      \@xp\gdef\csname r@tocindent\@tempa\endcsname{0pt}}{}%
  }%
}
```

`\@writetocindents` This function writes out the max toc indents to the aux file.

```
\def\@writetocindents{%
  \begingroup
  \@for\@tempa:=-1,0,1,2,3\do{%
    \immediate\write\@auxout{%
```

```
      \string\newlabel{tocindent\@tempa}{%
        \csname r@tocindent\@tempa\endcsname}}%
  }%
  \endgroup}
%
\AtEndDocument{\@writetocindents}
```

dentlabel This function is a no-op except in `\@tocwrite` where it is a pointer to `\@tochangmeasure`.

```
\let\indentlabel\@empty
```

ngmeasure This function measures the hangindent part of a toc entry and updates the current max for the given sectioning level, if necessary. The max's at the end of the document will be written in the form of a pseudo-label to the `.aux` file by `\@writetocindents`.

We can assume that `\@tochangmeasure` is already inside a group when called.

```
\def\@tochangmeasure#1{\sbox\z@{#1}%
  \ifdim\wd\z@>\csname r@tocindent\@toclevel\endcsname\relax
    \@xp\xdef\csname r@tocindent\@toclevel\endcsname{\the\wd\z@}%
  \fi
}
```

\@toclevel Initialize, for the record.

```
\def\@toclevel{0}
```

Since we don't have leader dots, we have `\@tocline` instead of `\@dottedtocline`.

```
\def\@tocline#1#2#3#4#5#6#7{\relax
  \ifnum #1>\c@tocdepth % then omit
  \else
    \par \addpenalty\@secpenalty\addvspace{#2}%
    \begingroup \hyphenpenalty\@M
    \@ifempty{#4}{%
      \@tempdima\csname r@tocindent\number#1\endcsname\relax
    }{%
      \@tempdima#4\relax
    }%
    \parindent\z@ \leftskip#3\relax \advance\leftskip\@tempdima\relax
    \rightskip\@pnumwidth plus4em \parfillskip-\@pnumwidth
    #5\leavevmode\hskip-\@tempdima #6\nobreak\relax
    \hfil\hbox to\@pnumwidth{\@tocpagenum{#7}}\par
    \nobreak
    \endgroup
  \fi}
\def\@tocpagenum#1{\hss{\mdseries #1}}
```

The function `\@tocwrite` writes the information of a section heading to the `.toc` file in a standard form. It allows different functions to be substituted for `\numberline`, to get greater control of toc formatting.

```
\def\@tocwrite#1{\@xp\@tocwriteb\csname toc#1\endcsname{#1}}
```

The \chapter command uses \@tocwriteb directly because of the need to write slightly different things to the toc file depending on the current value of \chaptername.

```
\def\@tocwriteb#1#2#3{%
  \begingroup
    \def\@tocline##1##2##3##4##5##6{%
      \ifnum##1>\c@tocdepth
      \else \sbox\z@{##5\let\indentlabel\@tochangmeasure##6}\fi}%
    \csname l@#2\endcsname{#1{\csname#2name\endcsname}{\@secnumber}{}}%
  \endgroup
  \addcontentsline{toc}{#2}%
    {\protect#1{\csname#2name\endcsname}{\@secnumber}{#3}}}
```

Specs for monograph toc are as follows (tocdepth is 1, i.e., subsections and lower are not listed in toc).

```
Part: Space above 12pt plus2pt, indent 0pt, "Part" + wordspace +
number + "." + 1em + title (raggedright, no hangindent) + 1em +
page number in column 1.6em wide.

Chapter: Space above 8pt, hangindent on ("Chapter 0" + "." + 1em),
+ title (raggedright) + 1em + page number in column 1.6em wide.

Appendix: Same as Chapter except for epithet "Appendix M"

Section: Space above 0pt, hangindent on (1pc + "0.0" + "." + 1em),
+ title (raggedright) + 1em + page number in column 1.6em wide.
```

Specs for article toc are as follows (tocdepth 2):

```
Section: Same as for monograph.

Subsection: Space above 0pt, hangindent 6pc (number + "." + 1em),
+ title (raggedright) + 1em + page number in column 1.6em wide.

Subsubsection: Same as subsection but hangindent 8pc
```

Typical invocation of \l@chapter:

```
\contentsline{chapter}{%
  \tocchapter{Chapter}{3}{Some title stuff}}{103}
-->\l@chapter-->
          #1 #2   #3  #4 #5
\@tocline{0}{8pt}{0pt}{}{\bfseries}
   #6           6a      6b 6c                 #7
  {\tocchapter{Chapter}{3}{Some title stuff}}{103}
```

The \tocchapter is a slightly more useful form than \numberline that allows control for optionally omitting strings like 'Chapter' or changing fonts for subcomponents of the toc entry. Note that it is allowed to have appendix and chapter at the same toc level, with \tocappendix instead of \tocchapter written in the .toc file.

The arguments of `\@tocline` are as follows:

```
\@tocline{LEVEL}{VSPACE}{INDENT}{NUMBERWIDTH}{EXTRA}%
  {TEXT}{PAGENUM}
```

where 'numberwidth' is the width of the box allotted to contain the section number, including any preceding word like 'Chapter' or 'Part'. If this width arg is empty then an automatically computed width (max over TOC of the numberwidths for this level) is used. The 'extra' argument is formatting such as font changes. The 'text' argument contains a section-command specific function like `\tocsection` or `\tocchapter` which takes in turn three arguments: epithet, number, topic.

```
\def\l@section{\@tocline{1}{0pt}{1pc}{}{}}
```

*** The indents do not agree between in-house and distributed ***
** versions; no changes; get specs before revising. ***

The use of `\ignorespaces` in `\tocsection` and its relatives means that if `#1` is empty, the following space will be also removed.

```
\newcommand{\tocsection}[3]{%
  \indentlabel{\@ifnotempty{#2}{\ignorespaces#1 #2.\quad}}#3}
\def\l@subsection{\@tocline{2}{0pt}{1pc}{5pc}{}}
\let\tocsubsection\tocsection
\def\l@subsubsection{\@tocline{3}{0pt}{1pc}{7pc}{}}
\let\tocsubsubsection\tocsection
\let\l@paragraph\l@subsubsection
\let\tocparagraph\tocsection
\let\l@subparagraph\l@subsubsection
\let\tocsubparagraph\tocsection
%
\def\l@part{\@tocline{-1}{12pt plus2pt}{0pt}{}{\bfseries}}
\let\tocpart\tocsection
\def\l@chapter{\@tocline{0}{8pt plus1pt}{0pt}{}{}}
\let\tocchapter\tocsection
```

In this case we are pretty sure the word "Appendix" or similar is present, so only check if arg 2 is empty:

```
\newcommand{\tocappendix}[3]{%
  \indentlabel{#1\@ifnotempty{#2}{ #2}.\quad}#3}
```

```
\def\l@figure{\@tocline{0}{3pt plus2pt}{0pt}{1.5pc}{}}
\let\l@table=\l@figure
```

2.19 Bibliography section or chapter

Following the `babel` package, we use `\refname` in articles and `\bibname` in books.

```
\def\refname{References}
\def\bibname{Bibliography}
```

Restudy the following code; `\bibsetup` isn't used anywhere although it's defined for the three different bibstyles. Because some publications have different default label styles, separate that out for easy tailoring of packages.

```
\def\@defaultbiblabelstyle#1{#1.}
\def\bibliographystyle#1{%
    \if@filesw\immediate\write\@auxout{\string\bibstyle{#1}}\fi
    \def\@tempa{#1}%
    \def\@tempb{amsplain}%
    \def\@tempc{}%
    \ifx\@tempa\@tempb
        \def\@biblabel##1{\@defaultbiblabelstyle{##1}}%
        \def\bibsetup{}%
    \else
        \def\bibsetup{\labelsep6\p@}%
        \ifx\@tempa\@tempc
            \def\@biblabel##1{}%
            \def\bibsetup{\labelwidth\z@ \leftmargin24\p@
                \itemindent-\leftmargin
                \labelsep\z@ }%
        \fi
    \fi}
```

Permit easy change of font size for unusual purpose, e.g., for an author's "life list" in collected works. [bnb, 2004/04/01]

```
\newcommand{\bibliofont}{\footnotesize}
```

`thebibliography` differs in some author packages only in the shape of the title; make this easy to change. [bnb, 2004/05/22]

```
\newcommand{\@bibtitlestyle}{%
⟨amsart | amsproc⟩  \@xp\section\@xp*\@xp{\refname}%
⟨amsbook⟩  \@xp\chapter\@xp*\@xp{\bibname}%
}
\newenvironment{thebibliography}[1]{%
  \@bibtitlestyle
  \normalfont\bibliofont\labelsep .5em\relax
  \renewcommand\theenumiv{\arabic{enumiv}}\let\p@enumiv\@empty
  \list{\@biblabel{\theenumiv}}{\settowidth\labelwidth{\@biblabel{#1}}%
    \leftmargin\labelwidth \advance\leftmargin\labelsep
    \usecounter{enumiv}}%
  \sloppy \clubpenalty\@M \widowpenalty\clubpenalty
  \sfcode`\.=\@m
}{%
```

Change error for empty list (no items) to warning, to allow authors to leave their bibliography temporarily empty during writing:

```
  \def\@noitemerr{\@latex@warning{Empty `thebibliography' environment}}%
  \endlist
}
```

The `\bysame` command prints a horizontal dash indicating repetition of the author's name in consecutive bibliography entries.

```
\def\bysame{\leavevmode\hbox to3em{\hrulefill}\thinspace}
```

We define `\newblock` even though it's not needed for AMS publication style, just to avoid error messages when a non-AMS `.bst` file is used. This is a convenience

for users; use of \newblock is not recommended for submissions to the AMS.

```
\def\newblock{}
```

\MR \MRhref Provide an MR number for a bibliography item. At the moment [mjd,1995/08/07] this only prints the MR number, but later we expect to extend it to write an HTML \special to the .dvi file.

The presentation of the MR number has been simplified (from using a bold volume number) coincident with the change in the MathSciNet database to a 7-digit reference number from the volume:number form.

Ensure that an old-style MR number does not break across lines if it contains a space; editorial request. [bnb; 2004/04/01] Countermanded, to permit break between reference number and an old-style number following in parentheses. [bnb; 2004/06/11]

```
\newcommand\MR[1]{\relax\ifhmode\unskip\spacefactor3000 \space\fi
  MR~\MRhref{#1}{#1}}
\let\MRhref\@gobble
```

\URL \URLhref \@URL Allows sticking in an arbitrary URL in a bibliography. Leading "http" is not assumed. Call \verb to ensure that special characters in the URL don't cause trouble.

```
\newcommand\URL{\begingroup
  \def\@sverb##1{%
    \def\@tempa####1##1{\@URL{####1}\egroup\endgroup}%
    \@tempa}%
  \verb}
\let\URLhref\@gobble
\def\@URL#1{\URLhref{#1}#1}
```

2.20 Index section or chapter

Chapter or section heading for an index. Index is set up to be two columns.

```
\newif\if@restonecol
```

⟨∗amsbook⟩

```
\def\indexchap#1{\global\topskip 7.5pc\relax
  \twocolumn[{\fontsize{\@xivpt}{18}\bfseries\centering
    \vskip\topskip\hbox{}\vskip-\baselineskip% adjust top space
    #1\par
```

After \twocolumn finishes operating, the top material is left in an insert register, and topskip will now be applied above the following material. So we should set it to the normal after-chapter-title space (34pt)—cf. \@makeschapterhead.

```
    \global\topskip 34\p@\relax
    \ifx\@empty\indexintro
    \else
      \begingroup \normalsize
        \skip@\topskip \advance\skip@ -\baselineskip
        \vskip\skip@
```

```
      \parbox[t]{24pc}{\normalfont\indexintro\par}%
    \endgroup
    \global\topskip 24\p@\relax
  \fi
 }]%
}
\newcommand{\indexintro}{}
⟨/amsbook⟩
```

`theindex` differs in some author packages only in the shape of the title; make this easy to change. [bnb, 2004/05/22]

```
\newcommand{\@indextitlestyle}{%
⟨*amsbook⟩
  \let\@makeschapterhead\indexchap
  \@xp\chapter\@xp*\@xp{\indexname}%
⟨/amsbook⟩
⟨amsart | amsproc⟩  \twocolumn[\@xp\section\@xp*\@xp{\indexname}]%
}
\def\theindex{\@restonecoltrue\if@twocolumn\@restonecolfalse\fi
  \columnseprule\z@ \columnsep 35\p@
  \@indextitlestyle
  \thispagestyle{plain}%
  \let\item\@idxitem
  \parindent\z@  \parskip\z@\@plus.3\p@\relax
  \raggedright
  \hyphenpenalty\@M
  \footnotesize}
\def\indexname{Index}
\def\@idxitem{\par\hangindent 2em}
\def\subitem{\par\hangindent 2em\hspace*{1em}}
\def\subsubitem{\par\hangindent 3em\hspace*{2em}}
\def\endtheindex{\if@restonecol\onecolumn\else\clearpage\fi}
\def\indexspace{\par\bigskip}
```

2.21 Footnotes

In books the footnote counter should reset to 0 at the beginning of each chapter:

```
⟨amsbook⟩\@addtoreset{footnote}{chapter}
```

Rule above footnotes is 5 picas wide.

```
\def\footnoterule{\kern-.4\p@
        \hrule\@width 5pc\kern11\p@\kern-\footnotesep}
```

A simple superscript doesn't work here; it fails on a minipage, where `\itshape` (which is invalid in math mode) is used for the footnote numbers. Cf. the definition of `\textprime`.

```
\def\@makefnmark{%
  \leavevmode
  \raise.9ex\hbox{\fontsize\sf@size\z@\normalfont\@thefnmark}%
}
%
```

```
\def\@makefntext{\indent\@makefnmark}
```

Add `\normalfont` before `\footnotesize` so that fonts will come out properly using the new font selection scheme.

```
\long\def\@footnotetext#1{%
  \insert\footins{%
    \normalfont\footnotesize
    \interlinepenalty\interfootnotelinepenalty
    \splittopskip\footnotesep \splitmaxdepth \dp\strutbox
    \floatingpenalty\@MM \hsize\columnwidth
```

Mostly `\@parboxrestore` does what we want; but not with respect to `\parindent` and `\tolerance`.

```
    \@parboxrestore \parindent\normalparindent \sloppy
    \protected@edef\@currentlabel{%
      \csname p@footnote\endcsname\@thefnmark}%
    \@makefntext{%
      \rule\z@\footnotesep\ignorespaces#1\unskip\strut\par}}}
```

We change `\sloppy` to keep it from overriding our normal value of 1pt for `\hfuzz` and `\vfuzz` with a LESS sloppy value (.5pt).

```
\hfuzz=1pt \vfuzz=\hfuzz
\def\sloppy{\tolerance9999 \emergencystretch 3em\relax}
```

2.22 Float placement parameters

These control the placing of floating objects like tables and figures. We use much more tolerant values than the LaTeX defaults; the LaTeX defaults are geared to fussier page breaks, at a price of requiring more manual intervention to deal with difficult page breaking problems.

When using LaTeX's twocolumn option, 'page' really means 'column', for the parameters that don't have a dbl prefix: that is, topnumber is then the maximum number of top figures allowed in each column, and so forth.

topnumber maximum number of top figures allowed per page

bottomnumber maximum number of bottom figures allowed per page

totalnumber maximum number of figures (top and bottom) allowed per page

dbltopnumber same as topnumber, but for two-column wide figures, when double-column format is used

```
\setcounter{topnumber}{4}
\setcounter{bottomnumber}{4}
\setcounter{totalnumber}{4}
\setcounter{dbltopnumber}{4}
```

Float fraction parameters.

`\topfraction` maximum part of the page allowed for top figures, expressed as a decimal fraction. The value of .97 means roughly 'accept pages that have only two lines of text, and the rest figures'.

`\bottomfraction` same as `\topfraction`, but for bottom figures

`\textfraction` *minimum* part of the page that must be occupied by text, if the page is to have any text at all. If this value cannot be achieved, LaTeX will turn the current figure or figures into a "float page", i.e., a page of figures without any text.

`\floatpagefraction` minimum amount (that is, total combined height) of figures needed before LaTeX will make a float page. This is expressed as a fraction of the normal page height.

`\dbltopfraction` like `\topfraction`, but applies only to figures that are two columns wide, when double-column format is used.

`\dblfloatpagefraction` minimum amount of double-column figure material needed before LaTeX will make a two-column wide "float page"

```
\renewcommand{\topfraction}{.97}
\renewcommand{\bottomfraction}{.97}
\renewcommand{\textfraction}{.03}
\renewcommand{\floatpagefraction}{.9}
\renewcommand{\dbltopfraction}{.97}
\renewcommand{\dblfloatpagefraction}{.9}
```

We also modify the default values for spacing around floating figures: (A) so that figures on a float page will not be vertically centered on the total page height but will be flush at the top of the page, and (B) so that there will be slightly more stretchability around figures, to help find better page breaks in difficult situations.

`\floatsep` Space between consecutive figures

`\textfloatsep` Space between text and top or bottom figures

`\intextsep` Space above and below a figure in the middle of the text (i.e., placed with the `[h]` option)

`\dblfloatsep` Space between consecutive figures that are two columns wide (when two-column format is used)

`\dbltextfloatsep` Space between double-column figures and text

`\@fptop` Space above the first figure on a float page

`\@fpsep` Space between figures on a float page

`\@fpbot` Space below the last figure on a float page

`\@dblfptop` Space above the first double-column figure on a two-column wide float page

`\@dblfptop` Space between double-column figures on a two-column wide float page

`\@dblfptop` Space below the last double-column figure on a two-column wide float page

```
\setlength{\floatsep}{12pt plus 6pt minus 4pt}
\setlength{\textfloatsep}{15pt plus 8pt minus 5pt}
\setlength{\intextsep}{12pt plus 6pt minus 4pt}
\setlength{\dblfloatsep}{12pt plus 6pt minus 4pt}
\setlength{\dbltextfloatsep}{15pt plus 8pt minus 5pt}
```

```
\setlength{\@fptop}{0pt}% removed "plus 1fil"
\setlength{\@fpsep}{8pt}% removed "plus 2fil"
\setlength{\@fpbot}{0pt plus 1fil}
\setlength{\@dblfptop}{0pt}% removed "plus 1fil"
\setlength{\@dblfpsep}{8pt}% removed "plus 2fil"
\setlength{\@dblfpbot}{0pt plus 1fil}
```

`\fps@figure`, `\fps@table`: placement specifications for `figure` and `table` environments. '`tbp`' means that a figure will be placed at the top or bottom of a page, or on a separate page with no text. This might be changed to '`tp`', for example, if you never want figures to appear at the bottom of a page.

```
\newcommand{\fps@figure}{tbp}
\newcommand{\fps@table}{tbp}
```

Some more setup for figures.

```
⟨amsart | amsproc⟩\newcounter{figure}
⟨amsbook⟩\newcounter{figure}[chapter]
\def\@captionheadfont{\scshape}
\def\@captionfont{\normalfont}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\ \thefigure}
\def\figurename{Figure}
\newenvironment{figure}{%
  \@float{figure}%
}{%
  \end@float
}
\newenvironment{figure*}{%
  \@dblfloat{figure}%
}{%
  \end@dblfloat
}
```

And similar for tables.

```
⟨amsart | amsproc⟩\newcounter{table}
⟨amsbook⟩\newcounter{table}[chapter]
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{\tablename\ \thetable}
\def\tablename{Table}
\newenvironment{table}{%
  \@float{table}%
}{%
  \end@float
}
\newenvironment{table*}{%
  \@dblfloat{table}%
}{%
  \end@dblfloat
}
```

Change \@floatboxreset to add \centering. Centering is always applied to tables and figures in AMS publications. It should not be necessary to throw in \begin{center} ... \end{center} in every instance to achieve this.

```
\def\@floatboxreset{\global\@minipagefalse \centering}
```

This is what we want \@makecaption to do: If the total width is greater than normal columnwidth we want to break the caption into lines using a line width of W = (columnwidth − 6pc), and center the resulting block between the margins. Otherwise we want to set the caption as a single line, centered between the margins.

To do this we set the caption as a vbox with line width W, except that we allow the last line (which may be the only line) to have width up to full columnwidth by adding a kern of -6pc. If the result is a single hbox (i.e., a single line) we need to unpack the hbox, remove rightskip, parfillskip, and the -6pc kern, and center the remaining material. If the caption is more than one line, then box 1 contains the last line, which we need to unpack in the same way, and run through the paragraphing process again (because this last line may be up to 6 picas wider than the desired width).

In practice this procedure tends to fail if there are any potential breakpoints near the end of the first line (in the window between short-width and full-width). Then TEX tends to choose a break (depending on spaceskip, tolerance, etc) at the last acceptable breakpoint before short-width is exceeded, *without considering any later material*—in particular, the negative kern. [This was pointed out by Donald Arseneau, May 2000.] Unfortunately, setting parfillskip to a negative value does not work either. I guess this is a special case of the existing limitation on parshape: you cannot specify a parshape in terms of number of lines from the bottom of the paragraph. (We would like to specify a parshape where the last line is 6 picas longer than the others.)

Finally, if the caption is for a figure, it will be set below the figure, so the separating space goes above the caption; otherwise the separating space goes below the caption.

```
\long\def\@makecaption#1#2{%
```

Measure the contents of the caption. If #2 is not empty, then we must add a period and an en-space before typesetting it. The \@caption macro adds an extra \ignorespaces at the beginning of #2, so in order to find out if the user-typed portion is empty we use \@cdr to pull off the \ignorespaces.

```
%    Use color-safe commands
  \setbox\@tempboxa\vbox{\color@setgroup
    \advance\hsize-2\captionindent\noindent
    \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
        {\@cdr#2\@nil}{.\@captionfont\upshape\enspace#2}%
    \unskip\kern-2\captionindent\par
    \global\setbox\@ne\lastbox\color@endgroup}%
  \ifhbox\@ne % the normal case
    \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
  \fi
```

If `\@tempboxa` is not empty at this point then the caption was more than one line long or there was extra vertical mode material, maybe a `\write` (from `\index` or something). Interestingly, we can't use `\ifvoid` to see if `\@tempboxa` is empty, because empty is not the same thing as void (as far as the `\ifvoid` test is concerned). So instead we measure the width of `\@tempboxa` to see if it's zero; this should suffice for non-bizarre cases.

```
\ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
  \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
\else % tempboxa contained more than one line
  \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
      \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
\fi
\ifnum\@tempcnta<64 % if the float IS a figure...
  \addvspace\abovecaptionskip
  \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
\else % if the float IS NOT a figure...
  \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
  \nobreak
  \vskip\belowcaptionskip
\fi
\relax
}
```

Allocate the skip registers for above and below caption space.

```
\newskip\abovecaptionskip \abovecaptionskip=12pt \relax
\newskip\belowcaptionskip \belowcaptionskip=12pt \relax
\newdimen\captionindent \captionindent=3pc
```

2.23 Miscellaneous

kingspace Change `~` to be more forgiving of accidental adjacent spaces. Note that this means multiple `~~~...` cannot be used to get multiple spaces in the output.

```
\def\nonbreakingspace{\unskip\nobreak\ \ignorespaces}
\def~{\protect\nonbreakingspace}
```

Redefine `\@biblabel` to do nothing if the argument is empty. We don't really care what the previous definition was so we don't check it.

```
\def\@biblabel#1{\@ifnotempty{#1}{[#1]}}
```

Changed `\@cite` to always use roman/upright, nonbold font, even in italic or bold text (following AMS style). Turn off `\mathsurround` just in case there are subscripts in the cite numbers.

```
\def\@citestyle{\m@th\upshape\mdseries}
⟨amsart⟩\let\citeform\@firstofone
⟨amsbook | amsproc⟩\def\citeform#1{{\bfseries#1}}
\def\@cite#1#2{{%
  \@citestyle[\citeform{#1}\if@tempswa, #2\fi]}}
```

Make `\cite` robust if it isn't already. Too many unsuspecting users get problems from this in a figure or table caption.

```
\@ifundefined{cite }{%
  \expandafter\let\csname cite \endcsname\cite
  \edef\cite{\@nx\protect\@xp\@nx\csname cite \endcsname}%
}{}
```

\fullwidthdisplay The function \fullwidthdisplay makes a displayed equation take up the full column width even if the current context is an indented list.

```
\def\fullwidthdisplay{\displayindent\z@ \displaywidth\columnwidth}
```

And we insert the \fullwidthdisplay function at the beginning of \everydisplay just in case any later code in \everydisplay needs to use the values of \displayindent or \displaywidth.

```
\edef\@tempa{\noexpand\fullwidthdisplay\the\everydisplay}
\everydisplay\expandafter{\@tempa}
```

A few odds and ends for indexes, based on makeindex. The definition of \see as "see also" is unfortunate, but of long standing, and cannot be changed without destroying backward compatibility, so an alternate command, \seeonly, is provided to cover the basic situation.

```
\newcommand*\seeonlyname{see}
\newcommand*\seename{see also}
\newcommand*\alsoname{see also}
\newcommand*\seeonly[2]{\emph{\seeonlyname} #1}
\newcommand*\see[2]{\emph{\seename} #1}
\newcommand*\seealso[2]{\emph{\alsoname} #1}
\newcommand\printindex{\@input{\jobname.ind}}
```

\textprime A text prime symbol, for applying primes to numbers such as list numbers or equation numbers that are not really math. Furthermore Cyrillic myagkii znak, or soft sign, is represented by a prime symbol in Russian names when they are transliterated into English.

\textprime uses the prime symbol from math, but because it's intended specifically for nonmath use, we avoid going through math mode with $...$. We must therefore call \check@mathfonts to ensure that scriptfont2 is actually defined. Otherwise, if \textprime were used in a document before the first math formula, there would be no adequate assignment yet for \scriptfont2.

The raise value of .8ex is just a reasonable guess at making the bottom of the prime symbol fall near the top of a preceding lowercase letter but still not fall too low on an uppercase letter. We could look up the \fontdimen values used in math mode for superscripts but I don't think it's worth the bother.

Cf. also the definition of \@makefnmark. The prime symbol here is not raised quite so high because I think that is desirable for the soft-sign usage.

```
\DeclareRobustCommand\textprime{\leavevmode
  \raise.8ex\hbox{\check@mathfonts\the\scriptfont2 \char48 }}
```

2.24 Book style variations

Here is the layout for a \maketitle in the amsbook class.

```
⟨*amsbook⟩
\def\maketitle{\par
  \@topnum\z@ % this prevents figures from falling at the top of page 1
  \begingroup
  \@maketitle
  \endgroup
  \c@footnote\z@
  \def\do##1{\let##1\relax}%
  \do\maketitle \do\@maketitle \do\title \do\@xtitle \do\@title
  \do\author \do\@xauthor \do\address \do\@xaddress
  \do\email \do\@xemail \do\curraddr \do\@xcurraddr
  \do\dedicatory \do\@dedicatory \do\thanks \do\thankses
  \do\keywords \do\@keywords \do\subjclass \do\@subjclass
}
\def\@maketitle{%
  \cleardoublepage \thispagestyle{empty}%
  \begingroup \topskip\z@skip
  \null\vfil
  \begingroup
  \LARGE\bfseries \centering
  \openup\medskipamount
  \@title\par\vspace{24pt}%
  \def\and{\par\medskip}\centering
  \mdseries\authors\par\bigskip
  \endgroup
  \vfil
  \ifx\@empty\addresses \else \@setaddresses \fi
  \vfil
  \ifx\@empty\@dedicatory
  \else \begingroup
    \centering{\footnotesize\itshape\@dedicatory\@@par}%
    \endgroup
  \fi
  \vfill
  \newpage\thispagestyle{empty}
  \begin{center}
    \ifx\@empty\@subjclass\else\@setsubjclass\fi
    \ifx\@empty\@keywords\else\@setkeywords\fi
    \ifx\@empty\@translators\else\vfil\@settranslators\fi
    \ifx\@empty\thankses\else\vfil\@setthanks\fi
  \end{center}
  \vfil
  \@setabstract
  \endgroup}
```

Define the desired form for translator names.

```
\def\@settranslators{\par
  \begingroup
  \translname: \andify\@translators \uppercasenonmath\@translators
  \@translators \@@par
```

```
  \endgroup}
\def\@setdate{\par\smallskip\@date\par\smallskip}
\def\@setsubjclass{\par\smallskip
  {\itshape\subjclassname.}\enspace\@subjclass\par\smallskip}
\def\@setkeywords{\par\smallskip
  {\itshape \keywordsname.}\enspace \@keywords\par\smallskip}
⟨/amsbook⟩
⟨/classes⟩
```

2.25 Setup for theorems, definitions, remarks, proofs

Intended usage

Here are some examples showing the kinds of theorem environment declarations that are possible.

```
\newtheorem{prop}{Proposition}
\newtheorem{thm}{Theorem}[section]
\newtheorem{lem}[thm]{Lemma}
\newtheorem*{Zorn}{Zorn's Lemma}

\theoremstyle{definition}
\newtheorem{dfn}{Definition}

\theoremstyle{remark}
\newtheorem*{rmk}{Remark}
```

The first four statements all define environments using the default theorem style ('plain'), since there is no prefatory `\theoremstyle` declaration. The first statement defines an automatically numbered `prop` environment whose headings will look like this: Proposition 1, Proposition 2, and so forth. The second statement defines an environment `thm` with numbers subordinate to section numbers, so the headings will look like this: Theorem 1.1, Theorem 1.2, Theorem 1.3, ..., (in section 2:) Theorem 2.1, Theorem 2.2, and so forth. The third statement defines a `lem` environment whose numbers will interleave in sequence with the theorem numbers: Theorem 1.3, Lemma 1.4, Lemma 1.5, Theorem 1.6, and so forth. The fourth statement defines a special unnumbered lemma named 'Zorn's Lemma'. The remaining two `\newtheorem` statements have no special features except for the `\theoremstyle` declarations that cause the `dfn` and `rmk` environments to have some differences in appearance.

There are three basic styles provided: The 'plain' style produces bold headings and italic body text; the 'definition' style produces bold headings and normal body text; the 'remark' style produces italic headings and normal body text.

A `\swapnumbers` command allows theorem numbers to be swapped to the front of the theorem headings. Putting `\swapnumbers` in your document preamble will cause *all following* `\newtheorem` statements to produce number-first headings. (To provide maximum control, `\swapnumbers` is designed so that it can be used more than once; each time it is used, theorem numbers will be swapped to the opposite side for all following `\newtheorem` statements. But rarely will it need to be invoked more than once per document.)

Custom theorem styles

There is a `\newtheoremstyle` command provided to make the creation of custom theoremstyles fairly easy.

Usage:

```
                  #1
\newtheoremstyle{NAME}%
    #2          #3          #4
  {ABOVESPACE}{BELOWSPACE}{BODYFONT}%
    #5      #6         #7          #8
  {INDENT}{HEADFONT}{HEADPUNCT}{HEADSPACE}%
    #9
  {CUSTOM-HEAD-SPEC}
```

Leaving the 'indent' argument empty is equivalent to entering `0pt`. The 'headpunct' and 'headspace' arguments are for the punctuation and horizontal space between the theorem head and the following text. There are two special values that may be used for 'headspace': a single space means that a normal interword space should be used; "`\newline`" means that there should be a line break after the head instead of horizontal space. The 'custom-head-spec' argument follows a special convention: it is interpreted as the replacement text for an internal three-argument function `\thmhead`, i.e., as if you were defining

```
\renewcommand{\thmhead}[3]{...#1...#2...#3...}
```

but omitting the initial `\renewcommand{\thmhead}[3]`. The three arguments that will be supplied to `\thmhead` are the name, number, and optional note components. Within the replacement text you can (and normally will want to) use other special functions `\thmname`, `\thmnumber`, and `\thmnote`. These will print their argument if and only if the corresponding argument of `\thmhead` is nonempty. For example

```
{\thmname{#1}\thmnumber{ #2}\thmnote{ (#3)}}
```

This would cause the theorem note `#3` to be printed with a preceding space and enclosing parentheses, if it is present, and if it is absent, the space and parentheses will be omitted because they are inside the argument of `\thmnote`.

Finally, if you have an extra bit of arbitrary code that you want to slip in somewhere, the best place to do it is in the 'body font' argument.

The `\newtheoremstyle` command is designed to provide, through a relatively simple interface, control over the style aspects that are most commonly changed. More complex requirements must be addressed through a separate LaTeX package.

When you set up custom theorem styles with `\newtheoremstyle` you should not use `\swapnumbers`. You have full control of the ordering of elements in the theorem head, just place them where you want. Or, if you do use `\swapnumbers`, you must look at the definition of `\swappedhead` and change it appropriately.

Implementation

The `\theoremstyle` command is very simple except for the need to warn about an unknown theoremstyle.

```
⟨*amsthm | classes⟩
\newcommand{\theoremstyle}[1]{%
  \@ifundefined{th@#1}{%
    \PackageWarning{amsthm}{Unknown theoremstyle `#1'}%
    \thm@style{plain}%
  }{%
    \thm@style{#1}%
  }%
}

\newtoks\thm@style
\thm@style{plain}
```

What's really needed is a full-fledged systematic approach for specifying the desired order and formatting of the three identified parts of a theorem head (name, number, note).

```
\newtoks\thm@bodyfont  \thm@bodyfont{\itshape}
\newtoks\thm@headfont  \thm@headfont{\bfseries}
\newtoks\thm@notefont  \thm@notefont{}
\newtoks\thm@headpunct \thm@headpunct{.}
```

Vertical spacing: initialize to current value of `\topsep`. If a document class loads the `amsthm` package it should take care to set these variables explicitly, if current `\topsep` is not the desired value.

```
\newskip\thm@preskip \newskip\thm@postskip
⟨*classes⟩
\def\thm@space@setup{%
  \thm@preskip=.5\baselineskip\@plus.2\baselineskip
                              \@minus.2\baselineskip
  \thm@postskip=\thm@preskip
}
⟨/classes⟩
⟨*amsthm⟩
\def\thm@space@setup{%
  \thm@preskip=\topsep \thm@postskip=\thm@preskip
}
⟨/amsthm⟩
```

Modify `\newtheorem` to add `*` option. If a `*` is found, pass it on to `\@xnthm` as the first argument. (This information enables us to handle two different possibilities in a single function `\@xnthm` instead of needing two separate functions.)

```
\renewcommand{\newtheorem}{\@ifstar{\@xnthm *}{\@xnthm \relax}}
```

Check to see if an optional arg is present after the first mandatory arg (which is `#2` at the moment since the `*` or non-`*` is `#1`).

```
\def\@xnthm#1#2{%
  \let\@tempa\relax
  \@xp\@ifdefinable\csname #2\endcsname{%
    \global\@xp\let\csname end#2\endcsname\@endtheorem
    \ifx *#1% unnumbered, need to get one more mandatory arg
      \edef\@tempa##1{%
```

```
        \gdef\@xp\@nx\csname#2\endcsname{%
          \@nx\@thm{\@xp\@nx\csname th@\the\thm@style\endcsname}%
            {}{##1}}}%
    \else % numbered theorem, need to check for optional arg
      \def\@tempa{\@oparg{\@ynthm{#2}}[]}%
    \fi
  }%
  \@tempa
}
```

Environment numbered together with a previously defined environment.
Arg1: env name, e.g. 'thm'
Arg2: optional sibling counter
Arg3: heading text e.g. 'Theorem'

```
\def\@ynthm#1[#2]#3{%
```

If optional arg `#2` is empty, call `\@xthm` to look for a possible optional arg in terminal position. Note that the two optional args are mutually exclusive. As `#2` is a counter name and must be processed by `\csname` anyway, we can use a simpler test instead of `\@ifempty`.

```
  \ifx\relax#2\relax
    \def\@tempa{\@oparg{\@xthm{#1}{#3}}[]}%
  \else
    \@ifundefined{c@#2}{%
      \def\@tempa{\@nocounterr{#2}}%
    }{%
      \@xp\xdef\csname the#1\endcsname{\@xp\@nx\csname the#2\endcsname}%
      \toks@{#3}%
      \@xp\xdef\csname#1\endcsname{%
        \@nx\@thm{%
          \let\@nx\thm@swap
            \if S\thm@swap\@nx\@firstoftwo\else\@nx\@gobble\fi
          \@xp\@nx\csname th@\the\thm@style\endcsname}%
            {#2}{\the\toks@}}%
      \let\@tempa\relax
    }%
  \fi
  \@tempa
}
```

Environment numbered relative to the counter given as `#3`. This function should really be named `\@znthm` but we're trying to save a bit of hash table and string pool by reusing one of the command names rendered obsolete by the `amsthm` package.

Arg1: env name e.g. 'thm'; Arg2: heading text e.g. 'Theorem'; Arg3: parent counter e.g. section.

```
\def\@xthm#1#2[#3]{%
```

Set up the counter `c@#1` and optionally add it to the reset list of counter `#3`. As `#3` is a counter name and must be processed by `\csname` anyway, we can use a

simpler test instead of \@ifempty.

```
  \ifx\relax#3\relax
    \newcounter{#1}%
  \else
    \newcounter{#1}[#3]%
```

Define \thexxx to be \theyyy.\arabic{xxx} (assuming default values of punctuation and numbering style). The use of \xdef here is inherited from the old LaTeX code, I'm not sure it's a good idea in general, but there should not be any problems unless someone changes the value of \@thmcountersep or \@thmcounter.

```
    \@xp\xdef\csname the#1\endcsname{\@xp\@nx\csname the#3\endcsname
      \@thmcountersep\@thmcounter{#1}}%
  \fi
  \toks@{#2}%
  \@xp\xdef\csname#1\endcsname{%
    \@nx\@thm{%
      \let\@nx\thm@swap
        \if S\thm@swap\@nx\@firstoftwo\else\@nx\@gobble\fi
      \@xp\@nx\csname th@\the\thm@style\endcsname}%
      {#1}{\the\toks@}}%
}
```

[mjd,1999/10/13] The following code doesn't handle the case where an equation is immediately followed by a theorem with no intervening \par—then the spacefactor is 1000.

```
% % \def\thm@check@break{%
% %   \ifhmode \unskip \unskip
% %     \edef\pre@thm@spacefactor{\the\spacefactor}\par
% %     \edef\thm@topbreak{%
% %       \ifnum\pre@thm@spacefactor<\sfcode`\!\relax
% %          % preceding text line did not end with end-of-sentence punctuation
% %         \ifnum\prevgraf=\@ne \penalty\widowpenalty
% %         \else \penalty9999\relax
% %         \fi
% %       \else
% %         \@nx\addpenalty{\@beginparpenalty}%
% %       \fi
% %     }%
% %   \else
% %     \def\thm@topbreak{\addpenalty\@beginparpenalty}%
% %   \fi
% % }
```

If arg #2 is empty, this is an unnumbered environment; otherwise #2 is the name of a counter. #3 is descriptive name such as "Theorem" or "Lemma". Arg #1 is the style function, for example \th@plain.

```
\def\@thm#1#2#3{%
  \ifhmode\unskip\unskip\par\fi
  \normalfont
```

```
  \trivlist
```

Explicitly set plain style here, then override parts as necessary in the function provided as `#1`.

```
  \let\thmheadnl\relax
  \let\thm@swap\@gobble
⟨amsart⟩  \let\thm@indent\noindent % no indent
⟨amsart⟩  \thm@headfont{\bfseries}% heading font bold
⟨amsbook | amsproc⟩  \let\thm@indent\indent % indent
⟨amsbook | amsproc⟩  \thm@headfont{\scshape}% heading font small caps
  \thm@notefont{\fontseries\mddefault\upshape}%
  \thm@headpunct{.}% add period after heading
  \thm@headsep 5\p@ plus\p@ minus\p@\relax
  \thm@space@setup
  #1% style overrides
  \@topsep \thm@preskip               % used by thm head
  \@topsepadd \thm@postskip           % used by \@endparenv
  \def\@tempa{#2}\ifx\@empty\@tempa
    \def\@tempa{\@oparg{\@begintheorem{#3}{}}[]}%
  \else
    \refstepcounter{#2}%
    \def\@tempa{\@oparg{\@begintheorem{#3}{\csname the#2\endcsname}}[]}%
  \fi
  \@tempa
}
```

The internal function `\@restorelabelsep` starts out as a no-op. I don't think this is needed any more [mjd,2000/10/26].

```
\def\@restorelabelsep{\relax}
```

This variation of the `\@thm` command is no longer needed. The variation `\@xthm` was commandeered for `\newtheorem` use.

```
\let\@ythm\relax
```

Init `\thmname` etc.

```
\let\thmname\@iden \let\thmnote\@iden \let\thmnumber\@iden
```

`\@upn` The function `\@upn` is used to force theorem numbers and similar elements to be upright even when the current font is italic. If a suitable italic font with upright numbers and punctuation is available, this function should be redefined to be a no-op.

```
\providecommand\@upn{\textup}
```

Definitions for theorem heads.

```
\def\thmhead@plain#1#2#3{%
```

To allow for the case where the thmname part is empty and the heading consists only of a number (don't laugh, we have examples from real mathematical manuscripts), we don't add the space at the beginning of thmnumber unless `#1` is nonempty.

```
  \thmname{#1}\thmnumber{\@ifnotempty{#1}{ }\@upn{#2}}%
```

In thmnote we always add a leading space, i.e., assuming that at least one of the preceding parts will always be present.

```
\thmnote{ {\the\thm@notefont(#3)}}}
\let\thmhead\thmhead@plain
```

Swappedhead is for the case where the number precedes the word "Theorem". When the numbers fall on the left, like section numbers, AMS journal style calls for them to be lightface. We get this by invoking `\@secnumfont`.

In handling the punctuation after the number we have to step delicately if we want to successfully handle cases where the theorem name is empty (this is very rare, yet it sometimes arises in practice).

```
\def\swappedhead#1#2#3{%
⟨amsthm⟩  \thmnumber{#2}%
⟨amsthm⟩  \thmname{\@ifnotempty{#2}{~}#1}%
⟨classes⟩  \thmnumber{\@upn{\@secnumfont#2\@ifnotempty{#1}{.~}}}%
⟨classes⟩  \thmname{#1}%
  \thmnote{ {\the\thm@notefont(#3)}}}
```

A customized definition of `\th@plain` written for version 1.x of the `amsthm` package might refer to `\swappedhead@plain`; this gives it a definition for backward compatibility.

```
\let\swappedhead@plain=\swappedhead
```

In `\@begintheorem` `\thmheadnl` is called after the theorem head: maybe a newline, otherwise a no-op.

```
\let\thmheadnl\relax
\let\thm@indent\noindent
\let\thm@swap\@gobble
```

If argument `#2` is empty, then this is an unnumbered environment. Otherwise `#2` is a numbering command such as `\thexyz`. We use `\deferred@thm@head` instead of `\item` in order to allow line breaking in the optional note argument.

```
\def\@begintheorem#1#2[#3]{%
  \deferred@thm@head{\the\thm@headfont \thm@indent
```

Changes to `\thmnumber` and `\thmnote` are local to the containing box.

```
    \@ifempty{#1}{\let\thmname\@gobble}{\let\thmname\@iden}%
    \@ifempty{#2}{\let\thmnumber\@gobble}{\let\thmnumber\@iden}%
    \@ifempty{#3}{\let\thmnote\@gobble}{\let\thmnote\@iden}%
```

The `\thm@swap` function selects either `\swappedhead` or `\thmhead`.

```
    \thm@swap\swappedhead\thmhead{#1}{#2}{#3}%
```

I can't think of any example where the after-head punctuation should be omitted so it seems correct not to use `\@addpunct` here.

```
    \the\thm@headpunct
```

If this *is* a newline (from a `\newtheoremstyle`), it gets lost if there isn't any text following the heading, since `\deferred@thm@head` packs the heading into an hbox with `\sbox\@labels`. Attempting to move the `\thmheadnl` outside the scope of `\deferred@thm@head` (just outside the ending brace below) results in

no improvement if no text follows the heading, and where there is text, runs it in. Inserting a space following such a heading results in an extra blank line. [bnb, 2004/06/30]

```
    \thmheadnl % possibly a newline.
    \hskip\thm@headsep
  }%
  \ignorespaces}

\newskip\thm@headsep
\thm@headsep=5pt plus1pt minus1pt\relax

\let\adjust@parskip@nobreak=\@nbitem

\newtoks\dth@everypar
\dth@everypar={%
  \@minipagefalse \global\@newlistfalse
  \@noparitemfalse
  \if@inlabel
    \global\@inlabelfalse
    \begingroup \setbox\z@\lastbox
     \ifvoid\z@ \kern-\itemindent \fi
    \endgroup
    \unhbox\@labels
  \fi
  \if@nobreak \@nobreakfalse \clubpenalty\@M
  \else \clubpenalty\@clubpenalty \everypar{}%
  \fi
}%

\def\deferred@thm@head#1{%
  \if@inlabel \indent \par \fi % eject a section head if one is pending
  \if@nobreak
```

This case normally arises when a theorem follows immediately after a section head. Then we leave the below-section-head space instead of adding above-theorem space; but some adjustment of parskip is needed.

```
    \adjust@parskip@nobreak
  \else
    \addpenalty\@beginparpenalty
    \addvspace\@topsep
    \addvspace{-\parskip}%
  \fi
  \global\@inlabeltrue
  \everypar\dth@everypar
  \sbox\@labels{\normalfont#1}%
  \ignorespaces
}
```

`\nonslanted` The `\nonslanted` command changes the current font to `\upshape` if it is `\itshape` or `\slshape`. This is used for document structure numbers that should be consistently upright in all contexts.

```
\def\nonslanted{\relax
```

Can't do a direct \ifx between \f@shape and \itdefault because the latter is \long (grumble grumble).

```
\@xp\let\@xp\@tempa\csname\f@shape shape\endcsname
\ifx\@tempa\itshape\upshape
\else\ifx\@tempa\slshape\upshape\fi\fi}
```

\swapnumbers The \swapnumbers command sets a switch \thm@swap that is used by \newtheorem. To conserve hash table we load \thm@swap with two uses; the first one is needed only in \newtheorem declarations and the second one is needed only in typesetting theorem environments.

```
\def\swapnumbers{\edef\thm@swap{\if S\thm@swap N\else S\fi}}
\def\thm@swap{N}%
```

\@opargbegintheorem not needed, \@oparg utility serves instead.

```
\let\@opargbegintheorem\relax
```

Except for the body font, default values are built into \@thm.

```
\def\th@plain{%
%%  \let\thm@indent\noindent % no indent
%%  \thm@headfont{\bfseries}% heading font is bold
%%  \thm@notefont{}% same as heading font
%%  \thm@headpunct{.}% add period after heading
%%  \let\thm@swap\@gobble
%%  \thm@preskip\topsep
%%  \thm@postskip\theorempreskipamount
  \itshape % body font
}
```

Theorem style 'definition' is the same as 'plain' except for the body font.

```
\def\th@definition{%
  \normalfont % body font
}
```

Theorem style 'remark' differs from 'plain' in head font and body font. Also smaller spacing above and below for AMS classes only.

```
\def\th@remark{%
⟨amsart | amsthm⟩  \thm@headfont{\itshape}%
  \normalfont % body font
⟨*amsthm⟩
  \thm@preskip\topsep \divide\thm@preskip\tw@
  \thm@postskip\thm@preskip
⟨/amsthm⟩
}
```

The standard definition of \@endtheorem is just \endtrivlist, but that doesn't automatically start a new paragraph, so we add \@endpefalse in order to ensure a new paragraph. This differs from the basic LaTeX behavior.

```
\def\@endtheorem{\endtrivlist\@endpefalse }
```

oremstyle An easy way to make a not too complicated variant theorem style. Usage:

```
                    #1
 \newtheoremstyle{NAME}%
     #2          #3          #4
   {ABOVESPACE}{BELOWSPACE}{BODYFONT}%
     #5       #6         #7          #8
   {INDENT}{HEADFONT}{HEADPUNCT}{HEADSPACE}%
     #9
   {CUSTOM-HEAD-SPEC}
```

```
\newcommand{\newtheoremstyle}[9]{%
```

Empty or 0pt for #5 is translated to \noindent.

```
  \@ifempty{#5}{\dimen@\z@skip}{\dimen@#5\relax}%
  \ifdim\dimen@=\z@
```

#4 is body font. Extra code could be included there if necessary.

```
    \toks@{#4\let\thm@indent\noindent}%
  \else
    \toks@{#4\def\thm@indent{\noindent\hbox to#5{}}}%
  \fi
```

Arg #8 is a glue spec for the space after the head. As a proper glue spec for 'normal interword space' is rather hard to write, we recognize an argument of { } as a special case and translate internally to the necessary fontdimen equivalent. Furthermore, if #8 consists entirely of \newline, then we will perform a line break after the theorem head instead of adding horizontal space. At the moment [1995/01/23] this is not perfectly well implemented because of complications with the way LaTeX's \item adds a heading to the vertical list; for best results there should not be anything (not even a blank line) after the \begin{xxx} command.

```
  \def\@tempa{#8}\ifx\space\@tempa
```

Notice that we disregard stretch and shrink for labelsep = interwordspace.

```
    \toks@\@xp{\the\toks@ \thm@headsep\fontdimen\tw@\font\relax}%
  \else
    \def\@tempb{\newline}%
    \ifx\@tempb\@tempa
      \toks@\@xp{\the\toks@ \thm@headsep\z@skip
        \def\thmheadnl{\newline}}%
    \else
      \toks@\@xp{\the\toks@ \thm@headsep#8\relax}%
    \fi
  \fi
  \begingroup
  \thm@space@setup
  \@defaultunits\@tempskipa#2\thm@preskip\relax\@nnil
  \@defaultunits\@tempskipb#3\thm@postskip\relax\@nnil
  \xdef\@gtempa{\thm@preskip\the\@tempskipa
    \thm@postskip\the\@tempskipb\relax}%
```

```
  \endgroup
  \@temptokena\@xp{\@gtempa
    \thm@headfont{#6}\thm@headpunct{#7}%
  }%
  \@ifempty{#9}{%
    \let\thmhead\thmhead@plain
  }{%
    \@namedef{thmhead@#1}##1##2##3{#9}%
    \@temptokena\@xp{\the\@temptokena
      \@xp\let\@xp\thmhead\csname thmhead@#1\endcsname}%
  }%
  \@xp\xdef\csname th@#1\endcsname{\the\toks@ \the\@temptokena}%
}
```

\qed Define \qed for end of proof. This command might occur in math mode, in a displayed equation, but it should never occur in inner math mode in ordinary paragraph text.

```
\DeclareRobustCommand{\qed}{%
  \ifmmode \mathqed
  \else
    \leavevmode\unskip\penalty9999 \hbox{}\nobreak\hfill
```

The hbox is to prevent a line break within the \qedsymbol if it is defined to be something composite— e.g., things like (Corollary 1.2) \openbox as are occasionally done.

```
    \quad\hbox{\qedsymbol}%
  \fi
}
```

Setup for QED stack. We use a dedicated iterator macro \qed@elt instead of the generic \@elt because the LaTeX output routine is not safe against changes in \@elt. Therefore it is not safe to use \@elt for any processing that might trigger the output routine. Although this does not seem very likely when adding a QED symbol to a horizontal list, we did in fact get a bug report for this kind of failure.

```
\let\QED@stack\@empty
\let\qed@elt\relax
```

Puts a QED symbol on the stack:

```
\newcommand{\pushQED}[1]{%
  \toks@{\qed@elt{#1}}\@temptokena\expandafter{\QED@stack}%
  \xdef\QED@stack{\the\toks@\the\@temptokena}%
}
```

Pops the QED stack and prints the result.

```
\newcommand{\popQED}{%
  \begingroup\let\qed@elt\popQED@elt \QED@stack\relax\relax\endgroup
}
\def\popQED@elt#1#2\relax{#1\gdef\QED@stack{#2}}
```

Prints the current QED symbol and replaces the top entry on the stack with a null entry.

```
\newcommand{\qedhere}{%
  \begingroup \let\mathqed\math@qedhere
    \let\qed@elt\setQED@elt \QED@stack\relax\relax \endgroup
}
```

In case the `amsmath` or `amstext` packages are loaded, we need the following two tests. (Redundantly declaring them is harmless.)

```
\newif\ifmeasuring@
\newif\iffirstchoice@ \firstchoice@true

\def\setQED@elt#1#2\relax{%
  \ifmeasuring@
  \else \iffirstchoice@ \gdef\QED@stack{\qed@elt{}#2}\fi
  \fi
  #1%
}
```

`\mathqed` When a QED occurs inside a math formula, well, it is presumably a displayed equation. In order to find out where to place the QED symbol, we need to check what kind of equation structure we are in.

```
\def\qed@warning{%
  \PackageWarning{amsthm}{The \@nx\qedhere command may not work
    correctly here}%
}

\newcommand{\mathqed}{\quad\hbox{\qedsymbol}}
```

The `\linebox@qed` function comes into play with the `fleqn` option.

```
\def\linebox@qed{\hfil\hbox{\qedsymbol}\hfilneg}
```

Two large sections of code follow. One for `amsmath` 2.0 and one for plain LaTeX.

```
\@ifpackageloaded{amsmath}{%

  \def\math@qedhere{%
    \@ifundefined{\@currenvir @qed}{%
      \qed@warning\quad\hbox{\qedsymbol}%
    }{%
      \@xp\aftergroup\csname\@currenvir @qed\endcsname
    }%
  }

  \def\displaymath@qed{%
    \relax
    \ifmmode
      \ifinner \aftergroup\linebox@qed
      \else
        \eqno
        \let\eqno\relax \let\leqno\relax \let\veqno\relax
```

```
      \hbox{\qedsymbol}%
    \fi
  \else
     \aftergroup\linebox@qed
  \fi
}
\@xp\let\csname equation*@qed\endcsname\displaymath@qed
```

If the equation has both an equation number and a qed, we've got trouble, but we can provide half-way decent for the simple cases.

```
\def\equation@qed{%
  \iftagsleft@
    \hbox{\phantom{\quad\qedsymbol}}%
    \gdef\alt@tag{%
      \rlap{\hbox to\displaywidth{\hfil\qedsymbol}}%
      \global\let\alt@tag\@empty
    }%
  \else
    \gdef\alt@tag{%
      \global\let\alt@tag\@empty
      \vtop{\ialign{\hfil####\cr
              \tagform@\theequation\cr
              \qedsymbol\cr}}%
      \setbox\z@
    }%
  \fi
}
\def\qed@tag{%
  \global\tag@true \nonumber
  &\omit\setboxz@h {\strut@ \qedsymbol}\tagsleft@false
  \place@tag@gather
  \kern-\tabskip
  \ifst@rred \else \global\@eqnswtrue \fi \global\advance\row@\@ne \cr
}
\def\split@qed{%
  \def\endsplit{\crcr\egroup \egroup \ctagsplit@false \rendsplit@
    \aftergroup\align@qed
  }%
}
\def\align@qed{%
  \ifmeasuring@ \tag*{\qedsymbol}%
  \else \let\math@cr@@@\qed@tag
  \fi
}
\@xp\let\csname align*@qed\endcsname\align@qed
\@xp\let\csname gather*@qed\endcsname\align@qed
%% Needs some patching up for amsmath 1.2
}{% end of amsmath branch, start plain LaTeX branch
```

The `\qedhere` handling for generic LaTeX classes (`article`, `book`) with the `amsthm` package (without `amsmath`) is fairly sketchy. When a qed symbol and an equation number are both applied to a single display, the results may not be entirely satisfactory, particularly when the `fleqn` and/or `leqno` options are used. [mjd,2000/01/17]

As for `\math@qedhere`, it is expected to occur only via `\qedhere`, where the `\aftergroup` makes sense.

```
\def\math@qedhere{%
  \@ifundefined{\@currenvir @qed}{%
    \qed@warning \aftergroup\displaymath@qed
  }{%
    \@xp\aftergroup\csname\@currenvir @qed\endcsname
  }%
}
```

The `\ifmmode \ifinner` case is expected to happen with the `fleqn` option, where we have something like:

```
\hbox to\linewidth{\hbox{$...$}\hfil}
```

In order to counteract the `\hfil` we must jump out two grouping levels.

```
\def\displaymath@qed{%
  \relax
  \ifmmode
    \ifinner \aftergroup\aftergroup\aftergroup\linebox@qed
    \else
      \eqno \def\@badmath{$$}%
      \let\eqno\relax \let\leqno\relax \let\veqno\relax
      \hbox{\qedsymbol}%
    \fi
  \else
     \aftergroup\linebox@qed
  \fi
}
```

This definition is a fallback definition that places the qed and then the equation number, on the right-hand side. For `leqno`, not so good; but then

```
\@ifundefined{ver@leqno.clo}{%
  \def\equation@qed{\displaymath@qed \quad}%
}{%
  \def\equation@qed{\displaymath@qed}%
}
```

If `amsmath` is not loaded, then we need to do some surgery on the `\[` macro. Normally it looks like this:

```
\[=macro:
->\relax \ifmmode \@badmath \else
\ifvmode \nointerlineskip \makebox [.6\linewidth ]\fi $$\fi
```

If arg 2 is `\m@th` when we make this test it indicates that the fleqn option is in effect. Perhaps try to do something there.

```
    \def\@tempa#1$#2#3\@nil#4{%
        \def#4{#1$#2\def\@currenvir{displaymath}#3}%
    }%
```

If `\[` has already been made robust (by `fixltx2e` or a LaTeX format from 2015 or later), patch the internal macro `\[<space>` instead.

```
    \expandafter\ifx\csname[ \endcsname\relax
        \expandafter\@tempa\[\@nil\[%
    \else
        \expandafter\expandafter\expandafter\@tempa\csname[
        \expandafter\endcsname\expandafter\@nil
        \csname[ \endcsname
    \fi
}
```

If an older version of `amsmath` is in use, we need to fall back to a simpler definition of `\math@qedhere`.

```
\@ifpackageloaded{amstex}{%
  \def\@tempa{TT}%
}{%
  \@ifpackageloaded{amsmath}{%
    \def\@tempb#1 v#2.#3\@nil{#2}%
    \ifnum\@xp\@xp\@xp\@tempb\csname ver@amsmath.sty\endcsname v0.0\@nil
       <\tw@
      \def\@tempa{TT}%
    \else
      \def\@tempa{TF}%
    \fi
  }{%
    \def\@tempa{TF}
  }%
}
\if\@tempa
  \renewcommand{\math@qedhere}{\quad\hbox{\qedsymbol}}%
\fi
```

The reason that we do not simply use the `\square` symbol from `msam` for the open-box qed symbol is that we want to avoid requiring users to have the AMSFonts font package. And the `lasy` `\Box` is too large.

```
\newcommand{\openbox}{\leavevmode
```

I think I got these numbers from measuring `msam`'s `\square` but I forgot to make notes at the time. [mjd,1995/01/25]

```
  \hbox to.77778em{%
  \hfil\vrule
  \vbox to.675em{\hrule width.6em\vfil\hrule}%
  \vrule\hfil}}

\DeclareRobustCommand{\textsquare}{%
  \begingroup \usefont{U}{msa}{m}{n}\thr@@\endgroup
}
```

```
⟨*classes⟩
\@ifclasswith{\@classname}{noamsfonts}{%
  \providecommand{\qedsymbol}{\openbox}%
}{}
\providecommand{\qedsymbol}{\textsquare}
⟨/classes⟩
⟨amsthm⟩\providecommand{\qedsymbol}{\openbox}
```

The proof environment is never numbered, and has a `\qed` at the end, which makes it inconvenient to use `\newtheorem` for defining it. Also authors frequently need to substitute an alternative heading text (e.g. 'Proof of Lemma 4.3') instead of the default 'Proof'. For all these reasons we define the proof environment here instead of leaving it for authors to define. Text after the end of a proof, like that after the end of a theorem, begins a new paragraph. This differs from basic LaTeX behavior. [bnb, 1999/09/27]

```
\newenvironment{proof}[1][\proofname]{\par
  \pushQED{\qed}%
  \normalfont \topsep6\p@\@plus6\p@\relax
  \trivlist
⟨amsbook | amsproc⟩  \itemindent\normalparindent
  \item[\hskip\labelsep
⟨amsbook | amsproc⟩        \scshape
⟨amsart | amsthm⟩         \itshape
    #1\@addpunct{.}]\ignorespaces
}{%
  \popQED\endtrivlist\@endpefalse
}
```

Default for `\proofname`:

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

For reference:

```
From: tycchow@math.mit.edu (Timothy Y. Chow)
Subject: Suppressing theorem numbering in LaTeX
Message-ID: <1994Aug11.234754.22523@galois.mit.edu>
Date: Thu, 11 Aug 94 23:47:54 GMT
To: tex-news@SHSU.EDU

A friend of mine wants numbering of theorems, conjectures, and so on
suppressed if there is only one of them in his article.  In other words
he wants "Conjecture 1" to appear as simply "Conjecture" if there is no
Conjecture 2.  What is the best way to go about doing this?
...
```

Maybe something clever can be done to make the desired behavior happen automatically. Note that this would seem to be a general numbering problem rather than a theorem-specific one, because similar behavior would be desirable for appendixes: according to standard publishing practice, if there's only one it is titled just 'Appendix', and if there are more than one they are titled 'Appendix A', 'Appendix B', and so on.

```
⟨/amsthm | classes⟩
⟨*classes⟩
```

2.26 Utility commands used with AMS author packages

\bb@skip Skip to result in base-to-base distance from previous to next box.

```
\def\bb@skip#1{%
  \skip@#1\relax \advance\skip@-\prevdepth \advance\skip@-\baselineskip
  \vskip\skip@}
```

\markleft Basic LaTeX has \markright and \markboth, but sometimes it's necessary to change just the left running head. This macro completes the set. From the altxext.sty module of Klaus Lagally's arabtex. [bnb, 2004/03/25]

```
\def\markleft#1{{\let\protect\noexpand
    \let\label\relax \let\index\relax \let\glossary\relax
    \expandafter\@markleft\@themark{#1}%
    \mark{\@themark}}%
  \if@nobreak\ifvmode\nobreak\fi\fi}
\def\@markleft#1#2#3{\gdef\@themark{{#3}{#2}}}
```

\DH \dh \DJ \dj The Icelandic thorn and eth and the Croatian barred D are part of the T1 font encoding, but aren't available in OT1. However, they are needed (rarely) in the author names or bibliographies. Provide emulations, using the thorn in msbm or D's barred with a macron. The lowercase eth requires an \edef to access msbm properly. Define it separately to permit checking for small caps. [bnb, 2004/04/05]

```
\@ifundefined{symAMSb}{%
  \def\@dh{dh}%
  \ClassWarning{\@classname}{\string\dh\space unavailable without amsfonts;
    replaced by 'dh'}%
}{%
  \edef\@dh{\noexpand\mathhexbox{\hexnumber@\symAMSb}67}%
}
\DeclareTextCommand{\dh}{OT1}{%
  \edef\@tempb{\scdefault}%
  \ifx\f@shape\@tempb
    \leavevmode
    \raisebox{-.8ex}{\makebox[\z@][l]{\hskip-.08em\accent"16\hss}}d%
  \else
    \@dh
  \fi
}
\DeclareTextCommand{\DH}{OT1}{%
  \leavevmode\raisebox{-.5ex}{\makebox[\z@][l]{\hskip-.07em\accent"16\hss}}D}
\DeclareTextCommand{\DJ}{OT1}{%
  \leavevmode\raisebox{-.5ex}{\makebox[\z@][l]{\hskip-.07em\accent"16\hss}}D}
\DeclareTextCommand{\dj}{OT1}{%
  \edef\@tempa{\f@shape}\edef\@tempb{\scdefault}%
  \ifx\@tempa\@tempb
    \leavevmode
```

```
    \raisebox{-.75ex}{\makebox[\z@][l]{\hskip-.08em\accent"16\hss}}d%
  \else
    \leavevmode\raisebox{.02ex}{\makebox[\z@][l]{\hskip.1em\accent"16\hss}}d%
  \fi}
```

2.27 Hyphenation exceptions

Some common hyphenation exceptions, based on the listing in TUGboat vol 10, no 3, November 1989, pp. 336–341. Many words from the TUGboat list that seemed less likely to occur in mathematical text have been omitted because hyphenation exceptions use up memory (most versions of TEX currently have a limit of 307 on hyphenation exceptions; each added hyphenation exception uses up something like 2 extra words of main memory as well). The list of hyphenations for proper names could be expanded forever if room permitted; we restrict ourselves to a rather short, extremely arbitrary list. Note that the hyphenation rules of British English differ in some particulars from the US rules, so some of the hyphenations given below may need to be overridden for proper UK hyphenation.

```
\hyphenation{acad-e-my acad-e-mies af-ter-thought anom-aly anom-alies
an-ti-deriv-a-tive an-tin-o-my an-tin-o-mies apoth-e-o-ses
apoth-e-o-sis ap-pen-dix ar-che-typ-al as-sign-a-ble as-sist-ant-ship
as-ymp-tot-ic asyn-chro-nous at-trib-uted at-trib-ut-able bank-rupt
bank-rupt-cy bi-dif-fer-en-tial blue-print busier busiest
cat-a-stroph-ic cat-a-stroph-i-cally con-gress cross-hatched data-base
de-fin-i-tive de-riv-a-tive dis-trib-ute dri-ver dri-vers eco-nom-ics
econ-o-mist elit-ist equi-vari-ant ex-quis-ite ex-tra-or-di-nary
flow-chart for-mi-da-ble forth-right friv-o-lous ge-o-des-ic
ge-o-det-ic geo-met-ric griev-ance griev-ous griev-ous-ly
hexa-dec-i-mal ho-lo-no-my ho-mo-thetic ideals idio-syn-crasy
in-fin-ite-ly in-fin-i-tes-i-mal ir-rev-o-ca-ble key-stroke
lam-en-ta-ble light-weight mal-a-prop-ism man-u-script mar-gin-al
meta-bol-ic me-tab-o-lism meta-lan-guage me-trop-o-lis
met-ro-pol-i-tan mi-nut-est mol-e-cule mono-chrome mono-pole
mo-nop-oly mono-spline mo-not-o-nous mul-ti-fac-eted mul-ti-plic-able
non-euclid-ean non-iso-mor-phic non-smooth par-a-digm par-a-bol-ic
pa-rab-o-loid pa-ram-e-trize para-mount pen-ta-gon phe-nom-e-non
post-script pre-am-ble pro-ce-dur-al pro-hib-i-tive pro-hib-i-tive-ly
pseu-do-dif-fer-en-tial pseu-do-fi-nite pseu-do-nym qua-drat-ic
quad-ra-ture qua-si-smooth qua-si-sta-tion-ary qua-si-tri-an-gu-lar
quin-tes-sence quin-tes-sen-tial re-arrange-ment rec-tan-gle
ret-ri-bu-tion retro-fit retro-fit-ted right-eous right-eous-ness
ro-bot ro-bot-ics sched-ul-ing se-mes-ter semi-def-i-nite
semi-ho-mo-thet-ic set-up se-vere-ly side-step sov-er-eign spe-cious
spher-oid spher-oid-al star-tling star-tling-ly sta-tis-tics
sto-chas-tic straight-est strange-ness strat-a-gem strong-hold
sum-ma-ble symp-to-matic syn-chro-nous topo-graph-i-cal tra-vers-a-ble
tra-ver-sal tra-ver-sals treach-ery turn-around un-at-tached
un-err-ing-ly white-space wide-spread wing-spread wretch-ed
wretch-ed-ly Eng-lish Euler-ian Feb-ru-ary Gauss-ian
```

```
Hamil-ton-ian Her-mit-ian Jan-u-ary Japan-ese Kor-te-weg
Le-gendre Mar-kov-ian Noe-ther-ian No-vem-ber Rie-mann-ian Sep-tem-ber}
```

2.28 Initialization

We define a function to do the normal calculations that we want for `\textheight` and `\textwidth`

`\calclayout` Subtract the height of the running heads:

```
\def\calclayout{\advance\textheight -\headheight
  \advance\textheight -\headsep
```

We set `\oddsidemargin` and `\evensidemargin` to center the text on the page.

```
  \oddsidemargin\paperwidth
  \advance\oddsidemargin -\textwidth
  \divide\oddsidemargin\tw@
```

Now we subtract the default margin provided by standard DVI drivers. But first we make sure that the final margin will be at least .5 inch.

```
  \ifdim\oddsidemargin<.5truein \oddsidemargin.5truein \fi
  \advance\oddsidemargin -1truein
  \evensidemargin\oddsidemargin
```

And we set `\topmargin` to get vertical centering as well.

```
  \topmargin\paperheight \advance\topmargin -\textheight
  \advance\topmargin -\headheight \advance\topmargin -\headsep
```

Height of running foot ignored: not present.

```
  \divide\topmargin\tw@
```

We provide a minimum of .5in (after compensating for the default margin—see next step).

```
  \ifdim\topmargin<.5truein \topmargin.5truein \fi
```

Now subtract the default margin provided by standard DVI drivers.

```
  \advance\topmargin -1truein\relax
}
```

Initialize the dimensions, page numbering, etc. For inhouse use, administrative stuff is isolated in separate files.

```
⟨amsart⟩\InputIfFileExists{amsart.cfg}{}{%
⟨amsproc⟩\InputIfFileExists{amsproc.cfg}{}{%
⟨amsbook⟩\InputIfFileExists{amsbook.cfg}{}{%
  \calclayout % initialize
  \pagenumbering{arabic}%
  \pagestyle{headings}%
  \thispagestyle{plain}%
}
```

If we are in compatibility mode, add some backward compatibility stuff below. Otherwise quit here.

```
\if@compatibility \else\endinput\fi
```

Compensate for changed meaning of `\tiny`:

```
\def\tiny{\Tiny}
```

The macro `\defaultfont` was provided in version 1.1 of amsart; retained for compatibility as a synonym of `\normalfont`. Resets everything except for size.

```
\def\defaultfont{\normalfont}
```

Macro for making non-slanted numbers and punctuation in italic or slanted text. This is to avoid visual inconsistencies between numbers or parentheses in math and adjacent numbers or parentheses in text.

```
\def\rom{\textup}
```

For backward compatibility with version 1.1 of `amsart`, we define `pf`, `pf*` environments. And undefine `\proof` just in case an existing document contains a `\newenvironment` or `\newcommand` for it, as that would now cause an error.

```
\let\@newpf\proof \let\proof\relax \let\endproof\relax
\newenvironment{pf}{\@newpf[\proofname]}{\popQED\endtrivlist}
\newenvironment{pf*}[1]{\@newpf[#1]}{\popQED\endtrivlist}
```

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
\endinput
```

⟨/classes⟩