

LaTeX 中的超文本标记：一份针对 hyperref 宏包的手册

Sebastian Rahtz

Heiko Oberdiek

February 2011

Contents

1 引言	1
2 隐式行为	2
3 宏包的选项	3
3.1 一般选项	3
3.2 配置选项	4
3.3 后端驱动	4
3.4 扩展选项	5
3.5 特定的 PDF 显示选项	6
3.6 PDF 显示和信息选项	7
3.7 PDF 属性信息选项	9
3.8 选项完整列表	9
4 其他用户宏	12
4.1 书签	15
4.1.1 设置书签	15
4.1.2 替代宏	16
4.2 实用宏	17
5 Acrobat 的具体行为	17
6 PDF 和 HTML 的交互环境	18
6.1 交互环境的参数	19
6.2 交互环境选项参数	20
7 定义一个新的驱动	21
8 对其他宏包的特别支持	21
9 历史与感谢	22

1 引言

¹ hyperref 宏包是源于和基于 HyperTeX 项目，详见 <http://xxx.lanl.gov/hypertex/>。它扩展了 LaTeX 中所有的交叉引用命令的功能 (如生成目录、参考文献等) 来为驱动生成超链接提

¹水平有限，先就这样了。随着学习深入我会慢慢改进。有任何意见建议，欢迎 E-mail:phileaslean@gmail.com

供`\special`²命令。`hyperref`宏包也为用户生成特别的超链接，如对外部文件和互联网网址等超链接提供了新的命令。

本手册只是提供了一个关于`hyperref`宏包的简短概述。想了解更多细节，你可以阅读与该宏包一起发布的更多文档，也可以阅读处理`hyperref.dtx`³得到的宏代码文档。你还可以阅读*The L^AT_EX Web Companion*中`hyperref`宏包的相关章节，在那里你可以找到额外的示例。

The HyperT_EX specification⁴中谈到阅读者和翻译者都必须认识如下的`\special`命令结构：

```
href: html:<a href = "href_string">
name: html:<a name = "name_string">
end: html:</a>
image: html:<img src = "href_string">
base_name: html:<base href = "href_string">
```

`href`、`name`和`end`命令常用于在文档的章节间建立基本的超链接。`image`命令（和当前 HTML 阅读器）是用来在页面中的当前位置放置任意格式的图片。`base_name`命令被用来为 DVI 阅读器提供当前文档中 URL 的完整位置，这样与关联 URL 的区域就可以被准确的寻回。

在 T_EX 文件中 `href`和 `name`命令必须结合后面的 `end`命令成对存在。这样在文档中两个成对 `end` 间的 T_EX 命令就形成了一个锚点。在一个 `href`命令情形下，DVI 阅读器中，锚点会被高亮显示，当点击它时，界面就会切换到由 `href_string`指定的地方。锚点结合一个 `name` 命令，代表一个其他超链接可能引用的位置，可能是本地引用（`name_string`用`href="#name_string"`的形式等同于它用 `name` 命令）或者作为 URL 的一部分（用 `URLname_string`的形式）。？这里的 `href_string`可以是一个确定的 URL 或本地标识符，而 `name_string`可以是任何字符串。惟一需要注意的是应避免“'”字符与反斜线（\）一起用，还有就是当 `name_string`看起来像一个 URL 的名字时可能会引起问题。

然而驱动趋向于用精确的 PostScript 或 PDF `\special commands` 产生 PDF。对于不同的驱动，在设置文件中这些命令已经被定义，它们由宏包的选项确定。就目前而言，支持如下驱动：

hypertex DVI 处理器遵循 HyperT_EX 的准则（例如：`xdvi`，`dvips`（需要 `-z`选项）`OzTeX`，和 `Textures`）

dvips 专门为 `dvips`提供`\special`命令

dvipsone 专门为 `dvipsone`提供`\special`命令

ps2pdf 由早期 Ghostscript's PDF 编辑器版本为进程输出合适的一种特殊情形。基本上与 `dvips`一样，但保留了一些在 5.21 版本前的变化。

tex4ht 为使用 T_EX4ht提供`\special`命令

pdftex pdfT_EX，Hàn Thê Thành's T_EX 变体，可以直接生成 PDF

dvipdfm 为 Mark Wicks' DVI 转向 PDF 编译器 `dvipdfm`提供`\special`命令

dvipdfmx 一个 `dvipdfm`继承版，为编译器 `dvipdfmx`提供`\special`命令

dviwindo 为 Y&Y's Windows 预览器提供`\special`命令，使超链接生效。

²【译注】T_EX 和 L^AT_EX 的输出用的是“.dvi”文件的格式，“.dvi”是“device-independent”的缩写。“dvi”文件是不完备的，因为它们依赖于外部的字体文件，但除此之外它描述文档的格式是与显示或打印设备无关的。“dvi”的形式可以任意延伸，全因为 T_EX 的“`\special`”命令，它允许新的显示命令嵌入到“.dvi”文件中。“`\special`”命令最初拓展 T_EX 用于实现画线，附图，彩色文本的功能。参见：<http://arxiv.org/hypertex/>

³【译注】扩展名为.dtx 文件为文档化 T_EX 文件。这也是 L^AT_EX 宏包发布的主要格式。通过处理一个.dtx 文件就可以得到该 L^AT_EX 宏包中所包括的宏代码文档。

⁴Arthur Smith 写的一篇文章，下面几段转自这里。详见：<http://www.dcs.ed.ac.uk/home/latex/latex/hyperref/hypertex.txt>

vtex 提供 `\special` 命令，使得在预览中 MicroPress' HTML 和 PDF-producing \TeX 的变化解释为超链接跳转。

textures 为 Textures 提供 `\special` 命令，使超链接在预览器中生效。

xetex 为 Xe \TeX 提供 `\special` 命令

dvips 和 **dvipsone** 的输出必须用 Acrobat Distiller 处理才能生成 PDF 文件。⁵ 通常先用 **hypertex** 驱动来处理再用 **dvips -z** 效果更好，但是 DVI 文件就不可移植的。使用 Hyper \TeX `\special` 命令的好处是，文档可以同时用于支持超文本的 DVI 阅读器中，例如，**xdvi**。

driverfallback 如果没有提供驱动，并且驱动也不能被自动检测到，那么就使用驱动选项，给 **driverfallback** 一个值来选择驱动。例如：

```
driverfallback=dvipdfm
```

能被自动检测驱动 (**pdftex**, **xetex**, **vtex**, **vtexpdfmark**) 在 \TeX 内部就被能识别，因而这些驱动不能作为 **driverfallback** 的值。然而 DVI 驱动程序在 \TeX 运行结束后才运行的，因而它不能在 \TeX 的宏层被检测到。那么 **hyperref** 宏包选择由 **driverfallback** 提供的驱动。如果已经指定了驱动或驱动可以被自动检测到，那么 **driverfallback** 的值就会被忽略。

2 隐式行为

或多或少任何一个 \LaTeX 文档都会在导言区用到 **hyperref** 宏包。

```
\usepackage{hyperref}
```

因为 **hyperref** 宏包会重新定义很多 \LaTeX 命令，所以请确保 **hyperref** 宏包写在最后面，这是最后的机会来保证它不会被重写。愿上天保佑你的所有作为超文本的交叉引用工作都是正确的。例如，`\section` 命令会生成书签和链接，然而 `\section*` 命令结合 `\addcontentsline` 命令时只会生成链接。

另外，**hyperindex** 的选项（见下文）尝试在索引中建立个项目，使得索引可以链接回原文中。而且 **backref** 选项为每个参考文献嵌入了链接回文中引用的链接。其他选项控制着链接的显示方式，并为 PDF 提供额外的输出控制。例如，**colorlinks**，正如字面意思所说的，链接的表示用颜色来取代用边框。这也是本文中采用的样式。

3 宏包的选项

所有关于 **hyperref** 宏包的用户设置，都是用“关键字 = 值”的形式（用 **keyval** 宏包）。这些选项可以在 `\usepackage` 命令的可选参数中设置，也可以用 `\hypersetup` 宏来设置。当宏包被加载了，就会去寻找 **hyperref.cfg** 文件，并读取它。这是一个很方便设置选项的地方。

下面是一个控制文档形式的例子：

- **hyperref.cfg** 设置链接的样式，增加往回引用，并设置 PDF 显示的默认参数。

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

- 文件中的全局选项，能传递到 **hyperref**，如：

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

⁵请确定关闭了 **dvips** 和 **dvipsone** 提供的部分字体下载功能，来支持 Distiller 自己的系统。

- 在`\usepackage` 设置的文档选项，优先于在 `hyperref.cfg` 中设置的选项。

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

正如前面例子中看到的，必须在宏包加载后设置那些 PDF 信息条目（PDF 的标题，PDF 的作者，等）。否则 `LATEX` 就过早的拓展了这些选项的值。`LATEX` 忽略选项中的空格。`pdfborder` 选项需要注意下。如果作为加载宏包时的选项值，这些值要写在大括号中。而在 `\hypersetup` 中不需要大括号。如下例：

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

宏包 ‘`kvoptions-patch`’ 修补了 `LATEX` 使其能够获取选项值并且阻止值过早的拓展。

首次使用前，一些选项可以在任何地方设置，但是很多是禁止的，只能用在 `\usepackage[...]{hyperref}` 中且在 `\begin{document}` 之前，等等。

在如下关键值的描述中，很多选项是不需要值的，假如用到，默认是值是真。下面是关于布尔类型的选项的一个表。这些选项值的真假都是可以人为指定的。

3.1 一般选项

首先这些选项指定了常见的功能和页面大小。

<code>draft</code>	boolean ⁶	<i>false</i>	使超文本选项失效
<code>final</code>	boolean	<i>true</i>	使超文本选项生效
<code>debug</code>	boolean	<i>false</i>	打印 log 诊断信息
<code>verbose</code>	boolean	<i>false</i>	与 <code>debug</code> 的功能一样
<code>implicit</code>	boolean	<i>true</i>	重新定义 <code>L^AT_EX</code> 内部
<code>hypertextnames</code>	boolean	<i>true</i>	为链接推测个名称
<code>naturalnames</code>	boolean	<i>false</i>	用 <code>L^AT_EX</code> 为链接自动生成名字
<code>setpagesize</code>	boolean	<i>true</i>	用特定的驱动命令设置页面尺寸

3.2 配置选项

<code>raiselinks</code>	boolean	<i>true</i>	在 <code>hypertex</code> 驱动中，链接的高度通常是由驱动简单的根据正文的底线来计算的，该参数强制 <code>\special</code> 命令反映链接（其中可能含有图片）的真实高度。
<code>breaklinks</code>	boolean	<i>false</i>	允许链接断成多条。在 PDF 中无效，如果用 <code>pdftex</code> 驱动，其值只能默认设置为真。它允许链接断成多个，但都指向同一个地方。
<code>pageanchor</code>	boolean	<i>true</i>	决定在每页的左上角是否安放一个隐藏的锚点。如果该选项关闭，则 <code>\printindex</code> 将不会形成有效的超链接。
<code>plainpages</code>	boolean	<i>false</i>	强制锚点的页码采用阿拉伯数字的形式来取代格式化的形式。
<code>nesting</code>	boolean	<i>false</i>	允许链接嵌套。但目前没有驱动支持它。

⁶ 【译注】数据类型，布尔类型，只有真和假两种。

注意**breaklinks** 选项：其值会被驱动的属性自动设置过。它会被那些无法支持断链的驱动重写。然而，在任何情况下，这个链接区域都可能会是错位的。

3.3 后端驱动

如果没有指定驱动，宏包试图按如下顺序找到一个驱动。

1. Autodetection (自动检测)，一些 $\text{T}_{\text{E}}\text{X}$ 驱动 ($\text{pdfT}_{\text{E}}\text{X}$, $\text{XeT}_{\text{E}}\text{X}$, $\text{V}_{\text{T}}\text{E}\text{X}$) 在 $\text{T}_{\text{E}}\text{X}$ 宏层可以自动被检测到。
2. `driverfallback`选项。假如已经设置了这个选项，那么它值就作为驱动的选择。
3. Makro `\Hy@defaultdriver`. makro 获取驱动文件名 (不含文件扩展名)。
4. 宏包默认的是驱动是 `hypertex`。

很多发行版都用驱动文件 `hypertex.cfg`和 `hdvips`来定义 `\Hy@defaultdriver`。这样的重新定义是因为对于 PDF 的生成，驱动 `dvips`比起 `hypertex`提供了更多的功能。

<code>driverfallback</code>	如果驱动没有被自动检测到，其值作为选择驱动的参数。
<code>dvipdfm</code>	为使用 <code>dvipdfm</code> 驱动，设置 <code>hyperref</code> 宏包。
<code>dvipdfmx</code>	为使用 <code>dvipdfmx</code> 驱动，设置 <code>hyperref</code> 宏包。
<code>dvips</code>	为使用 <code>dvips</code> 驱动，设置 <code>hyperref</code> 宏包。
<code>dvipsone</code>	为使用 <code>dvipsone</code> 驱动，设置 <code>hyperref</code> 宏包。
<code>dviwindo</code>	为使用 <code>dviwindo</code> 预览器，设置 <code>hyperref</code> 宏包。
<code>hypertex</code>	为使用 <code>HyperT_EX-compliant</code> 驱动设置 <code>hyperref</code> 宏包。
<code>latex2html</code>	为兼容 <code>compatibility</code> 重新定义了一些宏。
<code>nativepdf</code>	<code>dvips</code> 的另一名称。
<code>pdfmark</code>	<code>dvips</code> 的另一名称。
<code>pdftex</code>	为使用 <code>pdftex</code> 程序设置 <code>hyperref</code> 宏包。
<code>ps2pdf</code>	重新定义了一些宏来兼容 Ghostscript's PDF，其他就是跟 <code>dvips</code> 一样。
<code>tex4ht</code>	为用 <code>T_EX4ht</code>
<code>textures</code>	为用 <code>Textures</code>
<code>vtex</code>	使用 MicroPress' <code>V_TE_X</code> ；PDF 和 HTML backends 会被自动检测到。
<code>vtexpdfmark</code>	使用 <code>V_TE_X's</code> PostScript backend。
<code>xetex</code>	使用 <code>XeT_EX</code> 。(为 <code>dvipdfm</code> 使用 backend)

如果你使用 `dviwindo`，你可能就要重新定义宏 `\wwwbrowser` (默认的是 `C:\netscape\netscape7`) 来告诉 `dviwindo` 运行什么程序。因此，对于 Internet Explorer 用户就可能要在 `hyperref.cfg` 中加如下这些东西：

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

3.4 扩展选项

<code>extension</code>	<code>text</code>	如果用 <code>xr</code> 宏包的话，将文件的扩展名 (如， <code>dvi</code>) 附加到创建的文件链接中。
<code>hyperfigures</code>	<code>boolean</code>	

⁷netscape, 网景，一款网页浏览器。

<code>backref</code>	text		在参考文献的每个条目中增加链接回原文的链接，用章节标号表示。在每个 <code>\bibitem</code> 后要留有一空行，这个才可能生效。其值可以是 <code>section</code> 、 <code>slide</code> 、 <code>page</code> 、 <code>none</code> ，或者 <code>false</code> 。假如没有赋值，默认是 <code>section</code> ，即章节标号。
<code>pagebackref</code>	boolean	<i>false</i>	在参考文献的每个条目中增加链接回原文的链接，用页码表示。
<code>hyperindex</code>	boolean	<i>true</i>	使得索引条目中的页码成为超链接。继承自唯一的页码锚点 (<code>pageanchor</code> , ...) <code>pageanchors</code> 和 <code>plainpages=false</code> 。
<code>hyperfootnotes</code>	boolean	<i>true</i>	使得正文中的脚注标识成为超链接来链接到脚注。容易出问题 ...
<code>encap</code>			为超链接索引设置 <code>encap</code> 属性
<code>linktocpage</code>	boolean	<i>false</i>	使页码，非正文，链接到正文目录，图形目录，表格目录。
<code>breaklinks</code>	boolean	<i>false</i>	允许链接断裂成多条并指向同一地方。
<code>colorlinks</code>	boolean	<i>false</i>	使得链接和锚点具有颜色。其颜色取决于链接的类型。目前可区分的链接类型是引文，页码引用，URLs，本地引用，和其他链接。尽管边框也有颜色，但打印的时候链接文本的颜色也会保留。
<code>linkcolor</code>	color	<i>red</i>	设置内部普通链接的颜色，默认是红色。
<code>anchorcolor</code>	color	<i>black</i>	设置文中锚点的颜色，默认是黑色。
<code>citecolor</code>	color	<i>green</i>	设置文中参考文献引用的颜色，默认是绿色。
<code>filecolor</code>	color	<i>cyan</i>	设置打开本地文件的颜色，默认是蓝绿色。
<code>menucolor</code>	color	<i>red</i>	设置 Acrobat 菜单栏的颜色，默认使红色。
<code>runcolor</code>	color	<i>filecolor</i>	设置 <code>run</code> 链接（运行注释）的颜色，默认是文档颜色。
<code>urlcolor</code>	color	<i>magenta</i>	设置 URLs 的颜色，默认是品红色。
<code>allcolors</code>	color		设置所有颜色选项（边框和文档除外）。
<code>frenchlinks</code>	boolean	<i>false</i>	用小型大写字体代替用颜色表示链接
<code>hidelinks</code>			隐藏链接，即移除颜色和边框。

注意：在加载 `LATEX color` 宏包之后，在使用前，所有的颜色名称必须定义好。

3.5 特定的 PDF 显示选项

<code>bookmarks</code>	boolean	<i>true</i>	设置 Acrobat 的书签，用类似目录的方法，需要运行两次 <code>L^AT_EX</code> 。一些书签文件（扩展名为 <code>.out</code> ）可能需要转换成 <code>L^AT_EX</code> 码，然而书签必需用 <code>PDFEncoding</code> 。为此，如果加入一行 <code>\let\writebookmarks\relax</code> ， <code>.out</code> 将不会被 <code>L^AT_EX</code> 重写。
<code>bookmarksopen</code>	boolean	<i>false</i>	展开 PDF 目录书签的所有层次，默认是关闭。

bookmarksopenlevel	parameter		设置 PDF 书签目录展开的层次，默认是最高级，可选参数为，1，2，3，
bookmarksnumbered	boolean	false	将章节标号加入到 PDF 书签目录中，默认是关闭。
bookmarkstype	text	toc	指定 ‘toc’ 文件来模拟？
CJKbookmarks	boolean	false	该选项用于生成 CJK 书签。 hyperref 宏包支持 CJK 宏包的预处理模式和正常模式。在创建书签的时候， hyperref 宏包只是简单用相应的版本兼容的特征码代替 CJK 的宏集。注意如果没有 hyperref 宏包的 unicode 选项，你得到的 PDF 文件实际上是违背 PDF 的设定的，因为非 unicode 特征码被使用。同时也注意， CJKbookmarks 选项不能和 unicode 同时使用。对于 PDF 文档只能用 Unicode 编码，目前没有提供将非 Unicode 编码书签转成 Unicode 编码书签的功能。 ⁸
pdfhighlight	name	/I	设置链接被选中时的反映。默认的是/I，表示翻转；其他可能是/N，不产生特别反应；/O，使链接出列还有/P，设置高亮。
citebordercolor	RGB color	0 1 0	设置引用边框的颜色。
filebordercolor	RGB color	0 .5 .5	设置链接到文档的边框颜色。
linkbordercolor	RGB color	1 0 0	设置普通链接边框的颜色。
menubordercolor	RGB color	1 0 0	设置 Acrobat 菜单链接边框的颜色。
urlbordercolor	RGB color	0 1 1	设置 URLs 的边框的颜色。
runbordercolor	RGB color	0 .7 .7	设置 run 链接边框的颜色。
allbordercolors			设置所有边框的颜色。
pdfborder		0 0 1	设置链接边框的样式。默认边框是 1pt 的细线。但是当启用 colorlinks 选项，则 colorlinks 会去掉边框。

注意链接边框的颜色设置只能由 3 个数字组成，范围是 0 到 1，分别表示红色，绿色，蓝色，即 RGB 颜色值。在 6.76a 版本后可以在 **T_EX** 中定义颜色。特别是因为得到了 **xcolor**宏包的支持，在 **(x)color**宏包中的颜色都可以使用。想了解更多信息，详见 **hycolor**宏包。

这些书签命令都存储在一个叫 *jobname.out*的文件中。该文件不会被 **L^AT_EX** 处理，所以任何改动都会通过。如有需要你可以处理该文件，为了 *.out* 文件将不会在下次 **T_EX** 运行时被重写，但必需编辑时加入下面这行：

`\let\WriteBookmarks\relax`

3.6 PDF 显示和信息选项

baseurl	URL		为 PDF 文档设置基本的 URL。
pdfpagemode	text	empty	确定 PDF 在 Acrobat 中打开的样式。可选值有 UseNone ， UseThumbs (显示缩略图)， UseOutlines (显示书签)， FullScreen ， UseOC (PDF 1.5) 和 UseAttachments (PDF 1.6)。如果没有明确值，并且设置了书签，则默认为 UseOutlines ，即展开书签。
pdftitle	text		设置 PDF 属性中标题栏中的信息
pdfauthor	text		设置 PDF 属性中的作者栏中的信息

⁸解决书签乱码的编译顺序在 C_TEx 下按照 L^AT_EX，BibT_EX，pdf L^AT_EX，顺序可以完全编译。

pdfsubject	text		设置 PDF 属性中的主题栏中的信息
pdfcreator	text		设置 PDF 属性中的创建者栏中的信息
pdfproducer	text		设置 PDF 属性中的制作者栏中的信息
pdfkeywords	text		设置 PDF 属性中的关键字栏中的信息
pdftrapped	text	<i>empty</i>	设置 PDF 属性中文档的限制项目 其值可以是 True , False 和 Unknown 。 对于未设值, 则意味着没有设置限制项目。
pdfinfo	key value list	<i>empty</i>	另一种设置 PDF 文档属性信息的接口
pdfview	text	<i>XYZ</i>	为每个链接设置默认的 PDF 视野
pdfstartpage	text	<i>1</i>	设置打开 PDF 文档时显示第几页, 默认显示第一页。
pdfstartview	text	<i>Fit</i>	PDF 文件初始视图, 默认值为 <i>Fit</i> , 可选值为 <i>FitH</i> 页宽适合窗口, <i>FitV</i> 页高适合窗口, <i>FitR</i> 适合窗口对角线, <i>FitB</i> 版面适合窗口, <i>FitBH</i> 版面宽适合窗口, <i>FitBV</i> 版面高适合窗口, <i>XYZ</i> 自定义放大率
pdfremotestartview	text	<i>Fit</i>	开启阅读远程 PDF 文档时的初始视图。
pdfpagescrop	n n n n		设置剪切尺寸, 例如: 53 486 389 754
pdfcenterwindow	boolean	<i>false</i>	把文档窗口设置到屏幕中央。
pdfdirection	text	<i>empty</i>	设置方向
pdfdisplaydoctitle	boolean	<i>false</i>	在 PDF 阅读器的标题栏中用文档标题取代文件名称。 ⁹ (默认不启用)
pdfduplex	text	<i>empty</i>	打印对话框的纸张处理选项
pdfffitwindow	boolean	<i>false</i>	使文档窗口大小适合文档大小。
pdflang	text	<i>empty</i>	设置 PDF 语言的标识符 (RFC 3066)
pdfmenubar	boolean	<i>true</i>	使得 PDF 阅读器的菜单栏可见
pdfnewwindow	boolean	<i>false</i>	为链接的新 PDF 文件生成新的窗口
pdfnonfullscreenpagemode	boolean	<i>empty</i>	页面模式设置为非全屏模式
pdfnumcopies	integer	<i>empty</i>	设置打印份数
pdfpagelayout	text	<i>empty</i>	设置 PDF 页面布局
pdfpagelabels	boolean	<i>true</i>	设置 PDF 页码标签
pdfpagetransition	text	<i>empty</i>	设置页面过渡样式, 参数后可加子参数: <i>/Dm</i> 、 <i>/Bi</i> 、 <i>/M</i> 、 <i>/H</i> 、 <i>/V</i> 、 <i>/I</i> 、 <i>/O</i> (需全屏显示模式) <i>Dissolve</i> 马赛克散开, <i>Wipe</i> 下拉帘幕, <i>Split</i> 上下拉帘幕, <i>Glitter</i> 溶化, <i>Blinds</i> 百叶窗翻转, <i>Box</i> 渐缩框
pdfpicktraybypdfsize	text	<i>empty</i>	自动选择纸张
pdfprintarea	text	<i>empty</i>	设置打印范围, 参数与 <i>pdfviewarea</i> 相同
pdfprintclip	text	<i>empty</i>	设置剪切范围, 参数与 <i>pdfviewarea</i> 相同
pdfprintpagerange	n n (n n)*	<i>empty</i>	设置打印页码范围 n n
pdfprintscaling	text	<i>empty</i>	对打印对话框设置页面缩放选项 (<i>/PrintScaling</i> PDF 1.6, <i>NoneAppDefault</i> ,) <i>e</i> 显示 PDF 工具栏
pdftoolbar	boolean	<i>true</i>	
pdfviewarea	text	<i>empty</i>	设置显示区域, <i>MediaBox</i> 媒体框, <i>CropBox</i> 裁切框, <i>BleedBox</i> 出血框, <i>TrimBox</i> 修剪框, <i>ArtBox</i> 作品框
pdfviewclip	text	<i>empty</i>	设置剪贴区域, 参数与 <i>pdfviewarea</i> 相同
pdfwindowui	boolean	<i>true</i>	显示窗口控件

⁹觉得这很人性化, 因为文件名在传播过程中经常被改动。

`unicode` `boolean` `false` 对 PDF 的书签用 Unicode 编码

在 Acrobat 中的每个链接都有自己的放大倍数，被设置成使用 PDF 最适尺寸，这与 \TeX 的是不同的。该单元是 `bp` 和最初是在左下角。可参见 `\hypercalcbp`，在 17 中有解释。`pdf\TeX` 是来支持 XYZ (水平 \times 垂直 \times 放大) 和 `FitBH` 的默认值。然而，使用 `pdfmark` 的驱动不支持默认，所以 `hyperref` 用值 `-32768` 来运行，来生成 Acrobat 的默认值。下面是 `pdfview`, `pdfstartview` 和 `pdfremotestartview` 参数的值：

XYZ	<i>left top zoom</i>	设置坐标和缩放因子。如果任何一个 is <code>null</code> ，将使用源链接值。 <i>null null null</i> 将提供与当前页同样的值。
Fit		适合窗口
FitH	<i>top</i>	宽适合窗口
FitV	<i>left</i>	高适合窗口
FitR	<i>left bottom right top</i>	由指定的四个标参数来适合窗口
FitB		适合页面边框的窗口
FitBH	<i>top</i>	适合页面边框宽的窗口
FitBV	<i>left</i>	适合页面边框高的窗口

`pdfpagelayout` (页面布局) 可以使用如下值：

<code>SinglePage</code>	单页显示
<code>OneColumn</code>	连续单页
<code>TwoColumnLeft</code>	连续对开，奇数在左边
<code>TwoColumnRight</code>	连续对开，奇数在右边
<code>TwoPageLeft</code>	双页显示，奇数在左边 (PDF1.5 版本后)
<code>TwoPageRight</code>	双页显示，奇数在右边 (PDF1.5 版本后)

最后，`pdfpagetransition`，即 PDF 过渡样式选项可以有如下值：

<code>Blinds</code>	<code>/Dm</code>	百叶窗翻转
<code>Box</code>	<code>/M</code>	渐缩框
<code>Dissolve</code>		马赛克散开
<code>Glitter</code>	<code>/Di</code>	类似马赛克散开，溶化
<code>Split</code>	<code>/Dm /M</code>	上下拉帘幕
<code>Wipe</code>	<code>/Di</code>	下拉帘幕

3.7 PDF 属性信息选项

这些属性条目可以用 `pdftitle`, `pdfsubject` 等来设置。选项 `pdfinfo` 提供了另一个设置 PDF 属性信息的接口。它就是关键值的列表。这些关键值的名称就是 PDF 属性栏中的名称。这些关键值有 `Title`, `Subject`, `Trapped` 和其他对应 `pdftitle`, `subject`, `trapped` 等选项的参数。还有未知关键值加入到信息选项中。它们的值是文本字符串 (见 PDF 规范)。例如：

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

3.8 选项完整列表

以下是按字母排序的 `hyperref` 宏包选项的一个完整列表。¹⁰

<code>anchorcolor</code>	<code>black</code> ¹¹	设置锚点 ¹² 文本的颜色
<code>backref</code>	<code>false</code>	开启参考文献的反向链接（默认关闭）
<code>baseurl</code>	<code>empty</code>	设置文档中基本的 URL
<code>bookmarks</code>	<code>true</code>	生成书签
<code>bookmarksnumbered</code>	<code>false</code>	使书签目录中的章节带标号
<code>bookmarksopen</code>	<code>false</code>	展开书签目录
<code>bookmarksopenlevel</code>	<code>\maxdimen</code>	设置书签目录展开的层数，默认值 <code>\maxdimen</code> 为最高，可选 1, 2, 3...
<code>bookmarkstype</code>	<code>toc</code>	书签目录类型，默认是 <code>toc</code> 章节目录，可选 <code>lof</code> 图形目录， <code>lot</code> 表格目录
<code>breaklinks</code>	<code>false</code>	允许链接断行
<code>CJKbookmarks</code>	<code>false</code>	生成中文书签
<code>citebordercolor</code>	<code>0 1 0</code>	设置引用标记边框的颜色
<code>citecolor</code>	<code>green</code>	设置引用标记的颜色
<code>colorlinks</code>	<code>false</code>	使链接为彩色（默认是不启用）
	<code>true</code>	(<code>tex4ht</code> , <code>dviwindo</code>) 在 <code>tex4ht</code> , <code>dviwindo</code> 驱动下则默认是启用。
<code>debug</code>	<code>false</code>	提供锚点的详细信息，和 <code>verbose</code> 效果相同
<code>draft</code>	<code>false</code>	使超链接失效
<code>dvipdfm</code>		使用 <code>dvipdfm</code> 作为驱动
<code>dvipdfmx</code>		使用 <code>dvipdfmx</code> 作为驱动
<code>dvips</code>		使用 <code>dvips</code> 作为驱动
<code>dvipsone</code>		使用 <code>dvipsone</code> 作为驱动
<code>dviwindo</code>		使用 <code>dviwindo</code> 作为驱动
<code>encap</code>		为具有超链接特性的索引设置 <code>encap</code> 特性
<code>extension</code>	<code>dvi</code>	设置链接文件的扩展名，默认是 <code>dvi</code> ，可选 <code>ps</code> 、 <code>pdf</code>
<code>filebordercolor</code>	<code>0 .5 .5</code>	设置文件链接边框的颜色
<code>filecolor</code>	<code>cyan</code>	设置文件链接颜色
<code>final</code>	<code>true</code>	使超文本选项生效
<code>frenchlinks</code>	<code>false</code>	对链接用小型大写字体代替颜色
<code>hyperfigures</code>	<code>false</code>	对图表生成超链接
<code>hyperfootnotes</code>	<code>true</code>	对脚注建立超链接
<code>hyperindex</code>	<code>true</code>	对索引建立超链接
<code>hypertex</code>		使用 <code>HyperTeX</code> 作为驱动
<code>hypertextnames</code>	<code>true</code>	为链接推测个名称
<code>implicit</code>	<code>true</code>	重新定义 <code>LaTeX</code> 内部
<code>latex2html</code>		使用 <code>LaTeX2HTML</code> 作为驱动
<code>legalpaper</code>		使用 <code>legalpaper</code> , 8.5in×14in 纸张
<code>letterpaper</code>		使用 <code>letterpaper</code> , 8.5in×11in 纸张
<code>linkbordercolor</code>	<code>1 0 0</code>	设置内部链接边框的颜色
<code>linkcolor</code>	<code>red</code>	设置内部链接的颜色
<code>linktocpage</code>	<code>false</code>	使页码（不是文本）链接到书签目录，图形目录，表格目录
<code>menubordercolor</code>	<code>1 0 0</code>	设置菜单栏链接边框的颜色

¹⁰部分参考自 `LaTeX` 编辑部

¹¹注：中间的值默认值。

¹²名称很形象，超链接指向的点

menucolor	<i>red</i>	设置菜单栏链接的颜色
nativepdf	<i>false</i>	dvips的另一名称
naturalnames	<i>false</i>	用 L ^A T _E X 为链接自动生成名字
nesting	<i>false</i>	允许对链接嵌套
pageanchor	<i>true</i>	为每页安放一个锚点
pagebackref	<i>false</i>	在参考文献的每个条目中增加链接回原文的链接, 用页码表示
pdfauthor	<i>empty</i>	设置 PDF 属性中的作者栏中的信息
pdfborder	<i>0 0 1</i> <i>0 0 0</i>	PDF 链接边框的宽度 (colorlinks) 设置 PDF 链接边框的大小
pdfcenterwindow	<i>false</i>	把文档窗口设置到屏幕中央
pdfcreator	<i>LaTeX with</i> <i>hyperref</i> <i>package</i>	设置 PDF 属性中的创建者栏中的信息
pdfdirection	<i>empty</i>	设置方向
pdfdisplaydoctitle	<i>false</i>	在 PDF 阅读器的标题栏中用文档标题取代文件名称。 ¹³
pdfduplex	<i>empty</i>	打印对话框的纸张处理选项
pdfffitwindow	<i>false</i>	使文档窗口大小适合文档大小
pdfhighlight	<i>/I</i>	为 PDF 链接设置高亮
pdfinfo	<i>empty</i>	另一种设置 PDF 文档属性信息的接口
pdfkeywords	<i>empty</i>	设置 PDF 属性中的关键字栏中的信息
pdflang	<i>empty</i>	设置 PDF 语言的标识符 (RFC 3066)
pdfmark	<i>false</i>	dvips的另一名称
pdfmenubar	<i>true</i>	使得 PDF 阅读器的菜单栏可见
pdfnewwindow	<i>false</i>	为链接的新 PDF 文件生成新的窗口
pdfnonfullscreenpagemode	<i>empty</i>	页面模式设置为非全屏模式
pdfnumcopies	<i>empty</i>	设置打印份数
pdfpagelayout	<i>empty</i>	设置 PDF 页面布局
pdfpagemode	<i>empty</i>	设置默认的 PDF 显示模式
pdfpagelabels	<i>true</i>	设置底部页码样式
pdfpagescrop	<i>empty</i>	设置裁切尺寸, 例如: 53 486 389 754
pdfpagetransition	<i>empty</i>	设置页面过渡样式, 参数后可加子参数: /Dm、/Bi、/M、/H、/V、/I、/O(需全屏显示模式) Dissolve 马赛克散开, Wipe 下拉帘幕, Split 上下拉帘幕, Glitter 溶化, Blinds 百叶窗翻转, Box 渐缩框
pdfpicktraybypdfsize	<i>empty</i>	自动选择纸张
pdfprintarea	<i>empty</i>	设置打印范围, 参数与 pdfviewarea 相同
pdfprintclip	<i>empty</i>	设置剪切范围, 参数与 pdfviewarea 相同
pdfprintpagerange	<i>empty</i>	设置打印页码范围 n n
pdfprintscaling	<i>empty</i>	设置打印放大率
pdfproducer	<i>empty</i>	设置 PDF 属性中的制作者栏中的信息
pdfremotestartview	<i>Fit</i>	开时阅读远程 PDF 文档时的初始视图
pdfstartpage	<i>1</i>	设置打开 PDF 文档时显示第几页
pdfstartview	<i>Fit</i>	PDF 文件初始视图, 默认值为/Fit, 可选值为 FitH 页宽适合窗口, FitV 页高适合窗口, FitR 适合窗口对角线, FitB 版面适合窗口, FitBH 版宽适合窗口, FitBV 版高适合窗口, XYZ 自定放大率

¹³这也很人性话, 因为文件名在传播过程中经常会被改动。(默认不启用)

pdfsubject	<i>empty</i>	设置 PDF 属性中的主题栏中的信息
pdftex		使用 pdfTeX 作为后端驱动
pdftitle	<i>empty</i>	设置 PDF 属性中标题栏中的信息
pdftoolbar	<i>true</i>	显示 PDF 工具栏
pdftrapped	<i>empty</i>	设置 PDF 属性中文档的限制项目。其值可以是 True , False 和 Unknown , 对于未设值, 则意味着没有设置限制项目。
pdfview	<i>XYZ</i>	链接默认视图, 参数与 pdfstartview 相同
pdfviewarea	<i>empty</i>	设置显示区域, MediaBox 媒体框, CropBox 裁切框, BleedBox 出血框, TrimBox 修剪框, ArtBox 作品框
pdfviewclip	<i>empty</i>	设置剪贴区域, 参数与 pdfviewarea 相同
pdfwindowui	<i>true</i>	显示窗口控件
plainpages	<i>false</i>	用阿拉伯数字为页锚编号
ps2pdf		使用 ps2pdf 作为驱动
raiselinks	<i>false</i>	针对 HyperTeX 驱动, 抬高链接
runbordercolor	<i>0 .7 .7</i>	设置 “run” 链接边框的颜色, 默认 RGB 为 (0, 0.7, 0.7)
runcolor	<i>filecolor</i>	设置 “run” 链接的颜色, 默认为文本颜色
setpagesize	<i>true</i>	用命令设置页面尺寸
tex4ht		使用 TeX4ht 作为驱动
textures		使用 Textures 作为驱动
unicode	<i>false</i>	对 PDF 的书签用 Unicode 编码
urlbordercolor	<i>0 1 1</i>	设置网页边框的颜色
urlcolor	<i>magenta</i>	设置网页链接的颜色 (默认为品红)
verbose	<i>false</i>	附加信息
vtex		使用 VTeX 作为驱动
xetex		使用 XeTeX 作为驱动

4 其他用户宏

如果你需要生成对 URLs 的引用, 或者写精确的链接, 则可以用下面的低层次的用户宏集实现:

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

text 是链接到 *URL* 的文本。URL 必须是完整的地址 (如果定义了可以关联到基本 URL)。无论如何关键的特征符 # 和 ~ 是不能省略的。

其中的可选参数 *options* 能识别 hyperref 宏包的选项 pdfremotestartview, pdfnewwindow 和如下选项:

page: 指定 PDF 文档的开始页。默认第一页。

ismap: 布尔值。假如设置为真, PDF 阅读器应该给 URL 附加坐标作为查询参数。

nextactionraw: 行为参数 |/Next| 的值, 参见 PDF 的详述。

```
\url{URL}
```

类似于 `\href{URL}{\nolinkurl{URL}}`。取决于 `\href` 驱动的检测链接的类型。因而结果可能是 url 链接, 文档链接等。

```
\nolinkurl{URL}
```

像`\url`命令一样书写 *URL*，但只是文本，不是超链接。

```
\hyperbaseurl{URL}
```

预先建立 *URL*，使得写文档时更方便。

```
\hyperimage{imageURL}{text}
```

使用 *text* 作为锚点来对图片嵌入 *URL*。

对于那些能生成 HTML 的驱动，浏览器会把 *URL* 嵌入到图像之中，这时 *text* 就会被完全忽略。

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

标记文档中 (the *text*) 的一段文本为链接目标，并命名为 *category.name*。

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

使 *text* 中的文本链接到 *URL#category.name*。

```
\hyperref[label]{text}
```

使 *text* 链接到 `\ref{label}`。

```
\hyperlink{name}{text}
```

```
\hypertarget{name}{text}
```

由 `\hypertarget` 创建的一个简单内部链接，其中含有两个参数，分别是锚点名称 *name* 和锚点内容 *text*。`\hyperlink` 也有两个参数，*name* 与 `\hypertarget` 中的相同，确定对应的锚点，*text* 则是指向锚点的内容。

注意在 HTML 语言中，`\hyperlink` 命令中的每个链接前必须加上 `#`，使得与当前的文档关联。`\href` 则显示的 *URL* 全称。

```
\phantomsection
```

在本地设置一个锚点。其作用有点类似于 `\hypertarget{}{}` 命令加上一个自动给锚点命名的功能。经常与 `\addcontentsline` 一起用于像章节名称之类的东西（如索引、参考文献、前言）。`\addcontentsline` 与之前最近的锚点关联。

例如：

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

这时在目录中的这条项目指向的是索引页的开始，不是在这页之前的地方。

`\autoref{label}`

这是非常有用的命令`\ref`的替代版，它在引用前提供了一个上下文的标签。这给使用者提供了更大的超链接点击区域（例如将“章节 2”设置为超链接而不仅是数字“2”）。

该标签是由 `hyperref` 宏包使用下表（包括默认值）中的宏更改最初文中`\label`命令生成的。这些宏在文档中可以被`\(re)newcommand`命令定义或重新注意这里的一些宏已经在标准的文档类中定义过了。首字母的大小是区分的，作者要注意。

对如下每个宏，`hyperref`宏包在检查 `*name`命令之前先检查 `*autorefname`命令。例如，在`\figurename`命令前先寻找 `\figureautorefname`命令。

<i>Macro</i>	<i>Default</i>
<code>\figurename</code>	Figure
<code>\tablename</code>	Table
<code>\partname</code>	Part
<code>\appendixname</code>	Appendix
<code>\equationname</code>	Equation
<code>\Itemname</code>	item
<code>\chaptername</code>	chapter
<code>\sectionname</code>	section
<code>\subsectionname</code>	subsection
<code>\subsubsectionname</code>	subsubsection
<code>\paragraphname</code>	paragraph
<code>\Hfootnotename</code>	footnote
<code>\AMSname</code>	Equation
<code>\theoremname</code>	Theorem
<code>\page</code>	page

假如 `babel` 被用到，这是个重新定义的例子：

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

小贴士：`\autoref` 的工作原理是经由引用基于的计数器的名称。假如该计数器用于其他用途，有时候`\autoref`会选择错误的名称。例如，使用`\newtheorem`命令的时候，辅助定理和定理分享一个计数器。`aliascnt`宏包提供一种方法来提供相似的第二计数器来允许定理与辅助定理间的不同。

```
\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}`

它替代了`\pageref`命令并且在引用页码前增加了名称。当然了，先是检查`\pageautorefname`命令，再检查`\pagename`命令。

例如，某个地方的引用你想使用计数器，却不想生成链接，那可以用如下形式：

`\ref*{label}`

`\pageref*{label}`

`\autoref*{label}`

`\autopageref*{label}`

下面是一个经典的写法：

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

当用了`\hyperref`宏包的时候，我们可以用`\ref*{other}`命令来生成正确的序号，但又不生成一个链接，

`\pdfstringdef{macroname}{TeXstring}`

`\pdfstringdef`返回一个包含 PDF 字符串的宏。（目前这是全局范围内的，但不依赖于它。）以下的所有任务，定义和重新定义都是在一起的，来保持它们局域化：

- Switching to PD1 or PU encoding
- Defining the “octal sequence commands” (`\345`): `\edef\3{\string\3}`
- Special glyphs of TeX: `\{`, `\%`, `\&`, `\space`, `\dots`, etc.
- National glyphs (`german.sty`, `french.sty`, etc.)
- Logos: `\TeX`, `\eTeX`, `\MF`, etc.

- Disabling commands that do not provide useful functionality in bookmarks: `\label`, `\index`, `\glossary`, `\discretionary`, `\def`, `\let`, etc.
- L^AT_EX's font commands like `\textbf`, etc.
- Support for `\xspace` provided by the `xspace` package

另外，括弧被保护来避免 PDF 字符串中不安全和不稳定。有关进一步详情，参见和 `hyperref` 宏包一起发布的 Heiko Oberdiek 的 EuroT_EX 文章。

4.1 书签宏

4.1.1 设置书签

通常 `hyperref` 会自动将 `\section` 和类似的宏加入到书签中，但是它们也是可以人工设置的。

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

在给定的层次上（默认是 0 级）将指定的文本加入到书签中。其名称与层次一起作为内部锚点的名称。因此名称类似于 `\label` 必须是唯一的。

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

在当前层创建一个书签。

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

在书签层中创建一个次级书签。这时当前层就增加了一级。

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

在当前书签层中创建一个书签，但不改变当前层的级别。

小贴士：宏包 `bookmark` 用新的算法替代了 `hyperref` 宏包的书签组织方式，有如下好处：

- 通常只需运行一次 L^AT_EX。
- 提供更多书签显示形式（如，颜色、字体）。
- 支持不同的是书签链接（如，额外的文档链接、urls、...）。

因此我推荐使用这个宏包。

4.1.2 替代宏

`hyperref` 宏包从参数命令像 `\section` 之类来获取书签的文本，这些命令中可以含有数学、颜色、字体改变，然而这些都不能在书签中显示。

```
\texorpdfstring{TeXstring}{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}

```

在扩展字符串之前 `\pdfstringdef` 先执行。因而你可以使用它来实现额外的任务或关闭额外的命令。

```

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

```

然而使命令失效，有个简单方法是，经由 `\pdfstringdefDisableCommands` 给 `\pdfstringdefPreHook` 的定义增加参数（‘@’ 在这里被用作命令名称中的字母）。

```

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

```

4.2 实用宏

`\hypercalcbp{dimen specification}`

`\hypercalcbp` takes a TeX dimen specification and converts it to bp and returns the number without the unit. 这对选项 `pdfview`, `pdfstartview`, 和 `pdfremotestartview` 非常有用。例如：

```

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

```

The origin of the PDF coordinate system is the lower left corner.

Note, for calculations you need either package `|calc|` or ϵ -TeX. Nowadays the latter should automatically be enabled for L^AT_EX formats. Users without ϵ -TeX, please, look in the source documentation `hyperref.dtx` for further limitations.

`\hypercalcbp` 也不能用于 `\documentclass` 和 `\usepackage` 的选项中，因为 L^AT_EX 扩展了这些命令的选项列表。但是 `hyperref` 没有加载这些，可能会报错 “undefined control sequence error”。

5 Acrobat 的具体行为

假如你想设置 Acrobat Reader 或 Exchange 的菜单选项，在合适的驱动中，如下的宏可以提供：

<code>\Acrobatmenu{menuoption}{text}</code>

*text*的内容是用来创建一个按钮来激活合适的 *menuoption*。你可以用下表中的选项，比较这些在 Acrobat Reader 和 Exchange 菜单，你将知道它们是干什么的。很明显一些只适用于 Exchange。

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
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

6 PDF 和 HTML 的交互环境

你必须将你的文件放在一个 **Form**环境中（一个环境放一个文件）。这里有六个宏来准备文件：

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

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

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

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

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

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

形式和标签样式由如下决定:

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

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

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

这些宏默认是 #1 #2

实际在文件中显示的由如下决定:

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

```
\MakeCheckField{width}{height}
```

```
\MakeTextField{width}{height}
```

```
\MakeChoiceField{width}{height}
```

```
\MakeButtonField{text}
```

这些宏默认成 `\vbox to #2{\hbox to #1{\hfill}\vfill}`, 除了最后一个, 默认成 #1; 它是用于按钮, 和特殊宏 `\Submit` 和 `\Reset`。

你或许也想重新定义如下宏集:

```
\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}
```

6.1 交互环境的参数

action	URL	假如 Submit按钮被包含在 form 中，URL 将接受 form 的数据
encoding	name	设置 URL 字符串的 PDF 编码，PDF-encoding 是常见的，html是唯一有效值。
method	name	只是在生成 HTML 时使用，其值可以是 post 或者 get 。

6.2 交互环境选项参数

注意所有的颜色都是用 RGB 来表示，范围是 0 到 1。例如，color=0 0 0.5

accesskey	key		(as per HTML)
align	number	0	文字对齐，0 是左对齐，1 是居中，2 是右对齐。
altname	name		替代名称，该名称在用户界面中显示
backgroundcolor			边框颜色
bordercolor			边界颜色
bordersep			边框间距
borderwidth			边框宽度
calculate			JavaScript 编码来计算文档的价值
charsize	dimen		文本字体大小
checkboxsymbol	char	4 (✓)	用于复选框的符号 (ZapfDingbats)，其值是字符值或 \ding{number}，参见宏包 ‘psnfss’ 捆绑的宏包 ‘pifont’。
checked	boolean	false	是否选项默认选中
color			框中文本颜色
combo	boolean	false	选择菜单是 ‘combo’ 风格
default			默认值
disabled	boolean	false	文件失效
format			JavaScript code to format the field
height	dimen		文本框高度
hidden	boolean	false	隐藏文本
ketstroke			JavaScript code to control the keystrokes on entry
mappingname	name		the mapping name to be used when exporting the field data
maxlen	number	0	number of characters allowed in text field

menulength	number	4	number of elements shown in list
multiline	boolean	false	whether text box is multiline
name	name		name of field (defaults to label)
onblur			JavaScript code
onchange			JavaScript code
onclick			JavaScript code
ondblclick			JavaScript code
onfocus			JavaScript code
onkeydown			JavaScript code
onkeypress			JavaScript code
onkeyup			JavaScript code
onmousedown			JavaScript code
onmousemove			JavaScript code
onmouseout			JavaScript code
onmouseover			JavaScript code
onmouseup			JavaScript code
onselect			JavaScript code
password	boolean	false	text field is 'password' style
popdown	boolean	false	choice list is 'popdown' style
radio	boolean	false	choice list is 'radio' style
radiosymbol	char	H (★)	symbol used for radio fields (ZapfDingbats), the value is a character or <code>\ding{number}</code> , see package 'pifont' from bundle 'psnfss'
readonly	boolean	false	field is readonly
rotation	number	0	rotation of the widget annotation (degree, counterclockwise, multiple of 90)
tabkey			(as per HTML)
validate			JavaScript code to validate the entry
value			initial value
width	dimen		width of field box

7 定义一个新的驱动

一个 `hyperref` 宏包驱动必须提供如下八个宏集的定义：

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

如下定义宏的初步选项：

```
\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`

8 对其他宏包的特别支持

`hyperref`宏包的目的就是一起与其他宏包发挥作用，但是还是与一些宏包存在代码上的冲突。例如：

- 那些对参考文献操作的宏包。虽然能支持 PeterWilliam 的 `harvard` 的宏包。然而受欢迎的 Patrick Daly 的 `natbib`宏包必须与 `hyperref`宏包有特定的挂钩才允许可靠的交互。因为该宏包覆盖了非常广泛且多样的设计布局和引用风格，而这些都与 `hyperref`宏包有关。
- 那些具有排版目录标签和引用的宏命令的宏包，`showkeys`。如在 `hyperref`宏包重新定义这些命令之后，为了使这些宏包使用正常必须设置 `implicit=false`。
- 对索引有任何重大影响的宏包。

`hyperref`是和两个非常有用的变种宏包一起发布的，它们被设计成与 `hyperref` 宏包搭配非常好。它们是 `xr`和 `minitoc`宏包，都支持文档中的交叉引用，分别用的是 \LaTeX 下的 `\label/\ref` 机制和目录。

9 历史与感谢

`hyperref`宏包继承自 `hyperbasics.tex`和 `hypertex.sty`，它们的最初作者是 TanmoyBhattacharya (tanmoy@qcd.lanl.gov) 和 Thorsten Ohl (thorsten.ohl@physik.th-darmstadt.de)。`hyperref`宏包最初是作为他们的宏包与 \LaTeX 2_ε 标准联系的接口，但是最终我几乎重写了所有东西，因为我没有弄清楚很多源代码，只是一心想让它们能在 \LaTeX 下使用。非常感谢 Arthur Smith, Tanmoy Bhattacharya, Mark Doyle, Paul Ginsparg, David Carlisle, T. V. Raman 和 Leslie Lamport 是他们的评论、要求、想法和代码使得 `hyperref`宏包成为最有用的宏包之一。在这里还有很多人要感谢，是他们发现了问题，才使得我在最新版本中修改源代码。

Tanmoy 找出了许多 bug，并经常提供修改建议（有时非常好）。这使得 `hyperref`宏包更加健全。那些花在 `RevTeX` 上的时间完全是因为他！Bill Moss(bmoss@math.clemson.edu) 对新版本包括对本地 PDF 的支持的检测发现了很多 bug，并且他的检测是非常细致的。Hans Hagen (pragma@pi.net) 为 PDF 给出了很多深刻的见解。

对于 `dvipsone`和 `dviwindo`编译器，Berthold Horn 提供了帮助、鼓励和赞助。Sergey Lesenko 提供了针对 `dvipdf`的改动，还有 Hàn Thê Thành 提供了所以针对 `pdftex` 需要的信息。Patrick Daly 很乐意地更新了他的 `natbib`宏包使得 `hyperref`与它很容易一起用。Michael Mehlich 的 `hyper`宏包（和 `hyperref`宏包平行发展的）在一些问题上为我提供了解决方案。真希望某一天这两个宏包能合并在一起。

对于创建章节部分，要感谢 T. V. Raman 的鼓励、支持和想法。*Web Publishing with Acrobat/PDF*的作者 Thomas Merz 提供了关键的想法。D. P. Story 的关于 `pdfmarks` 文章解决了很多特殊的问题。Hans Hagen 解释了该宏包在 `pdftex`是如何运行的。

自从本手册弄丢后，Steve Peter 于 2003 年七月重写了手册的源文件。

特别感谢 David Carlisle 的 `backref`模型，还有对 `ps2pdf` 和 `dviwindo` 的支持，感谢他一次次重写我糟糕的代码，感谢他对 `xr`宏包的修改使适应 `hyperref`宏包的工作。