

L^AT_EX 中的超文本标记：hyperref 宏包手册

Sebastian Rahtz (塞巴斯蒂安·拉赫兹)*

Heiko Oberdiek (海科·奥伯迪克)[†]

L^AT_EX3 项目[‡]

黄旭华[§] 【译】 至 2023 年 02 月 07 日

张泓知 更新 至 2023 年 12 月 25 日

2023-11-26 v7.01g

目 录

1 前言	4	4 为类和包作者准备的接口	9
1.1 恢复已删除的补丁	4	4.1 计数器	9
2 介绍	5	4.2 宏包和 \hypersetup 选项的值	10
		4.3 链接命令	10
3 隐性行为	8	4.4 创建目标	11

* 已故

[†] 不活跃

[‡] <https://github.com/latex3/hyperref/issues>

[§] 一名业余 L^AT_EX 爱好者，现供职于赣南医学院第一附属医院神经内科。电子邮箱：hxh_828@163.com。

4.5	补丁以及如何抑制它们	12	7.10	\ifpdfstringunicode	51
5	宏包选项	14	7.11	使用 \nohyperpage 自定义索引样式文件	53
5.1	一般选项	16	7.12	“ocgcolorlinks” 实验性选项	53
5.2	目的地名称选项	17	7.13	“pdfa” 选项	54
5.3	页面锚点	20	7.14	已添加的 “linktoc” 选项 . . .	55
5.4	配置选项	21	7.15	已更改的 “pdfnewwindow” 选项	55
5.5	后端驱动程序	21	7.16	PDF 表单标志选项	56
5.6	扩展选项	22	7.17	“pdfversion” 选项	58
5.7	特定于 PDF 的显示选项	24	7.18	“name” 字段选项	59
5.8	PDF 显示和信息选项	26	7.19	“pdfencoding” 选项	59
5.9	pdfinfo 选项	31	7.20	颜色选项/hycolor 宏包	59
5.10	完整的选项列表, 按字母顺序排列	32	7.21	pdfusetitle 选项	60
6	其它用户宏	38	7.22	\autoref 的星号形式	60
6.1	书签宏	44	7.23	链接边框样式	60
6.1.1	设置书签	44	7.24	bookmarksdepth 选项	61
6.1.2	替换宏	45	7.25	pdfescapeform 选项	61
6.2	页面标签	46	7.26	默认驱动程序设置	62
6.3	实用程序宏	46	7.27	Backref 条目	62
7	新功能	48	7.28	\phantomsection	65
7.1	“pdflinkmargin” 选项	48	7.29	puenc 编码、puenc-greekbasic.def 和 puenc-extra.def	65
7.2	“calculatesortkey” 字段选项	48	8	特定于 Acrobat 的行为	66
7.3	“next-anchor” 选项	49	9	PDF 和 HTML 格式	68
7.4	“localanchorname” 选项 . .	49	9.1	表单环境参数	69
7.5	“customdriver” 选项	49	9.2	表单选项参数	70
7.6	“psdextra” 选项	50	10	定义新的驱动程序	73
7.7	\XeTeXLinkBox	50			
7.8	\IfHyperBooleanExists 和 \IfHyperBoolean	50			
7.9	\unichar	51			

11 对其他宏包的特殊支持	74		
11.1 宏包的兼容性	74	11.1.29 sidecap	80
11.1.1 algorithm	74	11.1.30 subfigure	80
11.1.2 amsmath	74	11.1.31 titleref	81
11.1.3 amsrefs	75	11.1.32 tabularx	81
11.1.4 arydshln、longtable	75	11.1.33 titlesec	81
11.1.5 babel/magyar.ldf	75	11.1.34 ucs/utf8x.def	81
11.1.6 babel/spanish.ldf	75	11.1.35 varioref	82
11.1.7 bibentry	75	11.1.36 verse	82
11.1.8 bigfoot	76	11.1.37 vietnam	82
11.1.9 chappg	76	11.1.38 XeTeX	82
11.1.10 countlto	76	12 局限性	83
11.1.11 dblacnt	77	12.1 换行/断开的链接支持	83
11.1.12 easyeqn	77	12.2 跨页面的链接	83
11.1.13 ellipsis	77	12.3 脚注	84
11.1.14 float	77	13 提示	85
11.1.15 endnotes	78	13.1 选项值中的空格	85
11.1.16 foiltex	78	13.2 Index with makeindex	85
11.1.17 footnote	78	13.3 警告“bookmark level for un-	
11.1.18 linguex	78	known <foobar> defaults to	
11.1.19 ltabptch	78	0”	86
11.1.20 mathenv	78	13.4 图形中的链接锚点	87
11.1.21 minitoc-hyper	79	13.5 书签和 pdf 信息条目中的其	
11.1.22 multind	79	他 unicode 字符	87
11.1.23 natbib	79	13.6 脚注	88
11.1.24 nomencl	79	13.7 从属计数器	90
11.1.25 ntheorem	79	14 历史和鸣谢	91
11.1.26 ntheorem-hyper	79	15 GNU 自由文档许可证	92
11.1.27 prettyref	80		
11.1.28 setspace	80		

1 前言

正如在下面的介绍 (introduction) 中看到的那样, `hyperref` 有着悠久的历史, 并且随着时间的推移发生了许多变化。下面的介绍中还提到了工作流 (workflows)、驱动程序, 以及不再 (或者只是在边缘情况 [edge case] 中) 相关的问题。该文档反映了这种不同的历史: 变化 (changes)、扩展 (extensions), 以及解释 (explanations)(这样的解释过去和现在都散布在各种论文和资料中, 或者后来被合并), 因此它们并不总是按照连贯的顺序出现, 彼此之间并不同步。

这一历史仍在继续: 如果您正在使用新的 L^AT_EX PDF 管理 (management), 它目前是以测试阶段宏包 (testphase package) `pdfmanagement-testphase` 的形式发布的, 那么 `hyperref` 将为 PDF 输出 (output) 使用一个新的通用驱动程序 (generic driver), 该驱动程序包含许多更改和新功能 (new features)。这个驱动程序的文档即 `hyperref-generic.pdf` 目前是 `pdfmanagement-testphase` 文档的一部分。新驱动程序的一个重要变化是: 它删除了旧的 `hyperref` 书签 (bookmarks) 代码, 改为使用 `bookmark` 宏包。因此, 要了解有关扩展书签 (extend the bookmarks) 的选项, 还应该查阅 `bookmark` 的文档¹。

1.1 恢复已删除的补丁

随着时间的推移, `hyperref` 已经修补了相当多的宏包, 以解决冲突 (clashes) 和不兼容性 (in-compabilities)。其中许多宏包要么不再需要, 要么应该由原始宏包来完成。这些补丁 (patches) 现在正慢慢地从 `hyperref` 中删除。这通常不会导致问题, 但如果应该恢复这些补丁, 则可以通过该捆绑包 (bundle package) 的 `hyperref-patches` 这部分来加载补丁。

¹ 译者注: 译者已将该宏包的文档译成了中文, 点击查看: [bookmark_ZH_CN.pdf](#)

2 介绍

该宏包源于 HyperTeX 项目的工作，并建立在该项目的基础上，在 <http://xxx.lanl.gov/hypertext/>² 有该项目的描述。它扩展了所有 L^AT_EX 交叉引用命令 (cross-referencing commands) 的功能，这些交叉引用命令包括目录 (table of contents)、参考文献 (bibliographies) 等，以生成驱动程序 (driver) 可以转换为超文本链接 (hypertext links) 的特殊命令；它还提供了新的命令，以允许用户编写 *ad hoc* 超文本链接，包括指向外部文档和 URLs 链接。

目前在 <https://github.com/latex3/hyperref/> 维护该宏包，应该在这里报告问题 (issues)。

本手册 (manual) 简要介绍了 hyperref 宏包。有关更详细的信息，请阅读随宏包分发的其他文档，以及编译 `hyperref.dtx` 来获得完整的文档。您还应该阅读 *The L^AT_EX Web Companion* 中关于 hyperref 的章节，在那里您可以找到其他示例。

HyperTeX 规范 (specification)³ 表示符合条件的查看器 (viewers)/翻译器 (translators) 必须识别以下 \special 结构集 (constructs set)：

href: html:

name: html:

end: html:

image: html:

base_name: html:<base href = "href_string">

href、*name* 和 *end* 命令用于执行在文档各部分之间建立链接 (links) 的基本超文本操作 (basic hypertext operations)。*image* 命令旨在 (与当前 HTML 查看器一样) 将任意图形格式 (graphical format) 的图像放置在页面的当前位置。*base_name* 命令用于向 DVI 查看器传递当前文档的完整 (URL) 位置，以便可以正确检索由相对 URLs 指定的文件。

在 T_EX 文件的后面，*href* 和 *name* 命令必须与 *end* 命令配对——一对命令的两端之间的 T_EX 命令在文档中形成锚点 (*anchor*)。在使用 *href* 命令的情况中，锚点将在 DVI 查看器 (*DVI viewer*) 中高亮显示 (*highlighted*)，单击锚点将导致场景 (*scene*) 转移到 *href_string* 指定的目标 (*destination*)。与 *name* 命令相关联的锚点表示其他超文本链接 (hypertext links) 可能引用的位置

² 目前: <https://ctan.org/tex-archive/support/hypertext/hypertext>

³ 这是从 Arthur Smith (亚瑟·史密斯) 的一篇文章中借来的。

(refer), 或者作为本地引用 (local references)(具有 *name_string* 的 `href="#name_string"` 形式与 *name* 命令中的 `href="#name_string"` 相同), 或者作为 URL 的一部分 (`URL#name_string` 形式)。在这里, *href_string* 是一个有效的 URL 或本地标识符 (local identifier), 而 *name_string* 可以是任意字符串 (string): 唯一的警告是, “`#`” 字符应该用反斜杠 (\) 转义, 如果它看起来像一个 URL 名称, 可能会导致问题。

但是, 用于只生成 PDF 的驱动程序使用 PostScript 或 PDF `\special` 命令。命令在不同驱动程序的配置文件 (configuration files) 中定义, 由宏包选项选择或者对于大多数当前的引擎进行自动检测; 目前, 支持以下驱动程序:

hypertex 符合 HyperTeX 指南 (guidelines) 的 DVI 处理器: `xdvi`、`dvips`(带 `-z` 选项)、`OzTeX` 和 `Textures`

dvips 生成 `dvips` 量身定制的 `\special` 命令。如果检测到 DVI 模式, 这将是默认的驱动程序。

dvipsone 生成 `dvipsone` 量身定制的 `\special` 命令

ps2pdf 适用于 Ghost 脚本 (Ghostscript)PDF 编写器 (writer) 早期版本处理的输出特例; 这与 `dvips` 的基本相同, 但在 5.21 版本之前保留了一些变体 (variations)

tex4ht 生成用于 `TeX4ht` 的 `\special` 命令, 自动检测。

pdftex `pdfTeX`, Hàn Thê Thành 的 `TeX` 变体, 直接编写 PDF, 自动检测。

luatex `luaTeX`, Unicode `TeX` 变体, 直接编写 PDF, 自动检测。

dvipdfm 为 Mark Wicks (马克·威克斯) 的 DVI 到 PDF 驱动程序 `dvipdfm` 生成 `\special` 命令

dvipdfmx 为 `dvipdfm` 的继承驱动程序 `dvipdfmx` 生成 `\special` 命令

dviwindo 生成 `\special` 命令, Y&Y 的 Windows 预览器 (previewer) 将这些命令解释为预览器中的超文本跳转 (hypertext jumps)

vtex 生成 `\special` 命令, MicroPress 的 HTML 和 PDF 生成 `TeX` 变体将其解释为预览器 (previewer) 中的超文本跳转 (hypertext jumps), 自动检测。

textures 生成 `\special` 命令, `Textures` 将这些命令解释为预览器 (previewer) 中的超文本跳转 (hypertext jumps)

xetex 为 `XeTeX` 生成 `\special` 命令, 自动检测。

hitex (2023 年新版) 适用于由 HiT_EX 引擎生成的提示格式, 会自动检测。这是一个相当实验性的引擎, 驱动文件不是 hyperref 套件的一部分, 而是由 hitex 宏包提供的。如有问题, 请报告至 <https://github.com/ruckertm/HINT>。

dvips 或 dvipsone 的输出必须使用 Acrobat Distiller 处理才能获得 PDF 文件⁴。结果通常比使用 hypertex 驱动程序, 然后使用 dvips -z 处理产生的结果要好, 但 DVI 文件是不可移植的。使用 HyperT_EX \special 命令的主要优点是, 您还可以在超文本 DVI 查看器 (hypertext DVI viewers) 例如 xdvi 中使用文档。

driverfallback 如果未提供驱动程序且无法自动检测, 则使用驱动程序选项 (driver option), 该选项的值为 driverfallback。例如:

```
driverfallback=dvipdfm
```

自动检测到的驱动程序 (pdftex, luatex, xetex, vtex, vtexpdfmark, hitex) 是从 T_EX 中识别的, 因此不能作为选项 driverfallback 的值。但是, 在 T_EX 运行完成后, 会运行 DVI 驱动程序。因此, 无法在 T_EX 宏级别 (macro level) 检测到它。然后, hyperref 宏包使用由 driverfallback 指定的驱动程序。如果已指定驱动程序或可以自动检测驱动程序, 则会忽略选项 driverfallback。

⁴ 请确保关闭 dvips 和 dvipsone 支持的部分字体下载, 以支持 Distiller 自己的系统。

3 隐性行为

通过在文档前言 (preamble) 中指使用下面的语句, 该宏包可以或多或少地与任何普通的 L^AT_EX 文档一起使用。

```
\usepackage{hyperref}
```

确保它出现在您加载的所有宏包的最后 (*last*), 给它一个不被改写的机会, 因为它的工作是重新定义许多 L^AT_EX 命令⁵。

不要将其加载到 `\AtBeginDocument` 或 `begindocument` 钩子 (*hook*) 中! 虽然这在过去经常起作用, 但官方并不支持这样做。由于 `hyperref` 和 `nameref` 也使用这个钩子来初始化命令, 如果在那里加载宏包, 代码执行的时机 (timing of code execution) 会很棘手 (tricky) 和脆弱 (fragile)。如果你想延迟加载, 请使用 `begindocument/before` 钩子。

希望你能发现所有的交叉引用 (cross-references) 都能以超文本 (hypertext) 的形式正常工作。例如, `\section` 命令将生成一个书签 (bookmark) 和一个链接 (link), 而 `\section*` 命令仅在与相应的 `\addcontentsline` 命令配对 (paired) 时显示链接。

此外, `hyperindex` 选项 (见下文) 试图通过超链接返回到文本来创建索引中的条目 (items in the index), `backref` 选项为每个条目在参考文献 (bibliography) 中插入额外的“返回 (back)”链接。其他选项控制链接的外观 (appearance), 并对 PDF 输出进行额外的控制。例如 `colorlinks`, 正如它的名字所暗示的那样, 为链接着色, 而不是使用方框 (boxes), 这是本文档中使用的选项。

⁵ 但是工作 (work) 已经开始减少重新定义的次数以及对加载顺序 (loading order) 的依赖性。

4 为类和包作者准备的接口

超链接功能 (hyperlink features) 现在是一种常见的需求。hyperref 从 L^AT_EX 内核和宏包中修补了相当多的命令，以添加此类功能。但这是相当脆弱的，它增加了对加载顺序 (loading order) 的依赖，如果外部宏包发生变化，它可能会中断。因此，如果宏包直接为其命令添加适当的支持，效果会好得多。实际上，很多宏包都这样做了，但由于缺少接口文档，他们经常查看代码，然后使用内部命令 (internal commands)，而不是公共命令 (public command)。

下面尝试描述现有的变量和命令，这些变量和命令可以被视为公共接口 (public interfaces)，应该或者可以被宏包设置为与 hyperref 命令保持兼容。文档化的用户命令 (Documented user commands) 自然也是接口，这里没有再明确提到它们。

这样的工作正在进行中。欢迎提出建议或意见。

4.1 计数器

计数器 (counters) 在代码中起着重要作用。它们用于创建目标名称 (destination names) 和定义书签 (bookmarks) 等层次结构 (hierarchies)。为了正常工作，他们需要一些额外的设置。

`\theH<counter>` hyperref 通常使用计数器的名称和 `\the<counter>` 值为链接锚点 (link anchor) 创建目标名称 (destination names)。这可能会失败，例如如果 `\the<counter>` 不是文档中唯一的，或者如果它是不可展开的 (not expandable)。在这种情况下，应该定义 `\theH<counter>`，以便它提供一个唯一的、可展开的值。即使没有加载 hyperref，定义它也没有害处。

`\toclevel@<counter>` 这是一个应该包含一个数字的变量。它用于书签中的级别。应该为类似 toc 的列表 (lists) 和 `\addcontentsline` 中使用的所有计数器定义它。典型值为：

```
\def\toclevel@part{-1}
\def\toclevel@chapter{0}
\def\toclevel@section{1}
\def\toclevel@subsection{2}
\def\toclevel@subsubsection{3}
\def\toclevel@paragraph{4}
\def\toclevel@subparagraph{5}
```

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

4.2 宏包和 \hypersetup 选项的值

7.00s 版本新增

当在宏包选项或 \hypersetup hyperref 中设置键 (key), 通常将结果存储在内部变量 (internal variables) 中, 或执行某些代码或设置内部布尔值 (internal boolean)。在这里, 包和类的作者不应该依赖于键处理的名称或细节。

但是由于其他宏包有时需要知道设置了哪个值, 所以可以使用可展开的 (expandable) \GetDocumentProperties 来检索某些值。这些值由 \exp_not:n 包围返回, 因此可以在 \edef 中安全地使用。例如, 你可以使用下面的语句来获得 pdf 作者 (pdfauthor)。

```
\edef\mypdfauthor{\GetDocumentProperties{hyperref/pdfauthor}}
```

这些值是按照用户输入的值返回的! 如果它们应该在 PDF 上下文 (context) 中使用, 则仍然必须应用 \pdfstringdef 或等效的东西 (something equivalent)。

目前, 这个接口可以用于键 pdfauthor、pdftitle、pdfproducer、pdfcreator、pdfsubject 和 pdfkeywords。如果与未知键一起使用, 则返回空值 (empty value)。如果使用 \Document-Metadata 加载了新的 PDF 管理 (management), 该接口也可以工作, 在这种情况下, 更多的键返回它们的值。

4.3 链接命令

所有驱动程序都提供以下命令来创建链接。如果用户命令不足 (not sufficient), 则宏包可以使用这些命令。新驱动程序必须为这些命令提供类似的参数。

```
\hyper@anchor {destination name}
\hyper@anchorstart {destination name}
\hyper@anchorend
\hyper@link {context}{destination name}{link text} %GoTo
\hyper@linkstart {context} {destination name} %GoTo
\hyper@linkend %GoTo
\hyper@linkfile {link text} {filename} {destname} %GoToR
\hyper@linkurl {link text}{url} %URI
\hyper@linklaunch{filename} {link text} {Parameters} %启动, 仅使用新的通用驱动程序
\hyper@linknamed {action}{link text} %已命名, 仅使用新的通用驱动程序
```

4.4 创建目标

内部链接 (internal links) 和书签 (bookmarks) 需要一些可以跳转到 (jump to) 的东西。在 PDF 中, 这通常被称为目的地 (*destination*)(因此原语被称为 `\pdfdest`), 在 HTML 中, 更常见的是将其称为锚点 (*anchor*)(因而 `hyperref` 为此使用 `\hyper@anchor`)。历史记录 (*history*) 无法撤消, 但未来的命令和描述将其称呼为通用的 (*generic*) 目标 (*target*), 除非是 PDF 特定的称呼即目的地 (*destination*)。

当使用 `\refstepcounter` 时, 会自动创建目标 (*target*), 在许多情况下, 这样做是正确的, 不需要更多的东西。但也有例外:

- 例如，如果使用星号版本 (starred version) 时分节命令 (sectioning command) 没有编号 (number)，或者由于 `secnumdepth` 的设置，则可能会缺少所需的目标 (needed target)。
- `\refstepcounter` 创建的目标可能位于错误的位置 (wrong place)。
- `\refstepcounter` 创建的目标可能会影响间距 (spacing)。
- 由 `\refstepcounter` 创建的目标名称 (target name) 不可用，例如，在 `\bibitem` 中，您需要基于 `bib` 键 (bib-key) 的目标名称。

宏包的作者和用户可以使用以下命令来创建和操作目标 (targets)。这些命令的更详细的描述请阅读 [hyperref-linktarget.pdf](#)。

- `\MakeLinkTarget`
- `\LinkTargetOff`
- `\LinkTargetOn`
- `\NextLinkTarget`
- `\SetLinkTargetFilter`

前四个命令也将在 `LaTeX` 中直接定义为没有操作 (no-op), 因此即使没有加载 `hyperref`, 也可以使用它们。

直到 L^AT_EX 被更新，宏包作者也可以直接提供这些定义：

```
\ProvideDocumentCommand\MakeLinkTarget{sO{ }m}{ }
\ProvideDocumentCommand\LinkTargetOn{ }{ }
\ProvideDocumentCommand\LinkTargetOff{ }{ }
\ProvideDocumentCommand\NextLinkTarget{m}{ }
```

4.5 补丁以及如何抑制它们

通过加载 `hyperref` 和 `implicit=false` 选项, 可以完全避免使用 `hyperref` 对外部命令 (external commands) 进行补丁。但是抑制一切 (suppressing everything) 往往过于极端。目前正在对补丁进行分类, 并提供接口 (interfaces) 以更精细的方式抑制它们。

分节命令 (sectioning commands)

- `hyperref` 修补 `\@sect`, `\@ssect`, `\@chapter`, `\@schapter`, `\@part`, `\@spart`。
- 它在星号命令 (starred commands) 中添加了链接的目标 (章节前缀为 `chapter*`, 其他前缀为 `section*`)。如果分节 [sectioning] 没有编号, 例如因为 `secnumdepth` 设置或在前面的内容 (front matter) 中, 它会在其他命令中添加链接的目标。
- 可以通过定义命令 `\hyper@nopatch@sectioning` 来抑制补丁。这通常只能由提供分节命令并添加目标本身的类或包来完成。目标在页面上有一个位置 (location), 例如, `section` 命令应该考虑缩进 (indents)。书签 (bookmarks) 和目录 (table of contents) 需要目标 (targets), 因此 `\@currentHref` 应该在使用 `\addcontentsline` 之前获得正确的含义 (correct meaning)。
- 请注意, `nameref` 宏包也会修补这些命令, 即添加命令来将标题文本 (title text) 存储在 `\@currentlabelname` 中。请查看 `nameref` 文档, 了解抑制这些补丁程序的方法。

脚注 (footnotes) 要启用 (部分) 脚注链接 (linking of footnotes), `hyperref` 将重新定义或修补各种命令, 这部分依赖于宏包。

- `hyperref` 重新定义 (redefine) `\@xfootnotenext`, `\@xfootnotemark`, `\@mpfootnotetext`, `\@footnotetext`, `\@footnotemark`。如果加载了 `tabularx`, 它将更改 `\TX@endtabularx`。如果加载了 `longtable`, 它将更改 `\LT@p@ftntext`。如果加载了 `fancyvrb`, 它将重新定义 `\V@@footnotetext`。它还重新定义了 `\footref` 和 `\maketitle`。
- 可以通过定义 `\hyper@nopatch@footnote` 来抑制上述所有 (All) 这些重新定义 (redefinitions)。请注意, 这可能会抑制链接 (suppress links), 但也会显现不需要的链接 (unwanted links)。

数学标记 (amsmath tags) `hyperref` 重新定义了 `amsmath` 的两个内部命令 (internal commands), 这两个内部命令与用于添加锚点 (anchor) 的 `\tag` 命令相关。可以通过定义

`\hyper@nopath@amsmath@tag` 来抑制此代码。(这在宏包中通常没有意义, 但当数学环境被更改为允许标记 [allow tagging] 时, 可能需要这样做。)

计数器 (counters) `hyperref` 修补内核命令 (kernel command) `\@definecounter`、`\@newctr`、`\@addtoreset` 和 `amsmath` 命令 `\numberwithin`, 以确保为每个计数器 (counter) 创建或重置正确的 `\theHcounter` 表达 (representation)。可以通过定义 `\hyper@nopath@counter` 来抑制此代码。(这在包中通常没有意义, 但当内核命令被更改为允许标记 [allow tagging] 时, 可能需要这样做。)

数学环境 (math environments) `hyperref` 修补 `\equation/\endequation`、`\eqnarray`、`\endeqnarray`。可以通过定义 `\hyper@nopath@mathenv` 来抑制此代码。

目录 (table of contents) `hyperref` 重新定义了 `\contentsline`, 以便能够添加到目录条目 (toc entries) 的链接。它重新定义了 `\addcontentsline` 以创建书签 (bookmarks) 并将目的地名称 (destination names) 传递给目录条目。可以通过定义 `\hyper@nopath@toc` 来抑制此代码。

标题 (captions) `hyperref` 重新定义了 `\caption` 和 `\@caption`, 插入了链接的目标。可通过定义 `\hyper@nopath@caption` 来抑制此代码 (附加的辅助命令不会被抑制)。由于各种包也重新定义标题 (例如 `caption` 包), 副作用必须小心测试!

长表 (longtable) `hyperref` 重新定义了 `\LT@start` 和 `\LT@array`, 将目标移到更好的位置。可通过定义 `\hyper@nopath@longtable` 来抑制此代码。

定理 (theorems) `hyperref` 对 `\@thm` 进行了修补。可通过定义 `\hyper@nopath@thm` 来抑制此代码。

引用与参考文献 (citations and bibliography) 如果未加载 `natbib`, `hyperref` 会重新定义 `\bibcite`、`\@lbibitem` 和 `\@bibitem`。可通过定义 `\hyper@nopath@bib` 来抑制这些重新定义。

5 宏包选项

所有用户可配置的 `hyperref` 方面都使用单一的 ‘key=value’ 方案（使用 `keyval` 宏包）来设置，键名为 `Hyp`。这些选项可以在 `\usepackage` 命令的可选参数中设置，或使用 `\hypersetup` 宏。加载包时，如果能找到文件 `hyperref.cfg`，则会读取该文件，这是在整个站点范围内设置选项的便利位置。

但请注意，某些选项（例如 `unicode`）只能作为包选项使用，而不能在 `\hypersetup` 中使用，因为选项设置在加载包时进行处理。以下表格列出了（希望是全部的）这些选项。请注意，某些选项可能无效或者在使用新的 `pdfmanagement` 以及新的通用 `hyperref` 驱动程序时有所改变。

hyperref 宏包的选项 (option)	说明 (remark)
所有驱动程序选项如 <code>pdftex</code>	通常不需要, 自动检测
<code>implicit</code>	无说明
<code>pdfa</code>	没有操作 (no-op) 新的 pdf 管理 (pdfmanagement), 请在 <code>\DeclareDocumentMetadata</code> 中设置标准 (standard)。
<code>unicode</code>	现在已经是默认值了
<code>pdfversion</code>	没有操作 (no-op) 新的 pdf 管理 (pdfmanagement), 请在 <code>\DeclareDocumentMetadata</code> 中设置版本 (version)。
<code>bookmarks</code>	这种情况可能会在某个时候改变。
<code>backref</code>	无说明
<code>pagebackref</code>	无说明
<code>destlabel</code>	无说明
<code>pdfusetitle</code>	无说明
<code>pdfpagelabels</code>	无说明
<code>hyperfootnotes</code>	无说明
<code>hyperfigures</code>	无说明
<code>hyperindex</code>	无说明
<code>encap</code>	无说明
<code>CJKbookmarks</code>	只有使用新的 pdf 管理 (pdfmanagement), 在其他情况下, 它可以在 <code>\hypersetup</code> 中使用
<code>psdextra</code>	只有使用新的 pdf 管理 (pdfmanagement), 在其他情况下, 它可以在 <code>\hypersetup</code> 中使用
<code>nesting</code>	只有使用新的 pdf 管理 (pdfmanagement), 在其他情况下, 它可以在 <code>\hypersetup</code> 中使用 (但不清楚它是否有用)

但是请注意, 有些选项 (例如 `unicode`) 只能用作宏包选项, 而不能在 `\hypersetup` 中使用, 因为在读取宏包时会处理选项设置。下面的表格列出了 (希望是所有的) 这些选项。请注意, 如果使用新的 pdf 管理 (pdfmanagement) 和新的通用 hyperref 驱动程序 (generic hyperref driver), 其中一些选项什么都不做, 或者已经改变了行为 (behaviour)。

例如, 特定文件 (particular file) 的行为 (behavior) 可以通过以下方式控制:

- 一个站点范围 (site-wide) 的 `hyperref.cfg` 设置链接的外观 (look of links), 添加反向引

用 (backreferencing), 并设置 PDF 的显示默认值 (display default):

```
\hypersetup{backref, pdfpagemode=FullScreen, colorlinks=true}
```

- 文件中的一个全局选项 (global option), 传递给 hyperref:

```
\documentclass[dvips]{article}
```

- `\usepackage` 命令中的特定于文件的选项 (file-specific options), 这些选项覆盖 `hyperref.cfg` 中设置的选项:

```
\usepackage[colorlinks=false]{hyperref}
\hypersetup{pdftitle={A Perfect Day}}
```

如前面的示例所示, 应该在加载宏包之后设置信息条目 (information entries)(`pdftitle`、`pdfauthor`、...)。否则, \LaTeX 会过早地展开这些选项的值。还有 \LaTeX 在选项中去掉空格 (spaces)。特别是选项 “`pdfborder`” 需要小心。如果作为宏包选项给定, 则需要用花括号 (braces) 保护值。它们在 `\hypersetup` 中不是必需的。

```
\usepackage[pdfborder={0 0 0}]{hyperref}
\hypersetup{pdfborder=0 0 0}
```

一些选项可以在任何时候给出, 但许多选项都是受限制的: 在 `\begin{document}` 之前, 仅在 `\usepackage[...]{hyperref}` 中, 在首次使用之前, 等等。

在下面的键描述 (key descriptions) 中, 许多选项不需要值, 因为如果使用, 它们的默认为值 `true`。这些是被归类为 “布尔型 (boolean)” 的。但是, 始终可以指定值 `true` 和 `false`。

5.1 一般选项

首先, 指定一般行为 (general behavior) 和页面尺寸 (page size) 的选项。

一般选项	数据类型	值	说明
<code>draft</code>	boolean	<i>false</i>	关闭 (turned off) 所有超文本选项 (hypertext options)
<code>final</code>	boolean	<i>true</i>	打开 (turned on) 所有超文本选项 (hypertext options)
<code>debug</code>	boolean	<i>false</i>	额外的诊断消息 (diagnostic messages) 打印在日志文件中
<code>verbose</code>	boolean	<i>false</i>	与调试 (debug) 相同
<code>implicit</code>	boolean	<i>true</i>	重新定义 L ^A T _E X 内部 (internals)
<code>setpagesize</code>	boolean	<i>true</i>	通过特殊驱动程序命令设置页面尺寸

5.2 目的地名称选项

目的地名称 (destinations names), 也包括锚点 (anchor)、目标 (target) 或链接 (link) 等的名称, 都是内部名称 (internal names), 用于标识 (identify) 文档页面中的位置。例如, 它们用于内部文档链接 (inner document links) 或书签 (bookmarks) 的链接目标 (link targets)。

如果调用 `\refstepcounter`, 通常会设置锚点 (anchor)。因此会有一个计数器名称和值 (counter name and value)。两者都用于构造目的地名称 (destination name)。默认情况下, 计数器值 (counter value) 跟在由句点 (dot) 分隔的计数器名称 (counter name) 后面。例如, (英文的) “第四章” 显示为: `chapter.4`。【译者注】: 要将 “chapter.4” 显示为中文的 “第四章”, 相应的汉化命令是: `\renewcommand{\chaptername}{第\thechapter章}`

此方案 (scheme) 用于:

`\autoref` 根据计数器名称 (counter name) 显示引用的描述标签 (description label)。

`\hyperpage` 被索引 (index) 用来获取页面链接 (page links)。页面锚点设置 (page anchor setting)(pageanchor) 不能被关闭。

目的地名称 (destination names) 的唯一性非常重要, 因为两个目的地不能共享相同的名称。计数器值 `\the<counter>` 对于计数器来说并不总是唯一的。例如, 表格 (table) 和图形 (figures) 可以在章中 (inside the chapter) 编号, 而不需要在其编号中包含章编号 (chapter number)。因此, `hyperref` 引入了 `\theH<counter>`, 它允许一个唯一的计数器值, 而不会扰乱计数器编号 (counter number) 的外观。例如, 第三章中第二个表格的编号可能排印为 2, 即 `\thetable` 的结果。但目的地名称 `table.2.4` 是唯一的, 因为它使用了 `\theHtable`, 在这种情况下给出了 2.4。

通常, 用户不需要设置 `\theH<counter>`。提供了标准案例 (standard cases)(chapter, ...)

的默认值。并且, 在加载了 `hyperref` 之后, 如果使用了 `amsmath` 宏包的 `\newcounter`、`\@addtoreset` 或 `\numberwithin`, 则带有父计数器 (parent counters) 的新计数器也会自动定义 `\theH<counter>`。

通常, 目的地名称 (destination names) 重复的问题可以通过适当定义 `\theH<counter>` 来解决。如果选项 `hypertextnames` 被禁用, 那么将使用唯一的人工数 (artificial number) 而不是计数器值 (counter value)。如果是页面锚点 (page anchors), 则使用绝对页面锚点 (absolute page anchor)。通过选项 `plainpages`, 页面锚点使用阿拉伯语形式 (arabic form)。在后两种情况中, 索引链接 (index links) 的 `\hyperpage` 都会受到影响, 可能无法正常工作。

如果一个未编号的实体 (unnumbered entity) 得到了一个锚点 (chapters、sections、... 的星形形式) 或使用了 `\phantomsection`, 则使用伪计数器名称 (dummy counter name) `section*` 和一个人工唯一编号 (artificial unique number)。

如果最终的 PDF 文件将与另一个文件合并, 那么目的地名称 (destination names) 可能会发生冲突, 因为两个文档可能都包含 `chapter.1` 或 `page.1`。此外, `hyperref` 在文档开头设置名为 `Doc-Start` 的锚点 (anchor)。这可以通过重新定义 `\HyperDestNameFilter` 来解决。`hyperref` 宏包每次调用此宏, 它使用了一个目的地名称。该宏必须是可展开的 (expandable), 并且只需要目的地名称作为参数。例如, 宏被重新定义为向所有目的地名称添加前缀 (prefix):

```
\renewcommand*{\HyperDestNameFilter}[1]{\jobname-#1}
```

在文档 `docA` 中, 目的地名称 `chapter.2` 变为 `docA-chapter.2`。

目的地名称也可以在 URIs 中从外部使用 (如果驱动程序没有删除或更改它们), 例如:

```
http://somewhere/path/file.pdf#nameddest=chapter.4
```

然而, 使用一个编号 (number) 似乎很不愉快。如果之前添加了另一章, 则编号会发生变化。但是, 很难将目的地的新名称传递给通常隐藏在内部的锚点设置过程 (anchor setting process)。锚点设置后的 `\label` 的第一个名称似乎是一个很好的近似值 (approximation):

```
\section{Introduction}  
\label{intro}
```

选项 `destlabel` 检查每个 `\label`, 如果有新的目的地名称处于活动状态, 则用标签名称 (label name) 替换目的地名称 (destination name)。由于锚点设置, 目的地名称已在使用中, 因此新名称将记录在 `.aux` 文件中, 并在随后的 `LATEX` 运行 (run) 中使用。重命名 (renaming) 是通过重新定义 `\HyperDestNameFilter` 来完成的。这样就保留了旧的目的地名称 (例如, `\autoref`

需要这些名称)。此重新定义也可用作 `\HyperDestLabelReplace`, 因此自己的重新定义 (own redefinition) 可以使用它。以下示例还为所有 (all) 目的地名称添加了前缀:

```
\renewcommand*{\HyperDestNameFilter}[1]{%
  \jobname-\HyperDestLabelReplace{#1}%
}
```

另一种情况是, 只有前缀为没有相应 `\label` 的文件才会更复杂, 因为 `\HyperDestLabelReplace` 需要未修改的目的地名称作为参数。这可以通过一个可展开的字符串测试 (pdfTeX 的 `\pdfstrcmp`, 或 XeTeX 的 `\strcmp`, pdfTeXcmds 宏包也支持 LuaTeX) 来解决:

```
\usepackage{pdfTeXcmds}
\makeatletter
\renewcommand*{\HyperDestNameFilter}[1]{%
  \ifcase\pdf@strcmp{#1}{\HyperDestLabelReplace{#1}} %
    \jobname-#1%
  \else
    \HyperDestLabelReplace{#1}%
  \fi
}
\makeatother
```

如果目的地尚未重命名, 还可以使用 `destlabel` 选项手动命名目的地:

```
\HyperDestRename{<destination>}{<newname>}
```

提示: 锚点 (anchors) 也可以由 `\hypertarget` 命名和设置。

目的地名称选项	数据类型	值	说明
<code>destlabel</code>	boolean	<i>false</i>	目的地 (destinations) 由锚点创建后的第一个 <code>\label</code> 命名
<code>hypertextnames</code>	boolean	<i>true</i>	对链接 (links) 使用可猜测的名称 (guessable names)
<code>naturalnames</code>	boolean	<i>false</i>	对链接使用 L ^A T _E X 计算的名称 (L ^A T _E X-computed names)
<code>plainpages</code>	boolean	<i>false</i>	强制使用页码 (page number) 的阿拉伯语 (Arabic) 形式而不是格式化 (formatted) 形式来命名页锚 (page anchors)

5.3 页面锚点

在 PDF 中, `hyperref` 为每一页的左上角添加了一个链接目标。这些目标用于诸如索引到条目页数之类的功能。与其他计数器生成的目标不同, 无法通过定义 `\theHpage` 命令来更改这些目标的名称: 目前无法将名称存储在标签中, 因此无法引用。尽管如此, 有一些选项可供选择, 在文档其他地方已经提到, 但在此提供一个总结:

pageanchor 一个布尔选项, 确定是否为每一页的左上角添加一个目标。如果关闭此选项, `\printindex` 将不包含有效的超链接。

hypertextnames 默认情况下, 目标使用 `\thepage` 构建名称: `page.\thepage`, 例如 `page.4` 或 `page.iii`。名称要求每一页有一个唯一的数字表示。一个常见的问题是标题页通常不显示页码, 但在内部使用与后续页面相同的数字。如果出现 `destination with the same identifier (namepage.1)` 的错误信息, 请更改标题页的数字表示方式或禁用页面目标。

如果布尔选项 `hypertextnames` 设置为 `false`, 将步进一个内部页计数器并将其用作阿拉伯数字。在大多数情况下, 这应该意味着您获得绝对页码 (异常情况可能出现在文档在 `shipout` 时丢弃或复制页面的情况下)。此选项避免了重复标识符的问题, 但在索引中不适用, 因为需要从传递给索引的表示中创建反向链接。

plainpages 这会强制页面锚点以阿拉伯数字形式命名, 但与前一个选项不同, 它不使用内部计数器而是使用页面计数器。这意味着, 例如, 如果您有罗马数字和阿拉伯数字页码, 将会得到重复的目标名称, 因为页面 `i` 和页面 `1` 都会设置为锚点 `page.1`。

`\@currentHpage`

从版本 7.01c 开始, `hyperref` 将刚刚设置的页面目标名称存储在 (全局) 变量 `\@currentHpage` 中。后备值为 `Doc-Start`。由于目标是在页面 `shipout` 时设置的, `\@currentHpage` 只能在 `shipout` 时可靠。从 2023-11-01 开始的 L^AT_EX 格式可以这样标记名称:

```
% 文档中的标记
\RecordProperties{mylabel}{pagetarget}

% 链接到页面:
\hyperlink{\RefProperty{mylabel}{pagetarget}}{一些文本}
```

5.4 配置选项

配置选项	数据类型	值	说明
<code>raiselinks</code>	boolean	<i>true</i>	在 <code>hypertex</code> 驱动程序中, 链接的高度通常由驱动程序简单地计算为包含文本的基线; 该选项强制 <code>\special</code> 命令反映链接的实际高度 (可能包含图形)
<code>breaklinks</code>	boolean	<i>both</i>	这个选项在 <code>hyperref</code> 中仅用于 <code>dviwindo</code> 驱动程序, 在所有其他情况下, 它不做任何明智的事情—它既不允许也不防止链接被破坏。 <code>Ocgx2</code> 宏包检查布尔值的状态。
<code>pageanchor</code>	boolean	<i>true</i>	确定是否在左上角为每一页提供隐式 (implicit) 锚点。如果关闭此选项, <code>\printindex</code> 将不包含有效的超链接。
<code>nesting</code>	boolean	<i>false</i>	允许嵌套链接; 目前没有驱动程序支持此功能。

选项 `breaklinks` 的注意事项: 正确的值是根据驱动程序功能自动设置的。对于不支持断开链接 (broken links) 的驱动程序, 它可以被覆盖。但是, 在任何情况下, 链接区域 (link area) 都会出错并被替换。

5.5 后端驱动程序

如果未指定驱动程序 (driver), 则宏包会尝试按以下顺序查找驱动程序:

1. 自动检测 (autodetection), 可以在 $\text{T}_{\text{E}}\text{X}$ 宏级别 (`pdfTEX`、`XeTEX`、`VTEX`) 检测到一些 $\text{T}_{\text{E}}\text{X}$ 处理器 (processors)。
2. 选项 `driverfallback`。如果设置了此选项, 则其值将作为驱动程序选项 (driver option)。
3. 宏 `\Hy@defaultdriver`。宏接受一个驱动程序文件名 (没有文件扩展名)。
4. 宏包的默认值是 `hypertex`。

许多发行版 (distributions) 都使用驱动程序文件 `hypertex.cfg`, 该驱动程序文件用 `hdvips` 来定义 `\Hy@defaultdriver`。建议这样做, 因为驱动程序 `dvips` 为 PDF 生成 (PDF generation) 提供了比 `hypertex` 多得多的功能。

驱动程序名	说明
<code>driverfallback</code>	如果未给定或未自动检测到驱动程序，则其值将用作驱动程序选项
<code>dvipdfm</code>	设置 <code>hyperref</code> 以便与 <code>dvipdfm</code> 驱动程序一起使用
<code>dvipdfmx</code>	设置 <code>hyperref</code> 以便与 <code>dvipdfmx</code> 驱动程序一起使用
<code>dvips</code>	设置 <code>hyperref</code> 以便与 <code>dvips</code> 驱动程序一起使用
<code>dvipsone</code>	设置 <code>hyperref</code> 以便与 <code>dvipsone</code> 驱动程序一起使用
<code>dviwindo</code>	设置 <code>hyperref</code> 以便与 <code>dviwindo</code> Windows 预览器 (previewer) 一起使用
<code>hypertex</code>	设置 <code>hyperref</code> 以便与 HyperTeX 兼容的驱动程序一起使用
<code>latex2html</code>	为了与 <code>latex2html</code> 兼容，重新定义了一些宏
<code>nativepdf</code>	<code>dvips</code> 的别名
<code>pdfmark</code>	<code>dvips</code> 的别名
<code>pdftex</code>	设置 <code>hyperref</code> 以便与 <code>pdftex</code> 程序一起使用
<code>ps2pdf</code>	为了与 Ghostscript 的 PDF 编写器兼容，重定义了一些宏，否则与 <code>dvips</code> 相同
<code>tex4ht</code>	用于 TeX4ht
<code>textures</code>	用于 Textures
<code>vtex</code>	为了与 MicroPress 的 VTeX 一起使用，PDF 和 HTML 后端将被自动检测到
<code>vtexpdfmark</code>	用于 VTeX 的 PostScript 后端
<code>xetex</code>	用于 XeTeX (使用 <code>dvipdfm</code> 的后端)

如果使用 `dviwindo`，可能需要重新定义宏 `\wwwbrowser` (默认为 `\wwwbrowser`) 来告诉 `dviwindo` 要启动什么程序 (program)。因此，Internet Explorer 的用户可能会在 `hyperref.cfg` 中添加以下内容：

```
\renewcommand{\wwwbrowser}{C:\string\Program\space
Files\string\Plus!\string\Microsoft\space
Internet\string\iexplore.exe}
```

5.6 扩展选项

扩展选项名	数据类型	值	说明
extension	text		设置文件的扩展名 (例如 dvi), 如果使用 xr 宏包, 它将被追加到创建的文件链接 (file links) 中
hyperfigures	boolean		
backref	text	<i>false</i>	将“反向链接 (backlink)”文本添加到参考文献中每个条目的末尾, 作为节编号 (section numbers) 列表。这只有在每个 \bibitem 后面有一个空行时才能正常工作。支持的值为 section、slide、page、none 或 false。如果未给定任何值, 则将 section 作为默认值。
pagebackref	boolean	<i>false</i>	将“反向链接 (backlink)”文本添加到参考文献中每个条目的末尾, 作为页码 (page numbers) 列表。
hyperindex	boolean	<i>true</i>	将索引条目 (index entries) 的页码制成超链接。在唯一的页面锚点 (pageanchor, ...) 上中继 pageanchors 和 plainpages=false
hyperfootnotes	boolean	<i>true</i>	将脚注标记 (footnote marks) 制作为指向脚注文本 (footnote text) 的超链接。很容易断开 ...
encap			为超级索引 (hyperindex) 设置封装字符 (encap character)
linktoc	text	<i>section</i>	在 TOC(目录)、LOF(图目当) 和 LOT(表目录) 上链接文本 (section)、页码 (page)、两者 (all) 或无 (none)
linktocpage	boolean	<i>false</i>	在 TOC(目录)、LOF(图目当) 和 LOT(表目录) 上链接页码 (page number) 而不是文本 (text)
breaklinks	boolean	<i>false</i>	通过将多行链接转换为指向同一目标的 PDF 链接, 允许链接换行

(后续下表)

(前接上表)

扩展选项名	数据类型	值	说明
colorlinks	boolean	<i>false</i>	为链接和锚点的文本着色。选择的颜色取决于链接的类型。目前唯一可以区分的链接类型是引用 (citations)、页面引用 (page references)、URL、本地文件引用 (local file references) 和其他链接。与彩色盒子不同，彩色文本在打印时保持不变
linkcolor	color	<i>red</i>	普通内部链接 (normal internal links) 的颜色
anchorcolor	color	<i>black</i>	锚点文本 (anchor text) 的颜色。被大多数驱动程序忽略
citecolor	color	<i>green</i>	文本中参考文献引用 (bibliographical citations) 的颜色
filecolor	color	<i>cyan</i>	打开本地文件的 URL 的颜色
menucolor	color	<i>red</i>	Acrobat 菜单项的颜色
runcolor	color	<i>filecolor</i>	运行链接的颜色 (启动注释)
urlcolor	color	<i>magenta</i>	链接 URL 的颜色
allcolors	color		设置所有颜色选项 (不带边框和字段选项)
frenchlinks	boolean	<i>false</i>	链接使用小写字母 (small caps) 代替颜色
hidelinks			隐藏链接 (删除颜色和边框)

请注意，在使用之前必须按照标准 L^AT_EX color 宏包的正常系统 (normal system) 定义所有颜色名称 (color names)。

5.7 特定于 PDF 的显示选项

特定于PDF的显示选项	数据类型	值	说明
bookmarks	boolean	<i>true</i>	以类似于目录的方式编写一组 Acrobat 书签, 需要运行两次 L ^A T _E X。对书签文件 (扩展名为 .out) 进行的一些后处理可能需要转换 L ^A T _E X 代码, 因为书签必须用 PDFEncoding 编写。为了帮助这个过程, .out 文件不会被 L ^A T _E X 重写, 如果它被编辑为包含一行 \let\WriteBookmarks\relax。
bookmarksopen	boolean	<i>false</i>	如果要求使用 Acrobat 书签, 则显示它们, 并展开所有子树。
bookmarksopenlevel	parameter		书签打开的级别 (\maxdimen)
bookmarksnumbered	boolean	<i>false</i>	如果要求使用 Acrobat 书签, 请包括章节编号。
bookmarkstype	text	<i>toc</i>	指定要模仿哪个 “toc” 文件
CJKbookmarks	boolean	<i>false</i>	此选项应用于生成 CJK 书签。hyperref 宏包支持 CJK 宏包的正常模式和预处理模式; 在创建书签的过程中, 它只是用特殊版本替换 CJK 的宏, 这些版本可以展开到相应的字符代码。请注意, 如果没有 hyperref 的 “unicode” 选项, 实际上您会得到不符合 PDF 规范的 PDF 文件, 因为使用了非 unicode 字符代码 — 一些为 CJK 语言本地化的 PDF 阅读器 (最值得注意的是 Acrobread 本身) 支持这一点。还要注意, 选项 “CJKbookmarks” 不能与选项 “unicode” 一起使用。没有提供将非 unicode 书签转换为 unicode 的机制; 对于可移植的 PDF 文档, 只应使用 Unicode 编码。
pdfhighlight	name	<i>/I</i>	选择链接按钮时的行为; /I 是反向的 (默认值); 其他可能性是 /N (无效果)、/O (大纲) 和 /P (插入高亮显示)。
citebordercolor	RGB color	<i>0 1 0</i>	引文周围边框的颜色
filebordercolor	RGB color	<i>0 .5 .5</i>	指向文件的链接周围边框的颜色

(后续下表)

(前接上表)

特定于PDF的显示选项	数据类型	值	说明
<code>linkbordercolor</code>	RGB color	<code>1 0 0</code>	正常链接周围边框的颜色
<code>menubordercolor</code>	RGB color	<code>1 0 0</code>	Acrobat 菜单链接周围边框的颜色
<code>urlbordercolor</code>	RGB color	<code>0 1 1</code>	URL 链接周围边框的颜色
<code>runbordercolor</code>	RGB color	<code>0 .7 .7</code>	“运行 (run)” 链接周围边框的颜色
<code>allbordercolors</code>			设置所有边框颜色选项
<code>pdfborder</code>		<code>0 0 1</code>	链接周围边框的样式; 默认情况下, 框的线条厚度为 1pt, 但 <code>colorlinks</code> 选项会将其重置为不产生边框。

链接边框的颜色过去仅指定为 0.1 范围内的 3 个数字, 给出一个 RGB 颜色。自版本 6.76a 以来, 如果已经加载了 `xcolor` 宏包, 则可以使用 `(x)color` 宏包的常用颜色规范 (color specifications)。有关更多信息, 请参阅 `hycolor` 宏包的描述 (description)。

书签命令 (bookmark commands) 存储在一个名为 `jobname.out` 的文件中。这些文件不由 \LaTeX 处理, 因此任何标记 (markup) 都会通过。您可以根据需要对该文件进行后处理; 为此, `.out` 文件在下次 \TeX 运行时不会被覆盖, 如果它被编辑为包含行:

```
\let\WriteBookmarks\relax
```

5.8 PDF 显示和信息选项

PDF 显示和信息选项	数据类型	值	说明
<code>baseurl</code>	URL		设置 PDF 文档的基本 URL
<code>pdfpagemode</code>	name	<code>empty</code>	确定如何在 Acrobat 中打开文件; 可能的模式 (mode) 为 <code>UseNone</code> 、 <code>UseThumbs</code> (显示缩略图)、 <code>UseOutlines</code> (显示书签)、 <code>FullScreen</code> 、 <code>UseOC</code> (PDF1.5) 和 <code>UseAttachments</code> (PDF1.6)。如果明确选择了无模式 (no mode), 但设置了书签选项, 则使用 <code>UseOutlines</code> 。

(后续下表)

(前接上表)

PDF 显示和信息选项	数据类型	值	说明
pdftitle	text		设置文档信息 “Title” (标题) 字段
pdfauthor	text		设置文档信息 “Author” (作者) 字段
pdfsubject	text		设置文档信息 “Subject” (主题) 字段
pdfcreator	text		设置文档信息 “Creator” (创建者) 字段
pdfcreationdate	date		设置创建日期, 请参阅下表了解格式说明。
pdfmoddate	date		设置修改日期, 请参阅下表了解格式说明。
addtopdfcreator	text		将其他文本添加到文档信息 “Creator” (创建者) 字段
pdfkeywords	text		设置文档信息 “Keywords” (关键字) 字段
pdftrapped	name	<i>empty</i>	设置文档信息 Trapped 条目。可能的值为 True、False 和 Unknown。空值 (empty value) 表示未设置条目。
pdfinfo	key value list	<i>empty</i>	用于设置文档信息的替代接口
pdfview	name	<i>XYZ</i>	为每个链接设置默认的 PDF “视图 (view)”
pdfstartpage	integer	<i>1</i>	确定打开 PDF 文件的页面。空值表示未设置条目。
pdfstartview	name	<i>Fit</i>	设置启动页面视图
pdfremotestartview	name	<i>Fit</i>	设置远程 PDF 文件的启动页面视图
pdfpagescrop	n n n n		设置页面的默认 PDF 裁剪框。这应该是一组四个数字
pdfcenterwindow	boolean	<i>false</i>	将文档窗口定位在屏幕中央
pdfdirection	name	<i>empty</i>	方向 (direction) 设置。可能的值: L2R(从左到右) 和 R2L(从右到左)
pdfdisplaydoctitle	boolean	<i>false</i>	在标题栏中显示文档标题而不是文件名

(后续下表)

(前接上表)

PDF 显示和信息选项	数据类型	值	说明
pdfduplex	name	<i>empty</i>	打印对话框的纸张处理 (paper handling) 选项。可能的值是: Simplex(单面打印), DuplexFlipShortEdge(双面打印并在纸张的短边上翻转), DuplexFlipLongEdge(双面打印并在纸张的长边上翻转)
pdffitwindow	boolean	<i>false</i>	调整文档窗口大小以适应文档大小
pdflang	name	<i>relax</i>	PDF 语言标识符 (RFC 3066)
pdfmenubar	boolean	<i>true</i>	使 PDF 查看器的菜单栏可见
pdfnewwindow	boolean	<i>false</i>	使打开另一 PDF 文件的链接启动一个新窗口
pdfnonfullscreenpagemode	name	<i>empty</i>	退出全屏模式时的页面模式设置。可能的值为: UseNone、UseOutlines、UseThumbs 和 UseOC
pdfnumcopies	integer	<i>empty</i>	打印份数
pdfpagelayout	name	<i>empty</i>	设置 PDF 页面的布局 (layout of PDF pages)。可能的值如下: SinglePage、OneColumn、TwoColumnLeft、TwoColumnRight、TwoPageLeft、TwoPageRight
pdfpagelabels	boolean	<i>true</i>	设置 PDF 页面标签
pdfpagetransition	name	<i>empty</i>	设置 PDF 页面过渡样式。可能的值为: Split、Blinds、Box、Wipe、Dissolve、Glitter、R、Fly、Push、Cover、Uncover、Fade。根据 PDF Reference, 默认值是 R, 它只是用新页面替换旧页面。
pdfpicktraybypdfsize	boolean	<i>false</i>	指定 PDF 页面尺寸是否用于选择打印对话框中的输入纸盘

(后续下表)

(前接上表)

PDF 显示和信息选项	数据类型	值	说明
pdfprintarea	name	<i>empty</i>	设置查看器首选项的 /PrintArea。可能的值为: MediaBox、CropBox、BleedBox、TrimBox 和 ArtBox。根据 PDF Reference, 默认值是 CropBox
pdfprintclip	name	<i>empty</i>	设置查看器首选项的 /PrintClip。可能的值为: MediaBox、CropBox、BleedBox、TrimBox 和 ArtBox。根据 PDF Reference, 默认值是 CropBox
pdfprintpagerange	n n (n n)*	<i>empty</i>	设置查看器首选项的 /PrintPageRange
pdfprintscaling	name	<i>empty</i>	打印对话框的页面缩放选项 (查看器首选项的选项 /PrintScaling, PDF 1.6); 有效值为 None 和 AppDefault
pdftoolbar	boolean	<i>true</i>	make PDF toolbar visible
pdfviewarea	name	<i>empty</i>	设置查看器首选项的 /ViewArea。可能的值为: MediaBox、CropBox、BleedBox、TrimBox 和 ArtBox。根据 PDF Reference, 默认值是 CropBox
pdfviewclip	name	<i>empty</i>	设置查看器首选项的 /ViewClip。可能的值为: MediaBox、CropBox、BleedBox、TrimBox 和 ArtBox。根据 PDF Reference, 默认值是 CropBox
pdfwindowui	boolean	<i>true</i>	使 PDF 用户接口元素可见
unicode	boolean	<i>true</i>	Unicode 编码的 PDF 字符串

通常情况下, `CreationDate` 和 `ModDate` 会在编译开始时自动设置为当前日期/时间。如果需要更改它们 (例如用于回归测试以生成可重现的文档), 可以使用 `\hypersetup` 设置 `pdfcreationdate`、`pdfmoddate` 键, 或通过设置 `SOURCE_DATE_EPOCH` 环境变量进行外部设置。

```
\hypersetup{pdfcreationdate=D:20010101205959-00'00'}
```

格式应为 PDF 格式的完整日期/时间，例如以下格式（数字可以自行更改）：

D:20010101205959-00'00'

D:20010101205959+00'00'

D:20010101205959Z

Acrobat 中的每个链接都有自己的放大级别 (magnification level)，该级别是使用 PDF 坐标空间 (coordinate space) 设置的，它与 \TeX 的坐标空间不同。单位为 bp，原点 (origin) 在左下角 (lower left corner)。另请参阅 `\hypercalcbp`，在第 46 页中对 `\hypercalcbp` 进行了解释。`pdf \TeX` 通过提供 XYZ (水平 \times 垂直 \times 缩放) 和 FitBH 的默认值来工作。但是，使用 `pdfmark` 的驱动程序不提供默认值，因此 `hyperref` 会传入 (passes in) -32768 的值，这会导致 Acrobat 设置 (通常) 合理的默认值。以下是 `pdfview`、`pdfstartview` 和 `pdfremotestartview` 参数的可能值。

名称	值	说明
XYZ	<i>left top zoom</i>	设置坐标和缩放因子。如果其中任何一个为 null，则使用源链接值。 <i>null null null</i> 将给出与当前页面相同的值。
Fit		使页面适合窗口
FitH	<i>top</i>	使页面的宽度与窗口相适应
FitV	<i>left</i>	使页面的高度与窗口相适应
FitR	<i>left bottom right top</i>	使四个坐标指定的矩形与窗口相适应
FitB		使页面边界框适合窗口
FitBH	<i>top</i>	使页面边界框的宽度与窗口相适应
FitBV	<i>left</i>	使页面边界框的高度与窗口相适应

`pdfpagelayout` 可能的值：

<code>pdfpagelayout</code> 的值	说明
SinglePage	显示单个页面；向前翻页 (advancing flips the page)。
OneColumn	以一系列 (column) 形式显示文档；连续滚动 (continuous scrolling)。
TwoColumnLeft	以两列形式显示文档，奇数页 (odd-numbered pages) 位于左侧。

TwoColumnRight	以两列形式显示文档，奇数页 (odd-numbered pages) 位于右侧。
TwoPageLeft	显示两页，左侧为奇数页 (从 PDF 1.5 开始)。
TwoPageRight	显示两页，右侧为奇数页 (从 PDF 1.5 开始)。

最后，pdfpagetransition 可以是以下值之一，其中 /Di 表示以度 (degrees) 为单位的运动方向 (direction of motion)，通常以 90 度为单位 (steps)，/Dm 表示水平 (/H) 或垂直 (/V) 维度 (例如 Blinds /Dm /V)，/M 表示运动，无论是向内 (/I) 还是向外 (/O)。

选项名	值	说明
Blinds	/Dm	多行均匀地分布在屏幕上，沿同一方向扫描，以显示新页面
Box	/M	一个盒子扫进扫出 (A box sweeps in or out)
Dissolve		页面图像以逐段的方式 (piecemeal fashion) 分解以显示新页面。
Glitter	/Di	类似于 Dissolve (溶解)，除了效果扫过屏幕。
Split	/Dm /M	两行 (lines) 横扫屏幕，显示新页面。
Wipe	/Di	一行 (lines) 横扫屏幕，显示新页面。
R		只需将旧页面替换为新页面即可。
Fly	/Di /M	除非 /Di 为 None，否则更改 (changes) 将以 /Di 指定的方向飞出或飞入 (由 /M 指定) 屏幕外的位置。
Push	/Di	当新页面滑入时，旧页面从屏幕上滑出，将旧页面按照 /Di 指定的方向推出。
Cover	/Di	新页面按照 /Di 指定的方向滑动到屏幕上，覆盖旧页面。
Uncover	/Di	旧页面按照 /Di 指定的方向从屏幕上滑出，并按照 /Di 指定的方向展开新页面。
Fade		新页面通过旧页面逐渐变得可见。

5.9 pdfinfo 选项

可以使用 pdftitle、pdfsubject、... 等设置信息条目 (information entries)。pdfinfo 选项提供了一个替代接口。它需要一个键值列表 (key value list)。键名称是直接出现在 PDF 信息字典 (information dictionary) 中的名称。已知键，如 Title、Subject、Trapped 和其他键被

映射到选项 `pdftitle`、`subject`、`trapped`、...。未知键被添加到信息字典中。它们的值是文本字符串 (text strings)(请参阅 PDF 规范)。示例:

```
\hypersetup{
  pdfinfo={
    Title={My Title},
    Subject={My Subject},
    NewKey={Foobar},
    % ...
  }
}
```

5.10 完整的选项列表，按字母顺序排列

下面是一个可用的 `hyperref` 选项的完整列表，按字母顺序 (alphabetically) 排列:

hyperref 的全部选项	值	说明
<code>allbordercolors</code>		设置所有边框颜色选项 (border color options)
<code>allcolors</code>		设置所有颜色选项 (不带边框和字段选项)
<code>anchorcolor</code>	<i>black</i>	设置锚点的颜色，被大多数驱动程序忽略。
<code>backref</code>	<i>false</i>	做回书目的引用 (do bibliographical back references)
<code>baseurl</code>	<i>empty</i>	设置文档的基本 URL
<code>bookmarks</code>	<i>true</i>	制作书签 (make bookmarks)
<code>bookmarksnumbered</code>	<i>false</i>	将章节编号 (section numbers) 放入书签
<code>bookmarksopen</code>	<i>false</i>	打开书签树 (bookmark tree)
<code>bookmarksopenlevel</code>	<code>\maxdimen</code>	书签打开的级别 (level)
<code>bookmarkstype</code>	<i>toc</i>	指定要模仿 (mimic) 哪个 “toc” 文件
<code>breaklinks</code>	<i>false</i>	允许链接换行
<code>CJKbookmarks</code>	<i>false</i>	生成 CJK 书签
<code>citebordercolor</code>	<i>0 1 0</i>	引用周围边界 (border around cites) 的颜色

(后续下表)

(前接上表)

hyperref 的全部选项	值	说明
citecolor	<i>green</i>	引用链接 (citation links) 的颜色
colorlinks	<i>false</i>	彩色链接 (color links)
	<i>true</i>	(tex4ht, dviwindo)
debug	<i>false</i>	提供所定义锚 (anchors) 的详细信息; 与 <i>verbose</i> 相同
destlabel	<i>false</i>	目的地 (destinations) 由锚点创建后的第一个 <code>\label</code> 命名
draft	<i>false</i>	不要进行任何超链接 (hyperlinking)
driverfallback		如果未指定或检测到驱动程序, 则默认
dvipdfm		使用 dvipdfm 后端 (backend)
dvipdfmx		使用 dvipdfmx 后端 (backend)
dvips		使用 dvips 后端 (backend)
dvipsone		使用 dvipsone 后端 (backend)
dviwindo		使用 dviwindo 后端 (backend)
encap		为超索引 (hyperindex) 设置封装字符 (encap character)
extension	<i>dvi</i>	链接文件的后缀 (suffix of linked files)
filebordercolor	<i>0 .5 .5</i>	文件链接周围边框的颜色
filecolor	<i>cyan</i>	文件链接的颜色
final	<i>true</i>	<i>draft</i> (草案) 选项的反面 (opposite)
frenchlinks	<i>false</i>	链接使用小型大写字母 (small caps) 代替颜色
hidelinks		隐藏链接 (删除颜色和边框)
hyperfigures	<i>false</i>	将图片 (figures) 用作超链接
hyperfootnotes	<i>true</i>	设置超链接脚注 (hyperlinked footnotes)
hyperindex	<i>true</i>	设置超链接索引 (hyperlinked indices)
hypertex		使用 HyperTeX 后端 (backend)

(后续下表)

(前接上表)

hyperref 的全部选项	值	说明
hypertextnames	<i>true</i>	将可猜测的名称 (guessable names) 用作超链接
implicit	<i>true</i>	重新定义 L ^A T _E X 内部结构 (internals)
latex2html		使用 L ^A T _E X2HTML 后端 (backend)
linkbordercolor	<i>1 0 0</i>	链接周围边框的颜色
linkcolor	<i>red</i>	链接的颜色
linktoc	<i>section</i>	使文本成为 TOC (目录)、LOF (图目录) 和 LOT (表目录) 上的链接
linktocpage	<i>false</i>	使页码而不是文本, 成为 TOC (目录)、LOF (图目录) 和 LOT (表目录) 上的链接
menubordercolor	<i>1 0 0</i>	菜单链接 ((menu links)) 周围边框的颜色
menucolor	<i>red</i>	菜单链接 (menu links) 的颜色
nativepdf	<i>false</i>	dvips 的别名
naturalnames	<i>false</i>	使用 L ^A T _E X 计算的名称作为链接
nesting	<i>false</i>	允许嵌套链接
next-anchor		允许设置下一个锚点 (anchor) 的名称
pageanchor	<i>true</i>	在每一页中加一个锚点 (anchor)
pagebackref	<i>false</i>	按页码反向引用 (backreference)
pdfauthor	<i>empty</i>	PDF 作者字段 (Author field) 的文本
pdfborder	<i>0 0 1</i> <i>0 0 0</i>	PDF 链接边框的宽度 (colorlinks)
pdfborderstyle		链接的边框样式 (border style)
pdfcenterwindow	<i>false</i>	将文档窗口放置在屏幕中央
pdfcreator	<i>LaTeX with hyperref</i>	PDF 创建者字段 (Creator field) 的文本
pdfcreationdate		设置创建日期, 请参阅第 5.8 节中的表格, 了解更多关于格式的信息。
pdfdirection	<i>empty</i>	方向设置 (direction setting)

(后续下表)

(前接上表)

hyperref 的全部选项	值	说明
pdfdisplaydoctitle	<i>false</i>	在标题栏中显示文档标题而不是文件名
pdfduplex	<i>empty</i>	打印对话框的纸张处理选项
pdffitwindow	<i>false</i>	调整文档窗口大小以适应文档大小
pdfhighlight	<i>/I</i>	设置 PDF 链接的高亮显示 (highlighting)
pdfinfo	<i>empty</i>	用于设置文档信息的替代接口
pdfkeywords	<i>empty</i>	PDF 关键字字段 (Keywords field) 的文本
pdflang	<i>relax</i>	PDF 语言标识符 (RFC 3066)
pdfmark	<i>false</i>	dvips 的一个别名
pdfmenubar	<i>true</i>	使 PDF 查看器 (viewer) 的菜单栏可见
pdfmoddate		设置修改日期, 请参阅第 5.8 节中的表格, 了解更多关于格式的信息。
pdfnewwindow	<i>false</i>	建立打开另一个 PDF 的链接 文件启动一个新窗口
pdfnonfullscreenpagemode	<i>empty</i>	退出全屏模式时的页面模式 (page mode) 设置
pdfnumcopies	<i>empty</i>	打印份数
pdfpagelabels	<i>true</i>	设置 PDF 页面标签 (page labels)
pdfpagelayout	<i>empty</i>	设置 PDF 页面的布局 (layout)
pdfpagemode	<i>empty</i>	设置 PDF 显示的默认模式 (default mode)
pdfpagescrop	<i>empty</i>	设置 PDF 文档的裁剪尺寸 (crop size)
pdfpagetransition	<i>empty</i>	设置 PDF 页面过渡样式 (transition style)
pdfpicktraybypdfsize	<i>empty</i>	设置打印对话框选项
pdfprintarea	<i>empty</i>	设置查看器首选项 (viewer preferences) 的 /PrintArea
pdfprintclip	<i>empty</i>	设置查看器首选项 (viewer preferences) 的 /PrintClip
pdfprintpagerange	<i>empty</i>	设置查看器首选项 (viewer preferences) 的 /PrintPageRange

(后续下表)

(前接上表)

hyperref 的全部选项	值	说明
pdfprintscaling	<i>empty</i>	打印对话框的页面缩放选项
pdfproducer	<i>empty</i>	PDF 创建者字段 (Producer field) 的文本
pdfremotestartview	<i>Fit</i>	远程 PDF 文档的起始视图 (starting view)
pdfstartpage	<i>1</i>	打开 PDF 文档时的页面
pdfstartview	<i>Fit</i>	PDF 文档的起始视图 (starting view)
pdfsubject	<i>empty</i>	PDF 主题字段 (Subject field) 的文本
pdftex		使用 pdfT _E X 后端 (backend)
pdftitle	<i>empty</i>	PDF 标题字段 (Title field) 的文本
pdftoolbar	<i>true</i>	使 PDF 工具栏可见
pdftrapped	<i>empty</i>	设置文档信息 Trapped 条目。可能的值为 True、False、Unknown。空值 (empty value) 表示未设置该条目。
pdfview	<i>XYZ</i>	链接遍历 (traversal) 时的 PDF“视图 (view)”
pdfviewarea	<i>empty</i>	设置查看器首选项 (viewer preferences) 的 /ViewArea
pdfviewclip	<i>empty</i>	设置查看器首选项 (viewer preferences) 的 /ViewClip
pdfwindowui	<i>true</i>	使 PDF 用户接口元素 (interface elements) 可见
plainpages	<i>false</i>	将页码锚定为普通阿拉伯文 (plain Arabic)
ps2pdf		使用 ps2pdf 后端 (backend)
psdextra	<i>false</i>	为 PDF 字符串命令定义更多的短名称 (short names)
raiselinks	<i>false</i>	建立链接 (用于 HyperT _E X 后端)
runbordercolor	<i>0.7.7</i>	“run (运行)” 链接周围边框的颜色
runcolor	<i>filecolor</i>	“run (运行)” 的颜色
setpagesize	<i>true</i>	通过特殊的驱动程序命令设置页面尺寸
tex4ht		使用 T _E X4ht 后端 (backend)

(后续下表)

(前接上表)

hyperref 的全部选项	值	说明
<code>textures</code>		使用 Textures 后端 (backend)
<code>unicode</code>	<i>true</i>	Unicode 编码的 pdf 字符串, 从 v7.00g 版开始, 所有引擎默认设置为 <code>true</code> 。将加载 <code>puenc.def</code> 中的很多定义。对于 <code>pdflatex</code> , 它可以设置为 <code>false</code> , 但是不推荐这样做。
<code>urlbordercolor</code>	<i>0 1 1</i>	URL 链接周围边框的颜色
<code>urlcolor</code>	<i>magenta</i>	URL 链接的颜色
<code>verbose</code>	<i>false</i>	很随意的 (be chatty)
<code>vtex</code>		使作 VTeX 后端 (backend)
<code>xetex</code>		使用 XeTeX 后端 (backend)

6 其它用户宏

如果您需要引用 URL 或编写显式链接 (explicit links), 以下低级用户宏 (low-level user macros) 可供使用:

`\href[options]{URL}{text}`

text 被创建为一个到 *URL* 的超链接; 这必须是一个完整的 URL (相对于基本 URL, 如果定义了这个 URL 的话)。特殊字符 # 和 % 不需要以任何方式转义 (除非该命令用于另一个命令的参数中)。

可选参数 *options* 识别 hyperref 选项 pdfremotestartview、pdfnewwindow 和以下键值选项 (key value options):

page: 指定远程 (remote)PDF 文档的起始页码 (start page number)。第一页是1。

ismap: 布尔键 (Boolean key), 如果设置为true, 则 URL 应附加坐标 (coordinates), 作为 PDF 查看器的查询参数 (query parameters)。

nextactionraw: 动作字典 (action dictionaries) 的 /Next 键的值, 请参阅 PDF 规范。

`\url{URL}`

类似于 `\href{URL}{\nolinkurl{URL}}`。根据驱动程序的不同, `\href` 也尝试检测链接类型。因此结果可以是 URL 链接、文件链接等等。实现过程中使用了 `url` 宏包, 其命令可用于格式化和微调 URL, 但是 `\url` 的行为与 `url` 宏包中的行为不同: 最重要的是, `hyperref` 的 `\url` 会展开命令。如果需要原始 `\url` 的行为或原样输出 URL, 应先加载 `url` 宏包, 然后将该命令复制到某个别名命令中使用。

`\nolinkurl{URL}`

使用与上面描述的 `\url` 相同的方式编写 *URL*, 但不创建超链接 (hyperlink)。

`\hyperbaseurl{URL}`

建立了一个基本的 (base) *URL*, 该 *URL* 预先设置在其他指定的 *URL* 之前, 以便更容易地编写可移植文档 (portable documents)。在创建 PDF 时, 该命令只能使用一次, 因为 *URL* 被写入目录中。

`\hyperimage{imageURL}{text}`

使用 *text* 作为锚点 (anchor) 插入 URL 引用的图像的链接。对于生成 HTML 的驱动程序，浏览器将插入图像本身，并完全忽略 *text*。

`\hyperdef{category}{name}{text}`

标记文档的目标区域 (target area)(*text*)，并将其命名为 *category.name*

`\hyperref{URL}{category}{name}{text}`

将 *text* 作为到 *URL#category.name* 的链接

`\hyperref[label]{text}`

text 被制作成一个链接，链接到的位置与 `\ref{label}` 被链接的位置相同。

`\hyperlink{name}{text}`

`\hypertarget{name}{text}`

使用 `\hypertarget` 创建一个简单的内部链接 (internal link)，其中包含两个参数：锚点 (anchor) *name* 和锚点 *text*。`\hyperlink` 有两个参数，一个是由 `\hypertarget` 定义的超文本对象 (hypertext object) 的名称，另一个是用作页面上链接的 *text*。

注意，在 HTML 用语 (parlance) 中，`\hyperlink` 命令在每个链接前面插入一个理论上的 (notional) #，使其与当前测试文档 (testdocument) 相关；`\href` 需要一个完整的 URL。

`\phantomsection`

这将在该位置设置 (location) 一个锚点。它的工作原理类似于 `\hypertarget{}{}`，使用自动选择的锚点名称 (anchor name)。它通常与 `\addcontentsline` 一起用于类似小节的东西 (sectionlike things) 如索引 (index)、参考文献 (bibliography)、序言 (preface)。`\addcontentsline` 指的是设置锚点的上一个最新位置 (latest previous location)。例如：

`\cleardoublepage`

`\phantomsection`

```
\addcontentsline{toc}{chapter}{\indexname}

\printindex
```

现在，索引目录 (和书签) 中的条目指向索引页 (index page) 的开头，而不是指向该页之前的位置。

`\hyperget{anchor}{label} \hyperget{pageanchor}{label}`

这将以可展开的方式 (expandable way) 从标签 (label) 中检索锚点或页面锚点 (page anchor)。它将 `\HyperDestNameFilter` 考虑在内。例如，它可以与 `bookmark` 宏包中的 `\bookmark` 一起使用，以将目的地 (destination) 设置为标签：

```
\bookmark[dest=\hyperget{anchor}{sec}]{section}
```

当 `pageanchor` 从标签中检索页码时，它不能与选项 `plainpages` 一起使用。

`\hyperget{currentanchor}{}`

这将检索已设置的最后一个锚点 (anchor)。它也考虑了 `\HyperDestNameFilter`。

`\autoref{label}`

这是对通常的 `\ref` 命令的替换，该命令将上下文标签 (contextual label) 放在引用 (reference) 前面。这为用户点击超链接提供了更大的目标 (bigger target)，例如“section 2 (第 2 节)”，而不仅仅是编号“2”。

标签 (label) 是通过 `hyperref` 使用下表列出的宏 (显示为默认值) 从原始 `\label` 命令的上下文 (context) 中计算出来的。可以使用 `\(re)newcommand` 在文档中 (重新) 定义这些宏；请注意，其中一些宏在标准文档类 (standard document classes) 中已经定义好了。小写和大写首字母的混合是经过深思熟虑的，符合作者的做法。

对于下面的每个宏 (macro)，`hyperref` 会在 `*name` 之前检查 `*autorefname`。例如，它在 `\figurename` 之前查找 `\figureautorefname`。

宏 (Macro)	默认显示为	汉化 (在前言中重新定义) ♣	汉化后显示为 ♣
<code>\figurename</code>	Figure	<code>\renewcommand{\figurename}{\heiti 图}</code>	图
<code>\tablename</code>	Table	<code>\renewcommand{\tablename}{\heiti 表}</code>	表
<code>\partname</code>	Part	<code>\renewcommand{\partname}{\heiti 第\thepart部分}</code>	第一部分
<code>\appendixname</code>	Appendix	<code>\renewcommand{\appendixname}{\heiti 附录}</code>	附录
<code>\equationname</code>	Equation	<code>\renewcommand{\equationname}{\heiti 方程}</code>	方程

(后续下表)

(前接上表)

宏 (Macro)	默认为显示为	汉化 (在前言中重新定义) ♣	汉化后显示为 ♣
\Itemname	item	一般不要汉化它	
\chaptername	chapter	\renewcommand{\chaptername}{第\thechapter章}	第一章
\sectionname	section	\renewcommand{\sectionname}{第\thesection节}	第一节
\subsectionname	subsection	\renewcommand{\subsectionname}{第\thesubsection小节}	第一小节
\subsubsectionname	subsubsection	\renewcommand{\subsubsection}{第\thesubsubsection小小节}	第一小小节
\paragraphname	paragraph	\renewcommand{\paragraphname}{第\theparagraph段}	第一段
\Hfootnotename	footnote	一般不要汉化它	
\AMSname	Equation	\renewcommand{\AMSname}{\heiti 方程}	方程
\theoremname	Theorem	\renewcommand{\thmname}{\heiti 定理}	定理
\page	page	\renewcommand{\pagename}{\heiti 第\thepage页}	第一页
以下为译者添加			
\algorithm	Algorithm	\renewcommand{\abstractname}{\heiti 算法}	算法
\abstractname	Abstract	\renewcommand{\abstractname}{\heiti 摘要}	摘要
\indexname	Index	\renewcommand{\indexname}{\heiti 索引}	索引
\refname	Reference	\renewcommand{\refname}{\heiti 参考文献}	参考文献
\bibname	Reference	\renewcommand{\bibname}{\heiti 专著}	专著
\contentsname	Contents	\renewcommand{\contentsname}{\heiti 目录}	目录
\listfigurename	Lists of figures	\renewcommand{\listfigurename}{\heiti 插图目录}	插图目录
\listtablename	Lists of tables	\renewcommand{\listtablename}{\heiti 表格目录}	表格目录
\indexname	page	\renewcommand{\pagename}{\heiti 第\thepage页}	第一页

♣ 表格的这两列系译者添加

如果使用 babel，则重新定义的示例如下：

```
\usepackage[ngerman]{babel}
\addto\extrasngerman{%
  \def\subsectionautorefname{Unterkapitel}%
}
```

提示：\autoref 通过引用所基于的计数器名称工作。如果计数器用于不同的用途，则 \autoref 有时会选择错误的名称。例如，如果一个引理 (lemma) 与定理 (theorems) 共享一个计数器，那么 \newtheorem 就会发生这种情况。然后，aliascnt 宏包提供了一种方法来生成一个模拟的第二计数器 (simulated second counter)，该计数器允许区分定理和引理：

```
\documentclass{article}

\usepackage{aliascnt}
\usepackage{hyperref}

\newtheorem{theorem}{Theorem}
```

```

\newaliascnt{lemma}{theorem}
\newtheorem{lemma}[lemma]{Lemma}
\aliascntresetthe{lemma}

\providecommand*{\lemmaautorefname}{Lemma}

\begin{document}

We will use \autoref{a} to prove \autoref{b}.

\begin{lemma}\label{a}
  Nobody knows.
\end{lemma}

\begin{theorem}\label{b}
  Nobody is right.
\end{theorem}.

\end{document}

```

`\autopageref{label}`

它将替换 `\pageref`，并在页面引用 (page reference) 前面添加页面的名称。首先检查 `\pageautorefname`，然后检查 `\pagename`。

对于希望引用 (reference) 使用正确的计数器但不希望创建链接 (create a link) 的实例 (instances)，有星号形式 (starred forms)(即使 `hyperref` 已加载 `implicit=false`，这些星号形式也存在)：

`\ref*{label}`

`\pageref*{label}`

`\autoref*{label}`

`\autopageref*{label}`

一个典型的用途是写：

`\hyperref[other]{that nice section (\ref*{other}) we read before}`

我们希望 `\ref*{other}` 生成正确的数字 (number)，但不要形成链接 (link)，因为我们自己使用 `\hyperref` 来实现这一点。

`\pdfstringdef{macroname}{TEXstring}`

`\pdfstringdef` 返回一个包含 PDF 字符串的宏。(目前这是在全球范围内完成的，但不依赖于它。) 以下所有任务 (tasks)、定义 (definitions) 和重新定义 (redefinitions) 都是在一个组 (group) 中完成的，以保持它们的本地性 (local)：

- 切换到 PD1 或 PU 编码
- 定义“八进制序列命令 (octal sequence commands)” (`\345`): `\edef\3{\string\3}`
- T_EX 的特殊字形 (special glyphs): `\{`、`\%`、`\&`、`\space`、`\dots` 等。
- 国家字形 (national glyphs)(`german.sty`、`french.sty` 等)
- 徽标 (Logos): `\TeX`、`\eTeX`、`\MF` 等。
- 禁用一些命令，这些命令不提供在书签 (bookmarks) 中有用的功能: `\label`、`\index`、`\glossary`、`\discretionary`、`\def`、`\let` 等。
- L^AT_EX 的字体命令如 `\textbf` 等。
- 支持 `xspace` 宏包提供的 `\xspace`

此外，(圆) 括号 (parentheses) 受到保护，以避免 PDF 字符串中不安全的不平衡括号 (unsafe unbalanced parentheses) 的危险。有关更多详细信息，请参阅 Heiko Oberdiek (海科·奥伯迪克) 的 Euro T_EX 论文，该论文与 `hyperref` 一起分发。

```
\begin{NoHyper}...\end{NoHyper}
```

有时我们只是不希望这个糟糕的宏包干扰我们。定义一个我们可以手动放入的环境，或者包含在一个样式文件 (style file) 中，这样可以阻止超文本函数 (hypertext functions) 做任何事情。例如，在 Elsevier 类中，这被用来阻止 `hyperref` 在前面大肆破坏。

6.1 书签宏

6.1.1 设置书签

通常 `hyperref` 会自动为 `\section` 和类似的宏 (similar macros) 添加书签。但它们也可以手动设置。

```
\pdfbookmark[level]{text}{name}
```

创建一个具有指定文本 (specified text) 和给定级别 (given level) 的书签 (默认值为 0)。作为内部锚点 (internal anchor) 的名称使用 (与级别一起使用)。因此，名称必须是唯一的 (类似于 `\label`)。

```
\currentpdfbookmark{text}{name}
```

创建当前级别 (current level) 的书签。

```
\subpdfbookmark{text}{name}
```

在书签层次结构 (bookmark hierarchy) 中向下一步 (one step down) 创建书签。在内部，当前级别增加一。

```
\belowpdfbookmark{text}{name}
```

在当前书签级别 (current bookmark level) 以下创建书签。但是，在该命令之后，当前书签级别没有改变。

提示： `bookmark` 宏包⁶用一种新算法替换 `hyperref` 的书签组织 (bookmark organization)：

⁶ 译者注：译者已将该宏包的文档译成了中文，点击查看：[bookmark_ZH_CN.pdf](#)

- 通常只需要运行 (run) 一次 \LaTeX 。
- 对书签外观 (颜色、字体) 进行更多控制。
- 支持不同的书签操作 (外部文件链接、URL、...)。

因此, 我建议使用这个 bookmark 宏包。

6.1.2 替换宏

hyperref 从 `\section` 这样的命令的参数中获取书签的文本 (text for bookmarks), 这些命令可以包含数学 (math)、颜色 (colors) 或字体更改 (font changes) 等内容, 但这些内容都不会按原样显示在书签中。

`\texorpdfstring{ $T_{\text{E}}\text{Xstring}$ }{PDFstring}`

例如:

```
\section{Pythagoras:
  \texorpdfstring{$ a^2 + b^2 = c^2 $}{%
    a\texttt{two}superior\ + b\texttt{two}superior\ =
    c\texttt{two}superior
  }%
}
\section{\texorpdfstring{\textcolor{red}{}}{}{Red} Mars}
```

`\pdfstringdef` 在展开字符串之前执行 `\pdfstringdefPreHook` 这个钩子 (hook)。因此, 您可以使用这个钩子来执行其它的任务 (tasks) 或禁用其它的命令 (commands)。

```
\expandafter\def\expandafter\pdfstringdefPreHook
\expandafter{%
  \pdfstringdefPreHook
  \renewcommand{\mycommand}[1]{}%
}
```

然而, 要禁用命令, 一种更简单的方法是通过 `\pdfstringdefDisableCommands`, 它将其参数添加到 `\pdfstringdefPreHook` 的定义中 (在这里, “@” 可以用作命令名称中的字母):

```

\pdfstringdefDisableCommands{%
  \let~\textasciitilde
  \def\url{\pdfstringdefWarn\url}%
  \let\textcolor\@gobble
}

```

6.2 页面标签

`\thispdfpagelabel{page number format}`

这允许更改 PDF 查看器 (viewer) 工具栏中显示的特定页面页码的格式, 例如

```
\thispdfpagelabel{Empty Page-\roman{page}}
```

该命令会影响执行该命令的页面, 因此应该考虑异步分页 (asynchronous page breaking)。它应该用于例如 `\thispagestyle` 也可以使用地方。

6.3 实用程序宏

`\hypercalsbp{dimen specification}`

`\hypercalsbp` 获取 \TeX 尺寸规范 (*dimen specification*) 并将其转换为大点 (bp)⁷, 然后返回不带单位的数字。这对选项 `pdfview`、`pdfstartview` 和 `pdfremotestartview` 很有用。例好:

```

\hypersetup{
  pdfstartview={FitBH \hypercalsbp{\paperheight-\topmargin-1in
    -\headheight-\headsep}
}

```

PDF 坐标系的原点 (origin) 位于左下角 (lower left corner)。

注意, 对于计算 (calculations), 您需要 `calc` 宏或 $\varepsilon\text{-}\TeX$ 。如今, 后者应该自动被 \LaTeX 格式启用。没有 $\varepsilon\text{-}\TeX$ 的用户, 请查看源文档 `hyperref.dtx` 以了解进一步的限制 (limitations)。

⁷ 译者注: 在 \LaTeX 中, bp 即大点, $1\text{ bp} = 0.353\text{ mm} > 1\text{ pt}$

此外, `\hypercalcbp` 不能用于 `\documentclass` 和 `\usepackage` 的选项规范 (option specifications), 因为 \LaTeX 扩展了这些命令的选项列表 (option lists)。但是宏包 `hyperref` 尚未加载, 并且会出现未定义的控制序列错误 (undefined control sequence error)。

7 新功能⁸

7.1 “pdflinkmargin” 选项

“pdflinkmargin” 选项是一个实验性选项 (experimental option), 用于指定链接边距 (link margin)(如果驱动程序支持的话)。支持驱动程序 (supporting drivers) 的默认值为 1pt。

pdfTeX

- 链接区域 (link area) 也取决于周围的方框 (surrounding box)。
- 设置具有局部效果 (local effect)。
- 当页面发送出去 (shipped out) 时, pdfTeX 对页面上的所有链接使用链接边距 (link margin) 的当前设置。

pdfmark

- 设置具有全局效果 (global effect)。

xetex

- 设置必须在前言或第一页中完成, 然后具有全局效果。键 (key) 插入新的 (x)dvipdfmx 专用 (special) `\special{dvipdfmx:config g #1}`(已删除单位 [unit])。

其他驱动程序 不支持。

7.2 “calculatesortkey” 字段选项

默认情况下, 具有计算值 (calculated values) 的字段 (fields) 按文档顺序 (document order) 计算。如果计算字段值取决于文档中稍后出现的其他计算字段 (calculated fields), 则可以使用选项 “calculatesortkey” 指定正确的计算顺序 (calculation order)。它的值被用作对计算字段进行字典排序 (lexicographically sort) 的关键字 (key)。排序键 (sort key) 不需要是唯一的。共享同一关键字的字段按文档顺序排序。

目前, 字段选项 (field option) “calculatesortkey” 仅受 pdfTeX 的驱动程序支持。

⁸ 本节从自述文件 (README file) 中删除, 需要更多地集成到手册中。

7.3 “next-anchor” 选项

此选项允许覆盖下一个锚点的锚点名称 (anchor name)。这使得可以为例如目录的标题 (heading of the table of contents) 提供一个锚点名称, 该名称可以用来自 bookmark 宏包的\bookmark命令引用。

```
\hypersetup{next-anchor=toc}
\tableofcontents
\bookmark[dest=\HyperDestNameFilter{toc},level=section]{\contentsname}
```

7.4 “localanchorname” 选项

设置锚点时 (例如, 通过 \refstepcounter), 锚点名称将全局设置为当前锚点名称。
例如:

```
\section{Foobar}
\begin{equation}\end{equation}
\label{sec:foobar}
```

使用默认全局设置 (localanchorname=false), 对 “sec:foobar” 的引用会跳转到前面的等式 (equation)。使用选项 “localanchorname”, 在环境之后会忘记等式的锚点 (anchor), 引用 “sec:foobar” 会跳到节标题 (section title)。

“localanchorname” 选项是一个试验性选项 (experimental option), 可能会出现锚点名称 (anchor name) 不可用的情况。

该选项已被弃用 (deprecated): 如果不清楚 \@currentHref 是局部设置 (locally set) 的还是全局设置 (globally set) 的, 则宏包作者很难添加链接的目标 (targets for links)。

已弃用 2022-04-27
v7.00o

7.5 “customdriver” 选项

选项 “customdriver” 的值是不带扩展名 “.def” 的外部驱动程序文件 (external driver file) 的名称。该文件的 \ProvidesFile 的版本日期 (version date) 和版本号 (version number) 必须与 “hyperref” 的日期和编号匹配, 否则将发出警告。

因为接口 (interface) 没有定义好 (该接口需要在驱动程序中定义), 而且相当混乱, 因此该选项主要用于简化驱动程序部分 (driver part) 的开发 (developing)、测试 (testing) 和调试 (debugging)。

7.6 “psdextra” 选项

LaTeX 的新字体选择框架 (NFSS) 用于协助将任意 TeX 字符串转换为 PDF 字符串如书签 (bookmarks)、PDF 信息条目 (information entries)。许多数学命令名称 (\geq 、 \notin 、...) 不受 NFSS 的控制，因此它们是用前缀 “text” ($\text{\texttt{\textit{textgeq}}}$ 、 $\text{\texttt{\textit{textnotin}}}$ 、...) 定义的。在处理 PDF 字符串的过程中，它们可以映射到短名称 (mapped to short names)。缺点是它们有数百个宏，需要为每个 PDF 字符串转换 (PDF string conversion) 重新定义。因此，可以启用或禁用这个选项 “psdextra”。默认情况下，该选项处于关闭状态 (设置为 “false”)。启用该选项意味着可以使用短名称。然后可以直接使用 \geq 来代替 $\text{\texttt{\textit{textgeq}}}$ 。

7.7 $\text{\texttt{\textit{XeTeXLinkBox}}}$

当 XeTeX 生成一个链接注释 (link annotation) 时，它不会查看盒子 (boxes)(与其他驱动程序一样)，而只查看字符字形 (character glyphs)。如果没有字形 (图像 [images]、线段 [rules]、...)，则不会生成链接注释。宏 $\text{\texttt{\textit{XeTeXLinkBox}}}$ 将其参数放在一个盒子中，并在左下角和右上角添加空格 (spaces)。可以通过将其设置为尺寸寄存器 (dimen register) $\text{\texttt{\textit{XeTeXLinkMargin}}}$ 来指定额外的边距 (additional margin)。默认值为 2pt。

例如：

```
% xelatex
\documentclass{article}
\usepackage{hyperref}
\setlength{\XeTeXLinkMargin}{1pt}
\begin{document}
\section{Hello World}
\newpage
\label{sec:hello}
\hyperref[sec:hello]{%
  \XeTeXLinkBox{\rule{10mm}{10mm}}}%
}
\end{document}
```

7.8 $\text{\texttt{\textit{IfHyperBooleanExists}}}$ 和 $\text{\texttt{\textit{IfHyperBoolean}}}$

```
\IfHyperBooleanExists{OPTION}{YES}{NO}
```

如果一个 hyperref OPTION 是一个布尔值 (boolean), 这意味着它取值为 “true” 或 “false”, 那么 \IfHyperBooleanExists 调用 YES, 否则调用 NO。

```
\IfHyperBoolean{OPTION}{YES}{NO}
```

宏 \IfHyperBoolean 调用 YES, 如果 OPTION 作为布尔值存在并且已启用。否则执行 NO。

这两个宏都是可展开的 (expandable)。此外, 还提供了 “stoppedearly(提前停止)” 选项。如果 \MaybeStopEarly 或 \MaybeStopNow 过早结束 hyperref, 则启用该选项。

7.9 \unichar

如果 puenc.def 不支持 Unicode 字符, 则可以使用 \unichar 来给定。它的名称和语法继承自 “ucs” 宏包。然而, 它是独立定义的, 用于 hyperref 的 \pdfstringdef (将任意 TeX 代码转换为 PDF 字符串或尝试这样做)。

\unichar 宏以 TeX 数字为参数, 例如 U+263A(白色笑脸):

```
\unichar{"263A}% hexadecimal notation
```

```
\unichar{9786}% decimal notation
```

“” 不能是 babel 简写字符 (shorthand character) 或以其他方式处于活动状态 (active)。否则, 请在其前面加上 \string:

```
\unichar{\string"263A}% converts `"' to `"' with catcode 12 (other)
```

(n)german 宏包或 babel 选项的用户可以使用 \dq 代替:

```
\unichar{\dq 263A}% \dq is double quote with catcode 12 (other)
```

7.10 \ifpdfstringunicode

PDF 规范 (specification) 的一些功能需要 PDF 字符串。例如书签 (bookmarks) 或信息字典 (information dictionary) 中的条目。PDF 规范允许两种编码即 “PDFDocEncoding” (8 位编码) 和 “Unicode” (UTF-16)。用户可以使用 \texorpdfstring 将复杂的 TeX 构造 (constructs) 替换为 PDF 字符串的表示 (representation)。但是, \texorpdfstring 并不能区分这两种编码。此程序 (gap) 关闭 \ifpdfstringunicode。它只允许在 \texorpdfstring 的第二个参数

中使用，并接受两个参数，第一个参数允许 Unicode 的全部范围 (full range)。第二个参数限制为 PDFDocEncoding 中可用的字符。

例如，我们对越南语中的 Hàn Thê Thành 的名字进行了宏定义 (macro definition)。正确书写它需要一些重音字符 (accented characters)，一个字符甚至带有双重重音 (double accent)。“tugboat.cls” 类为排版名称定义了一个宏：

```
\def\Thanh{%
  H\`an~%
  Th\^e\llap{\raise 0.5ex\hbox{'{}}}%
  ~Th\`anh%
}
```

这并不完全正确，“e” 上面的第二个重音 (accent) 不是锐音 (acute)，而是钩音 (hook)。然而，标准的 LaTeX 并没有提供这样的重音。

现在我们可以扩展定义 (extend the definition) 以支持 hyperref。已经自动支持第一个单词和最后一个单词。具有两个或两个以上重音的字符在 LaTeX 中是一项困难的业务 (difficult business)，因为 LaTeX 内核的 NFSS2 宏不支持多个重音。因此，puenc.def 也错过了对它们的支持。但我们可以使用 \unichar 来表示它。这个符号 (character) 是：

```
% U+1EC3 LATIN SMALL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
```

因此，我们可以将这些放在一起：

```
\def\Thanh{%
  H\`an~%
  \texorpdfstring{Th\^e\llap{\raise 0.5ex\hbox{'{}}}}%
  {\ifpdfstringunicode{Th\unichar{"1EC3}}{Th\^e}}%
  ~Th\`anh%
}
```

对于 PDFDocEncoding (PD1)，上面的变体 (variant) 去掉了第二个重音。或者，我们可以提供一个没有重音的表示 (representation)，而不是错误的重音 (wrong accents)：

```
\def\Thanh{%
  \texorpdfstring{%
    H\`an~%
```

```

Th\~e\lap{\raise 0.5ex\hbox{\'\{}}}\}%
~Th\`anh%
}{%
\ifpdfstringunicode{%
H\`an Th\unichar{"1EC3} Th\`anh%
}{%
Han The Thanh%
}%
}%
}

```

7.11 使用 \nohyperpage 自定义索引样式文件

自 2008/08/14 v6.78f 版本起。

为了在索引 (index) 中支持超链接 (hyperlink), `hyperref` 在索引宏 (index macros) 中插入 `\hyperpage`。使用 `Makeindex` 处理后, `\hyperpage` 分析其参数以检测页面范围 (page ranges) 和页面逗号列表 (page comma lists)。但是, 仅直接支持标准设置 (standard settings):

```

delim_r "--"
delim_n ", "

```

请参阅 `Makeindex` 的手册页/文档, 其中解释了可以在 `Makeindex` 的样式文件 (style files) 中使用的键 (keys)。`delim_r`、`delim_n`、`suffix_2p`、`suffix_3p`、`suffix_mp` 的自定义版本 (customized versions) 需要 `\hyperpage` 可以检测到的标记 (markup), 并且知道这些东西不属于页码 (page number)。`Makro \nohyperpage` 用作此标记。将这些键的自定义代码放在 `\nohyperpage` 中, 例如:

```

suffix_2p "\nohyperpage{f.}"
suffix_3p "\nohyperpage{ff.}"

```

根据排版传统, 在 `\nohyperpage` 内的第一个 `f` 之前应该放一些空格 (space) “\,” 或 “~”。

7.12 “ocgcolorlinks” 实验性选项

这个想法是彩色链接 (colored links), 当查看时, 但打印时没有颜色。这个新的实验性选项 “ocgcolorlinks” 使用了 PDF 1.5 中引入的可选内容组 (Optional Content Groups) 功能。

在 ocgx2 宏包中有一个更好的实现 (implementation), 它没有防止换行 (line breaks) 的缺点。有关如何使用它的详细信息, 请查看其文档。

- 必须为宏包加载提供选项: `\usepackage[ocgcolorlinks]{hyperref}`
- 主要缺点: 链接不能跨行断开 (broken across lines)。PDF 参考文献 1.7:4.10.2 “Making Graphical Content Optional (使图形内容可选)” : Graphics state operations (图形状态操作), 如设置颜色, ..., 仍然适用。因此, 将链接文本 (link text) 放在一个盒子 (box) 中并设置两次, 有颜色 and 没有颜色。
- 该特性 (feature) 可以通过文档内的 `\hypersetup{ocgcolorlinks=false}` 来切换。
- 支持的驱动程序: pdf_{tex}、dvipdfm
- PDF 版本最低为 1.5。它是为 pdf_{TeX}、Lua_{TeX} 和 dvipdfmx 自动设置的。

7.13 “pdfa” 选项

新选项 “pdfa” 试图避免在 hyperref 生成的代码中违反 PDF/A。然而, 结果通常不在 PDF/A 中, 因为许多特性 (feature) 不受 hyperref 控制 (XMP 元数据、字体、颜色、依赖于驱动程序的低级别内容、... 等)。

目前, 选项 “pdfa” 设置并禁用以下项目:

- 启用的注释标志 (annotation flags): Print、NoZoom、NoRotate [PDF/A 6.5.3]。
- 禁用的注释标志 (annotation flags): Hidden、Invisible、NoView [PDF/A 6.5.3]。
- 禁用: 启动操作 (Launch action) [PDF/A 6.6.1]。
- 受限: 命名操作 (Named actions) (NextPage、PrevPage、FirstPage、LastPage) [PDF/A 6.6.1]。
- PDF 公式 (formulas) 中禁用了很多东西:
 - JavaScript 动作 (actions) [PDF/A 6.6.1]
 - 触发事件 (Trigger events) (附加操作 [additional actions]) [PDF/A 6.6.2]
 - 按钮 (因为 JavaScript)
 - 交互式表单 (interactive Forms): NeedAppearances 标志 (flag) 是默认的 “false” (因此, hyperref 的表单实现 [implementation] 看起来很难看)。[PDF/A 6.9]

新选项“pdfa”的默认值为“false”。“pdfa”选项会影响宏包的加载,并且在加载 hyperref (\usepackage{hyperref}) 后无法更改。

7.14 已添加的“linktoc”选项

新的选项“linktoc”允许更多地控制目录中条目 (entry in the table of contents) 的哪一部分变成了链接:

- “linktoc=none” (没有链接)
- “linktoc=section” (默认行为, 与“linktocpage=false”相同)
- “linktoc=page” (与“linktocpage=true”相同)
- “linktoc=all” (节 [section] 和页面部分 [page part] 都是链接)

7.15 已更改的“pdfnewwindow”选项

6.77b 以前的版本:

- pdfnewwindow=true → /NewWindow true
- pdfnewwindow=false → (absent)
- unused pdfnewwindow → (absent)

自 6.77b 版本开始:

- pdfnewwindow=true → /NewWindow true
- pdfnewwindow=false → /NewWindow false
- pdfnewwindow= → (absent)
- unused pdfnewwindow → (absent)

理由: 设置为“false”和没有条目 (absent entry) 是有区别的。在前一种情况中, 新文档将取代旧文档, 在后一种情况中, PDF 查看器应用程序会考虑用户偏好 (user preference)。

7.16 PDF 表单标志选项

PDF 表单字段宏 (form field macros)(\TextField、\CheckBox、...) 支持布尔标志选项 (boolean flag options)。选项名称是 PDF 规范 (1.7) 中名称的小写版本 (lowercase version):

http://www.adobe.com/devnet/pdf/pdf_reference.html

http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference.pdf

选项 (转换为小写), 方括号中的标志 (flags) 除外:

- 表 8.16 注释标志 (Annotation flags)(第 608 页):

1 Invisible

2 Hidden (PDF 1.2)

3 Print (PDF 1.2)

4 NoZoom (PDF 1.3)

5 NoRotate (PDF 1.3)

6 NoView (PDF 1.3)

[7 ReadOnly (PDF 1.3)] 忽略小部件注释 (widget annotations), 请参阅表 8.70

8 Locked (PDF 1.4)

9 ToggleNoView (PDF 1.5)

10 LockedContents (PDF 1.7)

- 表 8.70 所有字段类型 (field types) 通用的字段标志 (field flags)(第 676 页):

1 ReadOnly

2 Required

3 NoExport

- 表 8.75 按钮字段 (button fields) 特有的字段标志 (field flags)(第 686 页):

15 NoToggleToOff (仅限单选按钮)

16 Radio (设置: 单选按钮, 清除: 复选框, 按钮: 清除)

17 Pushbutton

26 RadiosInUniso (PDF 1.5)

- 表 8.77 文本字段 (text fields) 特有的字段标志 (field flags)(第 691 页):

13 Multiline

14 Password

21 FileSelect (PDF 1.4)

23 DoNotSpellCheck (PDF 1.4)

24 DoNotScroll (PDF 1.4)

25 Comb (PDF 1.5)

26 RichText (PDF 1.5)

- 表 8.79 选择字段 (choice fields) 特有的字段标志 (field flags)(第 693 页):

18 Combo (设置: 组合框, 清除: 列表框)

19 Edit (仅在设置了 Combo [组合] 时有用)

20 Sort (排序) 用于创作工具 (authoring tools), 而不是 PDF 查看器

22 MultiSelect (PDF 1.4)

23 DoNotSpellCheck (PDF 1.4) (仅在设置了 Combo [组合] 和 Edit [编辑] 时有用)

27 CommitOnSelChange (PDF 1.5)

- 表 8.86 提交表单操作 (submit-form actions) 的标志 (第 704 页):

[1 Include/Exclude] 不支持, 请改用 “noexport” (表 8.70)

2 IncludeNoValueFields

[3 ExportFormat] 由选项 “export” 处理

4 GetMethod

5 SubmitCoordinates

[6 XFDF (PDF 1.4)] 由选项 “export” 处理

7 IncludeAppendSaves (PDF 1.4)

8 IncludeAnnotations (PDF 1.4)

[9 SubmitPDF (PDF 1.4)] 由选项 “export” 处理

10 CanonicalFormat (PDF 1.4)

11 ExclNonUserAnnots (PDF 1.4)

12 ExclFKey (PDF 1.4)

14 EmbedForm (PDF 1.5)

新选项 “export” 设置提交操作 (submit action) 的导出格式 (export format)。有效值如下 (大写或小写):

- FDF
- HTML
- XFDF
- PDF (Acrobat Reader 不支持)

7.17 “pdfversion” 选项

这是一个实验性的选项 (experimental option)。它会通知 “hyperref” 有关预期的 (intended)PDF 版本。目前, 这被用于 PDF 表单 (forms) 的代码中 (PDF 规范 1.7 的实现注释 [implementation notes]116 和 122)。

值: 1.2、1.3、1.4、1.5、1.6、1.7。不支持 1.2 以下的值, 因为大多数驱动程序需要更高的 PDF 版本。

该选项必须提前使用, 而不是在 `\usepackage{hyperref}` 之后。

理论上, 此选项还应设置 PDF 版本, 但通常不支持此选项。

- 1.10a 以下的 pdfTeX: 不支持。 $\geq 1.10a$ 且 < 1.30 的 pdfTeX: `\pdfoptionpdfminorversion`。
pdfTeX ≥ 1.30 : `\pdfminorversion`
- dvipdfm: 配置文件 (configuration file), 例如: TeX Live 2007, `texmf/dvipdfm/config/config`, entry “V 2”。
- dvipdfmx: 配置文件 (configuration file), 例如: TeX Live 2007, `texmf/dvipdfm/dvipdfmx.cfg`, entry “V 4”。
- Ghostscript: 选项 (option) `-dCompatibilityLevel` (这是在 “ps2pdf12”、“ps2pdf13”、“ps2pdf14” 中设置)。

如果可以检测到当前 PDF 版本, 则使用当前 PDF 版本作为默认版本 (仅 pdfTeX $\geq 1.10a$)。否则, 假设最低版本为 1.2。因此, “hyperref” 试图避免破坏这个版本的 PDF 代码, 但是可以自由地使用可忽略的 (ignorable) 更高的 PDF 特性 (features)。

7.18 “name” 字段选项

许多表单对象 (form objects) 使用 label 参数的目的有以下几个:

- 设计好的标签 (Layouted label)。
- 用作 HTML 结构 (structures) 中的名称。

适合使用 TeX 进行布局 (layouting) 的代码可能会破坏输出格式 (output format) 的结构。如果给定了选项 “name”, 则其值将用作不同输出结构 (output structures) 中的名称。因此, 该值应该仅由字母 (letters) 组成。

7.19 “pdfencoding” 选项

PDF 格式允许对书签 (bookmarks) 和信息字典 (information dictionary) 中的条目进行两种编码: PDFDocEncoding 和 Unicode(UTF-16BE)。pdfencoding 选项在这些编码之间进行选择:

- pdfdoc 使用 PDFDocEncoding, 每个字符 (character) 只使用一个字节 (byte), 但受支持的字符数有限 (在 PDF-1.7 中为 244 个)。
- unicode 设置 unicode。它被编码为 UTF-16BE。大多数字符使用两个字节, 代理 (surrogates) 需要四个字节。
- 如果字符串不包含编码以外的字符 (如果使用 Unicode 引擎, 则为 ascii 以外的字符), 则为 auto PDFDocEncoding, 否则为 Unicode。此选项不适用于 unicode 引擎。

所有驱动程序现在默认使用 unicode。如果应该强制执行另一个编码, 则应该在 hypersetup 中完成。

7.20 颜色选项/hycolor 宏包

请参阅 “hycolor” 宏包的文档。

7.21 pdfusetitle 选项

如果设置了 pdfusetitle 选项, 那么 hyperref 将尝试从 \title 和 \author 获得 (derive) pdftitle 和 pdfauthor 的值。支持 \title 和 \author 的可选参数 (amsart 类)。

7.22 \autoref 的星号形式

\autoref* 生成一个不带链接的引用, 如 \ref* 或 \pageref*。

7.23 链接边框样式

链接可以加下划线 (underlined), 而不是默认的矩形 (rectangle) 或选项 colorlinks、frenchlinks。这是通过选项 pdfborderstyle={/S/U/W 1} 完成的。

一些备注 (remarks):

- AR7/Linux 似乎有一个错误, 它不使用默认值 1 作为宽度, 而是使用零, 因此如果没有 /W 1, 下划线 (underline) 就不可见。这同样适用于虚线框 (dashed boxes), 例如:
pdfborderstyle=/S/D/D[3 2]/W 1
- 该语法在 PDF 规范中有描述, 请查找 “border style” (边框样式), 例如。表 8.13 “边框样式字典中的条目” (版本 1.6 的规范)
- 边框样式 (border style) 由 pdfborderstyle= 删除。如果启用了选项 colorlinks, 则会自动删除。
- 请注意, 并非所有 PDF 查看器都支持此功能, 甚至 Acrobat Reader 本身也不支持:

支持的有:

- AR7/Linux: underline 和 dashed, 但是必须给出边界宽度 (border width)。
- xpdf 3.00: underline 和 dashed。

不支持:

- AR5/Linux
- ghostscript 8.50

7.24 bookmarksdepth 选项

书签的深度 (depth) 可以由新选项 `bookmarksdepth` 控制。该选项在全局范围内发挥作用, 区分了三种情况:

- `bookmarksdepth` 没有值: 然后 `hyperref` 使用计数器的当前值 `tocdepth`。这是兼容的行为 (compatible behaviour), 也是默认的行为。
- `bookmarksdepth=<number>`: 值为数字 (也是负数), 书签的深度设置为这个数字。
- `bookmarksdepth=<name>`: `<name>` 是一个文档分割名称 (division name) (`part` [部分], `chapter` [章], ...)。它不能以数字 (digit) 或减号 (minus) 开头, 以免与数字大小写 (number case) 混淆。内部 `hyperref` 使用宏 `\toclevel@<name>` 的值。例如:

```
\hypersetup{bookmarksdepth=paragraph}
\hypersetup{bookmarksdepth=4} % 和以前一样
\hypersetup{bookmarksdepth}   % 使用计数器 “tocdepth”
```

7.25 pdfescapeform 选项

在许多地方, 任意字符串最终会变成 PS 或 PDF 字符串。括号形式 (parentheses form) 的 PS/PDF 字符串需要保护 (protection) 某些字符, 例如, 不匹配的左括号或右括号需要转义或转义字符本身 (反斜杠)。自 2006/02/12 v6.75a 开始, PS/PDF 驱动程序应该自动执行此操作。然而, 我认为兼容性存在问题, 尤其是在表单部分 (form part), 其中可能存在大量 JavaScript 代码。删除所有转义将是一件痛苦的事情, 因为额外的转义层 (additional escaping layer) 可能会伪造代码。

因此, 引入了一个新的选项 `pdfescapeform`:

- `pdfescapeform=false`: 公式 (formulas) 的转义被禁用, 这是兼容性行为, 因此这是默认行为。
- `pdfescapeform=true`: 然后 PS/PDF 驱动程序执行所有必要的转义。这是合乎逻辑的选择, 也是推荐的设置。例如, 用户以 JavaScript 的形式编写 JavaScript, 而不关心 PS/PDF 输出的转义字符。

7.26 默认驱动程序设置

(hyperref ≥ 6.72s) 如果没有给出驱动程序, hyperref 会尽力猜测最合适的驱动程序。因此, 如果检测到 pdfTeX 以 PDF 模式运行, 它将加载 hpdfTeX。或者它为 VTeX 的工作模式 (working modes) 加载相应的 VTeX 驱动程序。不幸的是, 许多驱动程序在 TeX 编译器之后运行, 因此 hyperref 没有机会 (dvips、dvipdfm、...)。在这种情况下, 加载了支持 HyperTeX 功能 (features) 的驱动程序 hypertex, 例如 xdv 可以识别这些功能。然而, 在配置文件 hyperref.cfg 中可以很容易地更改这种行为:

```
\providecommand*\Hy@defaultdriver{\hdvips}
```

对于 dvips, 或者

```
\providecommand*\Hy@defaultdriver{\hypertex}
```

对于 hyperref 的默认行为。

另请参阅新选项 “driverfallback”。

7.27 Backref 条目

用于格式化 backref 条目 (entries) 的替代接口 (alternative interface), 例如:

```
\documentclass[12pt,UKenglish]{article}
```

```
\usepackage{babel}
```

```
\usepackage[pagebackref]{hyperref}
```

% backref 宏包检测到一些语言选项 (language options), 这会影响以下宏:

```
% \backrefpagesname
```

```
% \backrefsectionsname
```

```
% \backrefsep
```

```
% \backreftwosep
```

```
% \backreflastsep
```

```
\renewcommand*\backref[1]{
```

```
% 默认接口
```

```

% #1: backref 列表
%
% 我们希望使用替代接口，因此此处的定义为空。
}

\renewcommand*{\backrefalt}[4]{%
% 替代接口
% #1: 不同的反向引用(back references)的数目
% #2: 具有不同条目的 backref 列表
% #3: 包括重复项(duplicates)在内的反向引用的数目
% #4: 包含重复项(duplicates)在内的 backref 列表
\par
#3 citation(s) on #1 page(s): #2,\par
\ifnum#1=1 %
  \ifnum#3=1 %
    1 citation on page %
  \else
    #3 citations on page %
  \fi
\else
  #3 citations on #1 pages %
\fi
#2,\par
\ifnum#3=1 %
  1 citation located at page %
\else
  #3 citations located at pages %
\fi
#4.\par
}

% 不同条目的列表可以进一步完善：
\renewcommand*{\backrefentrycount}[2]{%

```

```

% #1: 原始 backref 条目
% #2: 本条目的引用次数，如重复多于一次
#1%
\ifnum#2>1 %
  ~(#2)%
\fi
}

```

```

\begin{document}

```

```

\section{Hello}
\cite{ref1, ref2, ref3, ref4}
\section{World}
\cite{ref1, ref3}
\newpage

```

```

\section{Next section}
\cite{ref1}
\newpage

```

```

\section{Last section}
\cite{ref1, ref2}
\newpage

```

```

\pdfbookmark[1]{Bibliography}{bib}
\begin{thebibliography}{99}
\bibitem{ref1} Dummy entry one.

\bibitem{ref2} Dummy entry two.

\bibitem{ref3} Dummy entry three.

```



```

\bibitem{ref4} Dummy entry four.

\end{thebibliography}

\end{document}

```

7.28 \phantomsection

在此位置设置锚点 (anchor)。它通常与 `\addcontentsline` 结合使用，用于分节式的东西 (sectionlike things) 如索引 (index)、参考文献 (bibliography)、序言 (preface)。`\addcontentsline` 指的是设置锚点的上一个最新位置 (latest previous location)。

```

\cleardoublepage
\phantomsection
\addcontentsline{toc}{chapter}{\indexname}
\printindex

```

现在，用于索引的目录 (和书签) 中的条目指向索引页 (index page) 的开始，而不是指向该页之前的位置。

7.29 puenc 编码、puenc-greekbasic.def 和 puenc-extra.def

unicode 选项加载书签 `puenc.def`，`puenc.def` 包含相当多的书签命令定义。由于 unicode 现在适用于所有引擎，因此该文件现在也使用 `pdflatex` 加载。`puenc.def` 中的一些定义与其他用法冲突。为了减少影响，`hyperref` 使用了两种策略：

- 许多命令只能有条件地定义：如果 `\CYRDZE` 已定义，则为西里尔语块 (cyrillic block) 定义命令；如果 `\textBeta` 已定义，则为希腊语块 (greek block) 定义命令；如果 `\hebdalet` 已定义，则为希伯来语块 (hebrew block) 定义命令。

希腊语块 (greek block) 位于一个额外的文件 `puenc-greekbasic.def` 中，如果需要，可以手动加载该文件。

- 其他命令被移到一个额外的文件 `puenc-extra.def` 中，该文件不会自动加载，但如果需要，可以在前言中加载。目前，该文件包含重音符号 (accent) `\G` 的所有定义。

8 特定于 Acrobat 的行为

如果要访问 Acrobat Reader 或 Exchange 的菜单选项 (menu options), 相应的驱动程序 (appropriate drivers) 会提供以下宏 (macro):

`\Acrobatmenu{menuoption}{text}`

该文本 (*text*) 用于创建一个按钮 (button), 用于激活相应的菜单选项 (*menuoption*)。下表列出了您可以使用的选项名称 — 将其与 Acrobat Reader 或 Exchange 中的菜单进行比较将显示它们的功能。显然, 有些仅适用于 Exchange。

菜单	菜单选项 (<i>menuoption</i>)
File	Open, Close, Scan, Save, SaveAs, Optimizer:SaveAsOpt, Print, PageSetup, Quit
File→Import	ImportImage, ImportNotes, AcroForm:ImportFDF
File→Export	ExportNotes, AcroForm:ExportFDF
File→DocumentInfo	GeneralInfo, OpenInfo, FontsInfo, SecurityInfo, Weblink:Base, AutoIndex:DocInfo
File→Preferences	GeneralPrefs, NotePrefs, FullScreenPrefs, Weblink:Prefs, AcroSearch:Preferences(Windows) or, AcroSearch:Prefs(Mac), Cpt:Capture
Edit	Undo, Cut, Copy, Paste, Clear, SelectAll, Ole:CopyFile, TouchUp:TextAttributes, TouchUp:FitTextToSelection, TouchUp:ShowLineMarkers, TouchUp:ShowCaptureSuspects, TouchUp:FindSuspect, Properties
Edit→Fields	AcroForm:Duplicate, AcroForm:TabOrder
Document	Cpt:CapturePages, AcroForm:Actions, CropPages, RotatePages, InsertPages, ExtractPages, ReplacePages, DeletePages, NewBookmark, SetBookmarkDest, CreateAllThumbs, DeleteAllThumbs

(后续下表)

(前接上表)

菜单	菜单选项 (<i>menuoption</i>)
View	ActualSize, FitVisible, FitWidth, FitPage, ZoomTo, FullScreen, FirstPage, PrevPage, NextPage, LastPage, GoToPage, GoBack, GoForward, SinglePage, OneColumn, TwoColumns, ArticleThreads, PageOnly, ShowBookmarks, ShowThumbs
Tools	Hand, ZoomIn, ZoomOut, SelectText, SelectGraphics, Note, Link, Thread, AcroForm:Tool, Acro_Movie:MoviePlayer, TouchUp:TextTool, Find, FindAgain, FindNextNote, CreateNotesFile
Tools→Search	AcroSrch:Query, AcroSrch:Indexes, AcroSrch:Results, AcroSrch:Assist, AcroSrch:PrevDoc, AcroSrch:PrevHit, AcroSrch:NextHit, AcroSrch:NextDoc
Window	ShowHideToolBar, ShowHideMenuBar, ShowHideClipboard, Cascade, TileHorizontal, TileVertical, CloseAll
Help	HelpUserGuide, HelpTutorial, HelpExchange, HelpScan, HelpCapture, HelpPDFWriter, HelpDistiller, HelpSearch, HelpCatalog, HelpReader, Weblink:Home
Help(Windows)	About

9 PDF 和 HTML 格式

必须将字段 (fields) 放在 Form 环境中。该环境进行一些常规设置 (general setups), 因此在文档中只能使用一次。也可以在文档开头简单地使用 \Form。

有六个宏用于准备字段 (prepare fields):

```
\TextField[parameters]{label}
```

```
\CheckBox[parameters]{label}
```

```
\ChoiceMenu[parameters]{label}{choices}
```

```
\PushButton[parameters]{label}
```

```
\Submit[parameters]{label}
```

```
\Reset[parameters]{label}
```

表单 (forms) 及其标签 (labels) 的布局方式由以下因素决定:

```
\LayoutTextField{label}{field}
```

```
\LayoutChoiceField{label}{field}
```

```
\LayoutCheckField{label}{field}
```

这些宏默认为 #1 #2

字段 (field) 中实际显示的内容由以下因素决定:

```
\MakeRadioField{width}{height}
```

`\MakeCheckField{width}{height}`

`\MakeTextField{width}{height}`

`\MakeChoiceField{width}{height}`

`\MakeButtonField{text}`

这些宏默认为 `\vbox to #2{\hbox to #1{\hfill}\vfill}`，除了最后一个，它默认为 #1；它用于按钮 (buttons)，以及特殊的 `\Submit` 和 `\Reset` 宏。

您可能还需要重新定义以下宏：

```
\def\DefaultHeightofSubmit{12pt}
\def\DefaultWidthofSubmit{2cm}
\def\DefaultHeightofReset{12pt}
\def\DefaultWidthofReset{2cm}
\def\DefaultHeightofCheckBox{0.8\baselineskip}
\def\DefaultWidthofCheckBox{0.8\baselineskip}
\def\DefaultHeightofChoiceMenu{0.8\baselineskip}
\def\DefaultWidthofChoiceMenu{0.8\baselineskip}
\def\DefaultHeightofText{\baselineskip}
\def\DefaultHeightofTextMultiline{4\baselineskip}
\def\DefaultWidthofText{3cm}
```

9.1 表单环境参数

表单环境参数	值	说明
action	URL	如果表单 (form) 中包含 Submit 按钮，将接收表单数据的 URL
encoding	name	设置为 URL 的字符串的编码；FDF 编码是常见的，html 是唯一有效的值
method	name	仅在生成 HTML 时使用；值可以是 post 或 get

9.2 表单选项参数

请注意，所有颜色都必须表示为 RGB 三元组 (triples)，范围为 0..1(即 color=0 0 0.5)

表单选项参数	类型	值	说明
accesskey	键 (key)		(根据 HTML)
align	数字 (number)	0	文本字段 (text field) 内的对齐方式： 0 是左对齐，1 是居中对齐，2 是右对齐
altname	名称 (name)		备用 (alternative) 名称，显示在用户接口中的名称
backgroundcolor			方框的背景色 (color of box)
bordercolor			边框的颜色 (color of box)
bordersep			边框边界间距 (box border gap)
borderstyle	字符 (char)	S	方框边框的样式：S (实线) 为默认，B 是斜线， D 是虚线，I 是嵌图 (inset)，U 是下划线
borderwidth		1	边框的宽度，该值是一个尺寸 (dimension) 或一个默认单位为 bp 的数字
calculate			用于计算字段 (field) 值的 JavaScript 代码
charsize	尺寸 (dimen)		字段文本 (field text) 的字体尺寸
checkboxsymbol	字符 (char)	4 (✓)	用于复选框 (ZapfDingbats) 的符号， 该值是一个字符或 \ding{number}， 请参阅捆绑包 (bundle) psnfss 中的 pifont 宏包
checked	布尔值 (boolean)	false	是否选择默认选中
color			方框中文本的颜色
combo	布尔值 (boolean)	false	选项列表 (choice list) 为 “combo” 样式
default			默认值
disabled	布尔值 (boolean)	false	字段已禁用
format			用于格式化字段的 JavaScript 代码
height	尺寸 (dimen)		字段方框高度
hidden	布尔值 (boolean)	false	隐藏字段
keystroke			用于控制输入时按键的 JavaScript 代码
mappingname	名称 (name)		导出字段数据时使用的映射 (mapping) 名称
maxlen	数字 (number)	0	文本字段中允许的字符数

(后续下表)

(前接上表)

表单选项参数	类型	值	说明
menulength	数字 (number)	4	列表中显示的元素数 (number of elements)
multiline	布尔值 (boolean)	false	文本框是否为多行 (multiline)
name	名称 (name)		字段名称 (默认为 label[标签])
onblur			JavaScript 代码
onchange			JavaScript 代码
onclick			JavaScript 代码
ondblclick			JavaScript 代码
onfocus			JavaScript 代码
onkeydown			JavaScript 代码
onkeypress			JavaScript 代码
onkeyup			JavaScript 代码
onmousedown			JavaScript 代码
onmousemove			JavaScript 代码
onmouseout			JavaScript 代码
onmouseover			JavaScript 代码
onmouseup			JavaScript 代码
onselect			JavaScript 代码
password	布尔值 (boolean)	false	文本字段 (text field) 为 “password” 样式
popdown	布尔值 (boolean)	false	选项列表 (choice list) 为 “popdown” 样式
radio	布尔值 (boolean)	false	选项列表 (choice list) 为 “radio” 样式
radiosymbol	字符 (char)	H (★)	用于 radio 字段 (ZapfDingbats) 的符号, 该值是一个字符或 \ding{number}, 请参阅捆绑包 (bundle) psnfss 中的 pifont 宏包
readonly	布尔值 (boolean)	false	字段是只读的
rotation	数字 (number)	0	小部件注释 (widget annotation) 的 旋转 (rotation)(度, 逆时针, 90 的倍数)
tabkey			(根据 HTML)
validate			用于验证 (validate) 条目的 JavaScript 代码

(后续下表)

(前接上表)

表单选项参数	类型	值	说明
value			初始值 (initial value)
width	尺寸 (dimen)		字段方框 (field box) 的宽度

10 定义新的驱动程序

hyperref 驱动程序 (driver) 必须为以下八个宏提供定义:

1. `\hyper@anchor`
2. `\hyper@link`
3. `\hyper@linkfile`
4. `\hyper@linkurl`
5. `\hyper@anchorstart`
6. `\hyper@anchorend`
7. `\hyper@linkstart`
8. `\hyper@linkend`

draft 选项将宏定义如下:

```
\let\hyper@@anchor\@gobble
\gdef\hyper@link##1##2##3{##3}%
\def\hyper@linkurl##1##2{##1}%
\def\hyper@linkfile##1##2##3{##1}%
\let\hyper@anchorstart\@gobble
\let\hyper@anchorend\@empty
\let\hyper@linkstart\@gobbletwo
\let\hyper@linkend\@empty
```

11 对其他宏包的特殊支持

hyperref 包旨在与其他包协同工作，但长期目标是移除补丁并将链接支持移入核心、类和包中。然而，仍然存在多个可能引发冲突的来源，例如：

- 操纵文献机制的包。Peter William 的 harvard 包得到支持。但推荐使用 Patrick Daly 的 natbib 包，它具有特定的 hyperref 钩子以实现可靠的交互，或者使用 biblatex 包。这两个包涵盖了多种布局和引用样式，都与 hyperref 兼容。
- 改变 \label 和 \ref 宏的包。自 2023-06-01 的 L^AT_EX 核心和 hyperref/nameref 使用了相同的 \label 定义。 \label 具有外部包的钩子。因此，外部包和类不应该重新定义它们。
- 与索引有关的功能严肃的包。
- 与节标题命令和目录有关的功能严肃的包。

hyperref 包附带一个 xr 的变种 xr-hyper⁹，支持使用 L^AT_EX 的正常 \label/\ref 机制进行跨文档链接。

11.1 宏包的兼容性

目前只有按顺序加载的宏包 (package loading orders) 才可用：

11.1.1 algorithm

```
\usepackage{float}
\usepackage{hyperref}
\usepackage[chapter]{algorithm} % 举例
```

11.1.2 amsmath

对 equation 和 eqnarray 环境的支持不太好。例如，可能存在间距问题 (spacing problems)(无论如何都不建议使用 eqnarray，请参阅 CTAN:info/l2tabu/，equation 的情况尚不清楚，因为没有人对研究感兴趣)。考虑使用 amsmath 宏包提供的环境，例如，使用 equation gather (包围)。equation 环境甚至可以重新定义以使用 gather (包围)：

⁹它将很快与 xr 包合并

```
\usepackage{amsmath}
\let\equation\gather
\let\endequation\endgather
```

11.1.3 amsrefs

amsrefs 的文档声称该包必须在 hyperref 之后加载（目前还不确定是否真的需要这样），因此推荐的包加载顺序是：

```
\usepackage{hyperref}
\usepackage{amsrefs}
```

11.1.4 arydshln、longtable

longtable 宏包必须放在 hyperref 和 arydshln 之前，在 arydshln 之后的 hyperref 会产生错误，因此加载宏包顺序为：

```
\usepackage{longtable}
\usepackage{hyperref}
\usepackage{arydshln}
```

11.1.5 babel/magyar.ldf

旧版本 2005/03/30 v1.4j 将不起作用。您至少需要 1.5 版本，由 Péter Szabó (彼得·萨博) 维护，请参阅 CTAN:language/hungarian/babel/。

11.1.6 babel/spanish.ldf

Babel 的 spanish.ldf 重新定义了 “\.” 以支持 “\...”。在书签 (\pdfstringdef) 中，仅 “\.” 受到支持。如果需要 “\...”，则可以使用 \texorpdfstring{\...}{\dots} 来代替。

11.1.7 bibentry

解决办法：

```

\makeatletter
\let\saved@bibitem\@bibitem
\makeatother

\usepackage{bibentry}
\usepackage{hyperref}

\begin{document}

\begingroup
  \makeatletter
  \let\@bibitem\saved@bibitem
  \nobibliography{database}
\endgroup

```

11.1.8 bigfoot

hyperref 宏包不支持 “bigfoot” 宏包, 而 “bigfoot” 宏包不支持 hyperref 的脚注 (footnotes) 并禁用它们 (hyperfootnotes=false)。

11.1.9 chappg

“chappg” 宏包使用了由 “hyperref” 重新定义的 \@addtoreset。因此, 宏包的加载顺序是:

```

\usepackage{hyperref}
\usepackage{chappg}

```

11.1.10 count1to

“count1to” 宏包添加了几个 \@addtoreset 命令, 这些命令会使 “hyperref” 产生混淆。因此必须修正 \theH<...>:

```

\usepackage{count1to}

```

```

\AtBeginDocument{% *after* \usepackage{count1to}
  \renewcommand*{\theHsection}{\theHchapter.\arabic{section}}}%
  \renewcommand*{\theHsubsection}{\theHsection.\arabic{subsection}}}%
  \renewcommand*{\theHsubsubsection}{\theHsubsection.\arabic{subsubsection}}}%
  \renewcommand*{\theHparagraph}{\theHsubsubsection.\arabic{paragraph}}}%
  \renewcommand*{\theHsubparagraph}{\theHparagraph.\arabic{subparagraph}}}%
}

```

11.1.11 dblacnt

加载下列宏包之前必须先加载 `pd1enc.def` 或 `puenc.def`

```

\usepackage{hyperref}
\usepackage{dblacnt}

```

或参阅 `vietnam` 的条目 (entry)。

11.1.12 easyeqn

不兼容，会中断 (breaks)。

11.1.13 ellipsis

`ellipsis` 这个宏包在 `hyperref` 宏包 (应先加载 `pd1enc.def`/`puenc.def`) 之后重新定义了 `\textellipsis`:

```

\usepackage{hyperref}
\usepackage{ellipsis}

```

(这将导致书签中出现错误的省略号，因此需要 `\texorpdfstring`)。

11.1.14 float

```

\usepackage{float}
\usepackage{hyperref}

```

- 在一个浮动对象 (float object) 中不支持多个 `\caption` 命令。

- 如果浮动对象 (float object) 的样式由 float.sty 控制, 则锚点应设置在浮动对象的顶部 (top)。

11.1.15 endnotes

不受支持。

11.1.16 foiltex

更新至 2008/01/28 v2.1.4b 版本: 由于 6.77a 版本的 hyperref 没有挂接到 (hack into) \@begindvi, 因此它使用了 “atbegshi” 宏包, 该宏包挂接到 \shipout。因此, 关于 hyperref 的 “foils.cls” 补丁现在已经过时了, 该补丁会导致关于 \@hyperfixhead 的未定义错误消息。FoilTeX 2.1.4b 中对此进行了修复。

11.1.17 footnote

这个宏包不受支持, 您必须使用选项 hyperfootnotes=false 来禁用 hyperref 的脚注支持 (footnote support)。

11.1.18 linguex

```
\usepackage{hyperref}
\usepackage{linguex}
```

11.1.19 ltabptch

```
\usepackage{longtable}
\usepackage{ltabptch}
\usepackage{hyperref}
```

11.1.20 mathenv

不受支持。

“mathenv” 和 “hyperref” 都会扰乱环境 “eqnarray”。您可以在 “hyperref” 之后加载 “mathenv”, 以避免错误消息。但是 \label 在 “eqnarray” 环境中不能正常工作。

11.1.21 minitoc-hyper

minitoc-hyper 宏包已过时, 请使用最新的原始宏包 (up-to-date original package) minitocc。

11.1.22 multind

```
\usepackage{multind}  
\usepackage{hyperref}
```

11.1.23 natbib

```
\usepackage{natbib}  
\usepackage{hyperref}
```

11.1.24 nomenc1

引入页码链接 (links for the page numbers) 的示例:

```
\renewcommand*{\pagedeclaration}[1]{\unskip, \hyperpage{#1}}
```

11.1.25 ntheorem

这个包并未得到完全支持。不应该使用 `thref` 选项, 因为它会破坏 `\label` 命令。此外, 无法保证定理的链接始终正常工作, 因为该包重新定义了许多内部命令, 但并未提供适当的超链接接口。

11.1.26 ntheorem-hyper

ntheorem-hyper 宏包已过时, 请使用最新的原始宏包 (up-to-date original package) ntheorem。

对于等式 (equations), 以下方法可能有效:

```
\renewcommand*{\eqdeclaration}[1]{%  
  \hyperlink{equation.#1}{(Equation~#1)}%  
}
```

但是, 从等式编号 (equation number) 到锚点名称 (anchor name) 的映射 (mapping) 通常不可用。

11.1.27 prettyref

%%% 以下为 prettyref 的示例 %%%

```
\documentclass{article}
\usepackage{prettyref}
\usepackage{hyperref}

%\newreformat{FIG}{Figure~\ref{#1}} % 没有 hyperref
\newreformat{FIG}{\hyperref[#{1}]{Figure~\ref*{#1}}}

\begin{document}
  This is a reference to \prettyref{FIG:ONE}.
  \newpage
  \begin{figure}
    \caption{This is my figure}
    \label{FIG:ONE}
  \end{figure}
\end{document}
%%% 以上为 prettyref 的示例 %%%
```

11.1.28 setspace

```
\usepackage{setspace}
\usepackage{hyperref}
```

11.1.29 sidecap

再也不需要什么特别的东西了。

11.1.30 subfigure

这个宏包已经过时了。请使用 subfig 或 subcaption。

11.1.31 titleref

```
\usepackage{nameref}
\usepackage{titleref} % 没有 usetoc
\usepackage{hyperref}
```

11.1.32 tabularx

在“tabularx”环境中不支持链接脚注 (linked footnotes), 因为它们使用了 `\footnotetext` 的可选参数, 请参阅“限制 (Limitations)”一节。在 2011/09/28 6.82i 版本之前, `hyperref` 通过“`hyperfootnotes=false`”完全禁用了脚注 (footnotes)。

11.1.33 titlesec

`nameref` 支持 `titlesec`, 但 `hyperref` 不支持 (没有解决锚点设置 [anchor setting], 缺失未编号的节 [unnumbered section], 可能存在带编号的分页符问题)。

11.1.34 ucs/utf8x.def

注意: \LaTeX 现在默认使用 `utf8` 编码, 不再推荐使用 `ucs`。

第一次调用多字节 (multibyte) UTF8 序列 (sequence) 时, 它会进行一些计算, 并将结果存储在宏中, 以加快该 UTF8 序列的下次调用。然而, 这使得第一个调用不可展开, 并且如果在信息条目 (information entries) 或书签 (bookmarks) 中使用, 则会中断 (break)。ucs 宏包提供了 `\PrerenderUnicode` 或 `\PreloadUnicodePage` 来解决此问题:

```
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\usepackage{hyperref} % 或带有 unicode 选项
\PrerenderUnicode{^^c3^^b6} % 或 \PreloadUnicodePage{1}
\hypersetup{pdftitle={Umlaut example: ^^c3^^b6}}
```

带有两个插入符号 (carets) 的表示法 (notation) 避免了 README 文件中 8 位字节的麻烦, 您可以直接使用这些字符。

11.1.35 varioref

大部分与 varioref 相关的问题应该已解决。只有一个关于 `\vrefformat` 的问题仍然存在 (<https://github.com/latex3/hyperref/issues/225>)。

建议始终在加载 varioref 时使用 nospace 选项，请参阅文档。

```
\usepackage[nospace]{varioref}
```

11.1.36 verse

2005/08/22 v2.22 版本包含对 hyperref 的支持。

11.1.37 vietnam

% 应在 dblacnt 宏包之前加载 pdlenc.def:

```
\usepackage[PD1,OT1]{fontenc}
\usepackage{vietnam}
\usepackage{hyperref}
```

11.1.38 XeTeX

书签 (bookmarks) 的默认编码为 `pdfencoding=unicode`。这意味着字符串始终被视为 unicode 字符串。如果强制使用 `auto` 或 `pdfdoc`，则仅当字符串限制为可打印的 ASCII 集 (set) 时才适用。原因是 `\special` 不支持 `PDFDocEncoding`。

在旧版本中，hyperref 在 xetex 的多个位置包含从 UTF-16BE 返回 UTF-8 的特殊转换代码 (special conversion code)，以避免 `xdvipdfmx` 警告：

```
Failed to convert input string to UTF16... [未能将输入字符串转换为 UTF16...]
```

当前的 `xdvipdfmx` 不再需要这样做，因此此代码已被删除。

不应再使用 `\csname HyPsd@XeTeXBigCharstrue\endcsname`。

12 局限性¹⁰

12.1 换行/断开的链接支持

只有少数驱动程序支持自动换行/断开的链接 (automatically wrapped/broken links), 例如 pdftex、dvipdfmx、hypertex。其他驱动程序缺乏此功能, 例如 dvips、dvipsone。

变通方法:

- 对于目录或图形/表格列表中的长章节 (long section) 或标题 (caption titles), 可以使用 linktocpage 选项。然后页码 (page number) 将是一个链接, 过长的章节标题不会被强制转换为一个带有过满 (overflow) \hbox 警告的单行链接 (line link)。
- “\url” s 被 breakurl 宏包捕获。
- breaklinks 选项在内部使用。但是它可以用来强制链接换行 (link wrapping), 例如在打印文档时。然而, 当这样一个文档被转换成 PDF 并用 PDF 查看器查看时, 活动链接区域 (active link area) 将被放错位置 (misplaced)。

另一个局限性 (limitation) 是: 有些惩罚 (penalties) 是由 TeX “优化 (optimized)” 的, 因此缺少断点 (break points), 尤其是在 \url 中。(请参阅 comp.text.tex 2005-09 中的线索 “hyperref.sty, breaklinks and url.sty 3.2”)。

12.2 跨页面的链接

总的来说, 他们有问题:

- 有些驱动程序根本不支持它们 (见上文)。
- 驱动程序允许, 但链接结果可能包括页脚和/或页眉。目前 (2023 年) 可以通过使用 PDF 管理并至少加载 latex-lab 的 new-or-1 模块来避免这种情况:

```
\DocumentMetadata{testphase=new-or-1}
```

¹⁰ 本节从自述文件 (README file) 中删除, 需要更多地集成到手册中。

12.3 脚注

LaTeX 允许脚注标记 (footnote mark)(\footnotemark) 和脚注文本 (\footnotetext) 的分隔 (separation)。这个接口 (interface) 对于视觉排版 (visual typesetting) 来说可能已经足够了。但 \footnotemark 到 \footnotetext 之间的关系不如 \ref 到 \label 那么强。因此，通常不清楚哪个 \footnotemark 引用了哪个 \footnotetext。但这对于实现超链接是必要的。因此，hyperref 的实现 (implementation) 不支持 \footnotemark 和 \footnotetext 的可选参数 (optional argument)。

13 提示¹¹

13.1 选项值中的空格

不幸的是，如果选项是在 `\documentclass` 或 `\usepackage`(或`\RequirePackage`) 中给出的，LaTeX 会从选项中删除空格，例如：

```
\usepackage[pdfborder=0 0 1]{hyperref}
```

`hyperref` 宏包现在获得

```
pdfborder=001
```

结果是一个无效的 PDF 文件。作为一个变通办法，可以使用：

```
\usepackage[pdfborder={0 0 1}]{hyperref}
```

一些选项也可以在 `\hypersetup` 中给出

```
\hypersetup{pdfborder=0 0 1}
```

在 `\hypersetup` 中，选项被直接处理为键值选项 (key value options)(请参阅 `keyval` 宏包)，而在值部分 (value part) 没有空格剥离 (space stripping)。

或者，LaTeX 的选项处理系统 (option handling system) 可以通过 `kvoptions-patch` 宏包 (来自 `kvoptions` 项目) 或 `xkvltxp` 宏包 (来自 `xsetkeys` 项目) 之一来适应 (adapted to) 键值选项。

13.2 Index with makeindex

- 如果设置了 `hyperindex` 选项 (默认值)，则 `hyperref` 宏包通过 `encap` 机制添加 `\hyperpage` 命令 (请参阅 `Makeindex` 的文档)。`\hyperpage` 使用由 `hyperref` 在每个页面上 (默认值) 设置的页面锚点 (page anchors)。但是，在默认情况下，锚点名称中使用的页码采用阿拉伯格式 (arabic form)。如果使用其他格式的页码 (book 类用 `\frontmatter`、`\romannumbering`、...)，则页面锚点不是唯一的。因此，建议使用 `plainpages=false` 选项。
- `Makeindex` 的 `encap` 机制只允许使用一个命令 (请参阅 `Makeindex` 的文档)。如果用户设置了这样的命令，`hyperref` 将取消其 `\hyperpage` 命令。使用逻辑标记 (logical markup) 可以很容易地解决这种情况：

¹¹ 本节从自述文件 (README file) 中删除，需要更多地集成到手册中。

```

\usepackage{makeidx}

\makeindex

\usepackage[hyperindex]{hyperref}

\newcommand*{\main}[1]{\textbf{\hyperpage{#1}}}

...

\index{Some example|main}

```

- Scientific Word/Scientific WorkPlace 用户可以使用带有 `hyperindex=false` 的宏包健壮索引 (package `robustindex`)。
- 其他 `encap` 字符可以通过 `encap` 选项进行设置。“?” 的使用示例：

```

\usepackage[encap=?]{hyperref}

```

- 另一种可能性是通过 `makeindex` 的样式文件 (style file) 插入 `\hyperpage`。在这种情况下，`hyperref` 的插入 (insertion) 将被 `hyperindex=false` 禁用。`\hyperpage` 将被定义，而与 `hyperindex` 的设置无关。

```

%%% cut %%% hyperindex.ist %%% cut %%%

delim_0 ", \hyperpage{"
delim_1 ", \hyperpage{"
delim_2 ", \hyperpage{"
delim_n "}, \hyperpage{"
delim_t "}"
encap_prefix "}\"
encap_infix "{\hyperpage{"
encap_suffix "}"

%%% cut %%% hyperindex.ist %%% cut %%%

```

13.3 警告 “bookmark level for unknown <foobar> defaults to 0”

摆脱这样的警告，可以这样做：

```

\makeatletter
\providecommand*{\toclevel@<foobar>}{0}
\makeatother

```

13.4 图形中的链接锚点

`caption` 命令增加计数器, 这里是 `hyperref` 设置相应锚点的位置。不幸的是, 标题设置在图形 (figures) 的下方, 因此如果链接跳转到图形, 则图形不可见。在这种情况下, 请尝试 `hypcap` 宏包, 它实现了一种方法来避免 (circumvent) 这个问题。

13.5 书签和 pdf 信息条目中的其他 unicode 字符

```
\documentclass[pdftex]{article}
```

```
\usepackage[unicode]{hyperref}
```

支持其他 unicode 字符:

例如: `\{a}` and `\d{a}`

1. 获取包含 unicode 数据的列表 (list), 例如:

<http://www.unicode.org/Public/UNIDATA/UnicodeData.txt>

2. 识别字符 (`\{a}`、`\d{a}`):

```
0227;LATIN SMALL LETTER A WITH DOT ABOVE;...
```

```
1EA1;LATIN SMALL LETTER A WITH DOT BELOW;...
```

3. 计算八进制代码 (octal code):

文件中该行的第一个字符是十六进制值 (hex values), 转换每个字节, 并在前面加上反斜杠。(这将进入 PDF 文件。)

```
0227 -> \002\047
```

```
1EA1 -> \036\241
```

4. 转换为 `hyperref` 能够理解 (understand) 的形式:

`Hyperref` 必须知道第一个字节的起始位置, 这由 9(8 和 9 不能出现在八进制数字中) 标记:

```
\002\047 -> \9002\047
```

```
\036\241 -> \9036\241
```

可选: 8用于缩写 (abbreviations):

```
\900 = \80, \901 = \81, \902 = \82, ...
```

```
\9002\047 -> \82\047
```

5. 用 LaTeX 声明字符:

```
\DeclareTextCompositeCommand{\.}{PU}{a}{\82\047}
\DeclareTextCompositeCommand{\d}{PU}{a}{\9036\241}

\begin{document}
\section{\={a}, \d{a}, \'{a}, \.a}
\end{document}
```

13.6 脚注

对脚注 (footnote) 的支持相当有限。无序使用 `\footnotemark` 和 `\footnotetext` 或重复使用 `\footnotemark` 超出了范围 (beyond the scope)。在这里, 您可以通过 `hyperfootnotes=false` 禁用 `hyperref` 的脚注支持 (footnote support), 也可以篡改内部宏 (internal macros), 例如:

```
\documentclass{article}
\usepackage{hyperref}
\begin{document}
Hello%
\footnote{The first footnote}
World%
\addtocounter{footnote}{-1}%
\addtocounter{Hfootnote}{-1}%
\footnotemark.
\end{document}
```

或

```
\documentclass{article}

\usepackage{hyperref}
```



```

\begin{document}

\makeatletter

A%
\footnotemark
\let\saved@Href@A\Hy@footnote@currentHref
% remember link name
B%
\footnotemark
\let\saved@Href@B\Hy@footnote@currentHref
b%
\addtocounter{footnote}{-1}%
\addtocounter{Hfootnote}{-1}% generate the same anchor
\footnotemark
C%
\footnotemark
\let\saved@Href@C\Hy@footnote@currentHref

\addtocounter{footnote}{-2}%
\let\Hy@footnote@currentHref\saved@Href@A
\footnotetext{AAAA}%
\addtocounter{footnote}{1}%
\let\Hy@footnote@currentHref\saved@Href@B
\footnotetext{BBBBB}%
\addtocounter{footnote}{1}%
\let\Hy@footnote@currentHref\saved@Href@C
\footnotetext{CCCC}%

\end{document}

```

13.7 从属计数器

有些计数器没有唯一值,并要求其他计数器的值是唯一的。例如,节 (sections) 或图形 (figures) 可以在章 (chapters) 内编号, 或者 `\newtheorem` 与可选的计数器参数 (optional counter argument) 一起使用。在内部, LaTeX 使用 `\@addtoreset` 将依赖于另一个计数器的计数器重置。将 `hyperref` 钩住 (hooks into) `\@addtoreset` 以捕捉 (catch) 这种情况。此外, `amsmath` 宏包中的 `\numberwithin` 也被 `hyperref` 捕获。

但是, 如果从属计数器 (subordinate counters) 的定义发生在加载 `hyperref` 之前, 那么调用 `\@addtoreset` 的旧含义 (old meaning) 时不添加 `hyperref`。然后可以相应地重新定义伴随计数器宏 (companion counter macro) `\theH<counter>`。或者在加载 `hyperref` 之后移动 (move) 从属计数器的定义。

`\newtheorem` 的示例, 有问题的情况:

```
\newtheorem{corA}{CorollaryA}[section]
\usepackage{hyperref}
```

解决方案 a)

```
\usepackage{hyperref}
\newtheorem{corA}{CorollaryA}[section]
```

解决方案 b)

```
\newtheorem{corA}{CorollaryA}[section]
\usepackage{hyperref}
\newcommand*{\theHcorA}{\theHsection.\number\value{corA}}
```

14 历史和鸣谢

hyperref 宏包来自 hyperbasics.tex 和 hypertex.sty, 它们的原作者分别是 Tanmoy Bhattacharya (坦莫伊·巴塔查里亚) 和 Thorsten Ohl (托尔斯滕·奥尔)。hyperref 宏包一开始只是他们工作的一个简单端口 (simple port), 以达到 L^AT_EX 2_ε 标准, 但最终我几乎重写了所有内容, 因为我不理解很多原始内容, 只想让它与 L^AT_EX 一起工作。我要感谢 Arthur Smith (亚瑟·史密斯)、Tanmoy Bhattacharya (坦莫伊·巴塔查里亚)、Mark Doyle (马克·道尔)、Paul Ginsparg (保罗·金斯帕格)、David Carlisle (大卫·卡莱尔)、T. V. Raman (T·V·拉曼) 和 Leslie Lamport (莱斯利·兰伯特) 的评论 (comments)、请求 (requests)、想法 (thoughts) 和代码 (code), 使宏包进入第一个可用状态。在源代码中提到了许多其他人, 因为他们发现了问题, 我不得不在后来的版本中更改代码。

Tanmoy (坦莫伊) 发现了很多漏洞 (bug), 并且 (甚至更好的是) 经常修复这些漏洞, 这使得该宏包更加健壮 (robust)。在 RevT_EX 身上花费的时间完全归功于他! Bill Moss (比尔·摩斯) 对包括原生 PDF 支持在内的后续版本的调查发现了许多漏洞, 他的测试值得赞赏。Hans Hagen (汉斯·哈根) 对 PDF 提供了很多见解。

Berthold Horn (伯托尔德·霍恩) 为 dvipsone 和 dviwindo 驱动程序 (drivers) 提供了帮助、鼓励和赞助。Sergey Lesenko (谢尔盖·莱森科) 提供了 dvipdf 所需的更改, Hàn Thê Thành 提供了 pdftex 所需要的所有信息。Patrick Daly (帕特里克·戴利) 好心地更新了他的 natbib 宏包, 以便与 hyperref 轻松集成。Michael Mehlich (迈克尔·迈赫利希) 的 hyper 宏包 (与 hyperref 并行开发) 向我展示了一些问题的解决方案。希望有一天这两个宏包能合二为一。

表单创建 (forms creation) 这一节在很大程度上归功于 T. V. Raman (T·V·拉曼), 他给予了鼓励、支持和想法; Thomas Merz (托马斯·梅尔茨) 的著作 *Web Publishing with Acrobat/PDF* 提供了重要的见解; D. P. Story (D·P·斯托里), 他关于 pdfmarks 和表单 (forms) 的详细文章解决了许多实际问题; 以及 Hans Hagen (汉斯·哈根), 他在 pdftex 中解释了如何做到这一点。

Steve Peter (史蒂夫·彼得) 在 2003 年 7 月重新创建了手册的源代码, 此前该代码已经丢失。

特别感谢 David Carlisle (大卫·卡莱尔) 对 backref 模块、ps2pdf 和 dviwindo 的支持, 他经常重写我的错误代码, 以及对 xr 宏包进行修改以适应 hyperref。

15 GNU 自由文档许可证¹²

第 1.2 版, 2002 年 11 月

版权所有 © 2000,2001,2002 自由软件基金会

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

任何人都可以复制并分发本许可证文档的原始拷贝, 但不允许进行修改。

前言

本许可证用于使得手册、教材或其他功能的有用的文档在是“free”(自由)的: 保证任何人确实可以自由地拷贝与分发经过或未经改动的该文档, 无论是否用于商业目的。此外, 本许可证保护作者和出版者对他们的作品的信誉, 不需要对其他人的改动负责。

这个许可证是一种“copyleft”, 这意味着基于遵循 GPL 文档的衍生作品必须也是同样自由的。这个许可证是为自由软件设计的 copyleft 许可证 GPL 的补充。

我们为了用于自由软件的手册设计了这个许可证, 因为自由软件需要自由的文档: 一个自由的程序应该跟随着一个给出同样自由的文档一同发布。但这个许可证不仅仅限于软件的手册, 它可以被用于任何文本作品, 无论它的主题或它是否作为一本印刷的书籍被发行。我们建议将这个许可证主要用于说明或参考性的文献作品。

15.1 适用性与定义

本许可证适用于包含一个宣布该作品在本许可证下发布的声明的使用任何介质的任何手册或其他作品。这个声明授权在这里声明的条件下, 在任何时间、地点、无版权地使用作品的许可。下述的“文档”指任何这样的手册或作品。任何公众中的一员都是被授予许可的人, 这里被称为“你”。如果你在需要版权法律许可的情况下复制、修改或分发这个作品, 你就接受了这个许可证。

文档的“修订版本”意味着包含文档或文档的一部分的作品, 或者是原始拷贝, 或者进行了修改和/或翻译成为了其他的语言。

“次要章节”是文档中特定的附录或引言中的节, 专用于处理文档的出版者或作者与文档的全部主题(或相关问题)的关系, 不包含任何在全部主题之中的内容。(也就是说, 如果文档

¹² 这里的中译版出自: <http://www.linuxfocus.org/Chinese/team/fdl.html>, 来自: 王旭 <wangxu(at)linuxfocus.org>。以前我也翻译过, 但没有王旭老师的这个版本好, 因此照搬他的版本, 向他致敬! 还可以参考另外两个中译版 (V1.3):

① <https://zhuanlan.zhihu.com/p/603706108>; ② <https://book.huihoo.com/free-software-free-society/fdl/index.html>

是一本数学教材，“次要章节”可能不包含任何数学内容。) 其中的关系可能是和主题的历史关联，或者是相关问题，或者是关于主题的有关法律、商业、哲学、伦理或政治关系。

“不可变章节”是指定的标题的次要章节，这个标题在文档以本许可证发布的声明中被声明为不可变章节。如果一个章节不符合上述的次要章节的定义，它就不可以被声明为不可变章节。文档可以没有不可变章节。如果文档没有指定不可变章节，则视为没有不可变章节。

“封皮文本”是一些在文档以本许可证发布的声明中被列为封面文本或封底文本的特定的短段落。封面文本最多 5 个单词，封底文本最多 25 个单词。

文档的“透明”(Transparent) 拷贝是一个机器可读的拷贝，使用公众可以得到其规范的格式表达，这样的拷贝适合于使用通用文本编辑器、(对于像素构成的图像) 通用绘图程序、(对于绘制的图形) 广泛使用的绘画程序直接修改文档，也适用于输入到文本格式处理程序或自动翻译成各种适于适用于输入到文本格式处理程序的格式。一个用其他透明文件格式表示的拷贝，如果该格式的标记（或缺少标记）已经构成了对读者的后续的修改的障碍，那么就不是透明的。如果用一个图像格式表示确实有效的文本，不论数量多少，都不是透明各式的。不“透明”的拷贝称为“不透明”(Opaque)。

适于作为透明拷贝的格式的例子有：没有标记的纯 ASCII 文本、Texinfo 的输入格式、 \LaTeX 的输入格式、使用公众可用的 DTD 的 SGML 或 XML，符合标准的简单 HTML、可以手工修改的 PostScript 或 PDF。透明的图像格式的例子有 PNG、XCF 和 JPG。不透明的格式包括：尽可以被私有版权的字处理软件使用的私有版权格式、所用的 DTD 和/或处理工具不是广泛可用的 SGML 或 XML，机器生成的 HTML，一些字处理器生成的只用于输出目的的 PostScript 或 PDF。

对于被印刷的书籍，“扉页”(Title Page) 就是扉页本身以及随后的一些用于补充的页，本重要许可需要出现在扉页上。对于那些没有扉页的作品形式，扉页代表接近作品最突出的标题的、在文本正文之前的文本。

章节“Entitled XYZ”(特殊标题 XYZ) 表示文档的一个特定的子单元，其标题就是 XYZ 或包含 XYZ 其后面插入的文本将 XYZ 翻译为其他语言。(这里 XYZ 代表下面提及的特定章节的名字，比如“Acknowledgements”[致谢]，“Dedications”[献给]、“Endorsements”[签名]、或“History”[历史]。) 对这些章节“保护标题”(Preserve the Title) 就是依据这个定义保持这样一个“Entitled XYZ”章节。

文档可能在文档遵照本许可证的声明后面包含免责声明(Warranty Disclaimer)。这些免责声明被认为是包含在本许可证中的，但这仅本视为拒绝担保：免责声明中任何其它的暗示都是无用的并对本许可证的含义没有影响。

15.2 逐字地复制

你可以以任何媒质拷贝并分发文档，无论是否处于商业目的，只要保证本许可证、版权声明和宣称本许可证应用于文档的声明都在所有复制品中被完整地、无任何附加条件地给出即可。你不能使用任何技术手段阻碍或控制你制作或发布的拷贝的阅读或再次复制。不过你可以在复制品的交易中得到报酬。如果你发布足够多的拷贝，你必须遵循下面第 15.3 节中的条件。

你也可以在和上面相同的条件下出租拷贝和向公众放映拷贝。

15.3 大量复制

如果你发行文档的印刷版的拷贝 (或是有印制封皮的其他媒质的拷贝) 多于 100 份，而文档的许可证声明中要求封皮文本，你必须将它清晰明显地置于封皮之上，封面文本在封面上，封底文本在封底上。封面和封底上还必须标明你是这些拷贝的发行者。封面必须以同等显著、可见地完整展现标题的所有文字。你可以在标题上加入其他的材料。改动仅限于封皮的复制，只要保持文档的标题不变并满足这些条件，可以在其他方面被视为是的逐字的复制。

如果需要加上的文本对于封面或封底过多，无法明显地表示，你应该在封皮上列出前面的 (在合理的前提下尽量多)，把其它的放在邻近的页面上。

如果你出版或分发了超过 100 份文档的不透明拷贝，你必须在每个不透明拷贝中包含一份机器可读的透明拷贝，或是在每个不透明拷贝中给出一个计算机网络地址，通过这个地址，使用计算机网络的公众可以使用标准的网络协议没有任何附加条件的下载一个文档的完整的透明拷贝。如果你选择后者，你必须在开始大量分发非透明拷贝的时候采用相当谨慎的步骤，保证透明拷贝在其所给出的位置在 (直接或通过代理和零售商) 分发最后一次该版本的非透明拷贝的时间之后一年之内始终是有效的。

在重新大量发布拷贝之前，请你 (但不是必须) 与文档的作者联系，以便可以得到文档的更新版本。

15.4 修改

在上述第 15.2 节和第 15.3 节的条件下，你可以复制与分发文档的修改后的版本，只要严格的按照本许可证发布修改后的文档，在文档的位置填入修改后的版本，也就是许可任何得到这个修改版的拷贝的人分发或修改这个修改后的版本。另外，在修改版中，你需要做到以下几点：

- A. 使用和被修改文档与以前的各个版本 (如果有的话, 应该被列在文档的版本历史章节中) 有显著不同的扉页 (和封面, 如果有的话)。如果那个版本的原始发行者允许的话, 你可以使用和以前版本相同的标题。
- B. 与作者一样, 在扉页上列出承担修改版本中的修改的作者责任的一个或多个人或实体和至少五个文档的原作者 (如果原作者不足五个就全部列出), 除非他们免除了你的这个责任。
- C. 与原来的发行者一样, 在扉页上列出修改版的发行者的名字。
- D. 保持文档的全部版权声明不变。
- E. 在与其他版权声明邻近的位置加入恰当的针对你的修改的版权声明。
- F. 在紧接着版权声明的位置加入许可声明, 按照下面附录中给出的形式, 以本许可证给公众授于是用修订版本的权利。
- G. 保持原文档的许可声明中的全部不可变章节、封面文字和封底文字的声明不变。
- H. 包含一份未作任何修改的本协议的拷贝。
- I. 保持命名为特殊标题 “History” (版本历史) 的章节不变, 保持它的标题不变, 并在其中加入一项, 该项至少声明扉页上的修改版本的标题、年、新作者和新发行者。如果文档中没有一个特殊标题版本历史的章节, 就新建这一章节, 并加入声明原文档扉页上所列的标题、年、作者与发行者的项, 之后在后面加入如上句所说的描述修改版本的项。
- J. 如果问当中有用于公众访问的文档透明拷贝的网址的话, 保持网址不变, 并同样把它所基于的以前版本的网址给出。这些网址可以放在特殊标题版本历史章节。你可以不给出那些在原文档发行之前已经发行至少四年的版本给出的网址, 或者该版本的发行者授权不列出网址。
- K. 对于任何以特殊标题 “Acknowledgements” (致谢) 特殊标题 “Dedications” (献给) 命名的章节, 保持标题的章节不变, 并保持其全部内容和对每个贡献者的感谢与列出的奉献的语气不变。
- L. 保持文档的所有不可变章节不变, 不改变它们的标题和内容。章节的编号或等价的东西不被认为是章节标题的一部分。
- M. 删除以特殊标题 “Endorsements” (签名) 命名的章节。这样的章节不可以被包含在修改后的版本中。

N. 不要把一个已经存在的章节重命名为特殊标题“Endorsements”(签名) 或和任何不可变章节的名字相冲突的名字。

O. 保持任何免责声明不变。

如果修改版本加入了新的符合次要章节定义的引言或附录章节，并且不含有从原文档中复制的内容，你可以按照你的意愿将它标记为不可变。如果需要这样做，就把它们的标题加入修改版本的许可声明的不可变章节列表之中。这些标题必须和其他章节的标题相区分。

你可以加入一个命名为特殊标题“Endorsements”的章节，只要它只包含对你的修改版本由不同的各方给出的签名—例如书评或是声明文本已经被一个组织认定为一个标准的权威定义。

你可以加入一个最多五个字的段落作为封面文本和一个最多 25 个字的段落作为封底文本，把它们加入到修改版本的封皮文本列表的末端。一个实体之可以加入 (或通过排列制作) 一段封面或封底文本。如果原文档已经为该封皮 (封面或封底) 包含了封皮文本，由你或你所代表的实体先前加入或排列的文本，你不能再新加入一个，但你可以在原来的发行者的显示的许可下替换掉原来的那个。

作者和发行者不能通过本许可证授权公众使用他们的名字推荐或暗示认可任何一个修改版本。

15.5 合并文档

遵照第 15.4 节所说的修改版本的规定，你可以将文档和其他文档合并并以本许可证发布，只要你在合并结果中包含原文档的所有不可变章节，对它们不加以任何改动，并在合并结果的许可声明中将它们全部列为不可变章节，而且维持原作者的免责声明不变。

合并的作品仅需要包含一份本许可证，多个相同的不可变章节可以由一个来取代。如果有多个名称相同、内容不同的不可变章节，通过在章节的名字后面用包含在括号中的文本加以原作者、发行者的名字 (如果有的话) 来加以区别，或通过唯一的编号加以区别。并对合并作品的许可声明中的不可变章节列表中的章节标题做相同的修改。

在合并过程中，你必须合并不同原始文档中任何以特殊标题“版本历史”(History) 命名的章节，从而形成新的特殊标题“版本历史”的章节；类似地，还要合并特殊标题“致谢”(Acknowledgements) 和“献给”(Dedications) 命名的章节。你必须删除所有以特殊标题“签名”(Endorsements) 命名的章节。

15.6 文档的合集

你可以制作一个文档和其他文档的合集，在本许可证下发布，并在合集中将不同文档中的多个本许可证的拷贝以一个单独的拷贝来代替，只要你在文档的其他方面遵循本许可证的逐字地拷贝的条款即可。

你可以从一个这样的合集中提取一个单独的文档，并将它在本许可证下单独发布，只要你想这个提取出的文档中加入一份本许可证的拷贝，并在文档的其他方面遵循本许可证的逐字地拷贝的原则。

15.7 独立作品的聚合体

文档或文档的派生品和其它的与之相分离的独立文档或作品编辑在一起，在一个大包中或大的发布介质上，如果其结果著作权对编辑作品的使用者的权利的限制没有超出原来的独立作品的许可范围，称为文档的“聚合体”(aggregate)。当以本许可证发布的文档被包含在一个聚合体中的时候，本许可证不施加于聚合体中的本来不是该文档的派生作品的其他作品。

如果第 15.3 节中的封皮文本的需求适用于文档的拷贝，那么如果文档在聚合体中所占的比重小于全文的一半，文档的封皮文本可以被放置在聚合体内包含文档的部分的封皮上，或是电子文档中的等效部分。否则，它必须位于整个聚合体的印刷的封皮上。

15.8 翻译

翻译被认为是一种修改，所以你可以按照第 15.4 节的规定发布文档的翻译版本。如果要将文档的不可变章节用翻译版取代，需要得到著作权人的授权，但你可以将部分或全部不可变章节的翻译版附加在原始版本的后面。你可以包含一个本许可证和所有许可证声明、免责声明的翻译版本，只要你同时包含他们的原始英文版本即可。当翻译版本和英文版发生冲突的时候，原始版本有效。

在文档的特殊章节“致谢”(Acknowledgements)、“献给”(Dedications)、“版本历史”(History) 章节，保留其标题(第 15.1 节)的要求(第 15.4 节)通常需要更改实际标题。

15.9 许可的终止

除非确实遵从本许可证，你不可以对遵从本许可证发布的文档进行复制、修改、附加许可证或发布。任何其它的试图复制、修改、附加许可、发布本文档的行为都是无效的，并自动

终止本许可证所授予你的权利。然而其他从你这里依照本许可证得到的拷贝或权力的人 (或组织) 得到的许可证都不会终止, 只要他们仍然完全遵照本许可证。

15.10 本协议的未来修订版本

未来的某天, 自由软件基金会 (FSF) 可能会发布 GNU 自由文档许可证的修订版本。这些版本将会和现在的版本体现类似的精神, 但可能在解决某些问题和利害关系的细节上有所不同。参阅 <http://www.gnu.org/copyleft/>。

本许可证的每个版本都有一个唯一的版本号。如果文档指定服从一个特定的本协议版本“或任何之后的版本” (or any later version), 你可以选择遵循指定版本或自由软件基金会的任何更新的已经发布的版本 (不是草案) 的条款和条件来遵循。如果文档没有指定本许可证的版本, 那么你可以选择遵循任何自由软件基金会曾经发布的版本 (不是草案)。

附录：如何使用本许可证

要使用本许可证发布你写的文档, 请在文档中包含本许可证的一个副本, 并在紧接着扉页之后加入如下版权声明与许可声明:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

如果你有不可变章节、封面文本和封底文本, 请将 “with...Texts.” 一行替换为:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

如果你有不可变章节而没有封皮文本或上述三项的其他组合, 将两个版本结合使用以满足实际情况。

如果你的文档包含有意义的程序示例代码, 我们建议您同时将代码按照您的选择以自由软件许可证发布, 比如 GNU 通用公共许可证。以授权它们作为自由软件被使用。