How to Create PDF from LaTeX

Ki-Joo Kim*

Version 0.32
April 15, 2004

Abstract

This document describes how to create high quality PDF (portable document format) from LaTeX sources and also provides valuable tips.

Contents

* mailto:kijoo2000 at yahoo dot com

1 Introduction

1.1 Creation Methods

There are three main methods to create PDF from LaTeX:

- **ps2pdf** converts PS (postscript) to PDF version 1.3 (Adobe 4.x compatible)[1] ⇒ Section 2.
- **dvipdfm** converts DVI to PDF version 1.2 (Adobe 3.x compatible) and can handle EPS files on-the-fly ⇒ Section 3.
- **pdflatex** directly produces PDF version 1.4 (Adobe 5.x compatible) from LaTeX ⇒ Section 4.

You can also add PDF information (e.g., title, subject, keyword, author, etc), WWW links, internal links, cross-references, bookmarks, and more using the famous `hyperref` package, which is described in Section 5.

1.2 Font Issues for Best PDF Output

LaTeX fonts are very important in high quality PDF documents. You may have seen ugly PDF documents on screen even though the printed documents are very sharp.[2] This is due to the default bitmap CM (computer modern) fonts used in LaTeX, and can be solved by using PostScript Type 1 fonts.

As most TeX implementations come with PostScript Type 1 CM fonts, use those fonts. If you are going to use `ps2pdf`, the PS file should be obtained via:

```
dvips -Ppdf -G0 foo.dvi
```

[1] Of course you can change Adobe Acrobat version by explicitly using `ps2pdf12.bat`, `ps2pdf13.bat`, or `ps2pdf14.bat`.
[2] It is known that this display problem is corrected in Acrobat Reader version 6.

Instead of `-Ppdf`, you can use `-Poutline` option. Sometimes `-j0` option saves you by turning off partial font downloading.

You can use other PostScript Type 1 fonts. The PSNFSS collection, a collection of famous Adobe PostScript Type 1 fonts, is already installed in your system. Some of them are:

```
\usepackage{mathptmx,helvet,courier} for Times text and math
          + Helvetica + Courier + Times Math
\usepackage{mathpazo,helvet,courier} for Palatino text and math
          + Helvetica + Courier + Palatino Math
\usepackage{newcent} for New Century Schoolbook + Avant Garde + Courier
\usepackage{bookman} for Bookman + Avant Garde + Courier
```

`mathptmx` and `mathpazo` are replacements of `times` and `palatino`, respectively. An example to use these fonts is shown here:

```
\usepackage{mathpazo,courier}
\usepackage[scaled=0.92]{helvet} % Make thin width
\usepackage[T1]{fontenc}
```

Alternatively, you can use `txfonts` for Times text and math or `pxfonts` for Palatino text and math. However, equations in both fonts may be too tight! An example to load these fonts is shown below:

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

In addition, you are strongly advised to embed all the fonts used in your PDF documents for trouble-free reading and printing on different machines. For more info, see Section 8.2.

2 ps2pdf: From PS to PDF

2.1 How to Use

Any PS and PRN (file format from `print to file`) can be converted to PDF, using either `ps2pdf` in Ghostscript or `Adobe distiller`. This method is very useful when you only have the output PS of PRN file or you use PostScript-specific packages such as `PSTricks` in your source file.

The general structure of source code can be:

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

Even though many people did not try to use `hyperref` in this method, you can also use `hyperref` package. The one problem with `hyperref` is that `dvips` *cannot break hyper links.* Since this package has lots of powerful features and works with all three PDF creation methods, a separate section is devoted to this package (see Section 5).

As described earlier, the results can be horrible if certain simple precautions are omitted when you use CM fonts. For the best result from `dvips` use Type1 fonts in LaTeX and tpye:

```
dvips -Ppdf (or -Poutline) -G0 input.dvi
```

2.2 pd2pdf

Then you can convert the PS to PDF:

```
pd2pdf input.ps output.pdf
```

In order to get prepress quality use as shown below:

```
pd2pdf -dPDFSETTINGS=/prepress input.ps output.pdf
```

In Win32, use '#' instead of '=' in the above command. For more options for high graphics conversion, you may add the following options.

```
-dAutoFilterColorImages=false
-dAutoFilterGrayImages=false
-dEncodeColorImages=false
-dEncodeGrayImages=false
-dEncodeMonoImages=false
```

2.3 Adobe Distiller

If you have Adobe distiller, you are lucky since it is another powerful tool. If you use Adobe distiller for this task, it needs some optional settings for high quality PDF conversion. First select Press Optimize job option. Go to Settings - Job Options - Fonts. Make sure to select Embed all fonts and Subset fonts below 100%. This will ensure that only fonts for which 100% of all characters are used in the document are included completely and the distiller will remove font data for all unused characters from your PDF file. This will keep your PDF files small.

3 dvipdfm: From DVI to PDF

dvipdfm, developed by Mark A. Wicks, can convert DVI to PDF. The user guide is in your /texmf/doc/dvipdfm directory. To use dvipdfm begin your document with

```
\documentclass{article}
\usepackage[dvips]{graphicx,color}
\usepackage[dvipdfm,colorlinks]{hyperref}
```

and then type latex file.tex (may be several times) and then dvipdfm file.dvi. For hyperref please refer Section 5.

One of the advantages of dvipdfm is internal handling of EPS files, so you do not need to convert them to PDF figure files as in pdfTEX or pdfLATEX. However, if your EPS files contain PS special codes (e.g., your EPS is generated with PSTricks or PSfrag), they cannot be directly used with dvipdfm. Please refer Section 8.4.

An extended version of dvipdfm is **dvipdfmx**, which is developed by Jin-Hwan Cho and Shunsaku Hirata. The features of this program are:

- Support CJK bookamrks.
- Support CJK search.
- Support True Type and Open Type fonts.
- Support Omega, Lambda, HLATEX, ConTEXt.
- Support PDF encryption.

CJK bookmarks and CJK search can be possible only with TrueType fonts. To add CJK bookmarks, the preamble should be like this:

```
\usepackage[dvipdfm,CJKbookmarks]{hyperref}
\AtBeginDvi{\special{pdf:tounicode KSCms-UHC-UCS2}}
\hypersetup{bookmarksnumbered=true,pdftitle={..},pdfauthor={..}}
```

For more information please refer to http://project.ktug.or.kr/dvipdfmx/doc/tug2003.pdf.

4 pdfLaTeX: From LaTeX to PDF

4.1 How to Use

pdfTeX and pdfLaTeX, mainly developed by Han The Thanh, is a the most powerful tool (in LaTeX-related programs) to create PDF directly from LaTeX. To use `pdflatex`, begin your document with

```
\documentclass{article}
\usepackage[dvips]{graphicx,color} % or [pdftex]?
\usepackage[pdftex,colorlinks]{hyperref}
```

and then type `pdflatex file.tex` (may be several times) to produce `file.pdf`.

4.2 Color and Graphics

The standard color and graphicx packages also have pdfTeX options that are activated by adding `pdftex` to the document class options. pdfTeX natively support JPEG for continuous images such as photographs, PNG files for other bitmap images, and PDF for vector drawings. But it cannot directly handle EPS!

To deal with a document that has EPS figures[3]:

- Use **`epstopdf` package** to convert EPS figures to PDF figures on-the-fly. It needs to be loaded after `\usepackage[pdftex]{gr`
  However, if you have many EPS figures, compilation takes more time.
- Use the **`epstopdf` program** manually to convert the EPS figures to PDF figures. Ensure that you don't mention the filename's suffix in the `\includegraphics` commands. Add `pdftex` to the options in the document class line when you run `pdflatex`, remove it when you run latex.

  Some options for high quality conversion can be added:

  ```
  -dPDFSETTINGS=/prepress
  -dUseFlateCompression=true
  -dAutoFilterColorImages=false
  -dAutoFilterGrayImages=false
  -sColorImageFilter=FlateEncode
  -sGrayImageFilter=FlateEncode
  ```

Figure 1 is a PDF figure converted from EPS using the `epstopdf` program. There is no need to calculate bounding box information using `ebb.exe`.

Unlike LaTeX, pdfLaTeX supports JPEG, TIFF and PNG formats. Since pdfLaTeX does not know the bounding box of these figure formats, you should run `ebb.exe foo.jpg` to extract the bounding box information of the given file. The resulting file is `foo.bb` and should be in the same directory.

4.3 PDF Information

PDF information such as title, subject, keywords, author, date and more can be added in your PDF document. There are two main methods. The following first method is only valid in pdfLaTeX. Add somewhere near the beginning of a TeX or LaTeX document code like:

```
\special{! [ /Author ()
             /Title ()
             /Keywords ()
             /DOCINFO pdfmark}
```

[3] They do not have PS special codes. If they have, please see Section 8.4.

Figure 1: Figure insertion from EPS to PDF conversion using `epstopdf`.

to store the author, title, and keywords in the PDF file automatically (Note that the pdfmark command is understood only by Adobe distiller). If you are using pdfLaTeX, you can also include this information into the produced PDF file by adding:

```
\pdfinfo{                           % Info dictionary of PDF output;
                                    % all keys are optional.
   /Author (Han The Than)
   /CreationDate (D:20000012000000) % D:YYYYMMDDhhmmss (default: the actual date)
   /ModDate (D:20000012000000)      % ModDate is similar
   /Creator (TeX)                   % default: "TeX"
   /Producer (pdfTeX)               % default: "pdfTeX" + pdftex version
   /Title (example.pdf)             %
   /Subject (Example)               %
   /Keywords (PDF TeX)              %
}
```

A better approach under LaTeX is to use the `hyperref` package. See Section 5.

4.4 PDF Annotation

[The first three examples in this section is taken from the `samplepdf.pdf` file in your `\texmf\doc\pdftex\samplepdf`.]

This annotation (only on the screen) is created using the following generic annotation.

```
\pdfannot              % generic annotation
   width 10cm          % the dimension of the annotation can be controlled
   height 0cm          % via <rule spec>; if some of dimensions in
   depth  4cm          % <rule spec> is not given, the corresponding
                       % value of the parent box will be used.
{                      %
   /Subtype /Text      % text annotation
   % /Open true        % if given then the text annotation will be opened
   /Contents           % text co    nts
     (Hello, there! Can you see me? ^_^)
}%
```

To see the content click the right mouse button on this figure. You can also move this text icon using your mouse. If you want to define your own annotation command, see the following example:

```
\usepackage{hyperref}
\makeatletter
\newcommand{\notes}[1]{%
  \pdfstringdef\@tempa{#1}%
  \marginpar{%
  \leavevmode%
  \pdfannot width 12cm height \baselineskip depth 0pt{
        /Subtype /Text
        /Contents (\@tempa)
     }%
  }%
  \let\@tempa\relax%
}%
\makeatother
```

Then, `\notes{PDF notes are here}` is shown like ⇒

4.5 Attach Multimedia Files

You can also play sound and movie files in your PDF (You need Apple Quick Time Movie Player). See the following example:

```
  \leftline{%
  \pdfannot width 4in height 0in depth 3in {%
     /Subtype /Movie
     /Movie << /F (MovieFile.mov) >>}
     An example of movie annotation
  }%
  \leftline{%
  \pdfannot width 0in height 0in depth 0in {%
     /Subtype /Movie
     /Movie << /F (SoundFile.wav) >>}
     Sound can be also embedded using Movie annotation
  }%
```

To include AVI files you can use the following command in the preamble:

```
  \newcommand{\video}[5]{%
    \vspace{#3}\hspace{#2}{\pdfannot width #4 height #5 depth 0cm {%
    /Subtype /Movie /C [1 0 0] /Border [0 0 3] /Movie
    <> }}}
```

and `\video{g.avi}{4cm}{12cm}{12cm}{12cm}` where the first argument is the name of the file, and the 2nd and 3rd control the location of one corner, and 4th and 5th the width and height of the box. You can control the color of the frame and thickness by modifying `/Border`.

4.6 Attach Files

```
\usepackage[pdftex]{attachfile}
...
\attachfile[icon=Tag,mimetype=text/plain]{a.txt}
```

5 Hyperref Package

By using the `hyperref` package, developed by Sebastian Rahtz, with `ps2pdf`, `dvipdfm` or `pdflatex`, you automatically get PDF files with WWW links, bookmarks (if you have a table of contents) and cross-references and more.

5.1 How to Use

There is a golden rule that the `hyperref` package should be loaded as the last package, except when using `sidewaysfigures`.

```
\usepackage[dvips or dvipdfm or pdftex]{article}
\usepackage[dvips]{graphicx,color}
\usepackage[dvipdfm or pdftex,colorlinks]{hyperref}
\hypersetup{%
    pdftitle={How to Create PDF from LaTeX},
    pdfauthor={Ki-Joo Kim},
    pdfkeywords={pdf, latex, tex, ps2pdf, dvipdfm, pdflatex},
    bookmarksnumbered,
    pdfstartview={FitH},
    urlcolor=cyan,
}%
```

As shown above PDF information can added using `pdftitle={text}`,`pdfauthor={text}`, and `pdfkeywords={text}`.
Additional useful options are:

- `bookmarks=true`: Show bookmarks (default=`true`). For more sub-options please see Section 5.2.
- `breaklinks=true`: Allow link text to break across lines (default=`false`).
- `colorlinks=true`: Color the text of links (`true`) or put color frames over the links (`false`).
  - `linkcolor=red`: Color for normal internal links.
  - `anchorcolor=black`: Color for anchor text.
  - `citecolor=green`: Color for bibliographical citations in text.
  - `filecolor=magenta`: Color for URLs which open local files.
  - `menucolor=red`: Color for Acrobat menu items.
  - `pagecolor=red`: Color for links to other pages
  - `urlcolor=cyan`: Color for linked URLs.
- `pdfmenubar=true`: Show Acrobat Reader's menu (default=`true`).
- `pdftoolbar=true`: Show Acrobat Reader's toolbar (default=`true`).
- `pdfstartview={option}`: Set the startup page view. Possible options are:
  - `FitH`: Fit whole width of page
  - `FitV`: Fit whole height of page
  - `FitB`: Fit whole "Bounding Box" page
  - `FitBH`: Fit whole width of "Bounding Box" of page
  - `FitBV`: Fit whole height of "Bounding Box" of page
- `pdfpagescrop={53 486 389 754}`: Tell `hyperref` to leave cropping commands in the DVI and PDF. By cropping you can significantly increase screen display size of PDF.

5.2 Bookmark

5.2.1 Bookmark Options

The following options (with using `\hypersetup`) affect bookmarks:

- `bookmarks=true`: Make bookmarks (default=`true`). This option cannot be used after package has been loaded, thus use like this: `\usepackage[bookmarks=false]{hyperref}`.
- `bookmarksnumbered=true`: Put section numbers in bookmarks (default=`false`).
- `bookmarksopen=true`: Open up the bookmark trees (default=`false`).
- `bookmarksopenlevel=num`: Level to which bookmarks are open (default=`\maxdimen`).
- `bookmarkstype=toc`: Specify which ŚtocŠ file to mimic (default=`toc`).
- `pdfpagemode={None}`: Specify how document starts when opened (`{None}`). Possible options are:
  - `None`: Neither bookmarks nor thumbnails are visible.
  - `UseOutlines`: Bookmarks are visible.
  - `UseThumbs`: Thumbnails are visible.
  - `FullScreen`: Full-screen mode without bars, outlines, and thumbnails. Good for PDF presentations.
- `unicode=true`: Create Unicode bookmarks (default=`false`). After package has been loaded, it switches between Unicode and PDFDocEncoding.

When creating bookmarks `hyperref` writes them into the file `filename.out`. At the second run the bookmarks are set.

5.2.2 How to Add Bookmarks

Starred chapter and section headings are not inserted in the table of contents, and thus nor in the bookmark. For manual addition of bookmark,

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

where `level` is -1 for part, 0 for chapter, 1 for section, 2 for subsection, 3 for subsubsection, 4 for paragraph, and 5 for subparagraph.

It is also possible to change bookmarks level. See the following example:

```
\makeatletter
\renewcommand*{\toclevel@part}{0} % Set Part level from -1 to 0
\makeatother
```

5.2.3 Math Titles: • $a^2 + b^2 = \pi_s$

This section is taken from AcroTEX. Sometimes it is desired to have mathematics in the title of a section. Such a title does not translate to PDFDocEncoding automatically. In this case, use`\texorpdfstring{Body title}{Bookmark title}` to offer alternative for your bookmarks. Here is how the section title is generated:

```
\subsubsection{Math Titles: %
   \texorpdfstring{$\bullet\ a^2+b^2=\pi_s$}%
   {\textbullet\ a\texttwosuperior\ + b\texttwosuperior\ = \textpi\_s}%
}
```

TEX math (e.g., π) is not supported. The "unicode" option of hyperref only supports some greek, cyrillic, and some special glyphs. See puenc.def in `c:\texmf\tex\latex\hyperref` directory.

5.3 WWW Links

The hyperref package has several ways of defining www links. The command `\url{URL}` can be used to state and link a URL in the text.

```
\url{http://www.ctan.org/}
\url{ftp://ftp.gnu.org/}
\url{mailto:email@email.com}
```

A more sophisticated way of www links provides the command `\href{URL}{text}` where URL is the address and text is the text displayed in the document. For example, The MikTEX site. To include the graphics with a hypertext link, use`\href{URL}{fig filename}`.

The www links in PDF will be in a cyan-colored box (default), but will be changed to magenta texts when the `colorlinks` option is loaded as in this document.

5.4 Internal Links

`\hypertarget{target name}{text1}` creates an internal link target while `\hyperlink{target name}{text2}` creates an internal link. You can also link external targets. See the `xr` packages that comes with the `hyperref` package.[4]

The hyper links in PDF will be in a red-colored box (default), but will be changed to red texts when the `colorlinks` option is loaded as in this document.

5.5 Acrobat Reader Menu Control

You can also access the menu of Acrobat Reader by using `\Acrobatmenu{menuoption}{text}`. For example, if you want to see the document information of this PDF, click here that is created by `\Acrobatmenu{GeneralInfo}{here}`.

For more menu options, refer to Hyper_man.pdf (by Sebastian Rahtz) in your `\texmf\doc` tree.

6 PDF Manipulation

6.1 PDF Merge

We think that you are very interested in this topic. There may be tons of methods for PDF Merge out there. Here I only describe a few of them that I commonly use.

- **Ghostscript**.
  - To merge several PDF files to one PDF file, use `gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile=merged.pdf 1.pdf 2.pdf 3.pdf`.

    All bookmarks from each pdf file are copied to the merged pdf file, but only the top level bookmark (i.e., referring to each source pdf file) is working. All subsequent bookmarks are not working even though you can see them.
  - To merge several PDF files to one PDF file, write a PS file as shown below:

    ```
    %!
    (a.pdf) run
    (b.pdf) run
    (c.pdf) run
    ```

[4] LaTeX also has its own `xr` package. In this case disable this one.

Type `ps2pdf foo.ps foo.pdf` if the saved PS file is `foo.ps`.

All bookmarks from each pdf file are copied to the merged pdf file, but may not work properly.

- To merge several PS files to one PDF file, use `gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile=merged.pdf 1.ps 2.ps 3.ps`.
- To merge several PS files to one PS file, use `gs -dBATCH -dNOPAUSE -q -sDEVICE=pswrite -sOutputFile=merged 1.ps 2.ps 3.ps`. The only difference is `-sDEVICE=pswrite`.
- If you use MS Windows, use `gswin32` instead of `gs`.

- **Adobe Distiller**

  Adobe distiller can also be used to merge several PS files to one PDF file. Go to your `/Distiller/Xtras` directory and copy `Runfilex.ps` into your working directory (e.g., `c:/temp`). Open the file using any text editor, and change it (the default file path is for Mac). For example,

```
%!
(c:/temp/a.ps) prun
(c:/temp/b.ps) prun
(c:/temp/c.ps) prun
```

  Then distill the PS file.

- **PDCAT**

  An excellent shareware program is PDCAT (`http://www.glance.ch`) for Win32 and Unix systems. PDCAT is a command line tool for concatenating PDF files.

  All bookmarks from each source pdf file are copied to the merged pdf file with added top level bookmarks that refer to each source pdf file. But the title of the added top level bookmark is blank (default).

  To add bookmark title, use `-I "New Title"` in front of the source pdf file. For example

```
pdcat -I "1 Intro" 1.pdf -I "2 Background" 2.pdf merged.pdf
```

  It also features a variety of modification or enhancement functions like adding a background or logo to each page of an input file, adding hyper links and bookmarks, and so on.

- **pdfpages**

  Another excellent method is to use `pdfpages` package. For example,

```
\usepackage{pdfpages}
...
\includepdf[pages={1-10}, nup=2, scale=1.0]{imsi.pdf}
```

  Pages of 1-10 of `imsi.pdf` is inserted as 2 ups into the current position. This package provides various merging effects and can add links to the merged pdf files. But remember that the `pdfpages` package only works with pdfLaTeX.

  You can also include *PNG* even though its manual does not say about including PNG instead of PDF.

- **pdftk**

  Yet another excellent freeware is `pdftk` (`www.accesspdf.com`). The following example merges two PDFs in the order of Pages 1–5 of `in1.pdf`, whole pages of `in2.pdf`, and Pages 6–end of `in2.pdf`.

```
pdftk A=in1.pdf B=in2.pdf cat A1-5 B A6-end output out.pdf
```

6.2 PDF Split

- You can use pdftk:

```
pdftk A=input.pdf burst
```

- If you want to convert PDF to a single-paged graphic file (e.g., JPEG), Ghostscript can do.

```
gs -sDEVICE=jpeg -r100 -dNOPAUSE -sOutputFile=test%d.jpg input.pdf
```

  Then the input.pdf file is converted to test1.jpg, test2.jps, and so on.

6.3 PDF to PS

I do not know if this conversion is actually needed, but there are some methods to do this.

- **pdf2ps** in GhostScript.

```
pdf2ps input.pdf output.ps
```

  The font in the PS file is Type 3 font.
- **pdftops** in xpdf. Its windows version is also available at http://www.foolabs.com/xpdf/download.html.

```
pdftops -f 3 -l 10 input.pdf output.ps
```

  This program also supports lots of options. For example, the above example converts only pages 3 to 10 of PDF into PS. For more information, type `pdftops -help`.

6.4 PDF/PS to Text

`ps2ascii` extracts ASCII text from PS and PDF, using ghostscript.

```
ps2ascii input.ps [output.txt]
ps2ascii input.pdf [output.txt]
```

`pstotext` in xpdf extracts ASCII text PDF.

```
pstotext input.pdf output.txt
```

7 PDF Presentation

PDF can also be used as a presentation tool. You can use all the benefits of LaTeX as well as new presentation features such as overlay, transition, and animation.

Visit Screen Presentation Tools to see a very long list of screen presentation tools. In this section only Prosper and Beamer classes are described because they support PSTricks that is the most powerful graphics tool inside LaTeX.

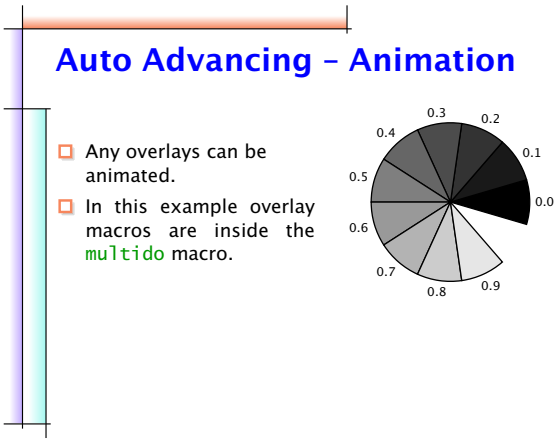

Figure 2: Prosper screen shot.

7.1 Prosper

Prosper (http://prosper.sourceforge.net), built on seminar package, is a very powerful PDF presentation tool.

- LaTeX class for fancy PDF presentation
- Only supports `dvips/ps2pdf` due to its full support for PSTricks!
- Output formats: transparency (in PS) and screen (in PDF)
- Supports transitions and overlays (No need of PDF enhancement tools such as PPower4 or TeXPower)
- Easy to use.
- Easy to expand. Create your own style files!

A screen shot of Prosper is in Figure 2. A Prosper guide file is available from our home page, http://www.geocities.com/kijoo2000/index.html

7.2 Beamer

Beamer (http://tal.cs.tu-berlin.de/tantau/software/index.html) is another very powerful PDF presentation tool.

- LaTeX class for fancy PDF presentation
- Supports `dvips/ps2pdf` and `pdftex`
- Output formats: screen, transparency, notes, handouts (in PDF)
- Supports transitions and overlays (No need of PDF enhancement tools such as PPower4 or TeXPower)
- Create your own style files?

A screen shot of Prosper is in Figure 3. Beamer guide files (with and without PSTricks) are available from our home page, http://www.geocities.com/kijoo2000/index.html

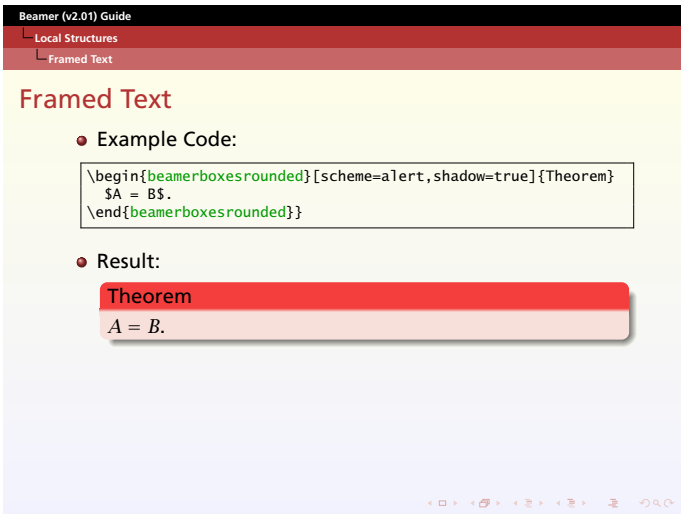

Figure 3: Beamer screen shot.

8 Additional PDF Topics

8.1 Zipping PDF Files?

Since PDF is already compressed, do not try to zip PDF.

8.2 Font Embedding Check

Before you distribute your PDF, embed all fonts except some basic PostScript fonts such as Times, Courier, and Helvetica. These fonts are not embedded by default.

To check fonts used in your PDF, go to File - Document Properties - Fonts. Click here to see the fonts used in this PDF document.[5]

Another way to check fonts is type `gswin32 foo.pdf`. Ghostscript shows the first page of your PDF after checking fonts. If a font is not embedded, it shows "Loading ....". If you see this, the specified font is not embedded. Check each page.

8.3 Replacing Type 3 Fonts in PS

This problem is pointed out several times throughout this document. If your (downloaded) PS file has type 3 fonts (pk fonts or bitmap fonts), the converted PDF file is really illegible on your computer screen. To replace bitmap fonts with the outline Type 1 fonts, you may use one of the following methods.

- I heard that Adobe Reader Version 6.0 can automatically replace bitmap fonts with Type 1 fonts. I did not try this method since Adobe Reader 6.0 is really slow in loading.
- The program **pkfix** (perl script by Heiko Oberdiek[6]) can work fine. For MS Windows,

```
perl pkfix.pl --verbose input.ps output.ps
```

[5] This menu is not working with Acrobat Reader 6.0. The menu might be changed. Will be corrected later.
[6] You know who he is.

Now the `output.ps` is ready for ps2pdf conversion. The PS file that was generated by older dvips may have problems to exchange the fonts since older dvips does not include the informative line "`%DVIPSBitmapFont: ...`".

- In this case you may work with PStill, a shareware program. An evaluation version will put demo logo to your output PDF!

8.4 PS Graphics

If you have LaTeX and/or EPS files created with PSTricks or PSfrag, you cannot directly use `dvipdfm` or `pdflatex` since these packages put PS special codes. Of course there is no problem in this case if you use `ps2pdf`, a robust converter.

Here are some (partial) remedies:

- Use `ps4pdf` package, which wraps PS special codes inside a pdflatex-processed document. Look at the following example:

```
\documentclass{article}
\usepackage{ps4pdf}
\usepackage{pstricks}
\begin{document}
\PSforPDF{%
  \begin{pspicture}(0,0)(2,2)
    \pscircle(1,1){1}
  \end{pspicture}
}%
\PSforPDF{%
  \begin{equation}
    f(x) = \sum U_i
  \end{equation}
}
\savePics % save all \PSforPDF stuffs to filename-pics.pdf
\end{document}
```

Each PS special is enclosed by `\PSforPDF{...}`, resulting in PDF image in a separate file (e.g., `imsi-pics.pdf` if the source is `imsi.tex`). If you have n `\PSforPDF{...}` macros in the source file, there will be n pages (with exact bounding box) in the `imsi-pics.pdf` file. Non PSTricks commands (e.g., equation in this example) can also be enclosed by `\PSforPDF{...}` for producing PDF image.[7] To compile this

```
latex imsi.tex
dvips -Ppdf -G0 -o imsi-pics.ps imsi.dvi
ps2pdf imsi-pics.ps imsi-pics.pdf % PSforPDF is converted to PDF images
pdflatex imsi.tex                 % Comipile with created PDF images
```

- Use `pdftricks` package. There may be difficulty in determining a bounding box in some PS environments. For more information, refer http://sarovar.org/projects/pdftricks/.
- Use the `unpsfrag` command (available from http://www.gts.tsc.uvigo.es/~fiz/unpsfrag) to convert a LaTeX document that uses the `PSfrag` into one that doesn't, thus letting you use `pdflatex`.
- Use `pgf` (portable graphics format) package. This package is similar to PSTricks but less powerful. However, it enables you to create graphics for both PS and PDF. So you can use this package with `dvips`, `dvipdfm`, or `pdftex`. For more information, refer http://gd.tuwien.ac.at/publishing/latex/CTAN/graphics/pgf/.

[7] So equations can be converted to other graphics formats easily.

8.5 Get "Acrobat Reader" Icon and Link

Get "Acrobat Reader" Icon and Link in Your WWW

The following HTML fragment creates a link like this to the appropriate page.

```
<a href="http://www.adobe.com/prodindex/acrobat/readstep.html"
target="_blank"><img src="http://www.adobe.com/images/getacro.gif"
width=88 height=31 border=0 alt="ACROBAT READER"></a>
```

8.6 PDF with Java Script

Another option is to use `insDLJS` package developed by D.P. Story (developer of AcroTEX). This package defines a new environment, `insDLJS`, used for inserting Java Script into a PDF file created from a LaTeX source. This package works correctly for users of `pdftex` or `dvipdfm`. For those that use the Acrobat Distiller (specifically, those that use either dvips or dripstone to produce a postscript file, which is then distilled), you are required to have Acrobat 5.0 (or later).

The following is a quick illustration of the use of the new environment.

```
\documentclass{article}
\usepackage[pdftex]{hyperref}
\usepackage[pdftex]{insdljs}

\newcommand\tugHello{Welcome to TUG 2001!}
\begin{insDLJS}[HelloWorld]{mydljs}{My Private DLJS}
function HelloWorld() { app.alert("\tugHello", 3); }
\end{insDLJS}

\begin{document}
\begin{Form} % needed for \PushButton
\section{Test of the \texttt{insDLJS} Package}
% use built in form button of hyperref
Push \PushButton[name=myButton, onclick={HelloWorld();}]{Button}
\end{Form}
\end{document}
```

8.7 Security Restrictions?

Suppose that you have purchased eBooks or secured PDFs. Some security features may constantly annoy you or block your legal right of proper use. Even you may forget the password. A technical web site, Gallery of Adobe Remedies, may help you in this situation.

9 Useful Links

- Effective Scientific Electronic Publishing (http://www.cl.cam.ac.uk/~mgk25/publ-tips.html).
- Producing HTML and PDF files with LaTeX (http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/makingWWWdocs.html).
- Using LaTeXto Create Quality PDF Documents over the World Wide Web (http://www.math.uakron.edu/~dpstory/latx2pdf.html).
- AcroTeX (by D.P.Story) (http://www.math.uakron.edu/~dpstory/acrotex.html).

10 Note

If you find any errors or have suggestions, please contact me at `kijoo2000 at yahoo dot com`.