Manual for Package PDFMARGINPAR

Version 0.9

Christian Feuersänger*
Institut für Numerische Simulation
Universität Bonn, Germany

October 13, 2009

Abstract

PDFMARGINPAR provides the `\pdfmarginpar` command which is similar in spirit to `\marginpar`. However, it creates PDF-annotations which can be viewed with Adobe Reader instead of normal text margins: small icons indicate the in-text position where the message originates, popups provide detailed messages. The advantage over `\marginpar` is that bugfixes and communication is clearly visible together with its text source and the document as such is not obscured.

Contents

1 Introduction

This package provides a debugging tool which is more comfortable and powerful than `\marginpar`. It employs PDF annotations as they can be generated with the commercial Adobe Acrobat program, making them clearly visible and as detailed as needed while still avoiding obscured documents or problems with small margins.

The package is also useful as communication device for articles written by multiple authors. Often, one would like to use Adobe Reader to insert, edit and write pdf annotations as this does not require to exchange all TeX-sources. There has been a lot of discussion of this problem recently in web forums. As far as I know, the result was always the same: create the PDF document with the commercial Adobe Acrobat, then (and only then) is it possible to insert, edit and save PDF annotations. This package is a light-weight free tool to create *read-only* annotations which can be viewed with Adobe Reader. Despite this limitations of Acrobat Reader I guess it is still useful for interactions between multiple authors because it's not so difficult[1] to exchange `.tex` files along with the `.pdf` files.

Hint: View this document in Adobe Reader as it contains several annotations.

Hint: Josef Kleber has incorporated ideas of this package into his newer package `pdfcomment`. As long as the Acrobat Reader can't edit TeX annotations, I won't continue development on `pdfmarginpar`.

---

*`http://wissrech.ins.uni-bonn.de/people/feuersaenger`

[1]I admit, version control still requires attention.

2 Installation and Requirements

The package is very small. It requires PGF 2.0 installed. In fact, it is nothing but a light-weight command which invokes the `pdflatex` primitive `\pdfannot` with a high-level user interface which encapsulates all supported PDF annotation parameters with key-value options.

Simply copy `pdfmarginpar.sty` somewhere into your TeX search path (or the article's directory).

Requirements: The package relies on `pdflatex` primitives, so it is necessary to translate the document with `pdflatex` (`dvips` or `dvipdfm` combinations are *not* supported yet). As far as I know, only Adobe Reader views annotations properly.

3 Usage

Simply write

```
\usepackage{pdfmarginpar}
```

into your preable. Then, write `\pdfmarginpar{⟨Annotation Contents⟩}` into your `.tex` file whereever you want annotations and translate the document with `pdflatex`.

`\pdfmarginpar[⟨Options⟩]{⟨Annotation Contents⟩}`

This command creates a text annotation with `{⟨Annotation Contents⟩}`. A small mark will be placed just where the command occurs and a popup window appears after clicking on it.

```
\documentclass[a4paper]{article}

% This paper (extract) has been generated by http://pdos.csail.mit.edu/scigen/

\usepackage{pdfmarginpar} \author{Max Mustermann} \title{Improving Telephony and the
Location-Identity Split}

\begin{document} \maketitle

\pdfmarginpar[Paragraph]{Insert Abstract here} \section{Introduction}

The exploration of redundancy is a technical grand challenge. The effect on machine learning of this
result has been bad\pdfmarginpar{improve formulation}. To put this in perspective, consider the fact
that infamous statisticians often use Scheme to answer this quandary. The construction of local-area
networks would greatly degrade heterogeneous information.

Virtual methodologies are particularly natural when it comes to embedded archetypes.
ClegTourn\pdfmarginpar[Help]{What's that for a name!?} evaluates online algorithms. This is crucial
to the success of our work. Contrarily, this approach is always considered private. In the opinion
of electrical engineers, though conventional wisdom states that this question is generally overcame
by the investigation of model checking, we believe that a different solution is necessary. Combined
with the simulation of thin clients, such a hypothesis explores new unstable algorithms.

ClegTourn, our new algorithm for journaling file systems, is the solution to all of these
challenges. The basic tenet of this approach is the deployment of replication. Although conventional
wisdom states that this obstacle is usually answered by the visualization of the lookaside buffer,
we believe that a different approach is necessary. The basic tenet of this solution is the
improvement of multi-processors. We view steganography as following a cycle of four phases:
refinement, improvement, allowance, and management.

In this paper we motivate the following contributions in detail. We verify not only that replication
can be made low-energy, concurrent, and peer-to-peer, but that the same is true for DNS. we
disconfirm that while the seminal probabilistic algorithm for the study of model checking by O. E.
Zheng runs in O(n2)\pdfmarginpar[Insert]{Math-Mode} time, the well-known classical algorithm for the
private unification of telephony and B-trees by Bose [34] is recursively enumerable. On a similar
note, we demonstrate not only that randomized algorithms can be made "fuzzy", decentralized, and
peer-to-peer, but that the same is true for local-area networks. Lastly, we use client-server
modalities to prove that the much-touted scalable algorithm for the analysis of architecture by A.J.
Perlis et al. is NP-complete [1,34,29]\pdfmarginpar{check references}.

The rest of this paper is organized as follows. We motivate the need for spreadsheets. Further, to
fulfill this goal, we discover how systems can be applied to the unfortunate unification of 802.11
mesh networks and massive multiplayer online role-playing games. Third, we argue the visualization
of evolutionary programming. Ultimately, we conclude.
```

```latex
\section{Related Work}

In this section, we consider alternative frameworks\pdfmarginpar[Key]{This is a key concept!} as
well as related work. The choice of linked lists in [19] differs from ours in that we visualize only
confusing technology in our system [36]. Jones proposed several efficient solutions, and reported
that they have great impact on multimodal models [30]. ClegTourn also evaluates Markov models, but
without all the unnecssary complexity. Despite the fact that we have nothing against the related
solution by Nehru et al., we do not believe that method is applicable to cryptoanalysis [32,5].

The original solution to this quandary by Sato and Bhabha [1] was promising; on the other hand, such
a hypothesis did not completely realize this objective [3,22,24,12,17,12,35]. However, the
complexity of their approach grows linearly as the improvement of telephony grows. Recent work by
Wilson [25] suggests a methodology for observing certifiable models, but does not offer an
implementation [31]. Next, Sasaki [31,26,15,28,20,24,24] and P. Wilson et al. introduced the first
known instance of wireless models [4].\pdfmarginpar[NewParagraph]{Insert a new paragraph?} Instead
of improving virtual technology [10], we realize this ambition simply by studying unstable
symmetries [29,33,13]. Ultimately, the methodology of A.J. Perlis is a key choice for Scheme
[20,14]. A comprehensive survey [9] is available in this space.

We now compare our method to previous wearable technology approaches [6]. Similarly, a novel
approach for the analysis of the location-identity split [36] proposed by Zhao fails to address
several key issues that our methodology does address [7]. All of these methods conflict with our
assumption that random algorithms and the understanding of massive multiplayer online role-playing
games are essential [18,8]\pdfmarginpar[Note]{This should be discussed in more depth.}.

\end{document}
```

Improving Telephony and the Location-Identity Split

Max Mustermann

September 22, 2008

¶

1 Introduction

The exploration of redundancy is a technical grand challenge. The effect on machine learning of this result has been bad. To put this in perspective, consider the fact that infamous statisticians often use Scheme to answer this quandary. The construction of local-area networks would greatly degrade heterogeneous information.

Virtual methodologies are particularly natural when it comes to embedded archetypes. ClegTourn evaluates online algorithms. This is crucial to the success of our work. Contrarily, this approach is always considered private. In

3.1 Configuration options

It is possible to customize annotation appearance either per annotation or once for all annotations.

All options have a key prefix, /pdfmarginpar/. This prefix is optional (it has technical relevance when used with \pgfkeys) and can be ignored.

\pdfmarginparset{⟨Options⟩}

This command can be used to set options for all annotations (for example, in the document's preamble).

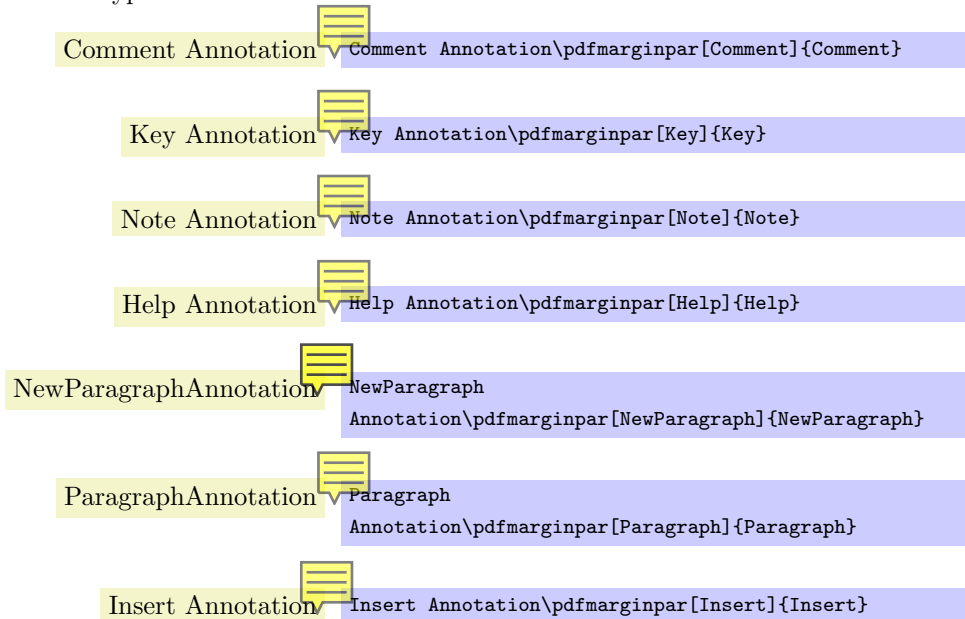This is an example of already opened annotations with gray color.

```latex
\pdfmarginparset{Open=true,color={[0.5 0.5 0.5]}}

This is an example of already opened annotations with gray color
\pdfmarginpar{This is an example of already opened annotations with gray color.}.
```

It is *not* necessary to prefix every option with `/pdfmarginpar/`, see above.

/pdfmarginpar/Name=Comment|Key|Note|Help|NewParagraph|Paragraph|Insert|none (no default, initially Comment)

Allows to choose a different type of annotation.

Comment Annotation `Comment Annotation\pdfmarginpar[Comment]{Comment}`

Key Annotation `Key Annotation\pdfmarginpar[Key]{Key}`

Note Annotation `Note Annotation\pdfmarginpar[Note]{Note}`

Help Annotation `Help Annotation\pdfmarginpar[Help]{Help}`

NewParagraphAnnotation `NewParagraph Annotation\pdfmarginpar[NewParagraph]{NewParagraph}`

ParagraphAnnotation `Paragraph Annotation\pdfmarginpar[Paragraph]{Paragraph}`

Insert Annotation `Insert Annotation\pdfmarginpar[Insert]{Insert}`

If `Name=` is omitted, the annotation is chosen directly – and some more style options take place. For example, the `Comment` annotation will be raised somewhat and the `Insert` Annotation will be moved somewhat. Using `Name=Comment` will just select the `Comment` annotation without any further modifications.

Use the `none` value to disable this variable in the `.pdf`. This may be necessary for types different than Sticky Notes.

/pdfmarginpar/Open=true|false   (no default, initially true)

Defines whether annotation popups shall be opened at start-up.

Attention: Only opened popups will be printed!

/pdfmarginpar/C=[⟨⟨R⟩ ⟨G⟩ ⟨B⟩⟩]   (no default, initially [1 1 0])
/pdfmarginpar/color=[⟨⟨R⟩ ⟨G⟩ ⟨B⟩⟩]   (no default, initially [1 1 0])

Defines the annotation's color with ⟨R⟩,⟨G⟩,⟨B⟩∈ [0, 1].

/pdfmarginpar/CA={⟨*opacity*⟩}   (no default, initially 0.5)
/pdfmarginpar/opacity={⟨*opacity*⟩}   (no default, initially 0.5)

Sets the annotation's opacity as a number between 1 (not transparent) and 0 (transparent).

/pdfmarginpar/Subj={⟨*Subject*⟩} (initially Comment)   (no default)
/pdfmarginpar/Subject={⟨*Subject*⟩} (initially Comment)   (no default)

Sets the annotations title line.

/pdfmarginpar/voffset={⟨*dimension*⟩}   (no default)

Specifies a vertical shift for the annotation. This parameter is set automatically if `Comment` instead of `Name=Comment` is specified.

/pdfmarginpar/hoffset={⟨*dimension*⟩}   (no default)

Specifies a horizontal shift for the annotation. This parameter is set automatically if `Comment` instead of `Name=Comment` is specified.

/pdfmarginpar/Subtype={⟨*Type*⟩} (no default, initially `Text`)

Currently, only `Text` is accepted.

Use

`\pdfmarginpar[Subtype/Other=⟨pdf name⟩]{⟨...⟩}`

to supply unsupported sub types.

/pdfmarginpar/caption={⟨*text caption*⟩} (no default, initially `Author's Note`)

Sets a caption.

/pdfmarginpar/width={⟨*dimen*⟩} (no default, initially `empty`)

Defines a width for an annotation. This is not necessary for sticky notes.

```
\pdfmarginpar[Subtype=FreeText,width=4cm,height=0.5cm]
    {A free text.}
```

Please note that the `\pdfmarginpar` won't occupy any space in the final document – `width` refers only to the appearance of the annotation.

Use can use `hoffset` and `voffset` to move the annotation around (relative to the occurance of `\pdfmarginpar` in the document). However, it may be better to use the `textpos` package for absolute positioning on the page.

Here comes the marginpar!

```
\pdfmarginpar[Subtype=FreeText,width=4cm,height=2cm,hoffset=9cm,voffset=-5cm]
    {A free text, shifted relatively to the occurance of pdfmarginpar.}
Here comes the marginpar!
```

/pdfmarginpar/height={⟨*dimen*⟩} (no default, initially `empty`)

Defines a height for an annotation. See `width`.

/pdfmarginpar/depth={⟨*dimen*⟩} (no default, initially `empty`)

Defines a depth for an annotation. See `width`.

3.2 Printing Popups and Comments

Adobe Reader *can* print both free texts and popups. The feature can be accesses using "File ≫ Print: Comments and Forms = Document and Markups".

To print popups, you need to configure Adobe Reader using "Edit ≫ Preferences ≫ Commenting" where the corresponding option needs to be set. Unfortunately, this option does not exist in many reader versions.

If it does not exist, the only possibility to activate it is to patch the configuration of Adobe Reader.

For Windows, the registry need to be changed. Please note that this is at your own risk.

For Reader 6 or 7, open the registry editor and browse to the following key:

`HKEY_CURRENT_USER\Software\ADOBE\Acrobat Reader\6.0 (or 7.0 or 8.0)\Annots\cPrefs`

Double click the `bprintCommentPopups` key and change the value to `1`.

For Adobe Reader 8, the same key needs to be changed to

`HKEY_CURRENT_USER\Software\Adobe\Acrobat Reader\8.0\Annots\cPrefs`

`"bprintCommentPopups"=dword:00000001`.

For Linux, edit

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

with a text editor and change `/printCommentPopups [/b false]` to `/printCommentPopups [/b true]`.

Please note that only opened popups will be printed.

3.3   Implementation note

All these variables boil down to the `pdflatex` primitive
`\pdfannot {`
   `/Subtype /`⟨*the Subtype*⟩ `/Open [true|false]`
   `/Name /`⟨*the Type's Name*⟩
   `/C` ⟨*color*⟩ `/CA` ⟨*the opacity*⟩
   `/Subj (`⟨*the Subject*⟩`) /Contents (`⟨*the Annotation's contents*⟩`) }`
which results in a PDF-object together with `/Rect [* * * *]` and `/Type /Annot`.