

面向作者的 L^AT_EX— 历史版本

© 版权 2020-2023, L^AT_EX Project Team.

版权所有*

张泓知 翻译

2022 年 08 月 30 日

目 录

1	介绍	2
1.1	L ^A T _E X 2 _ε —全新的 L ^A T _E X 版本	3
1.2	L ^A T _E X 3—L ^A T _E X 的长期未来	3
1.3	概述	4
1.4	更多信息	4
2	类与宏包	5
2.1	什么是类和宏包?	5
2.2	类和宏包选项	6
2.3	标准类	7
2.4	标准宏包	8
2.5	相关软件	9
3	命令	11
3.1	导言区命令	11
3.2	用于写出支持文件的环境	13
3.3	文档结构	14
3.4	定义	14
3.5	盒子	16

*本文件可根据 L^AT_EX 项目公共许可证的条件进行分发和/或修改, 可以选择本许可证的 1.3c 版本或 (自选) 以后的版本 (LPPL v1.3c)。请参阅源文件 `usrguide.tex` 以获取完整详情。

3.6	测量尺寸	18
3.7	行尾处理	18
3.8	控制分页	18
3.9	浮动体	19
3.10	字体更改: 文本	20
3.11	字体更改: 数学模式	21
3.12	确保数学模式	21
3.13	设置文本上标	22
3.14	文本命令: 所有编码	22
3.15	文本命令: T1 编码	24
3.16	Logos	25
3.17	图形命令	26
3.18	老的命令	26
4	L^AT_EX 2.09 文档	27
4.1	警告	27
4.2	字体选择问题	28
4.3	原生模式	29
5	局部修改	29
6	问题	30
6.1	新的错误信息	30
6.2	老的内部命令	32
6.3	老的文件	33
6.4	寻求更多帮助	33
7	尽情享受!	33

1 介绍

欢迎使用 L^AT_EX 2_ε, 这是 L^AT_EX 文档准备系统的新标准版本。

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

因为 L^AT_EX 2_ε 在 1994 年问世, 所以它现在有点成为了历史文档。

1.1 L^AT_EX 2_ε—全新的 L^AT_EX 版本 (已经超过 10 年了)

之前的 L^AT_EX 版本被称为 L^AT_EX 2.09。多年来，为 L^AT_EX 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 L^AT_EX 格式。这包括了“标准 L^AT_EX 2.09”、使用 *New Font Selection Scheme* (NFSS) 构建的 L^AT_EX、S_LT_EX、 $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX 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 L^AT_EX 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 L^AT_EX 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 L^AT_EX 2.09 方言的蔓延。在 L^AT_EX 2_ε 中，“新字体选择方案”是标准的，例如，`amsmath`（原为 $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX 格式）或者 `slides`（原为 S_LT_EX 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 L^AT_EX 3—L^AT_EX 的长期未来

L^AT_EX 2_ε 是对 L^AT_EX 系统全面重新实现的巩固步骤。下一个主要版本的 L^AT_EX 将是 L^AT_EX 3，它将包括对文档设计者和宏包编写者接口的彻底改造。

L^AT_EX 3 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 L^AT_EX 2_ε。因此，从生产和维护 L^AT_EX 2_ε 中获得的经验将对 L^AT_EX 3 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 L^AT_EX 项目基金捐款；这个基金旨在通过资助与当前 L^AT_EX 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 T_EX 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>

L^AT_EX 3 项目在万维网上有它的首页：

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

该页面描述了 L^AT_EX 和 L^AT_EX3 项目，并包含指向其他 L^AT_EX 资源的链接，例如用户指南、T_EX 常见问题解答和 L^AT_EX 错误数据库。

早期涵盖 L^AT_EX3 项目各个方面的文章也可通过 Comprehensive T_EX 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 L^AT_EX 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 L^AT_EX 特性。您应该结合一个 L^AT_EX 的介绍来阅读本文档。

第2节 包含了 L^AT_EX 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 L^AT_EX 一同提供的标准宏包和类。

第3节 描述了 L^AT_EX 2_ε 中作者可以使用的新命令。

第4节 展示了如何用 L^AT_EX 2_ε 处理老的 L^AT_EX 文档。

第6节 包含了处理在运行 L^AT_EX 2_ε 时可能遇到的问题的建议。列出了 L^AT_EX 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 L^AT_EX 的一般介绍，包括 L^AT_EX 2_ε 的新特性，您应该阅读 Leslie Lamport 的著作《[L^AT_EX: A Document Preparation System](#)》[4]。

关于 L^AT_EX 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 L^AT_EX 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 L^AT_EX Graphics Companion](#)》[1] 中找到。

在《[The L^AT_EX Web Companion](#)》[2] 中提供了使用 L^AT_EX 在万维网上发布的解决方案。

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

每份 L^AT_EX 复制品都附带了一些文档文件。每六个月发布的《[L^AT_EX 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《[L^AT_EX 2_ε for Class and Package Writers](#)》描述了写文档类和宏包的新 L^AT_EX 特性；它在 `clsguide.tex` 中。指南《[L^AT_EX 2_ε Font Selection](#)》描述了文档类和宏包作者的 L^AT_EX 字体选择方案；它在 `fntguide.tex` 中。关于在 L^AT_EX 中支持西里尔语言的信息描述在《[Cyrillic languages support in L^AT_EX](#)》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 L^AT_EX 2_ε Sources](#)》。这个非常庞大的文档还包含了 L^AT_EX 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 T_EX 和 L^AT_EX 的信息，请联系您所在地的 T_EX 用户组，或国际 T_EX 用户组（见第 3 页）。

2 类与宏包

本节描述了 L^AT_EX 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？

L^AT_EX 2.09 和 L^AT_EX 2_ε 之间的主要区别在于 `\begin{document}` 之前的命令。

在 L^AT_EX 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：

```
\documentstyle[options]{style}
```

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_ϵ 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[\langle options \rangle]{\langle class \rangle}
\usepackage[\langle options \rangle]{\langle packages \rangle}
```

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_ϵ 仍然理解 \LaTeX 2.09 的 `\documentstyle` 命令。这个命令会使 \LaTeX 2_ϵ 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_ϵ 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_ϵ 配合良好。

2.2 类和宏包选项

在 \LaTeX 2.09 中，只有文档样式才能拥有诸如 `twoside` 或 `draft` 等选项。在 \LaTeX 2_ϵ 中，类和宏包都可以有选项。例如，要指定一个使用 `dvips` 驱动程序的双面文章并包含图形，您可以这样写：

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

宏包可以共享常见的选项。例如，您还可以通过指定以下方式加载 color 宏包：

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

但是因为 \usepackage 允许列出多个宏包，所以这可以简写为：

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

此外，宏包还将使用给 \documentclass 命令的每个选项（如果它们知道如何处理它），因此您也可以这样写：

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

类和宏包选项在《[The L^AT_EX Companion](#)》和《[L^AT_EX 2_ε for Class and Package Writers](#)》中有更详细的介绍。

2.3 标准类

以下类是与 L^AT_EX 一起分发的：

article 在《[L^AT_EX: A Document Preparation System](#)》中描述的 article 类。

book 在《[L^AT_EX: A Document Preparation System](#)》中描述的 book 类。

report 在《[L^AT_EX: A Document Preparation System](#)》中描述的 report 类。

letter 在《[L^AT_EX: A Document Preparation System](#)》中描述的 letter 类。

slides 在《[L^AT_EX: A Document Preparation System](#)》中描述的 slides 类，以前是 S_LT_EX。

proc 基于 article 的用于会议录文档类。以前是 proc 宏包。

ltxdoc 用于文档化 L^AT_EX 程序的文档类，基于 `article`。

ltxguide 《[L^AT_EX 2_ε for Authors](#)》和《[L^AT_EX 2_ε for Class and Package Writers](#)》的文档类，基于 `article`。您正在阅读的文档使用 `ltxguide` 类。这个类的布局可能会在未来的 L^AT_EX 版本中发生变化。

ltnews 《[L^AT_EX News](#)》信息单的文档类，基于 `article`。这个类的布局可能会在未来的 L^AT_EX 版本中发生变化。

minimal 这个类是一个最基本的类（3 行），它是一个 L^AT_EX 类文件所需的最少内容。它只设置了文本的宽度和高度，并定义了 `\normalsize`。主要用于在不需加载像 `article` 这样的“完整”类的情况下调试和测试 L^AT_EX 代码。但是，如果您正在设计一个完全针对与 `article` 类提供的结构根本不同的文档的全新类，那么使用这个作为基础，并添加实现所需结构的代码，可能比从 `article` 开始并修改那里的代码更有意义。

新的特色
1995/12/01

2.4 标准宏包

以下宏包是与 L^AT_EX 一起分发的：

alltt 该宏包提供了 `alltt` 环境，类似于 `verbatim` 环境，但 `\{` 和 `}` 有它们的通常含义。它在 `alltt.dtx` 和《[L^AT_EX: A Document Preparation System](#)》中有描述。

新的特色
1994/12/01

doc 用于排版 L^AT_EX 程序文档的基本宏包。在 `doc.dtx` 和《[The L^AT_EX Companion](#)》中有描述。

exscale 提供了数学扩展字体的缩放版本。在 `exscale.dtx` 和《[The L^AT_EX Companion](#)》中有描述。

fontenc 用于指定 L^AT_EX 应该使用哪种字体编码。在 `ltoutenc.dtx` 中有描述。

graphpap 该宏包定义了 `\graphpaper` 命令；可以在 `picture` 环境中使用。

新的特色
1994/12/01

ifthen 提供形式为“如果……则做……否则做……”的命令。在 `ifthen.dtx` 和《[The L^AT_EX Companion](#)》中有描述。

inputenc 用于指定 L^AT_EX 应该使用哪种输入编码。在 `inputenc.dtx` 中有描述。

新的特色
1994/12/01

latexsym L^AT_EX 2_ε 不再默认加载 L^AT_EX 符号字体。要访问它，应该使用 latexsym 宏包。在 latexsym.dtx 和 《[The L^AT_EX Companion](#)》中有描述；另见第 6 节。

makeidx 提供用于生成索引的命令。在 《[L^AT_EX: A Document Preparation System](#)》和 《[The L^AT_EX Companion](#)》中有描述。

newlfont 用于用新字体选择方案模拟 L^AT_EX 2.09 的字体命令。在 《[The L^AT_EX Companion](#)》中有描述。

oldlfont 用于模拟 L^AT_EX 2.09 的字体命令。在 《[The L^AT_EX Companion](#)》中有描述。

showidx 导致每个 \index 命令的参数都打印在它出现的页面上。在 《[L^AT_EX: A Document Preparation System](#)》中有描述。

syntonly 用于处理文档而不对其进行排版。在 syntonly.dtx 和 《[The L^AT_EX Companion](#)》中有描述。

tracefmt 允许您控制显示有关 L^AT_EX 字体加载的信息的程度。在 《[The L^AT_EX Companion](#)》中有描述。

2.5 相关软件

以下软件应该从与您的 L^AT_EX 2_ε 副本相同的发行商处获取。为了拥有 《[L^AT_EX: A Document Preparation System](#)》中描述的所有文件，您应该至少获取 graphics 和 tools 集合。amsmath 宏包（amslatex 的一部分，以前称为 amstex）和 babel 也在该书的第 C.5.2 节的“标准宏包”列表中提到。

新的说明
1998/12/01

amslatex 美国数学学会提供的高级数学排版。其中包括 amsmath 宏包；它提供了许多用于排版更复杂数学公式的命令。由美国数学学会制作和支持，描述在 《[The L^AT_EX Companion](#)》中。

babel 此宏包及相关文件支持多种语言的排版。在 《[The L^AT_EX Companion](#)》中有描述。

cyrillic 使用西里尔字体排版所需的一切（除了字体本身）。

新的特色
1998/12/01

graphics 包含了 graphics 宏包，提供对图形的包含和转换的支持，包括其他软件生成的文件。还包括 color 宏包，提供彩色排版的支持。这两个宏包在 《[L^AT_EX: A Document Preparation System](#)》中有描述。

psnfss 使用大范围 Type 1 (PostScript) 字体排版所需的一切 (除了字体本身)。

tools L^AT_EX3 项目团队编写的各种杂项宏包。

这些宏包附带有文档, 并且每个宏包至少在 《[The L^AT_EX Companion](#)》和 《[L^AT_EX: A Document Preparation System](#)》中有介绍。

2.5.1 工具

此套件至少包含以下内容 (某些文件在特定系统上可能具有稍有不同的名称):

array 对 `array`、`tabular` 和 `tabular*` 环境的扩展版本, 具有许多额外功能。

calc 在指定长度和计数器的值时, 启用某些代数表示法。

dcolumn 在表格条目中对齐“小数点”。需要 `array` 宏包。

delarray 在数组周围添加“大定界符”。需要 `array` 宏包。

hhline 对表格中的水平线条有更精细的控制。需要 `array` 宏包。

longtable 多页表格。(不需要 `array`, 但如果同时加载了两者, 则使用了扩展功能。)

tabularx 定义了一个 `tabularx` 环境, 类似于 `tabular*`, 但它修改了列宽, 而不是列间距, 以达到所需的表格宽度。

afterpage 将文本放置在当前页面之后。

bm 访问粗体数学符号。

enumerate `enumerate` 环境的扩展版本。

fontsmpl 用于生成“字体示例”的宏包和测试文件。

ftnright 在双栏模式下将所有脚注放在右列中。

indentfirst 缩进章节等的第一个段落。

layout 显示当前文档类定义的页面布局。

multicol 在多列中排版文本, 并使列的长度“平衡”。

rawfonts 使用 L^AT_EX 2.09 的老的内部字体名称预加载字体。参见第 6.2 节。

新的特色
1996/12/01

somedefs 选择性处理宏包选项。（被 `rawfonts` 宏包使用。）

showkeys 打印由 `\label`、`\ref`、`\cite` 等使用的“键”；在起草时很有用。

theorem “定理类”环境的灵活声明。

varioref 智能处理页面引用。

verbatim `verbatim` 环境的灵活扩展。

xr 交叉引用其他“外部”文档。

xspace “智能空格”命令，可帮助您避免在命令名称后漏掉常见的空格错误。

3 命令

本节描述了 \LaTeX 2_ϵ 中可用的新命令。详细内容可参阅 《[\LaTeX: A Document Preparation System](#)》和 《[The \LaTeX Companion](#)》。

3.1 导言区命令

对导言区命令的更改是有意设计的，旨在使 \LaTeX 2_ϵ 文档在外观上与老的文档明显不同。这些命令应该仅用于 `\begin{document}` 之前。

`\documentclass [<option-list>] {<class-name>} [<release-date>]`

此命令替代了 \LaTeX 2.09 的 `\documentstyle` 命令。

文档中必须有且仅有一个 `\documentclass` 命令，并且通常应该放在其他命令之前。（有一些例外情况，比如，你可以在它之前放置 `filecontents` 环境或 `\RequirePackage` 命令，但这些应该仅在特殊情况下使用，如其他地方所述。）

<option-list> 是一系列选项，每个选项都可以修改 *<class-name>* 文件中定义的元素格式，以及所有后续的 `\usepackage` 命令中定义的元素格式（参见下文）。

可选参数 *<release-date>* 可以用于指定类文件的最早期望发布日期；它应该包含格式为 `YYYY/MM/DD` 的日期。如果找到的类文件版本早于此日期，则会发出警告。

例如，要指定使用在 1994 年 6 月之后发布的版本的两栏文章，你可以写成：

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

`\documentstyle [<option-list>] {<class-name>}`

此命令仍然为了与老的文件兼容而提供支持。它与 `\documentclass` 基本相同，除了它会启用 *L^AT_EX 2.09* 兼容模式。它还会导致未在 *<option-list>* 中处理的任何选项在加载类文件后被作为宏包加载。更多关于 *L^AT_EX 2.09* 兼容模式的细节请参阅第 4 节。

`\usepackage [<option-list>] {<package-name>} [<release-date>]`

任意数量的 `\usepackage` 命令是允许的。每个宏包文件（由 *<package-name>* 表示）定义了新的元素（或修改了由 `\documentclass` 命令加载的 *<class-name>* 参数所定义的元素）。因此，宏包文件扩展了可以处理的文档范围。

<option-list> 参数可以包含一系列选项，每个选项都可以修改此 *<package-name>* 文件中定义的元素格式。

与上述相同，*<release-date>* 可以以格式 *YYYY/MM/DD* 包含宏包文件的最早期望发布日期；如果找到的宏包版本早于此日期，则会发出警告。

例如，要加载 dvips 驱动程序的 `graphics` 宏包，使用的是在 1994 年 6 月之后发布的 `graphics.sty` 版本，你可以写成：

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

每个宏包只加载一次。如果同一个宏包被多次请求，则在第二次或后续尝试中将不会发生任何事情，除非该宏包被请求时给出了原始 `\usepackage` 中没有的选项。如果指定了这样额外的选项，则会产生错误消息。请参见第 6 节如何解决此问题。

除了处理 `\usepackage` 命令的 *<option-list>*，每个宏包还处理 `\documentclass` 命令的 *<option-list>*。这意味着每个宏包都会处理 `\documentclass` 命令中给出的任何应该由每个宏包（准确说，是每个指定了它的动作的宏包）处理的选项，而不是针对每个需要它的宏包重复给出它。

`\listfiles`

如果此命令放置在导言区中，那么在运行结束时，会在终端（以及日志文件）上显示一个读入文件的列表（因为处理文档而读入的文件）。在可能的情况下，还会产生简短的描述。

警告：此命令只会列出使用 L^AT_EX 命令（比如 `\input{file}` 或 `\include{file}`）读取的文件。如果文件是使用原始 T_EX 语法 `\input file`（在文件名周围没有 `{ }` 大括号）读取的，则不会列出它；未使用带有大括号的 L^AT_EX 形式可能会导致更严重的问题，可能会覆盖重要文件，所以**始终加上大括号**。

新的说明
1995/12/01

```
\setcounter{errorcontextlines}{num}
```

T_EX 3 引入了一个名为 `\errorcontextlines` 的新原始命令，它控制错误消息的格式。L^AT_EX 2_ε 通过标准 `\setcounter` 命令提供了对此的接口。由于大多数 L^AT_EX 用户在出现错误时不想看到 L^AT_EX 命令的内部定义，因此 L^AT_EX 2_ε 将此默认设置为 `-1`。

3.2 用于写出支持文件的环境

直到 2019 年的 L^AT_EX 发布版中，`filecontents` 环境被限制在 `\documentclass` 命令之前的位置。如今它可以在任何地方使用，尽管我们仍认为，在大多数情况下，最好只在文档的顶部或导言区使用它。

新的特色
2019

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

`filecontents` 环境旨在将包、选项或其他文件的内容捆绑到单个文档文件中。当文档文件通过 L^AT_EX 2_ε 运行时，此环境的主体部分（前面带有一行注释）会被写入一个名为该环境唯一参数的文件中。然而，如果该文件已经存在，则除了发出信息消息之外，什么也不会发生。

如今，大多数 UTF-8 文本字符都可以在 `filecontents` 环境中使用——它们将不经改动地写入输出文件。但是，制表符和换页符会产生警告，说明它们分别被转换为空格或空行。

默认情况下，该环境不会覆盖现有文件，即使存在路径中 T_EX 在输入文件时搜索的文件。使用 `nosearch` 选项可以要求它仅查找当前目录，并使用 `overwrite`（或 `force`）选项可以请求它写入文件。但是，为了避免覆盖自身，它永远不会写入到 `\jobname.tex`。

`filecontents` 环境用于包含 \LaTeX 文件。对于其他纯文本文件（如 Encapsulated PostScript 文件），应使用 `filecontents*` 环境，它不会添加注释行。

3.3 文档结构

`book` 文档类引入了新的命令来表示文档结构。

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

这些命令表示前言（标题页、目录和前言）、正文（主要文本）和尾声（参考文献、索引和勘误表）的开始。

3.4 定义

在 \LaTeX 中，命令可以同时具有必选和可选参数，例如：

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

中的 `11pt` 参数是可选的，而 `article` 类名是必需的。

在 \LaTeX 2.09 中，用户可以定义带有参数的命令，但这些参数必须是必需的。而在 \LaTeX 2_ε 中，用户现在可以定义具有一个可选参数的命令和环境。

```
\newcommand {\<cmd>} [\<num>] [\<default>] {\<definition>}
\newcommand* {\<cmd>} [\<num>] [\<default>] {\<definition>}
\renewcommand {\<cmd>} [\<num>] [\<default>] {\<definition>}
\renewcommand* {\<cmd>} [\<num>] [\<default>] {\<definition>}
```

这些命令具有新的第二个可选参数；这用于定义自己带有一个可选参数的命令。最好通过一个简单（因此不太实用）的例子来介绍这个新参数：

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

这定义了 `\example` 为一个具有两个参数的命令，在 `{\<definition>}` 中被称为 `#1` 和 `#2`——到目前为止没什么新鲜的。但是通过给这个 `\newcommand` 加上第二个

可选参数 ([YYY])，新定义的 `\example` 的第一个参数 (#1) 变成了可选的，其默认值为 YYY。

因此，`\example` 的使用方式要么是：

```
\example{BBB}
```

它会打印：

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

要么是：

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

它会打印：

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

可选参数的默认值是 YYY。这个值被指定为创建 `\example` 的 `\newcommand` 的 [*default*] 参数。

作为另一个更有用的例子，定义：

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

意味着输入 `\seq{a}` 会产生公式 $\{a_0, \dots, a_n\}$ ，而输入 `$\seq[k]{x}$` 会产生公式 $\{x_0, \dots, x_k\}$ 。

总结一下，命令：

```
\newcommand {<cmd>} [<num>] [<default>] {<definition>}
```

定义了 *<cmd>* 为一个带有 *<num>* 个参数的命令，其中第一个是可选的，并且默认值为 *<default>*。

请注意，只能有一个可选参数，但是与以前一样，总共可以有最多九个参数。

```

\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>}

```

L^AT_EX 2_ε 也支持具有一个可选参数的环境创建。因此，这两个命令的语法与 `\newcommand` 相同被扩展了。

```

\providecommand {<cmd>} [<num>] [<default>] {<definition>}
\providecommand* {<cmd>} [<num>] [<default>] {<definition>}

```

这两个命令接受与 `\newcommand` 相同的参数。如果 `<cmd>` 已定义，则保留现有定义；但如果当前未定义，`\providecommand` 的作用是定义 `<cmd>`，就像使用了 `\newcommand` 一样。

上述五个“定义命令”现在都有带有 `*` 的形式，通常用于定义带有参数的命令，除非某些参数意图包含整段文字。此外，如果你发现自己需要使用非星号形式，则应考虑将该参数视为适当定义环境的内容。

新的特色
1994/12/01

上述五个“定义命令”生成的命令现在是健壮的。

新的特色
1995/12/01

3.5 盒子

下面三个用于制作 LR 盒子的命令都存在于 L^AT_EX 2.09 中。它们已在两个方面进行了增强。

```

\makebox [<width>] [<pos>] {<text>}
\framebox [<width>] [<pos>] {<text>}
\savebox {<cmd>} [<width>] [<pos>] {<text>}

```

对于 L^AT_EX 2_ε 来说，一个小但影响深远的变化是，在 `<width>` 参数中，可以使用四个特殊长度。这些都是由简单地使用 `\mbox{<text>}` 产生的盒子的尺寸：

`\height` 它在基线之上的高度；
`\depth` 它在基线之下的深度；
`\totalheight` `\height` 和 `\depth` 的总和；
`\width` 它的宽度。

因此，要将“hello”放在宽度为其自然宽度两倍的盒子中心，你可以使用：


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

或者你可以将 f 放入一个正方形盒子中，像这样： f

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

请注意，设置为 `\totalheight` 的带框框的盒子的总宽度包括边框。

另一个变化是对于 $\langle pos \rangle$ ：在 `l` 和 `r` 中增加了一个 `s` 的新选择。如果 $\langle pos \rangle$ 是 `s`，则文本将被拉伸到整个盒子的长度，利用盒子内容中的任何“可伸展长度”（包括任何词间距）。如果没有这样的“可伸展长度”，可能会产生“未充满盒子”。

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

与上述盒子命令类似，`\height`、`\width` 等可以用于 $[\langle height \rangle]$ 参数，表示盒子的自然尺寸。

$\langle inner-pos \rangle$ 参数是 L^AT_EX 2_ε 中的新内容。它是对于 `\makebox` 等的 $\langle pos \rangle$ 参数的垂直对应，决定 $\langle text \rangle$ 在盒子内的位置。 $\langle inner-pos \rangle$ 可以是 `t`、`b`、`c` 或 `s` 中的任意一个，分别表示顶部、底部、居中或“拉伸”对齐。当未指定 $\langle inner-pos \rangle$ 参数时，L^AT_EX 将其设为与 $\langle pos \rangle$ 相同的值（这可能是后者的默认值）。

```
\begin{lrbox} {\langle cmd \rangle}
\langle text \rangle
\end{lrbox}
```

这是一个不直接输出任何内容的环境。它的效果是将排版好的 $\langle text \rangle$ 保存在箱子 $\langle cmd \rangle$ 中。因此，它类似于 `\sbox {\langle cmd \rangle} {\langle text \rangle}`，但会忽略 $\langle text \rangle$ 内容前后的任何空格。

这非常有用，因为它使得可以在 $\langle text \rangle$ 中使用 `\verb` 命令和 `verbatim` 环境。

它还使得可以定义例如“带框盒子”环境。首先在此环境中将一些文本保存在箱子 $\langle cmd \rangle$ 中，然后调用 `\fbox{\usebox{\langle cmd \rangle}}`。

以下示例定义了一个名为 `fmpage` 的环境，它是 `minipage` 的带框版本。

```

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

```

3.6 测量尺寸

接下来的这些命令中，第一个命令在 L^AT_EX 2.09 中已存在。而两个新命令则是明显的类比。

```

\settowidth {<length-cmd>} {<lr text>}
\settoheight {<length-cmd>} {<lr text>}
\settodepth {<length-cmd>} {<lr text>}

```

3.7 行尾处理

命令 `\` 用于指示各处的行尾，在诸如章节标题之类的参数中使用，现在是一个健壮的命令。

新的说明
1994/12/01

同时，因为通常需要区分要结束的是哪种类型的行，我们引入了以下新命令；它具有与 `\` 相同的参数语法。

```

\tabularnewline [ <vertical-space> ]

```

新的特色
1994/12/01

其用法之一是，在 `tabular` 环境的最后一列中设置了 `\raggedright` 的文本；这时可以使用 `\tabularnewline` 来指示 `tabular` 的行结束，而 `\` 将指示列内段落文本的行结束。这个命令可以在 `array` 环境中使用，以及由 `array` 和 `longtable` 工具包提供的这些环境的扩展版本中也可以使用。

3.8 控制分页

有时候，为了文档的最终版本，需要“协助” L^AT_EX 以最佳方式分页。在 L^AT_EX 2.09 中有各种命令处理这种情况：`\clearpage`、`\pagebreak` 等。此外，L^AT_EX 2_ε 还提供了能够产生更长页和更短页的命令。

```

\enlargethispage {<size>}
\enlargethispage* {<size>}

```

这些命令通过指定的刚性长度 $\langle size \rangle$ 增加页面的高度（从其正常值 \texttheight ）。这个变化仅影响 当前页面。

例如，可以使用它来允许额外的一行放在页面上，或者使用负长度来生成比正常更短的页面。

星号形式还会尽可能地收缩页面上的垂直空白，以便在页面上容纳最大量的文本。

这些命令不会改变页脚文本的位置；因此，如果页面长度过长，主文本可能会重叠页脚。

新的说明
1995/12/01

3.9 浮动体

有一个新命令 `\suppressfloats` 和一个新的“浮动体说明符”。这些将使人们能够更好地控制 \LaTeX 的浮动体排版算法。

`\suppressfloats [$\langle placement \rangle$]`

此命令阻止在当前页面上进一步放置任何浮动体环境。带有可选参数，应该是 `t` 或 `b`（不能同时有），此限制仅适用于在页面顶部或底部放置进一步的浮动体。任何通常放置在此页面上的浮动体将被放置在下一页上。

额外的浮动体位置说明符： `!`

可以与至少一个 `h`、`t`、`b` 和 `p` 一起使用，在浮动体的位置可选参数中使用。

如果存在 `!`，那么对于此特定浮动体，在其由浮动体机制处理时会忽略以下内容：

- 所有限制出现的浮动体数量；
- 所有明确限制文本页面上可被浮动体占用的空间或必须被文本占用的空间。

然而，该机制仍然会尝试确保页面不会过满，并确保相同类型的浮动体按正确顺序打印。

请注意，它的存在对于生成浮动体页面没有影响。

`!` 说明符会覆盖对此特定浮动体的任何 `\suppressfloats` 命令的影响。

3.10 字体更改：文本

在 \LaTeX 2_ϵ 中使用的字体选择方案与 \LaTeX 2.09 大不相同。在本节中，我们简要描述了新命令。更详细的说明和示例请参阅 《[The \$\text{\LaTeX}\$ Companion](#)》，而针对类和宏包编写者的接口描述请参阅 《 [\$\text{\LaTeX 2}_\epsilon\$ Font Selection](#)》。

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

这些是字体命令，其使用方式与 `\rm`、`\bf` 等命令相同。不同之处在于每个命令仅更改字体的一个属性（更改的属性是命名的一部分）。由此产生的一个结果是，例如，`\bfseries\itshape` 会同时改变字体的系列和形状，得到一个粗体斜体字体。

```
\textnormal{<text>  
\textrm{<text>  
\textsf{<text>  
\texttt{<text>  
\textmd{<text>  
\textbf{<text>  
\textup{<text>  
\textit{<text>  
\textsl{<text>  
\textsc{<text>  
\emph{<text>
```

这些是一参数命令；它们接受一个参数，该参数是要以特定字体排版的文本。它们还会自动插入适当的斜体修正；如果你不喜欢结果，可以使用 `\/` 添加斜体修正，或者使用 `\nocorr` 删除它。`\nocorr` 应该始终是 `{<text>}` 参数中的第一项或最后一项。

3.11 字体更改：数学模式

在数学模式内使用的大多数字体不需要显式调用；但为了使用来自一系列字体的字母，提供了以下命令。

```
\mathrm {\langle letters \rangle}
\mathnormal {\langle letters \rangle}
\mathcal {\langle letters \rangle}
\mathbf {\langle letters \rangle}
\mathsf {\langle letters \rangle}
\mathtt {\langle letters \rangle}
\mathit {\langle letters \rangle}
```

这些也是一参数命令，接受的参数是要以特定字体排版的字母。参数在数学模式下处理，因此其中的空格将被忽略。只有字母、数字和重音会改变其字体，例如 `$\mathbf{\tilde A \times 1}$` 会产生 $\mathbf{\tilde A} \times 1$ 。

3.12 确保数学模式

```
\ensuremath {\langle math commands \rangle}
```

在 L^AT_EX 2.09 中，如果希望命令在数学模式和文本模式下都起作用，建议的方法是定义类似于：

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

不幸的是，`\mbox` 会在上标、下标或分数（例如）中停止 `\Gp` 正确地更改大小。

在 L^AT_EX 2_ε 中，你可以这样定义它：

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

现在 `\Gp` 在所有上下文中都能正确工作了。

这是因为当 `\Gp` 在数学模式中使用，`\ensuremath` 不会产生任何效果，简单地生成 `Gp`；但当 `\Gp` 在文本模式中使用，它确保按需进入（和退出）数学模式。

3.13 设置文本上标

`<text>`

在 \LaTeX 2.09 中，诸如脚注标记之类的文本上标是通过内部进入数学模式，并将数字排版为数学上标来产生的。这在使用计算机现代字体时通常看起来不错，因为数学字体中的数字与文本字体中的相同。但是当选择了不同的文档字体（例如 Times）时，结果看起来可能有些奇怪。因此，引入了命令 `\textsuperscript`，它在当前文本字体中排版其参数，在上标位置以及正确的大小。

新的特色
1995/06/01

3.14 文本命令：所有编码

\LaTeX 2_ε 与 \LaTeX 2.09 的主要区别之一是 \LaTeX 2_ε 能够处理任意的字体编码。（字体编码是字体中的字符序列——例如，西里尔字体的编码会与希腊字体的编码不同。）

新的说明
1994/12/01

用于英语或德语等拉丁语言的两个主要字体编码是 OT1（Donald Knuth 的 7 位编码，在大多数 \TeX 的使用时间内使用）和 T1（新的 8 位“Cork”编码）。

\LaTeX 2.09 仅支持 OT1 编码，而 \LaTeX 2_ε 内置对 OT1 和 T1 的支持。接下来的部分将涵盖如果你使用了 T1 编码字体所拥有的新命令。本节描述的命令可在所有编码中使用。

这些命令中的大多数提供的字符在 \LaTeX 2.09 中已经存在。例如 `\textemdash` 会产生一个“长破折号”，在 \LaTeX 2.09 中可以通过输入 --- 实现。然而，某些字体（例如希腊字体）可能没有 --- 这个连字，但你仍然可以通过输入 `\textemdash` 来访问长破折号。

`\r{<text>}`

这个命令会给出一个“环”重音，例如 ‘ö’ 可以通过 `\r{o}` 输入。

新的特色
1994/12/01

`\SS`

此命令产生德语的大写“SS”，即大写的“ß”。这个字母在连字时可能与“SS”有不同的连字符，因此在输入全大写的德语时需要它。

新的特色
1994/12/01

`\textcircled{<text>}`

新的特色
1994/12/01

此命令用于制作“带圈字符”，例如 `\copyright`。例如 `\textcircled{a}` 产生 ©。

`\textcompwordmark`

新的特色
1994/12/01

此命令用于分隔通常会连字的字母。例如 ‘fi’ 可以通过 `f\textcompwordmark i` 输入。注意，“f”和“i”没有连字生成“fi”。在英语中很少使用这个功能（‘shelfful’ 是可能使用它的罕见示例），但在德语等语言中会用到。

`\textvisiblespace`

新的特色
1994/12/01

此命令产生一个“可见空格”字符 ‘`\textvisiblespace`’。有时在计算机代码中使用，例如 ‘type hello`\textvisiblespace`world’。

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

新的特色
1994/12/01

这些命令产生本来需要通过连字访问的字符：

<i>ligature</i>	<i>character</i>	<i>command</i>
---	—	<code>\textemdash</code>
--	–	<code>\textendash</code>
!`	¡	<code>\textexclamdown</code>
?`	¿	<code>\textquestiondown</code>
``	“	<code>\textquotedblleft</code>
''	”	<code>\textquotedblright</code>
`	‘	<code>\textquoteleft</code>
'	’	<code>\textquoteright</code>

将这些字符直接提供的原因是为了使它们能够在没有这些字符的编码中正常工作。

`\textbullet` `\textperiodcentered`

新的特色
1994/12/01

这些命令允许访问以前只能在数学模式中使用的字符：

<i>math command</i>	<i>character</i>	<i>text command</i>
<code>\bullet</code>	•	<code>\textbullet</code>
<code>\cdot</code>	·	<code>\textperiodcentered</code>

`\textbackslash \textbar \textless \textgreater`

新的特色
1995/12/01

这些命令允许访问以前只能在抄录或数学模式中使用的 ASCII 字符:

<i>math command</i>	<i>character</i>	<i>text command</i>
<code>\backslash</code>	<code>\</code>	<code>\textbackslash</code>
<code>\mid</code>	<code> </code>	<code>\textbar</code>
<code><</code>	<code><</code>	<code>\textless</code>
<code>></code>	<code>></code>	<code>\textgreater</code>

`\textasciicircum \textasciitilde`

新的特色
1995/12/01

这些命令允许访问以前只能在抄录中使用的 ASCII 字符:

<i>verbatim</i>	<i>text command</i>
<code>^</code>	<code>\textasciicircum</code>
<code>~</code>	<code>\textasciitilde</code>

`\textregistered \texttrademark`

新的特色
1995/12/01

这些命令提供了“注册商标”(R)和“商标”(TM)符号。

3.15 文本命令: T1 编码

OT1 字体编码用于英文排版很好,但在排版其他语言时存在问题。T1 编码通过提供额外字符(如“eth”和“thorn”)解决了部分问题,并且允许包含重音字母的单词断字(只要你使用像 babel 这样允许非美式断字的包)。

新的说明
1994/12/01

本节描述了如果你有 T1 字体时可用的命令。要使用它们,你需要获取“ec 字体”,或者 psnfss 使用的 T1 编码的 PostScript 字体。所有这些字体都可以通过 Comprehensive T_EX Archive 的匿名 ftp 获得,也可以在 CD-ROMs *4all T_EX* 和 *T_EX Live* 上获得(这两者都可以从 T_EX Users Group 获得)。

然后你可以通过以下方式选择 T1 字体:

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

这将允许你使用本节中的命令。

注意：由于本文档必须能够在任何运行最新 L^AT_EX 的站点上处理，它不包含任何只存在于 T1 编码字体中的字符。这意味着本文档无法展示这些字形！如果你想看到它们，请在文档 fontsmpl 上运行 L^AT_EX，并在提示输入字族名称时输入 “cmr”。

`\k{<text>}`

这个命令产生一个 “ogonek” 重音符号。

新的特色
1994/12/01

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

这些命令产生 “eth”、“dbar”、“eng” 和 “thorn” 字符。

新的特色
1994/12/01

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

这些命令产生各种引号符号。它们的粗略表示如下：«a» <a> „a“ ,a‘ 和 "a"。

新的特色
1994/12/01

因此，在只会使用 T1 编码字体的文档中，有一些额外的短格式连字可供使用。

新的说明
2001/06/01

尖括号 `\guillemotleft` 和 `\guillemotright`¹ 可以通过输入 << 和 >> 获得，`\quotedblbase` 可以通过输入 ,, 获得。

另外，与 OT1 编码字体产生意外结果不同，< 和 > 现在会产生 < 和 >。

还要注意，单个字符 " 不再产生”，而是产生 `\textquotedbl`。

3.16 Logos

`\LaTeX`
`\LaTeXe`

`\LaTeX`（生成 ‘L^AT_EX’）仍然是“主要”的标志命令，但如果需要引用新特性，可以使用 `\LaTeXe`（生成 ‘L^AT_EX 2_ε’）。

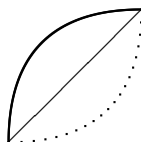
¹我们再次为保留 Adobe 混淆潜水鸟与标点符号的巨大自我中心主义（注：原文 solipsism）而道歉！

3.17 图形命令

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

`\qbezier` 命令可以在 `picture` 模式中绘制二次贝塞尔曲线，从位置 $(\langle AX \rangle, \langle AY \rangle)$ 到 $(\langle CX \rangle, \langle CY \rangle)$ ，控制点为 $(\langle BX \rangle, \langle BY \rangle)$ 。可选参数 $\langle N \rangle$ 给出曲线上的点数。

例如，以下图表：



是用以下代码绘制的：

```
\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}
```

`\bezier` 命令与之相同，不同之处在于参数 $\langle N \rangle$ 不是可选的。它提供了与 L^AT_EX 2.09 的 `bezier` 文档样式选项兼容性。

3.18 老的命令

`\samepage`

`\samepage` 命令仍然存在，但不再维护。这是因为它总是运行不稳定；它不能保证在其作用范围内不发生分页；它可能导致脚注和边注放置错误。

我们建议结合页面分隔命令，如 `\newpage` 和 `\pagebreak`，使用 `\enlargethispage` 来控制分页。

`\SLiTeX`

由于 `SLiTeX` 不再存在，所以在 `LaTeX` 内核中不再定义此标志。一个合适的替代是 `\textsc{Sli\TeX}`。`SLiTeX` 标志在 `LaTeX 2.09` 兼容模式下定义。

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

这些符号包含在 `LaTeX` 符号字体中，在 `LaTeX 2.09` 中会自动加载此字体。但是，`TeX` 只有十六个数学字体族的空间；因此，许多用户发现字体用完了。因此，除非使用 `latexsym` 包，否则 `LaTeX` 不会加载 `LaTeX` 符号字体。

这些符号也由 `amsmath` 包提供，该包还定义了大量其他符号，并使用不同的字体提供这些符号。

在 `LaTeX 2.09` 兼容模式下，`latexsym` 包会自动加载。

4 `LaTeX 2.09` 文档

`LaTeX 2ε` 可以处理（几乎）任何 `LaTeX 2.09` 文档，通过进入 *LaTeX 2.09* 兼容模式。操作方式与以往相同，你运行 `LaTeX` 将会得到大致相同的结果。

“几乎”不能处理的原因是，一些 `LaTeX 2.09` 的包使用了不被支持的底层特性。如果发现这样的包，你应该查看是否已更新为与 `LaTeX 2ε` 兼容。大多数包仍然可以在 `LaTeX 2ε` 中工作，最简单的方法是尝试运行它！

`LaTeX 2.09` 兼容模式是对 `LaTeX 2.09` 的全面模拟，但代价是时间。文档在兼容模式下的运行速度可能比在 `LaTeX 2.09` 下慢 50%。

4.1 警告

这个 *LaTeX 2.09* 兼容模式仅用于处理 2.09 文档，也就是（我们希望是很久以前）为非常老的系统编写的文档，因此可以使用真正的古老的 `LaTeX 2.09` 系统处理。

新的说明
1995/12/01

因此，此模式不旨在提供对 `LaTeX 2ε` 的增强功能的访问。因此，不能将其用于处理伪装成 2.09 文档（即以 `\documentstyle` 开头）但在真正的古老的 `LaTeX 2.09` 系统中无法处理的新文档，因为它们包含了一些新的，仅在 `LaTeX 2ε` 中才有的命令或环境。

为防止对系统的误用，以及因分发这些误导性编码的文档而导致的麻烦，*L^AT_EX 2.09* 兼容模式关闭了大多数这些新功能和命令。任何尝试使用它们的操作都会产生错误消息，而且许多操作根本不起作用，而其他一些则会产生不可预测的结果。因此，如果发现此类情况，请不要给我们发送任何错误报告，因为这些都是故意的。

4.2 字体选择问题

当使用兼容模式时，可能会在一些老的文档中遇到字体变化命令的问题。这些问题有两种类型：

- 引发错误信息；
- 没有产生预期的字体变化。

如果出现错误信息，可能是因为文档（或其中使用的老的样式文件）包含对不再定义的老的内部命令的引用，请参阅第 6.2 节了解更多信息。

一个意外的例子是，如果你使用新的数学模式字体更改命令如下：

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

你可能会发现，这会表现得好像你输入了：

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

所有内容都会变成粗体，包括 *A*。

L^AT_EX 2.09 允许站点自定义其 *L^AT_EX* 安装，这导致文档在不同 *L^AT_EX* 安装上产生不同结果。*L^AT_EX 2_ε* 不再允许这么多的自定义，但为了与老的文档兼容，每次进入 *L^AT_EX 2.09* 兼容模式时都会加载本地配置文件 `latex209.cfg`。

例如，如果你的站点定制使用新字体选择方案（NFSS）和 `oldfont` 选项，那么你可以通过创建包含以下命令的 `latex209.cfg` 文件来使 *L^AT_EX 2_ε* 模拟这种情况：

```
\ExecuteOptions{oldfont}\RequirePackage{oldfont}
```

类似地，要模拟带有 `newfont` 选项的 NFSS，你可以创建包含以下内容的 `latex209.cfg` 文件：

```
\ExecuteOptions{newfont}\RequirePackage{newfont}
```

新的说明
1995/12/01

4.3 原生模式

为了更快地运行老的文档并使用 \LaTeX 2_ϵ 的新特性，你应该尝试使用 \LaTeX 2_ϵ 原生模式。这可以通过将命令：

```
\documentstyle[\langle options \rangle,\langle packages \rangle]{\langle class \rangle}
```

替换为：

```
\documentclass[\langle options \rangle]{\langle class \rangle}  
\usepackage{latexsym,\langle packages \rangle}
```

来实现。然而，一些可以在 \LaTeX 2.09 兼容模式下处理的文档可能在原生模式下无法工作。一些 \LaTeX 2.09 的宏包只能在 \LaTeX 2_ϵ 的 2.09 兼容模式下使用。一些文档会因为 \LaTeX 2_ϵ 改进的错误检测能力而导致错误。

但大多数 \LaTeX 2.09 文档可以通过上述更改在 \LaTeX 2_ϵ 的原生模式下处理。同样，确定你的文档是否能在原生模式下处理的最简单方法就是尝试一下！

5 局部修改

有两种常见的局部修改类型可以非常简单地实现。不要忘记，使用这些修改生成的文档在其他地方可能无法使用（这样的文档被称为“非便携”）。

新的说明
1995/12/01

一种修改类型是使用个人命令来表示常用符号或结构。这些应该放入一个宏包文件中（例如，名为 `mymacros.sty` 的文件），并在文档导言中使用 `\usepackage{mymacros}` 加载它们。

另一种类型是一个本地文档类，它与标准类之一非常相似，但包含一些直接的修改，比如额外的环境，某些参数的不同值等。这些应该放入一个类文件中；这里我们将使用一个名为 `larticle` 的类作为示例，它与 `article` 类非常相似。

名为 `larticle.cls` 的类文件应该（在预备识别命令之后）开始如下：

新的特色
1995/12/01

```
\LoadClassWithOptions{article}
```

这个命令应该跟随着你希望对读入文件 `article.sty` 的结果进行的任何添加和更改。

使用上述 `\LoadClassWithOptions` 命令的效果是加载标准类文件 `article` 并使用文档请求的任何选项。因此，使用你的 `larticle` 类的文档可以指定在使用标准 `article` 类时可以指定的任何选项；例如：

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

6 问题

本节描述了在使用 \LaTeX 2_ϵ 时可能出现的一些问题以及解决方法。

6.1 新的错误信息

\LaTeX 2_ϵ 具有许多新的错误信息。还请注意，如果在错误提示中按下 `h` 键，许多错误信息现在会提供进一步的有用信息。

Option clash for package `\package`.

已加载的特定宏包使用了不同的选项加载了两次。如果输入 `h`，你会得知这些选项是什么，例如，如果你的文档包含了：

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

那么你将得到错误信息：

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

在 `?` 提示符下键入 `h` 将会显示：

```
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.
```

解决方法是，如建议的那样，使用这两组选项加载宏包。需要注意的是，由于 \LaTeX 宏包可以调用其他宏包，即使没有显式地两次请求相同的宏包，也可能出现宏包选项冲突的情况。

Command $\langle command \rangle$ not provided in base NFSS.

在默认情况下, $\langle command \rangle$ 不在 \LaTeX 2_ϵ 中提供。使用以下命令会导致此错误:

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

这些命令现在属于 `latexsym` 宏包的一部分。解决方法是在文档导言中添加:

```
\usepackage{latexsym}
```

\LaTeX 2_ϵ command $\langle command \rangle$ in \LaTeX 2.09 document.

$\langle command \rangle$ 是一个 \LaTeX 2_ϵ 命令, 但这是一个 \LaTeX 2.09 的文档。解决方法是将该命令替换为 \LaTeX 2.09 的命令, 或者按照第 4.3 节的描述在原生模式下运行文档。

NFSS release 1 command `\newmathalphabet` found.

命令 `\newmathalphabet` 是 New Font Selection Scheme Release 1 中的命令, 但现在已被 `\DeclareMathAlphabet` 取代, 其使用方法在 《[\text{\LaTeX 2}_\epsilon Font Selection](#)》中有描述。

最好的解决方法是更新包含 `\newmathalphabet` 命令的宏包。查看是否有该包的新版本, 或者(如果你自己编写了该包)参考 《[\text{\LaTeX 2}_\epsilon Font Selection](#)》查看字体命令的新语法。

如果没有该包的更新版本, 你可以通过使用 `newlfont` 或 `oldlfont` 宏包来解决这个错误, 告诉 \LaTeX 应该模拟哪个版本的 `\newmathalphabet`。

如果文档选择数学字体的语法如下:

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

则使用 `oldlfont`。如果文档的语法类似于:

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

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

`\verb` 命令已经开始但在该行未结束。这通常意味着你忘记在 `\verb` 命令中添加结束字符。

Illegal use of `\verb` command.

`\verb` 命令已经被用在另一个命令的参数中。这在 \LaTeX 中从未被允许——通常会导致不正确的输出而不提供任何警告——因此 \LaTeX 2_ϵ 会产生错误消息。

6.2 老的内部命令

许多 \LaTeX 2.09 的内部命令已被移除，因为它们的功能现在以不同的方式提供。详细信息请参阅《 [\$\text{\LaTeX 2}_\epsilon\$ for Class and Package Writers](#)》，了解类和宏包作者的新支持接口。

```
\tenrm \elvrm \twlrm ...  
\tenbf \elvbf \twlbf ...  
\tensf \elvsv \twlsf ...  
:
```

这些命令提供了访问 \LaTeX 2.09 预加载的七十种字体的方式。相比之下， \LaTeX 2_ϵ 通常最多预加载十四种字体，这节省了大量字体内存；但结果是任何使用上述命令直接访问字体的 \LaTeX 文件将不再起作用。

它们的使用通常会产生错误信息，例如：

```
! Undefined control sequence.  
1.5 \tenrm
```

解决方法是更新文档以使用 \LaTeX 2_ϵ 提供的新的字体更改命令；这些命令在《 [\$\text{\LaTeX 2}_\epsilon\$ Font Selection](#)》中有描述。

如果不可能进行更新，作为最后的手段，你可以使用 `rawfonts` 宏包，它加载了七十种 \LaTeX 2.09 字体，并使用老的命令直接访问它们。这将消耗时间和内存。如果你不想加载所有七十种字体，你可以通过 `rawfonts` 的 `only` 选项选择其中一些。例如，仅加载 `tenrm` 和 `tenbf`，你可以写成：

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

`rawfonts` 宏包作为 \LaTeX 工具软件的一部分分发，详见第 2.4 节。

6.3 老的文件

在运行 \LaTeX 时，较常见的错误之一是读取老版本的宏包而不是新版本。如果从标准宏包中得到了难以理解的错误消息，请确保加载的是宏包的最新版本。你可以在日志文件中找到类似以下行的信息来确认加载了哪个版本的宏包：

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

你可以使用 `\documentclass` 和 `\usepackage` 的 `\release-date` 选项来确保获取适当最新的文档类或宏包副本。当向其他站点发送文档时，这非常有用，因为其他站点可能使用过时的软件。

6.4 寻求更多帮助

如果在这里找不到解决问题的答案，请尝试查阅《 [\$\text{\LaTeX}\$: A Document Preparation System](#)》或《[The \$\text{\LaTeX}\$ Companion](#)》。如果在安装 \LaTeX 方面遇到问题，请查阅发行版附带的安装指南文件。

如果这些方法都不起作用，请联系你当地的 \LaTeX 专家或 \LaTeX 邮件列表。

如果你认为自己发现了一个 bug，请报告！首先，你应该确定问题是否与第三方宏包或类相关。如果问题是由于除了第 2 节中列出的类和宏包之外的某个包或类引起的，请将问题报告给该包或类的作者，而不是 \LaTeX 3 项目团队。

如果 bug 真的是与核心 \LaTeX 有关的，你应该创建一个简短、自包含的文件来展示问题。你应该在该文件上运行一个最新（不超过一年的版本）的 \LaTeX ，然后运行 \LaTeX 在 `latexbug.tex` 文件上。这将创建一个错误报告，你应该将其与示例文档和日志文件一起发送到 \LaTeX 的 bug 地址，该地址可以在文件 `latexbug.tex` 或 `bugs.txt` 中找到。

7 尽情享受！

我们当然希望你会喜欢使用新的标准 \LaTeX ，但如果不可能的话，我们希望你能够因为它帮助你创建的文档而取得成功和成就感。

如果你发现 \LaTeX 对你的生活有所贡献，以至于你想支持项目团队的工作，那么请阅读第 1.2 节，并发现实际的支持方法。

参考文献

- [1] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997.
- [2] Michel Goossens and Sebastian Rahtz. *The L^AT_EX Web Companion*. Addison-Wesley, Reading, Massachusetts, 1999.
- [3] Donald E. Knuth. *The T_EXbook*. Addison-Wesley, Reading, Massachusetts, 1986. Revised to cover T_EX3, 1991.
- [4] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.
- [5] Frank Mittelbach and Michel Goossens. *The L^AT_EX Companion second edition*. With Johannes Braams, David Carlisle, and Chris Rowley. Addison-Wesley, Reading, Massachusetts, 2004.