面向作者的 LaTeX— 历史版本

© 版权 2020-2023, LaTeX Project Team.
版权所有*

张泓知　　翻译

2022 年 08 月 30 日

目　录

---

1   介绍

欢迎使用 LaTeX 2ε，这是 LaTeX 文档准备系统的新标准版本。

本文档描述了如何利用 LaTeX 的新特性，以及如何使用 LaTeX 2ε 处理你的旧 LaTeX 文档。但是，本文档只是对新功能的简要介绍，面向已经熟悉旧版 LaTeX 的作者。这并不是 LaTeX 2ε 的参考手册，也不是对 LaTeX 的完整介绍。

因为 LaTeX 2ε 在 1994 年问世，所以它现在有点成为了历史文档。

1.1 LaTeX 2_ε——全新的 LaTeX 版本（已经超过 10 年了）

之前的 LaTeX 版本被称为 LaTeX 2.09。多年来，为 LaTeX 开发了许多扩展。这当然是它持续受欢迎的一个明确迹象，但也导致了一个不幸的结果：在不同地点出现了不兼容的 LaTeX 格式。这包括了"标准 LaTeX 2.09"、使用 *New Font Selection Scheme* (NFSS) 构建的 LaTeX、SLITeX、$\mathcal{AMS}$-LaTeX等。因此，为了处理来自不同地点的文档，站点维护者被迫保留多个版本的 LaTeX 程序。此外，查看源文件时并不总是清楚文档是为哪种格式编写的。

为了结束这种不令人满意的局面，产生了 LaTeX 2_ε；它将所有这些扩展重新统一到一个格式下，防止了相互不兼容的 LaTeX 2.09 方言的蔓延。在 LaTeX 2_ε 中，"新字体选择方案"是标准的，例如，amsmath（原为 $\mathcal{AMS}$-LaTeX 格式）或者 slides（原为 SLITeX 格式）只是扩展，可以被使用相同基础格式的文档加载。

新版本的引入也使得添加了少量经常请求的功能成为可能，并使得编写宏包和文档类的任务更加简单。

1.2 LaTeX3——LaTeX 的长期未来

LaTeX 2_ε 是对 LaTeX 系统全面重新实现的巩固步骤。下一个主要版本的 LaTeX 将是 LaTeX3，它将包括对文档设计者和宏包编写者接口的彻底改造。

LaTeX3 是一个长期的研究项目，但在它完成之前，项目团队致力于积极维护 LaTeX 2_ε。因此，从生产和维护 LaTeX 2_ε 中获得的经验将对 LaTeX3 的设计产生重大影响。该项目的简要描述可在文档 `ltx3info.tex` 中找到。

如果您想支持该项目，欢迎向 LaTeX 项目基金捐款；这个基金旨在通过资助与当前 LaTeX 的维护和进一步开发相关的各种费用来帮助研究团队进行这项自愿工作。

该基金由 TeX 用户组和各地区用户组管理。有关捐款和加入这些组织的信息，请访问：

http://www.tug.org/lugs.html

LaTeX3 项目在万维网上有它的首页：

http://www.latex-project.org/

该页面描述了 LaTeX 和 LaTeX3 项目，并包含指向其他 LaTeX 资源的链接，例如用户指南、TeX 常见问题解答和 LaTeX 错误数据库。

早期涵盖 LaTeX3 项目各个方面的文章也可通过 Comprehensive TeX Archive 的匿名 ftp 获取，位于以下目录：

```
ctan:info/ltx3pub
```

该目录中的文件 ltx3pub.bib 包含每个文件的摘要。

1.3   概述

本文档包含了 LaTeX 新结构和新特性的概述。它并非一个独立的文档，因为它仅包含了自 2.09 版本以来发生变化的 LaTeX 特性。您应该结合一个 LaTeX 的介绍来阅读本文档。

第2节 包含了 LaTeX 文档的新结构概述。描述了类和宏包的工作原理，以及如何使用类和宏包选项。列出了随 LaTeX 一同提供的标准宏包和类。

第3节 描述了 LaTeX 2_ε 中作者可以使用的新命令。

第4节 展示了如何用 LaTeX 2_ε 处理旧的 LaTeX 文档。

第6节 包含了处理在运行 LaTeX 2_ε 时可能遇到的问题的建议。列出了 LaTeX 2_ε 中的一些新错误消息，并描述了一些常见问题的解决方法，或者可以找到进一步信息的位置。

1.4   更多信息

要了解 LaTeX 的一般介绍，包括 LaTeX 2_ε 的新特性，您应该阅读 Leslie Lamport 的著作《 LaTeX: A Document Preparation System 》[4]。

关于 LaTeX 新特性的更详细描述，包括对 200 多个宏包和近 1000 个示例的概述，可在 Frank Mittelbach 和 Michel Goossens 的著作《 The LaTeX Companion, second edition 》[5] 中找到。

有关生成和处理图形的宏包和程序的详细讨论可在 Michel Goossens、Sebastian Rahtz 和 Frank Mittelbach 的著作 《 The LaTeX Graphics Companion 》[1] 中找到。

在《The LaTeX Web Companion》[2] 中提供了使用 LaTeX 在万维网上发布的解决方案。

要了解更多关于众多新的 LaTeX 宏包的信息，您应该阅读宏包的文档，这些文档应该从您的 LaTeX 复制品的同一来源获取。

每份 LaTeX 复制品都附带了一些文档文件。每六个月发布的《LaTeX News》将与之同行；它会在文件 `ltnews*.tex` 中找到。类和宏包编写者指南《LaTeX 2ε for Class and Package Writers》描述了写文档类和宏包的新 LaTeX 特性；它在 `clsguide.tex` 中。指南《LaTeX 2ε Font Selection》描述了文档类和宏包作者的 LaTeX 字体选择方案；它在 `fntguide.tex` 中。关于在 LaTeX 中支持西里尔语言的信息描述在《Cyrillic languages support in LaTeX》中。

现在已经可以获取文档化的源代码（通过用于生成核心格式的文件 `latex.ltx`）。它被命名为《The LaTeX 2ε Sources》。这个非常庞大的文档还包含了 LaTeX 命令的索引。您可以使用 `base` 目录中的源文件和类文件 `ltxdoc.cls`，从文件 `source2e.tex` 中排版它。

要了解更多关于 TeX 和 LaTeX 的信息，请联系您所在地的 TeX 用户组，或国际 TeX 用户组（见第 3 页）。

2 类与宏包

本节描述了 LaTeX 文档的新结构以及新类型的文件：类和宏包。

2.1 什么是类和宏包？

LaTeX 2.09 和 LaTeX 2ε 之间的主要区别在于 `\begin{document}` 之前的命令。

在 LaTeX 2.09 中，文档具有样式，比如 article 或 book，以及选项，比如 twoside 或 epsfig。这些由 `\documentstyle` 命令表示：

$$\text{\textbackslash documentstyle[}\langle options\rangle\text{]\{}\langle style\rangle\text{\}}$$

例如，要指定一个双面的带有封装的 PostScript 图形的文章，您可以这样写：

```
\documentstyle[twoside,epsfig]{article}
```

然而，文档样式选项有两种不同的类型：内置选项比如 twoside；以及宏包比如 epsfig.sty。这些非常不同，因为任何 LaTeX 文档样式都可以使用 epsfig 宏包，但只有声明了 twoside 选项的文档样式才能使用该选项。

为了避免这种混淆，LaTeX 2$_\varepsilon$ 在内置选项和宏包之间做了区分。这由新的 \documentclass 和 \usepackage 命令给出：

```
\documentclass[⟨options⟩]{⟨class⟩}
\usepackage[⟨options⟩]{⟨packages⟩}
```

例如，要指定一个双面的文章并包含封装的 PostScript 图形，您现在应该这样写：

```
\documentclass[twoside]{article}
\usepackage{epsfig}
```

您可以使用单个 \usepackage 命令加载多个宏包；例如，而不是写成：

```
\usepackage{epsfig}
\usepackage{multicol}
```

您可以指定为：

```
\usepackage{epsfig,multicol}
```

请注意，LaTeX 2$_\varepsilon$ 仍然理解 LaTeX 2.09 的 \documentstyle 命令。这个命令会使 LaTeX 2$_\varepsilon$ 进入 *LaTeX 2.09* 兼容模式，该模式在第4节中描述。

然而，您不应该为新文档使用 \documentstyle 命令，因为这个兼容模式非常慢，并且在这种模式下无法使用 LaTeX 2$_\varepsilon$ 的新特性。

为了帮助区分类和宏包，文档类现在以 .cls 结尾，而不是 .sty。宏包仍然以 .sty 结尾，因为大多数 LaTeX 2.09 宏包与 LaTeX 2$_\varepsilon$ 配合良好。

2.2   Class and package options

In LaTeX 2.09, only document styles could have options such as twoside or draft. In LaTeX 2$_\varepsilon$, both classes and packages are allowed to have options. For example, to specify a two-sided article with graphics using the dvips driver, you write:

```
\documentclass[twoside]{article}
\usepackage[dvips]{graphics}
```

It is possible for packages to share common options. For example, you could, in addition, load the color package by specifying:

```
\documentclass[twoside]{article}
\usepackage[dvips]{graphics}
\usepackage[dvips]{color}
```

But because `\usepackage` allows more than one package to be listed, this can be shortened to:

```
\documentclass[twoside]{article}
\usepackage[dvips]{graphics,color}
```

In addition, packages will also use each option given to the `\documentclass` command (if they know what to do with it), so you could also write:

```
\documentclass[twoside,dvips]{article}
\usepackage{graphics,color}
```

Class and package options are covered in more detail in *The LaTeX Companion* and in *LaTeX 2ε for Class and Package Writers*.

2.3 Standard classes

The following classes are distributed with LaTeX:

article The `article` class described in *LaTeX: A Document Preparation System.*

book The `book` class described in *LaTeX: A Document Preparation System.*

report The `report` class described in *LaTeX: A Document Preparation System.*

letter The `letter` class described in *LaTeX: A Document Preparation System.*

slides The `slides` class described in *LaTeX: A Document Preparation System*, formerly SLITEX.

7

proc A document class for proceedings, based on `article`. Formerly the `proc` package.

ltxdoc The document class for documenting the LaTeX program, based on `article`.

ltxguide The document class for *LaTeX 2ε for Authors* and *LaTeX 2ε for Class and Package Writers*, based on `article`. The document you are reading now uses the `ltxguide` class. The layout for this class is likely to change in future releases of LaTeX.

ltnews The document class for the *LaTeX News* information sheet, based on `article`. The layout for this class is likely to change in future releases of LaTeX.

minimal This class is the bare minimum (3 lines) that is needed in a LaTeX class file. It just sets the text width and height, and defines `\normalsize`. It is principally intended for debugging and testing LaTeX code in situations where you do not need to load a 'full' class such as `article`. If, however, you are designing a completely new class that is aimed for documents with structure radically different from the structure supplied by the article class, then it may make sense to use this as a base and add to it code implementing the required structure, rather than starting from `article` and modifying the code there.

新的特色
1995/12/01

2.4 Standard packages

The following packages are distributed with LaTeX:

alltt This package provides the `alltt` environment, which is like the `verbatim` environment except that \, {, and } have their usual meanings. It is described in `alltt.dtx` and *LaTeX: A Document Preparation System.*

新的特色
1994/12/01

doc This is the basic package for typesetting the documentation of LaTeX programs. It is described in `doc.dtx` and in *The LaTeX Companion.*

exscale This provides scaled versions of the math extension font. It is described in `exscale.dtx` and *The LaTeX Companion.*

fontenc This is used to specify which font encoding LaTeX should use. It is described in `ltoutenc.dtx`.

graphpap This package defines the `\graphpaper` command; this can be used in a `picture` environment.

ifthen Provides commands of the form 'if...then do...otherwise do...'. It is described in `ifthen.dtx` and *The LaTeX Companion*.

inputenc This is used to specify which input encoding LaTeX should use. It is described in `inputenc.dtx`.

latexsym LaTeX 2ε no longer loads the LaTeX symbol font by default. To access it, you should use the `latexsym` package. It is described in `latexsym.dtx` and in *The LaTeX Companion*; see also Section 6.

makeidx This provides commands for producing indexes. It is described in *LaTeX: A Document Preparation System* and in *The LaTeX Companion*.

newlfont This is used to emulate the font commands of LaTeX 2.09 with the New Font Selection Scheme. It is described in *The LaTeX Companion*.

oldlfont This is used to emulate the font commands of LaTeX 2.09. It is described in *The LaTeX Companion*.

showidx This causes the argument of each `\index` command to be printed on the page where it occurs. It is described in *LaTeX: A Document Preparation System*.

syntonly This is used to process a document without typesetting it. It is described in `syntonly.dtx` and in *The LaTeX Companion*.

tracefnt This allows you to control how much information about LaTeX's font loading is displayed. It is described in *The LaTeX Companion*.

2.5 Related software

The following software should be available from the same distributor as your copy of LaTeX 2ε. You should obtain at least the `graphics` and `tools` collections in order to have all the files described in *LaTeX: A Document Preparation System*. The `amsmath` package (part of `amslatex` and formerly known as `amstex`) and `babel` are also mentioned in the list of 'standard packages' in section C.5.2 of that book.

amslatex Advanced mathematical typesetting from the American Mathematical Society. This includes the `amsmath` package; it provides many commands for typesetting mathematical formulas of higher complexity. It is produced and supported by the American Mathematical Society and it is described in *The LaTeX Companion.*

babel This package and related files support typesetting in many languages. It is described in *The LaTeX Companion.*

cyrillic Everything you need (except the fonts themselves) for typesetting with Cyrillic fonts.

新的特色
1998/12/01

graphics This includes the `graphics` package which provides support for the inclusion and transformation of graphics, including files produced by other software. Also included, is the `color` package which provides support for typesetting in colour. Both these packages are described in *LaTeX: A Document Preparation System.*

psnfss Everything you need (except the fonts themselves) for typesetting with a large range of Type 1 (PostScript) fonts.

tools Miscellaneous packages written by the LaTeX3 project team.

These packages come with documentation and each of them is also described in at least one of the books *The LaTeX Companion* and *LaTeX: A Document Preparation System.*

2.5.1 Tools

This collection of packages includes, at least, the following (some files may have slightly different names on certain systems):

array Extended versions of the environments `array`, `tabular` and `tabular*`, with many extra features.

calc Enables the use of certain algebraic notation when specifying values for lengths and counters.

新的特色
1996/12/01

dcolumn Alignment on 'decimal points' in tabular entries. Requires the `array` package.

delarray Adds 'large delimiters' around arrays. Requires `array`.

hhline Finer control over horizontal rules in tables. Requires `array`.

longtable Multi-page tables. (Does not require `array`, but it uses the extended features if both are loaded.)

tabularx Defines a `tabularx` environment that is similar to `tabular*` but it modifies the column widths, rather than the inter-column space, to achieve the desired table width.

afterpage Place text after the current page.

bm Access bold math symbols.

enumerate Extended version of the `enumerate` environment.

fontsmpl Package and test file for producing 'font samples'.

ftnright Place all footnotes in the right-hand column in two-column mode.

indentfirst Indent the first paragraph of sections, etc.

layout Show the page layout defined by the current document class.

multicol Typeset text in columns, with the length of the columns 'balanced'.

rawfonts Preload fonts using the old internal font names of LaTeX 2.09. See Section 6.2.

somedefs Selective handling of package options. (Used by the rawfonts package.)

showkeys Prints the 'keys' used by `\label`, `\ref`, `\cite` etc.; useful whilst drafting.

theorem Flexible declaration of 'theorem-like' environments.

varioref 'Smart' handling of page references.

verbatim Flexible extension of the verbatim environment.

xr Cross reference other 'external' documents.

xspace 'Smart space' command that helps you to avoid the common mistake of missing spaces after command names.

3 Commands

This section describes the new commands available in LaTeX 2ε. They are covered in more detail in *LaTeX: A Document Preparation System* and in *The LaTeX Companion.*

3.1 Preamble commands

The changes to the preamble commands are intentionally designed to make LaTeX 2ε documents look clearly different from old documents. The commands should be used only before `\begin{document}`.

`\documentclass [⟨`*option-list*`⟩] {⟨`*class-name*`⟩} [⟨`*release-date*`⟩]`

This command replaces the LaTeX 2.09 command `\documentstyle`.

There must be exactly one `\documentclass` command in a document; and it should normally come before any other command. (There are some exceptions, e.g., you can have `filecontents` environments before it or `\RequirePackage` but these should be only used in special scenarios as discussed elsewhere.)

The ⟨*option-list*⟩ is a list of options, each of which may modify the formatting of elements which are defined in the ⟨*class-name*⟩ file, as well as those in all following `\usepackage` commands (see below).

The optional argument ⟨*release-date*⟩ can be used to specify the earliest desired release date of the class file; it should contain a date in the format YYYY/MM/DD. If a version of the class older than this date is found, a warning is issued.

For example, to specify a two-column article, using a version of `article.cls` released after June 1994, you specify:

`\documentclass[twocolumn]{article}[1994/06/01]`

`\documentstyle [⟨`*option-list*`⟩] {⟨`*class-name*`⟩}`

This command is still supported for compatibility with old files. It is essentially the same as `\documentclass` except that it invokes *LaTeX 2.09 compatibility mode.* It also causes any options in the ⟨*option-list*⟩ that are not processed

12

by the class file to be loaded as packages after the class has been loaded. See Section 4 for more details on LaTeX 2.09 compatibility mode.

`\usepackage` [⟨*option-list*⟩] {⟨*package-name*⟩} [⟨*release-date*⟩]

Any number of `\usepackage` commands is allowed. Each package file (as denoted by ⟨*package-name*⟩) defines new elements (or modifies those defined in the class file loaded by the ⟨*class-name*⟩ argument of the `\documentclass` command). A package file thus extends the range of documents which can be processed.

The ⟨*option-list*⟩ argument can contain a list of options, each of which can modify the formatting of elements which are defined in this ⟨*package-name*⟩ file.

As above, ⟨*release-date*⟩ can contain the earliest desired release date of the package file in the format YYYY/MM/DD; if an older version of the package is found, a warning is issued.

For example, to load the `graphics` package for the `dvips` driver, using a version of `graphics.sty` released after June 1994, you write:

```
\usepackage[dvips]{graphics}[1994/06/01]
```

Each package is loaded only once. If the same package is requested more than once, nothing happens in the second or following attempt unless the package has been requested with options that were not given in the original `\usepackage`. If such extra options are specified then an error message is produced. See Section 6 how to resolve this problem.

As well as processing the options given in the ⟨*option-list*⟩ of the `\usepackage` command, each package processes the ⟨*option-list*⟩ of the `\documentclass` command as well. This means that any option which should be processed by every package (to be precise, by every package that specifies an action for it) can be specified just once, in the `\documentclass` command, rather than being repeated for each package that needs it.

`\listfiles`

If this command is placed in the preamble then a list of the files read in (as a result of processing the document) will be displayed on the terminal (and in

the log file) at the end of the run. Where possible, a short description will also be produced.

Warning: this command will list only files which were read using LaTeX commands such as `\input{⟨file⟩}` or `\include{⟨file⟩}`. If the file was read using the primitive TeX syntax `\input` *file* (without `{ }` braces around the file name) then it will not be listed; failure to use the LaTeX form with the braces can cause more severe problems, possibly leading to overwriting important files, so **always put in the braces**.

`\setcounter{errorcontextlines} {⟨num⟩}`

TeX 3 introduced a new primitive `\errorcontextlines` which controls the format of error messages. LaTeX 2_ε provides an interface to this through the standard `\setcounter` command. As most LaTeX users do not want to see the internal definitions of LaTeX commands each time they make an error, LaTeX 2_ε sets this to -1 by default.

3.2   Environments to write out support files

Until the LaTeX release in 2019 the `filecontents` environment was restricted to a place before the `\documentclass` command. These days it can be used anywhere, though we still think that in most cases it is best to only use it a the top of your document or in the preamble.

`\begin{filecontents} [⟨option-list⟩] {⟨file-name⟩}`
`⟨file-contents⟩`
`\end{filecontents}`

The `filecontents` environment is intended for bundling within a single document file the contents of packages, options, or other files. When the document file is run through LaTeX 2_ε the body of this environment is written verbatim (preceded by a comment line) to a file whose name is given as the environment's only argument. However, if that file already exists then nothing happens except for an information message.

These days most UTF-8 text characters can be used in a `filecontents` environment—they will be written unchanged to the output file. However, tabs

and form feeds produce a warning, explaining that they are turned into spaces or blank lines, respectively.

By default the environment does not overwrite an existing file and it even refuses to write out the data if there exists a file that is anywhere in the path that TeX searches when inputting files. With the option `nosearch` you can ask it to look only into the current directory and with the option `overwrite` (or `force`) you can request it to write the file regardless. It will, however, never write to `\jobname.tex` to avoid overwriting itself.

The `filecontents` environment is used for including LaTeX files. For other plain text files (such as Encapsulated PostScript files), you should use the `filecontents*` environment which does not add a comment line.

3.3   Document structure

The `book` document class introduces new commands to indicate document structure.

```
\frontmatter
\mainmatter
\backmatter
```

These commands indicate the beginning of the front matter (title page, table of contents and prefaces), main matter (main text) and back matter (bibliography, indexes and colophon).

3.4   Definitions

In LaTeX, commands can have both mandatory and optional arguments, for example in:

```
\documentclass[11pt]{article}
```

the `11pt` argument is optional, whereas the `article` class name is mandatory.

In LaTeX 2.09 users could define commands with arguments, but these had to be mandatory arguments. With LaTeX 2_ε, users can now define commands and environments which also have one optional argument.

```
\newcommand {⟨cmd⟩} [⟨num⟩] [⟨default⟩] {⟨definition⟩}
\newcommand* {⟨cmd⟩} [⟨num⟩] [⟨default⟩] {⟨definition⟩}
\renewcommand {⟨cmd⟩} [⟨num⟩] [⟨default⟩] {⟨definition⟩}
\renewcommand* {⟨cmd⟩} [⟨num⟩] [⟨default⟩] {⟨definition⟩}
```

These commands have a new, second, optional argument; this is used for defining commands which themselves take one optional argument. This new argument is best introduced by means of a simple (and hence not very practical) example:

```
\newcommand{\example}[2][YYY]{Mandatory arg: #2;
                             Optional arg: #1.}
```

This defines `\example` to be a command with two arguments, referred to as `#1` and `#2` in the `{⟨definition⟩}`—nothing new so far. But by adding a second optional argument to this `\newcommand` (the `[YYY]`) the first argument (`#1`) of the newly defined command `\example` is made optional with its default value being `YYY`.

Thus the usage of `\example` is either:

```
\example{BBB}
```

which prints:

```
    Mandatory arg: BBB; Optional arg: YYY.
```

or:

```
\example[XXX]{AAA}
```

which prints:

```
    Mandatory arg: AAA; Optional arg: XXX.
```

The default value of the optional argument is `YYY`. This value is specified as the `[⟨default⟩]` argument of the `\newcommand` that created `\example`.

As another more useful example, the definition:

```
\newcommand{\seq}[2][n]{\lbrace #2_{0},\ldots,\,#2_{#1} \rbrace}
```

means that the input `$\seq{a}$` produces the formula $\{a_0, \ldots, a_n\}$, whereas the input `$\seq[k]{x}$` produces the formula $\{x_0, \ldots, x_k\}$.

In summary, the command:

\newcommand {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*definition*⟩}

defines ⟨*cmd*⟩ to be a command with ⟨*num*⟩ arguments, the first of which is optional and has default value ⟨*default*⟩.

Note that there can only be one optional argument but, as before, there can be up to nine arguments in total.

\newenvironment {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*beg-def*⟩} {⟨*end-def*⟩}
\newenvironment* {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*beg-def*⟩} {⟨*end-def*⟩}
\renewenvironment {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*beg-def*⟩} {⟨*end-def*⟩}
\renewenvironment* {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*beg-def*⟩} {⟨*end-def*⟩}

LaTeX 2_ε also supports the creation of environments that have one optional argument. Thus the syntax of these two commands has been extended in the same way as that of \newcommand.

\providecommand {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*definition*⟩}
\providecommand* {⟨*cmd*⟩} [⟨*num*⟩] [⟨*default*⟩] {⟨*definition*⟩}

This takes the same arguments as \newcommand. If ⟨*cmd*⟩ is already defined then the existing definition is kept; but if it is currently undefined then the effect of \providecommand is to define ⟨*cmd*⟩ just as if \newcommand had been used.

All the above five 'defining commands' now have *-forms that are usually the better form to use when defining commands with arguments, unless any of these arguments is intended to contain whole paragraphs of text. Moreover, if you ever do find yourself needing to use the non-star form then you should ask whether that argument would not better be treated as the contents of a suitably defined environment.

新的特色
1994/12/01

The commands produced by the above five 'defining commands' are now robust.

新的特色
1995/12/01

3.5 Boxes

These next three commands for making LR-boxes all existed in LaTeX 2.09.
They have been enhanced in two ways.

```
\makebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}
\framebox [⟨width⟩] [⟨pos⟩] {⟨text⟩}
\savebox {⟨cmd⟩} [⟨width⟩] [⟨pos⟩] {⟨text⟩}
```

One small but far-reaching change for LaTeX 2_ε is that, within the ⟨*width*⟩
argument only, four special lengths can be used. These are all dimensions of the
box that would be produced by using simply `\mbox{`⟨*text*⟩`}`:

`\height`   its height above the baseline;

`\depth`   its depth below the baseline;

`\totalheight`   the sum of `\height` and `\depth`;

`\width`   its width.

Thus, to put 'hello' in the centre of a box of twice its natural width, you would
use:

```
\makebox[2\width]{hello}
```

Or you could put *f* into a square box, like this: $\boxed{f}$

```
\framebox{\makebox[\totalheight]{\itshape f\/}}
```

Note that it is the total width of the framed box, including the frame, which is
set to `\totalheight`.

The other change is a new possibility for ⟨*pos*⟩: `s` has been added to `l` and
`r`. If ⟨*pos*⟩ is `s` then the text is stretched the full length of the box, making use
of any 'rubber lengths' (including any inter-word spaces) in the contents of the
box. If no such 'rubber length' is present, an 'underfull box' will probably be
produced.

```
\parbox [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩} {⟨text⟩}
\begin{minipage} [⟨pos⟩] [⟨height⟩] [⟨inner-pos⟩] {⟨width⟩}
⟨text⟩
\end{minipage}
```

As for the box commands above, `\height`, `\width`, etc. may be used in the [⟨*height*⟩] argument to denote the natural dimensions of the box.

The ⟨*inner-pos*⟩ argument is new in LaTeX 2ε. It is the vertical equivalent to the ⟨*pos*⟩ argument for `\makebox`, etc, determining the position of ⟨*text*⟩ within the box. The ⟨*inner-pos*⟩ may be any one of `t`, `b`, `c`, or `s`, denoting top, bottom, centered, or 'stretched' alignment respectively. When the ⟨*inner-pos*⟩ argument is not specified, LaTeX gives it same value as ⟨*pos*⟩ (this could be the latter's default value).

```
\begin{lrbox} {⟨cmd⟩}
⟨text⟩
\end{lrbox}
```

This is an environment which does not directly print anything. Its effect is to save the typeset ⟨*text*⟩ in the bin ⟨*cmd*⟩. Thus it is like `\sbox {⟨cmd⟩}` `{⟨text⟩}`, except that any white space before or after the contents ⟨*text*⟩ is ignored.

This is very useful as it enables both the `\verb` command and the `verbatim` environment to be used within ⟨*text*⟩.

It also makes it possible to define, for example, a 'framed box' environment. This is done by first using this environment to save some text in a bin ⟨*cmd*⟩ and then calling `\fbox{\usebox{⟨cmd⟩}}`.

The following example defines an environment, called `fmpage`, that is a framed version of `minipage`.

```
\newsavebox{\fmbox}
\newenvironment{fmpage}[1]
  {\begin{lrbox}{\fmbox}\begin{minipage}{#1}}
  {\end{minipage}\end{lrbox}\fbox{\usebox{\fmbox}}}
```

3.6 Measuring things

The first of these next commands was in LaTeX 2.09. The two new commands are the obvious analogues.

```
\settowidth {⟨length-cmd⟩} {⟨lr text⟩}
\settoheight {⟨length-cmd⟩} {⟨lr text⟩}
\settodepth {⟨length-cmd⟩} {⟨lr text⟩}
```

3.7 Line endings

The command \\, which is used to indicate a line-end in various places, is now a robust command when used within arguments such as section titles.

新的说明
1994/12/01

Also, because it is often necessary to distinguish which type of line is to be ended, we have introduced the following new command; it has the same argument syntax as that of \\.

```
\tabularnewline [⟨vertical-space⟩]
```

新的特色
1994/12/01

One example of its use is when the text in the last column of a `tabular` environment is set with `\raggedright`; then `\tabularnewline` can be used to indicate the end of a row of the `tabular`, whilst \\ will indicate the end of a line of text in a paragraph within the column. This command can be used in the `array` environment as well as `tabular`, and also the extended versions of these environments offered by the array and longtable packages in the tools collection.

3.8 Controlling page breaks

Sometimes it is necessary, for a final version of a document, to 'help' LaTeX break the pages in the best way. LaTeX 2.09 had a variety of commands for this situation: `\clearpage`, `\pagebreak` etc. LaTeX 2ε provides, in addition, commands which can produce longer pages as well as shorter ones.

```
\enlargethispage {⟨size⟩}
\enlargethispage* {⟨size⟩}
```

These commands increase the height of a page (from its normal value of \texttheight) by the specified amount ⟨*size*⟩, a rigid length. This change affects *only* the current page.

This can be used, for example, to allow an extra line to be fitted onto the page or, with a negative length, to produce a page shorter than normal.

The star form also shrinks any vertical white space on the page as much as possible, so as to fit the maximum amount of text on the page.

These commands do not change the position of the footer text; thus, if a page is lengthened too far, the main text may overprint the footer.

新的说明
1995/12/01

3.9 Floats

There is a new command, \suppressfloats, and a new 'float specifier'. These will enable people to gain better control of LaTeX's float placement algorithm.

\suppressfloats [⟨*placement*⟩]

This command stops any further floating environments from being placed on the current page. With an optional argument, which should be either t or b (not both), this restriction applies only to putting further floats at the top or at the bottom. Any floats which would normally be placed on this page are placed on the next page instead.

The extra float location specifier: !

This can be used, along with at least one of h, t, b and p, in the location optional argument of a float.

If a ! is present then, just for this particular float, whenever it is processed by the float mechanism the following are ignored:

- all restrictions on the number of floats which can appear;

- all explicit restrictions on the amount of space on a text page which may be occupied by floats or must be occupied by text.

The mechanism will, however, still attempt to ensure that pages are not overfull and that floats of the same type are printed in the correct order.

Note that its presence has no effect on the production of float pages.

A ! specifier overrides the effect of any `\suppressfloats` command for this particular float.

3.10   Font changing: text

The font selection scheme used in LaTeX 2_ε differs a lot from that used in LaTeX 2.09. In this section, we give a brief description of the new commands. A more detailed description with examples is given in *The LaTeX Companion*, and the interface for class- and package-writers is described in *LaTeX 2_ε Font Selection*.

```
\normalfont
\rmfamily
\sffamily
\ttfamily
\mdseries
\bfseries
\upshape
\itshape
\slshape
\scshape
```

These are font commands whose use is the same as the commands `\rm`, `\bf`, etc. The difference is that each command changes just one attribute of the font (the attribute changed is part of the name). One result of this is that, for example, `\bfseries\itshape` produces both a change of series and a change of shape, to give a bold italic font.

```
\textnormal{⟨text⟩}
\textrm{⟨text⟩}
\textsf{⟨text⟩}
\texttt{⟨text⟩}
\textmd{⟨text⟩}
\textbf{⟨text⟩}
\textup{⟨text⟩}
\textit{⟨text⟩}
\textsl{⟨text⟩}
\textsc{⟨text⟩}
\emph{⟨text⟩}
```

These are one-argument commands; they take as an argument the text which is to be typeset in the particular font. They also automatically insert italic corrections where appropriate; if you do not like the result, you can add an italic correction with `\/` or remove it with `\nocorr`. The `\nocorr` should always be the first or last thing within the `{⟨text⟩}` argument.

3.11   Font changing: math

Most of the fonts used within math mode do not need to be explicitly invoked; but to use letters from a range of fonts, the following class of commands is provided.

```
\mathrm {⟨letters⟩}
\mathnormal {⟨letters⟩}
\mathcal {⟨letters⟩}
\mathbf {⟨letters⟩}
\mathsf {⟨letters⟩}
\mathtt {⟨letters⟩}
\mathit {⟨letters⟩}
```

These are also one-argument commands which take as an argument the letters which are to be typeset in the particular font. The argument is processed in math mode so spaces within it will be ignored. Only letters, digits and accents have their font changed, for example `$\mathbf{\tilde A \times 1}$` produces $\tilde{\mathbf{A}} \times \mathbf{1}$.

3.12 Ensuring math mode

<div style="border:1px solid blue; display:inline-block; padding:4px">

\ensuremath {⟨*math commands*⟩}

</div>

In LaTeX 2.09, if you wanted a command to work both in math mode and in text mode, the suggested method was to define something like:

```
\newcommand{\Gp}{\mbox{$G_p$}}
```

Unfortunately, the `\mbox` stops `\Gp` changing size correctly in (for instance) subscripts or a fraction.

In LaTeX 2_ε you can define it thus:

```
\newcommand{\Gp}{\ensuremath{G_p}}
```

Now `\Gp` will work correctly in all contexts.

This is because the `\ensuremath` does nothing, producing simply `G_p`, when `\Gp` is used within math mode; but it ensures that math mode is entered (and exited) as required when `\Gp` is used in text mode.

3.13 Setting text superscripts

<div style="border:1px solid blue; display:inline-block; padding:4px">

\textsuperscript {⟨*text*⟩}

</div>

In LaTeX 2.09 textual superscripts such as footnote markers were produced by internally entering math mode and typesetting the number as a math superscript. This normally looked fine since the digits in math fonts are the same as those in text fonts when Computer Modern fonts are used. But when a different document font (such as Times) is selected, the results look rather strange. For this reason the command `\textsuperscript` has been introduced which typesets its argument in the current text font, in a superscript position and in the correct size.

新的特色
1995/06/01

24

3.14　Text commands: all encodings

One of the main differences between LaTeX 2_ε and LaTeX 2.09 is that LaTeX 2_ε can deal with fonts in arbitrary *encodings*. (A font encoding is the sequence of characters in the font—for example a Cyrillic font would have a different encoding from a Greek font.)

The two major font encodings that are used for Latin languages such as English or German are `OT1` (Donald Knuth's 7-bit encoding, which has been used during most of TeX's lifetime) and `T1` (the new 8-bit 'Cork' encoding).

LaTeX 2.09 only supported the `OT1` encoding, whereas LaTeX 2_ε has support for both `OT1` and `T1` built-in. The next section will cover the new commands which are available if you have `T1`-encoded fonts. This section describes new commands which are available in all encodings.

Most of these commands provide characters which were available in LaTeX 2.09 already. For example `\textemdash` gives an 'em dash', which was available in LaTeX 2.09 by typing `---`. However, some fonts (for example a Greek font) may not have the `---` ligature, but you will still be able to access an em dash by typing `\textemdash`.

`\r{`⟨*text*⟩`}`

This command gives a 'ring' accent, for example 'o̊' can be typed `\r{o}`.

`\SS`

This command produces a German 'SS', that is a capital 'ß'. This letter can hyphenate differently from 'SS', so is needed for entering all-caps German.

`\textcircled{`⟨*text*⟩`}`

This command is used to build 'circled characters' such as `\copyright`. For example `\textcircled{a}` produces ⓐ.

`\textcompwordmark`

This command is used to separate letters which would normally ligature. For example 'fi' is produced with `f\textcompwordmark i`. Note that the 'f' and

'i' have not ligatured to produce 'fi'. This is rarely useful in English ('shelfful' is a rare example of where it might be used) but is used in languages such as German.

```
\textvisiblespace
```

This command produces a 'visible space' character '␣'. This is sometimes used in computer listings, for example 'type hello␣world'.

```
\textemdash \textendash \textexclamdown \textquestiondown
\textquotedblleft \textquotedblright \textquoteleft \textquoteright
```

These commands produce characters which would otherwise be accessed via ligatures:

ligature	character	command
---	—	\textemdash
--	–	\textendash
!`	¡	\textexclamdown
?`	¿	\textquestiondown
``	"	\textquotedblleft
''	"	\textquotedblright
`	'	\textquoteleft
'	'	\textquoteright

The reason for making these characters directly accessible is so that they will work in encodings which do not have these characters.

```
\textbullet \textperiodcentered
```

These commands allow access to characters which were previously only available in math mode:

math command	character	text command
\bullet	•	\textbullet
\cdot	·	\textperiodcentered

```
\textbackslash \textbar \textless \textgreater
```

These commands allow access to ASCII characters which were only available in verbatim or math mode:

math command	character	text command
\backslash	\	\textbackslash
\mid	|	\textbar
<	<	\textless
>	>	\textgreater

\textasciicircum \textasciitilde

These commands allow access to ASCII characters which were previously only available in verbatim:

verbatim	text command
^	\textasciicircum
~	\textasciitilde

\textregistered \texttrademark

These commands provide the 'registered trademark' (R) and 'trademark' (TM) symbols.

3.15   Text commands: the T1 encoding

The `OT1` font encoding is fine for typesetting in English, but has problems when typesetting other languages. The `T1` encoding solves some of these problems, by providing extra characters (such as 'eth' and 'thorn'), and it allows words containing accented letters to be hyphenated (as long as you have a package like `babel` which allows for non-American hyphenation).

This section describes the commands you can use if you have the `T1` fonts. To use them, you need to get the 'ec fonts', or the `T1`-encoded PostScript fonts, as used by psnfss. All these fonts are available by anonymous ftp in the Comprehensive TeX Archive, and are also available on the CD-ROMs *4all TeX* and *TeX Live* (both available from the TeX Users Group).

You can then select the `T1` fonts by saying:

```
\usepackage[T1]{fontenc}
```

This will allow you to use the commands in this section.

Note: Since this document must be processable on any site running an up-to-date LaTeX, it does not contain any characters that are present only in `T1`-encoded fonts. This means that this document cannot show you what these glyphs look like! If you want to see them then run LaTeX on the document `fontsmpl` and respond '`cmr`' when it prompts you for a family name.

```
\k{⟨text⟩}
```

This command produces an 'ogonek' accent.

<div style="text-align: right">新的特色
1994/12/01</div>

```
\DH \DJ \NG \TH \dh \dj \ng \th
```

These commands produce characters 'eth', 'dbar', 'eng', and 'thorn'.

<div style="text-align: right">新的特色
1994/12/01</div>

```
\guillemotleft \guillemotright \guilsinglleft \guilsinglright
\quotedblbase \quotesinglbase \textquotedbl
```

These commands produce various sorts of quotation mark. Rough representations of them are: «a» ‹a› „a" ‚a' and "a".

<div style="text-align: right">新的特色
1994/12/01</div>

There are therefore some extra short-form ligatures available for use in documents that will only be used with `T1`-encoded fonts.

<div style="text-align: right">新的说明
2001/06/01</div>

The guillemets `\guillemotleft` and `\guillemotright`[1] can be obtained by typing `<<` and `>>` and `\quotedblbase` by typing `,,`.

Also, unlike the unexpected results with `OT1`-encoded fonts, `<` and `>` will produce < and >.

Note also that the single character `"` will no longer produce " but rather `\textquotedbl`.

3.16 Logos

---

[1] We apologise once again for maintaining Adobe's enormous solipsism (sic) of confusing a diving bird with punctuation marks!

```
\LaTeX
\LaTeXe
```

\LaTeX (producing 'LATEX') is still the 'main' logo command, but if you need to refer to the new features, you can write \LaTeXe (producing 'LATEX 2_ε').

3.17   Picture commands

```
\qbezier[⟨N⟩](⟨AX⟩,⟨AY⟩)(⟨BX⟩,⟨BY⟩)(⟨CX⟩,⟨CY⟩)
 \bezier{⟨N⟩}(⟨AX⟩,⟨AY⟩)(⟨BX⟩,⟨BY⟩)(⟨CX⟩,⟨CY⟩)
```

The \qbezier command can be used in picture mode to draw a quadratic Bezier curve from position ($\langle AX\rangle$,$\langle AY\rangle$) to ($\langle CX\rangle$,$\langle CY\rangle$) with control point ($\langle BX\rangle$,$\langle BY\rangle$). The optional argument $\langle N\rangle$ gives the number of points on the curve.

For example, the diagram:
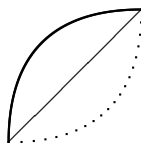
is drawn with:

```
\begin{picture}(50,50)
   \thicklines
   \qbezier(0,0)(0,50)(50,50)
   \qbezier[20](0,0)(50,0)(50,50)
   \thinlines
   \put(0,0){\line(1,1){50}}
\end{picture}
```

The \bezier command is the same, except that the argument $\langle N\rangle$ is not optional. It is provided for compatibility with the LATEX 2.09 bezier document style option.

3.18   Old commands

The `\samepage` command still exists but is no longer being maintained. This is because it only ever worked erratically; it does not guarantee that there will be no page-breaks within its scope; and it can cause footnotes and marginals to be wrongly placed.

We recommend using `\enlargethispage` in conjunction with page-break commands such as `\newpage` and `\pagebreak` to help control page breaks.

Since SLiTEX no longer exists, the logo is no longer defined in the LATEX kernel. A suitable replacement is `\textsc{Sli\TeX}`. The SLiTEX logo is defined in LATEX 2.09 compatibility mode.

These symbols are contained in the LATEX symbol font, which was automatically loaded by LATEX 2.09. However, TEX has room for only sixteen math font families; thus many users discovered that they ran out. Because of this, LATEX does not load the LATEX symbol font unless you use the latexsym package.

These symbols are also made available, using different fonts, by the amsfonts package, which also defines a large number of other symbols. It is supplied by the American Mathematical Society.

The latexsym package is loaded automatically in LATEX 2.09 compatibility mode.

4   LATEX 2.09 documents

LATEX 2_ε can process (almost) any LATEX 2.09 document, by entering *LATEX 2.09 compatibility mode*. Nothing has changed, you run LATEX in the same way you always did, and you will get much the same results.

The reason for the 'almost' is that some LATEX 2.09 packages made use of low-level unsupported features of LATEX. If you discover such a package, you should find out if it has been updated to work with LATEX 2_ε. Most packages

will still work with LaTeX 2ε—the easiest way to find out whether a package still works is to try it!

LaTeX 2.09 compatibility mode is a comprehensive emulation of LaTeX 2.09, but at the cost of time. Documents can run up to 50% slower in compatibility mode than they did under LaTeX 2.09. In addition, many of the new features of LaTeX 2ε are not available in LaTeX 2.09 compatibility mode.

4.1   Warning

This *LaTeX 2.09 compatibility mode* is provided solely to allow you to process 2.09 documents, i.e. documents that were written (we hope, a long time ago) for a very old system and therefore could be processed by using a genuine antique LaTeX 2.09 system.

This mode is therefore *not* intended to provide access to the enhanced features of LaTeX 2ε. Thus it must not be used to process new documents which masquerade as 2.09 documents (i.e. they begin with \documentstyle) but which could not be processed using that genuine antique LaTeX 2.09 system because they contain some new, LaTeX 2ε-only, commands or environments.

To prevent such misuse of the system, and the consequent trouble it causes when such misleadingly encoded documents are distributed, the *LaTeX 2.09 compatibility mode* turns off most of these new features and commands. Any attempt to use them will give you an error message and, moreover, many of them simply will not work, whilst others will produce unpredictable results. So don't bother sending us any bug reports about such occurrences since they are intentional.

4.2   Font selection problems

When using compatibility mode, it is possible that you will find problems with font-changing commands in some old documents. These problems are of two types:

- producing error messages;

- not producing the font changes you expected.

In case of error messages it is possible that the document (or an old style file used therein) contains references to old internal commands which are no longer defined, see Section 6.2 for more information if this is the case.

One example of the unexpected is if you use one of the new style of math-mode font changing command as follows:

```
$ \mathbf{xy} A $
```

You may well find that this behaves as if you had put:

```
$ \bf {xy} A $
```

everything including the *A* coming out bold.

LaTeX 2.09 allowed sites to customize their LaTeX installation, which resulted in documents producing different results on different LaTeX installations. LaTeX 2_ε no longer allows so much customization but, for compatibility with old documents, the local configuration file `latex209.cfg` is loaded every time LaTeX 2_ε enters LaTeX 2.09 compatibility mode.

For example, if your site was customized to use the New Font Selection Scheme (NFSS) with the `oldlfont` option, then you can make LaTeX 2_ε emulate this by creating a `latex209.cfg` file containing the commands:

```
\ExecuteOptions{oldlfont}\RequirePackage{oldlfont}
```

Similarly, to emulate NFSS with the `newlfont` option, you can create a `latex209.cfg` file containing:

```
\ExecuteOptions{newlfont}\RequirePackage{newlfont}
```

4.3 Native mode

To run an old document faster, and use the new features of LaTeX 2_ε, you should try using *LaTeX 2_ε native mode*. This is done by replacing the command:

```
\documentstyle[⟨options⟩,⟨packages⟩]{⟨class⟩}
```

with:

```
\documentclass[⟨options⟩]{⟨class⟩}
\usepackage{latexsym,⟨packages⟩}
```

However, some documents which can be processed in LATEX 2.09 compatibility mode may not work in native mode. Some LATEX 2.09 packages will only work with LATEX 2ε in 2.09 compatibility mode. Some documents will cause errors because of LATEX 2ε's improved error detection abilities.

But most LATEX 2.09 documents can be processed by LATEX 2ε's native mode with the above change. Again, the easiest way to find out whether your documents can be processed in native mode is to try it!

5   Local modifications

There are two common types of local modifications that can be done very simply. Do not forget that documents produced using such modifications will not be usable at other places (such documents are called 'non-portable').

新的说明
1995/12/01

One type of modification is the use of personal commands for commonly used symbols or constructions. These should be put into a package file (for example, one called `mymacros.sty`) and loaded by putting `\usepackage{mymacros}` in the document preamble.

Another type is a local document class that is very similar to one of the standard classes but contains some straightforward modifications such as extra environments, different values for some parameters, etc. These should be put into a class file; here we shall describe a simple method of constructing such a file using, as an example, a class called larticle that is very similar to the article class.

The class file called `larticle.cls` should (after the preliminary identification commands) start as follows:

新的特色
1995/12/01

```
\LoadClassWithOptions{article}
```

This command should be followed by whatever additions and changes you wish to make to the results of reading in the file `article.sty`.

The effect of using the above `\LoadClassWithOptions` command is to load the standard class file article with whatever options are asked for by the document. Thus a document using your larticle class can specify any option that could be specified when using the standard article class; for example:

```
\documentclass[a4paper,twocolumn,dvips]{larticle}
```

6   Problems

This section describes some of the things which may go wrong when using LaTeX 2_ε, and what you can do about it.

6.1   New error messages

LaTeX 2_ε has a number of new error messages. Please also note that many error messages now produce further helpful information if you press h in response to the error prompt.

`Option clash for package ⟨package⟩.`

The named package has been loaded twice with different options. If you enter h you will be told what the options were, for example, if your document contained:

```
\usepackage[foo]{fred}
\usepackage[baz]{fred}
```

then you will get the error message:

```
Option clash for package fred.
```

and typing h at the ? prompt will give you:

```
The package fred has already been loaded with options:
  [foo]
There has now been an attempt to load it with options:
```

```
    [baz]
Adding the line:
  \usepackage[foo,baz]{fred}
to your document may fix this.
Try typing <return> to proceed.
```

The cure is, as suggested, to load the package with both sets of options. Note that since LaTeX packages can call other packages, it is possible to get a package option clash without explicitly requesting the same package twice.

Command ⟨*command*⟩ not provided in base NFSS.

The ⟨*command*⟩ is not provided by default in LaTeX 2ε. This error is generated by using one of the commands:

```
\mho \Join \Box \Diamond \leadsto
\sqsubset \sqsupset \lhd \unlhd \rhd \unrhd
```

which are now part of the latexsym package. The cure is to add:

```
\usepackage{latexsym}
```

in the preamble of your document.

LaTeX2e command ⟨*command*⟩ in LaTeX 2.09 document.

The ⟨*command*⟩ is a LaTeX 2ε command but this is a LaTeX 2.09 document. The cure is to replace the command by a LaTeX 2.09 command, or to run document in native mode, as described in Section 4.3.

NFSS release 1 command \newmathalphabet found.

The command \newmathalphabet was used by the New Font Selection Scheme Release 1 but it has now been replaced by \DeclareMathAlphabet, the use of which is described in *LaTeX 2ε Font Selection*.

The best cure is to update the package which contained the \newmathalphabet command. Find out if there is a new release of the package, or (if you wrote

the package yourself) consult *LATEX 2ε Font Selection* for the new syntax of font commands.

If there is no updated version of the package then you can cure this error by using the newlfont or oldlfont package, which tells LATEX which version of `\newmathalphabet` should be emulated.

You should use oldlfont if the document selects math fonts with syntax such as this:

```
{\cal A}, etc.
```

Use newlfont if the document's syntax is like this:

```
\cal{A}, etc.
```

```
Text for \verb command ended by end of line.
```

The `\verb` command has been begun but not ended on that line. This usually means that you have forgotten to put in the end-character of the `\verb` command.

```
Illegal use of \verb command.
```

The `\verb` command has been used inside the argument of another command. This has never been allowed in LATEX—often producing incorrect output without any warning—and so LATEX 2ε produces an error message.

6.2   Old internal commands

A number of LATEX 2.09 internal commands have been removed, since their functionality is now provided in a different way. See *LATEX 2ε for Class and Package Writers* for more details of the new, supported interface for class and package writers.

```
\tenrm \elvrm \twlrm …
\tenbf \elvbf \twlbf …
\tensf \elvsf \twlsf …
⋮
```

These commands provided access to the seventy fonts preloaded by LaTeX 2.09. In contrast, LaTeX 2_ε normally preloads at most fourteen fonts, which saves a lot of font memory; but a consequence is that any LaTeX file which used the above commands to directly access fonts will no longer work.

Their use will usually produce an error message such as:

```
! Undefined control sequence.
l.5 \tenrm
```

The cure for this is to update the document to use the new font-changing commands provided by LaTeX 2_ε; these are described in *LaTeX 2_ε Font Selection*.

If this is not possible then, as a last resort, you can use the rawfonts package, which loads the seventy LaTeX 2.09 fonts and provides direct access to them using the old commands. This takes both time and memory. If you do not wish to load all seventy fonts, you can select some of them by using the `only` option to rawfonts. For example, to load only `tenrm` and `tenbf` you write:

```
\usepackage[only,tenrm,tenbf]{rawfonts}
```

The rawfonts package is distributed as part of the LaTeX tools software, see Section 2.4.

6.3   Old files

One of the more common mistakes in running LaTeX is to read in old versions of packages instead of the new versions. If you get an incomprehensible error message from a standard package, make sure you are loading the most recent version of the package. You can find out which version of the package has been loaded by looking in the log file for a line like:

```
Package: fred 1994/06/01 v0.01 Fred's package.
```

You can use the ⟨*release-date*⟩ options to `\documentclass` and `\usepackage` to make sure that you are getting a suitably recent copy of the document class or package. This is useful when sending a document to another site, which may have out-of-date software.

6.4 Where to go for more help

If you can't find the answer for your problem here, try looking in *LaTeX: A Document Preparation System* or *The LaTeX Companion*. If you have a problem with installing LaTeX, look in the installation guide files which come with the distribution.

If this doesn't help, contact your local LaTeX guru or local LaTeX mailing list.

If you think you've discovered a bug then please report it! First, you should find out if the problem is with a third-party package or class. If the problem is caused by a package or class other than those listed in Section 2 then please report the problem to the author of the package or class, not to the LaTeX3 project team.

If the bug really is with core LaTeX then you should create a *short*, *self-contained* document which exhibits the problem. You should run a *recent* (less than a year old) version of LaTeX on the file and then run LaTeX on `latexbug.tex`. This will create an error report which you should send, together with the sample document and log file, to the LaTeX bugs address which can be found in the file `latexbug.tex` or `bugs.txt`.

7 Enjoy!

We certainly hope you will enjoy using the new standard LaTeX but, if this is not possible, we hope that you will enjoy success and fulfillment as a result of the documents which it will help you to create.

If you find that the contribution of LaTeX to your life is such that you would like to support the work of the project team, then please read Section 1.2 and discover practical ways to do this.

参考文献

[1] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997.

[2] Michel Goossens and Sebastian Rahtz. *The LaTeX Web Companion*. Addison-Wesley, Reading, Massachusetts, 1999.

[3] Donald E. Knuth. *The TeXbook*. Addison-Wesley, Reading, Massachusetts, 1986. Revised to cover TeX3, 1991.

[4] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.

[5] Frank Mittelbach and Michel Goossens. *The LaTeX Companion second edition*. With Johannes Braams, David Carlisle, and Chris Rowley. Addison-Wesley, Reading, Massachusetts, 2004.