

pdfcomment.sty v1.4a*

A user-friendly interface to PDF annotations†

Josef Kleber‡

July 14, 2009

Abstract

For a long time pdf_latex has offered the command `\pdfannot` for inserting arbitrary PDF annotations. However, the command is presented in a form where additional knowledge of the definition of the PDF format is indispensable. This package is an answer to the – occasional – questions in newsgroups, about how one could use the comment function of Adobe Reader. At least for the writer of L^AT_EX code, the package offers a convenient and user-friendly means of using `\pdfannot` to provide comments in PDF files. Since version v1.1, pdfcomment.sty also supports:

L^AT_EX → dvips → ps2pdf, L^AT_EX → dvipdfmx¹ and XeL^AT_EX.

Unfortunately, support of PDF annotations by PDF viewers is sparse to nonexistent. The reference viewer for the development of this package is Adobe Reader.

If you can't see this annotation  you are definitely using the wrong PDF viewer!

Required packages for using pdfcomment.sty

hyperref (v6.76a [2007/04/09]), zref (v1.8 [2007/04/22]), xkeyval, ifpdf, ifthen, calc, marginnote and the packages loaded by them.

*pdfcomment.sty@pdfcomment.josef-kleber.de

†<http://pdfcomment.josef-kleber.de>

‡josef.kleber@gmx.de

¹only with style option dvipdfmx

Contents

1 Options	4		
1.1 Global options	4		
1.1.1 final	4		
1.1.2 draft	4		
1.1.3 dvipdfmx	4		
1.2 Local options	4		
1.2.1 subject	4		
1.2.2 author	4		
1.2.3 opacity	4		
1.2.4 color	4		
1.2.5 icolor	5		
1.2.6 icon	5		
1.2.7 open	5		
1.2.8 hspace	5		
1.2.9 type	6		
1.2.10 font	6		
1.2.11 fontsize	6		
1.2.12 fontcolor	6		
1.2.13 line	6		
1.2.14 linewidth	7		
1.2.15 linebegin	7		
1.2.16 lineend	7		
1.2.17 linesep	7		
1.2.18 borderstyle	7		
1.2.19 dashstyle	7		
1.2.20 bse	7		
1.2.21 bsei	7		
1.2.22 caption	7		
1.2.23 captionhoffset	7		
		1.2.24 captionvoffset	8
		1.2.25 voffset	8
		1.2.26 hoffset	8
		1.2.27 width	8
		1.2.28 height	8
		1.2.29 markup	8
		1.2.30 avatar	8
		1.2.31 style	8
2 Environments	8		
2.1 Comment environments	8		
2.1.1 \pdfsidelinecomment	8		
3 Commands	9		
3.1 Comment commands	9		
3.1.1 \pdfcomment	9		
3.1.2 \pdfmargincomment	9		
3.1.3 \pdfmarkupcomment	9		
3.1.4 \pdffreetextcomment	10		
3.1.5 \pdfsquarecomment	10		
3.1.6 \pdfcirclecomment	10		
3.1.7 \pdflinecomment	11		
3.2 Misc. commands	11		
3.2.1 \pdfcommentsetup	11		
3.2.2 \defineavatar	11		
3.2.3 \definestyle	11		
4 Printing comments and pop-ups	11		
References	13		

Acknowledgment

I want to thank the following persons for contributions to the development of this package:

- **Alexander Grahn** for contributing a patch for other drivers

$\text{\LaTeX} \rightarrow \text{dvips} \rightarrow \text{ps2pdf}$, $\text{\LaTeX} \rightarrow \text{dvi\text{p}dfmx}$ and Xe\LaTeX .

- **Gabriel Cardona** for pointing out problems with older versions of `hyperref` (v6.76a [2007/04/09])
- **Ulrike Fischer** for answering my stupid questions on d.c.t.t. for making possible the avatar and style system and much more.
- **Christian Feuersänger** for contributing new ideas in form of his package `pdfmarginpar.sty` and for solving the printing problem² of PDF annotations and popups.
- **Heiko Oberdiek** also for answering my stupid questions on d.c.t.t., as well as for the development of dozens of very usefull packages, especially `hyperref.sty`, `hycolor.sty` and `zref.sty`, which made this package possible.
- **Herbert Voß** also for answering my stupid questions on d.c.t.t.

²see section 4

1 Options

1.1 Global options

1.1.1 `final`

The option `final` will set the package to final mode. The PDF annotations will not be typeset and will not influence line breaking.

1.1.2 `draft`

The option `draft` (default) will set the package to draft mode. Therefore, the PDF annotations will be typeset.

1.1.3 `dvipdfmx`

If you want to use the driver `dvipdfmx` for creating your documents, you have to use the option `dvipdfmx`. The other drivers are recognized automatically.

1.2 Local options

The following options are useable as options for the commands presented in sections 2.1 and 3.1, as well as style options. As style options they have global effect, whereas they have only local effect when used in commands. Furthermore global options are not overwritten.

1.2.1 `subject`

You can use the option `subject` for defining the subject of the PDF popup annotations.

1.2.2 `author`

You can use the option `author` for defining the author of the PDF popup annotations.

1.2.3 `opacity`

You can use the option `opacity` for defining the opacity of PDF annotations with values between 0 (transparent) and 1 (not transparent, default). If you want to print PDF popup annotations with transparency you will have to use the option `open = true`.

1.2.4 `color`

You can use the option `color` for defining the color of PDF annotations in the form `{0.34 0.56 0.12}` (RGB). If you are using the additional package `xcolor` you can use predefined color names, as well as the available optional color names. Furthermore you can use the command `\definecolor` to define your



own named colors. Please take a look at the attached example `example.tex`. It shows the different possibilities of defining colors.

By definition the PDF specification allows four different color spaces with different numbers of color values:

#	color space
0	transparent
1	grey scale
3	RGB
4	CMYK

1.2.5 icolor

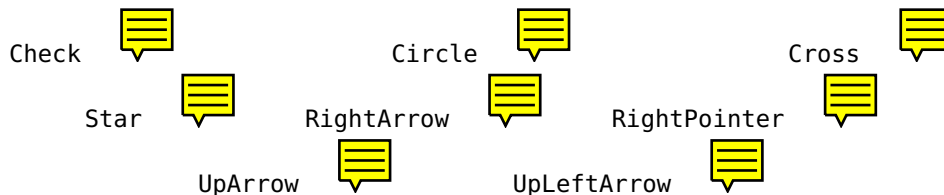
You can use the option `icolor` for defining the so called 'inner color', which is used by some PDF annotations, like arrowheads or the inner area of PDF circle annotation. For the rest, the provisions of the option `color` shall apply.

1.2.6 icon

You can use the option `icon` for defining the graphic used for the PDF text annotations. The following icons are defined as mandatory by the PDF reference:



Furthermore the following icons are also supported by Adobe Reader and some other PDF viewers:



1.2.7 open

You can use the option `open` for defining the opening status of the PDF popup annotations. Possible values are `true` or `false` (default).

If you want to print the PDF popup annotations (with transparency) you will have to use the option `open = true`.

1.2.8 hspace


You can use the option `hspace` for defining the horizontal space after the PDF text annotations, otherwise the PDF text annotations will overlay the text.

1.2.9 type

You can use the option `type` for defining the type of PDF annotation, if a comment command supports more than one PDF annotations:

comment command	possible types
<code>\pdfreetextcomment</code>	freetext, callout, typewriter
<code>\pdflinecomment</code>	line, polyline, polygon

1.2.10 font

You can use the option `font` for defining the font of PDF freetext annotations, like all fonts installed in the system, which don't contain a space in their name. Although Adobe Reader shows these fonts as embedded, they are not embedded in reality, but the PDF file just contains a reference to the font (default: Helvetica)! Therefore you should use this option quite carefully. In the case of document exchange between several authors you should just use fonts, which are available on all computer systems, like the so called 'standard 14 fonts'. 

1.2.11 fontsize

You can use the option `fontsize` for defining the fontsize of PDF freetext annotations.

1.2.12 fontcolor

You can use the option `fontcolor` for defining the font color in PDF freetext annotations, which must be a RGB color.

1.2.13 line

You can use the option `line` for defining the coordinates of lines in certain PDF annotations, e. g. in the form $\{x_1 \ y_1 \ x_2 \ y_2\}$. The origin is in the bottom left page corner. The given numbers will be interpreted as Postscript points (L^AT_EX unit: bp (big points)), as usual in PDF documents.

type of line	# points
line	$n = 2$
polyline, polygon	$n > 2$
callout line	$n = 3$


In general, the points must be given from line begin to line end. For callout lines the points must be given from line end to line begin. Please note the example file `example.tex`!

To avoid the method 'trail and error' you can load the PDF file with the Ghostscript viewer, which is capable of showing the cursor position in it's status line.

1.2.14 linewidth

You can use the option `linewidth` for defining the line width of PDF annotations.

1.2.15 linebegin

You can use the option `linebegin` for defining the arrow type at the line begin. The example file `example.pdf` shows all possible types of arrows. 

1.2.16 lineend

You can use the option `lineend` for defining the arrow type at the line end.

1.2.17 linesep

You can use the option `linesep` for defining the horizontal space between text and line in the command `\pdfsidelinecomment`.

1.2.18 borderstyle

You can use the option `borderstyle` for defining the line style. Possible values are: `solid` (default) and `dashed`

1.2.19 dashstyle

You can use the option `dashstyle` for defining the dash style, e.g. `{5 3}` (5 points line, 3 points space).

1.2.20 bse

You can use the option `bse` for defining the 'border style effect'. Possible values are: `none` (default) and `cloudy`

1.2.21 bsei

You can use the option `bsei` for defining the 'border style effect intensity' (size of cloud elements). The PDF reference suggests values between 1 and 2.

1.2.22 caption

You can use the option `caption` for defining the caption type of lines. Possible values are: `none`, `inline` (default), `top`

The options `inline` and `top` shows the comment 'inline' and on top of the line. While using `none` the comment will be shown in a PDF popup annotation.

1.2.23 captionoffset

You can use the option `captionoffset` for defining a horizontal offset, that is a horizontal shift of the line caption.

1.2.24 captionvoffset

You can use the option `captionvoffset` for defining a vertical offset of the line caption.

1.2.25 voffset

You can use the option `voffset` for defining a vertical offset of the PDF annotations, that is a vertical shift for the given length.

1.2.26 hoffset

You can use the option `hoffset` for defining a horizontal offset of the PDF annotations.

1.2.27 width

You can use the option `width` for defining the width of PDF annotations, e. g. the width of FreeText annotations. PDF text annotations have a width of 0pt by definition.

1.2.28 height

You can use the option `height` for defining the height of PDF annotations. PDF text annotations have a height of `\baselineskip` by definition.

1.2.29 markup

You can use the option `markup` for defining the type of the PDF text markup annotation. Possible values are: Highlight, Underline, Squiggly und StrikeOut.

1.2.30 avatar

With the option `avatar` you can load the option lists, that were predefined with the command `\defineavatar` to avoid annoying typing.

1.2.31 style

With the option `style` you can also load predefined option lists for splitting personal and stylistic options, e. g. `avatar=Josef`, `style=MyComment`.

2 Environments

2.1 Comment environments

2.1.1 \pdfsidelinecomment

<code>\begin{pdfsidelinecomment}</code>	Possible options: <code>avatar</code> , <code>style</code> , <code>subject</code> , <code>author</code> , <code>color</code> , <code>icolor</code> , <code>opacity</code> ,
<code> [<options>]{<comment>}</code>	<code>linewidth</code> , <code>linebegin</code> , <code>lineend</code> , <code>linesep</code> , <code>borderstyle</code> , <code>dashstyle</code> , <code>caption</code> ,
<code> ...</code>	<code>captionhoffset</code> , <code>captionvoffset</code>
<code>\end{pdfsidelinecomment}</code>	

With the environment `pdfsidelinecomment` you can comment complete parts of a page in the form of two lines in the margins.

Limitations:

- The PDF sideline annotation must not be longer than one page, otherwise the recognition of the page break will fail.
- While using dvi files in the meantime you have to use a L^AT_EX distribution, which is using `pdf(e)latex` as engine in a version \geq v1.40.0³!
- While using XeL^AT_EX you have to make sure that page dimensions are written to the xdv file, e. g. with the option `pagesize` of the KoMa-Script classes, or as option of the package `typearea.sty` respectively.

3 Commands


3.1 Comment commands

You can use the following commands for commenting your documents.

3.1.1 `\pdfcomment`

`\pdfcomment`
[*<options>*]{*<comment>*}

Possible options: `avatar`, `style`, `subject`, `author`, `icon`, `color`, `opacity`, `open`, `hspace`, `voffset`, `hoffset`

`\pdfcomment` will typeset an annotation into the text at the current  position.

3.1.2 `\pdfmargincomment`

`\pdfmargincomment`
[*<options>*]{*<comment>*}

Possible options: `avatar`, `style`, `subject`, `author`, `icon`, `color`, `opacity`, `open`, `hspace`, `voffset`, `hoffset`



`\pdfmargincomment` will typeset an annotation into the margin. Please note the positioning of the annotation in this example. This documentation was written with the L^AT_EX class `ltxdoc`. Therefore the annotation is typeset into the left margin.

3.1.3 `\pdfmarkupcomment`

`\pdfmarkupcomment`[*<options>*]
{*<markup text>*}{*<comment>*}

Possible options: `avatar`, `style`, `subject`, `author`, `color`, `opacity`, `markup`

`\pdfmarkupcomment` will typeset so called PDF text markup annotations over the text specified in the argument {*<markup text>*}. Possible types for the option `markup` are: Highlight, Underline, Squiggly and ~~StrikeOut~~

³older versions are not capable of providing x/y coordinates of the current position in dvi mode.

Limitations:

- As the name PDF text markup annotation might lead to suggest, this form of comment is suitable for text only.
- The specified text with the argument $\{\langle markup\ text\rangle\}$ must not be longer than one paragraph.
- The PDF text markup annotation must not be longer than one page, otherwise the recognition of the page break will fail.
- While using dvi files in the meantime you have to use a L^AT_EX distribution, which is using pdf(e)latex as engine in a version $\geq v1.40.0^4$!
- While using XeL^AT_EX you have to make sure that page dimensions are written to the xdvi file, e. g. with the option pagesize of the KoMa-Script classes, or as option of the package typearea.sty respectively.

3.1.4 \pdfreetextcomment

`\pdfreetextcomment`
`[\langle options\rangle]{\langle comment\rangle}`

Possible options: avatar, style, subject, author, color, font, fontsize, fontcolor, opacity, line, linewidth, lineend, borderstyle, dashstyle, bse, bsei, type, height, width, voffset, hoffset

`\pdfreetextcomment` (type=freetext) will typeset a comment in form of a freely positioned box on the wished spot of the page. By using the option type = callout the box will have an additional so called callout line, to bind the box to a certain element of the page. With the option type = typewriter you can write a comment everywhere on the page with a virtual ‘typewriter’. These PDF freetext annotation do have no frame and a transparent background for ‘overwriting’ arbitrary parts of the page. Please note the example file example.pdf!



This is a FreeText an

3.1.5 \pdfsquarecomment

`\pdfsquarecomment`
`[\langle options\rangle]{\langle comment\rangle}`

Possible options: avatar, style, subject, author, color, icolor, opacity, linewidth, borderstyle, dashstyle, bse, bsei, height, width, voffset, hoffset

`\pdfsquarecomment` will typeset a rectangular box at the wished spot of a page (see: example.pdf).

3.1.6 \pdfcirclecomment

`\pdfcirclecomment`
`[\langle options\rangle]{\langle comment\rangle}`

Possible options: avatar, style, subject, author, color, icolor, opacity, linewidth, borderstyle, dashstyle, bse, bsei, height, width, voffset, hoffset

`\pdfsquarecomment` will typeset a comment in form of a circle or ellipse at the wished spot of the page (see: example.pdf).

⁴older versions are not capable of providing x/y coordinates of the current position in dvi mode.

3.1.7 `\pdflinecomment`

`\pdfsidelinecomment`
`[\langle options \rangle]{\langle comment \rangle}` Possible options: `avatar`, `style`, `subject`, `author`, `type`, `color`, `icolor`, `opacity`, `caption`, `captionhoffset`, `captionvoffset`, `linewidth`, `linebegin`, `lineend`, `line`, `borderstyle`, `dashstyle`

`\pdflinecomment` will typeset a comment in form of a line, polyline or polygon at the wished spot of the page (see: `example.pdf`).

3.2 Misc. commands

3.2.1 `\pdfcommentsetup`

`\pdfcommentsetup{\langle options \rangle}` With the command `\pdfcommentsetup` you can reset the global options at any time.

3.2.2 `\defineavatar`

`\defineavatar{\langle name \rangle}{\langle options \rangle}` With the command `\defineavatar` you can create named predefined option lists, which can be later used in the comment commands with the option `avatar`. With this option, it's easy for several authors of the same document to switch between different avatars, that is their graphical representation.

3.2.3 `\definestyle`

`\definestyle{\langle name \rangle}{\langle options \rangle}` With the command `\definestyle` you can split up lists in personal and stylistic option lists (see `example.tex`). This option list can be loaded with the option `style`.

4 Printing comments and popups

Basically, Adobe Reader is able to print PDF annotations. Therefore, you have to choose 'Document and Markups' in the field 'Comments and Forms' of the print dialog. Additionally, you have to configure the menu 'Edit → Preferences → Commenting'. Unfortunately, this option does not exist in many versions of Adobe Reader. If it does not exist, the only possibility is to edit the configuration files itself.

Windows: the registry needs to be changed (on your own risk!)

Start the registry editor (`regedit.exe`) and open the following key, e. g. for Adobe Reader 9.0:

HKEY_CURRENT_USER/Software/Adobe/Acrobat Reader/9.0/Annots/
cPrefs

The key for other version will be similar, except version number and some letters in lower- or uppercase!

Open or create (with closed Adobe Reader) the key
`bprintCommentPopups` and change the value to 1 (dword).

Linux: open the configuration file:

`~/ .adobe/Acrobat/8.0/Preferences/reader_prefs`

with a text editor and change the entry

`/printCommentPopups [/b false] to /printCommentPopups [/b true]`

Please note that only opened popups will be printed!

References

- [Ado05] Adobe Systems Incorporated: *pdfmark Reference Manual*. 2005. – URL: http://www.adobe.com/devnet/acrobat/pdfs/pdfmark_reference.pdf
- [Ado06] Adobe Systems Incorporated: *PDF Reference sixth edition*. 2006. – URL: http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf
- [Adr08] Adriaens, Hendri: *The xkeyval package*. 2008. – TEXMF://doc/latex/xkeyval/xkeyval.pdf
- [Car99] Carlisle, David: *The file ltxdoc.dtx for use with L^AT_EX*. 1999. – TEXMF://doc/latex/base/ltxdoc.dvi
- [Gün02] Günther, Karsten: *L^AT_EX Ge-packt*. 1. Auflage. mitp, 2002. – ISBN 3-8266-0785-6
- [Ker07] Kern, Uwe: *Extending L^AT_EX 's color facilities: the xcolor package*. 2007. – TEXMF://doc/latex/xcolor/xcolor.pdf
- [Mer02] Merz, Thomas: *Die PostScript & PDF Bibel*. PDFLib edition. dpunkt-Verlag, 2002. – URL: http://www.pdflib.com/fileadmin/pdflib/pdf/Bibel/bibel_d_pdfmark_lx1.pdf
- [MG05] Mittelbach, Frank ; Goossens, Michel: *Der L^AT_EX Begleiter*. 2., überarbeitete und erweiterte Auflage. Pearson Studium, 2005. – ISBN 3-8273-7166-X
- [Oga01] Ogawa, Arthur: *Extensions to the ltxdoc class*. 2001. – TEXMF://doc/latex/revtex4/ltxdocext.pdf