Seminar demonstration files

Introduction

Denis Girou
July 2002

1 – Introduction

- These files want to demonstrate the various capabilities of the **Seminar** package (from Timothy VAN ZANDT) to produce screen oriented presentations
- They would like to show that **Seminar** (with all the interactive features added by the 'hyperref' package) has by itself all the major required features, since it beginning
- They emphasize only the features adapted to **screen** presentations. They are not at all devoted to show all the capabilities and options of **Seminar** itself, which are supposed here to be known. For them, you must refer to the **Seminar** documentation[a] and the **Seminar** FAQ[b].

[a] See `http://www.loria.fr/services/tex/classes/sem-user.pdf` or the file `sem-user.dvi` on your local installation.
[b] See `http://www.tug.org/applications/Seminar/Seminar-FAQ.html` and `http://www.tug.org/applications/Seminar/Seminar-FAQ.ps.gz`

2 – Warnings and advices

- If you want to reuse some parts of these demonstration files, **first read all this introductory document with care!**
- **No support** of any kind is provided. Do not write me about these files, for any additional explanations or to adapt them to your own needs, except if you have to send me informations of general interest or new innovative examples that you want to share.
- The source files are commented and try to be self-explanatory
- Just use these files as **pedagogical examples**. They will clearly require real understanding, work and some knowledge of (LA)TEX programming to be adapted to your own needs. Do not try to use them if you are not convinced that you must make these **efforts**...
- A lot of packages (around 10...)[a] have been written these last years in the LATEX world to produce screen oriented presentations. Look at them before to make your own choice[b].
- Some examples are only included for demonstration purpose, but are really not to be used in *real* situations (because they will generate too huge files, because, even if the effect is rather aesthetic, the slides themselves will be difficult to read, etc.). So, use these examples with care and study them before to copy parts of them in your own presentations!

[a] See `http://www.miwie.org/presentations/presentations.html` from Michael WIEDMANN for an attempt to offer a summary.
[b] In my personal opinion, only the presentation styles from ConTEXt offer in the TEX world significative new ideas, new layouts and new features over what **Seminar** with 'hyperref' provide since 1999. See `http://www.pragma-ade.nl/show-pre.pdf`

Seminar demonstration files: introduction

- `Makefiles` are given for Unix systems:
  - for other systems, just execute the same commands in the adapted way...
  - we use here `dvips` and GhostScript as softwares to generate both the PDF and PostScript resulting files. Adapt the `Makefiles` if you use other tools.
  - the main file `Makefile` allow to repeat automatically the specified actions on all the demonstration files
  - the main entries of each individual file `Makefile-sem-demN` are:
    - `pdf-bw`: black and white PDF version,
    - `pdf-color`: color PDF version,
    - `pdf`: alias for `pdf-color`,
    - `postscript-bw`: black and white PostScript version,
    - `postscript-color`: color PostScript version,
    - `postscript`: alias for `postscript-bw`,

Seminar demonstration files: introduction

- `pdf-draft`: color PDF version, but with only one compilation (so the references would be wrong if the auxiliary files have not been generated by previous compilations) and no thumbnails generation,
- `postscript-draft`: black and white PostScript version, but with only one compilation,
- `latex`: LaTeX compilation only,
- `dvips`: generation of the dvi file only, with `dvips`,
- `ps2pdf`: generation of the pdf file only, with GhostScript,
- `mpage1`: generation of a PostScript file suited for printing, if needed (the original ones have very reduced margins, so can be truncated by printers), with one logical page for each physical one,
- `mpage2`: generation of a PostScript file with two logical pages for each physical one,
- `mpage4`: generation of a PostScript file with four logical pages for each physical one,
- `clean`: removing of all the auxiliary files,
- `cleanall`: removing of all the generated files (auxiliary, PDF and PostScript ones).

Seminar demonstration files: introduction

- Always think **both** to the **screen** and the **paper** versions. Look at the way they are managed in the different files, sometimes by inclusion of specific tests relative to the version. The main goal is to have only one source file for the two versions, **without** any change of any character to generate both. The `\SeminarPaperVersion` macro can be tested to generate different behavior:

```
\ifx\SeminarPaperVersion\AnswerYes
  ...
\else
  ...
\fi
```

- The generation of **color** and **black and white** versions are managed in the same way. The goal is again to have only one source file to handle. Look also at the way a second level coding system is used (with the `\HL...` macros defined both, but differently, in the files `hcolor.sty` and `hbaw.sty`). And the `\SeminarColorVersion` macro can be tested to generate different behavior:

```
\ifx\SeminarColorVersion\AnswerYes
  ...
\else
  ...
\fi
```


Seminar demonstration files: introduction

- Choosing the **physical format** of the generated files is not straightforward. What is sure is that you must not change of format between the screen and the paper version, otherwise you will got infinite formatting problems, with each time different behaviors between the two versions. You must also been able to generate a paper version with several logical slides on each physical page, as by definition a slide doesn't contain too many informations.
- Several choices can be done:
  - to generate only PDF versions, a color one for screen and a black and white for paper, using Acroread to generate the PostScript version, or the powerful and flexible 'pdfpages' LaTeX package[a] from Andreas MATTHIAS. In this case, you can choose a physical format different from A4 or Letter, and resize the files when the PostScript ones are generated.

    I do not choose this way here, for speed reasons. The generation of a PDF version is really slow comparing to the PostScript one, both using a PostScript to PDF converter or a TeX to PDF compiler. But if your own files are rather small or even huge but simple, or if you have a very fast computer, you can use this way.

[a] See CTAN:macros/latex/contrib/supported/pdfpages

Seminar demonstration files: introduction

- to generate the screen and the paper version in a different output format (PDF and PostScript ones)
- to use an A4, Letter, screen-like or other size fomat. Here I choose to generate all the files in the A4 format (this would be the same for the Letter format, just adapting some dimensions and positions), to have them formatted for printing[a].
- to generate a PostScript version with several logical pages for each physical one, which is often recommanded as each slide will generally contain few informations. Several tools can be used:
  - the already mentionned 'pdfpages' LaTeX package if you start from a PDF file,
  - the PostScript tool `psnup`[b] from Angus DUGGAN, part of the so-called *PostScript Utilities*,
  - the PostScript tool `mpage`[c] from Marcel J. E. MOL and Mark P. HAHN, which generally require to shuffle the pages, for which `pstops` from the *PostScript Utilities* is useful.

[a] This is not straightforward to start from a non paper physical format of a PostScript file to resize it later to a paper format. You would see that there are various pitfalls trying to do this, and that some tools have not the relevant options to allow that (`mpage` would have problems that `psnup` with `psresize` would not have, but they do not allow exactly the same things, etc.).
[b] See CTAN:support/psutils and `http://www.tardis.ed.ac.uk/~ajcd/psutils`
[c] See `http://www.mesa.nl/download.html`

Seminar demonstration files: introduction

- If the compiler that you use does not allow to import vector graphical formats of files (like JPEG or PNG), take care to not use a converter in the encapsulated PostScript format which will generate huge files (this is the case of most of them). Convert first your files in the JPEG format then use a tool which can produce compressed encapsulated PostScript files:
  - `convert` from `ImageMagick`[a] (but using the EPS2 format and not the EPS one!)
  - `jpeg2ps`[b] from Thomas MERZ,
  - `xwpick`[c] from Evgeni CHERNYAEV.
- If the converter that you use to produce the PDF files is not able to include only one time an external file loaded many times[d], take special care! Do not use backgrounds with external images, do not use logos with external images, do not use navigation buttons with external images, etc. Rather use each time graphical objects generated with a graphical (La)TeX package. Nevertheless, if you really need to do that, first try to apply the technique explained by Keith RECKDAHL in his very useful document *Using Imported Graphics in LaTeX 2ε*, in the chapter 15, *Including an EPS File Multiple Times* (see CTAN:info/epslatex.pdf or epslatex.ps).

[a] See `http://www.imagemagick.org/`
[b] See CTAN:nonfree/support/jpeg2ps
[c] See CTAN:support/xwpick which has only the version 2.10, but a version 2.20 is available on the mirrors of the X11 contributions (see `http://ftp.x.org/contrib/applications/xwpick-2.20.tar.gz`).
[d] Typically, this is the case with GhostScript.

Seminar demonstration files: introduction

- For the paper version, load the 'nohyperref' package (part of the distribution of the 'hyperref' one), to deactivate the interactive features, as done here in the `seminar.con` file. Otherwise the PostScript resulting file will contain unnecessary extra commands nearly everywhere.

We can of course put **notes** and require as usual to print only slides, notes or both, using the `slidesonly` and `notesonly` parameters of the document class.

But, for instance, we can also ask to print the notes only in the paper version, as we have done here:

```
% To print the notes in the paper version only
\ifx\SeminarPaperVersion\AnswerYes
\else
  \slidesonlytrue
\fi
```

3 – Pitfalls

- Unfortunately, you can found several kinds of **pitfalls** if you try to compile yourself these files:
  - First, take care that several files require rather huge ressources in **memory** (and in CPU time if you have not a recent processor) to be compiled. You could have to increase the memory allowed to your TeX compiler.
  - The other major source of problems would be that your installation could have some **too old versions** of some packages[a], which have incompatibilities with some other packages or not yet the required features. So, in case of problems not related to the ressources used, first identify the package concerned and check it version number compared to the one I used myself (given below). If your one is older, update it from your nearest CTAN mirror. From what I can guess, the most common problems you can have would be related to:
    - 'fancybox': a version $\geq$ 1.3 is required,
    - 'hyperref': a version $\geq$ 6.71x is required,
    - 'listings': a version $\geq$ 1.0 patch 1.0d (for the old version 0.21 patch $\geq$ 0.21q) is required,
    - 'semhelv': a version $\geq$ 1.4 is required,
    - 'thumbpdf': a version $\geq$ 2.0 is required,

[a] From my experience, never trust your distribution! Check by yourself and make by hand the necessary updates from CTAN.

- 'pst-eucl' is not on CTAN: you must download it from it Web page (see below) or comment the example which use it in the file `sem-dem6.tex`[b],
- 'pst-fr3d' is not yet on CTAN: until this moment, you must download it from the PSTricks Web page (see below).
- If you use `dvips`, take care to use the `-Ppdf -G0` options
- If you use GhostScript as PDF generator, take care to use a version $\geq$ 6.50
- If you use GhostScript as PDF generator, you could see that the versions $\geq$ 7.0 introduce a new problem and genarate rotated Postscript files. A simple workaround[c] is to use commands like: `cat FILE.ps | ps2pdf - > FILE.pdf` rather than `ps2pdf FILE.ps`
- If you do not use `dvips`, of course think to adapt, move, rename or delete the `hyperref.cfg` and `thumbpdf.cfg` configuration files
- To be able to compile the file `sem-dem4.tex` which use the 'listings' package, take care that you need a patch $\geq$ 1.0d for version 1.0 (and a patch$\geq$ 0.21q[d] if you still use the old version 0.21)

[b] Just comment the line `\PstGaussPolygon` in the `figure` environment of this example.
[c] From Erik Frisk.
[d] Available on `http://www.atscire.de/products/listings/021/lstpatch.sty`

- If you use the old version 0.21 of the 'listings' package, you must change the `showstringspaces` keyword by the `stringspaces` one in the file `sem-dem4.tex`
- `dvips` cannot break hyperlinks (you can see such unpleasant effects on some URL in some screen versions of the files). If you really cannot accept this, you must use another tool (like VTeX) to generate the PDF files.
- If you use `mpage`, take care that, even with it `-k` option, if failed in various cases when the file contain some kinds of encapsulated PostScript files (most of the time, these problems can be easily bypassed, removing the lines `%%EOF`, `%%Page: N N`, `%%Trailer` and `%%EndTrailer`)
- If you use the `article` Seminar mode, do not use the `-t landscape` parameter for `dvips`
- I format all the files for the A4 format: make the various adaptations if you want to use another one
- Under Linux, you can have a strange behavior and got strange error messages about fonts if you use the old 4.05 version of Acrobat Reader and if your environment has set a national language different of the english/american one. This was a bug in this Linux version of Acroread[a], so with it you must start Acroread in an english/american environment or modify it shell script as described in the PDF FAQ.

[a] See the PDF FAQ `http://www.stillhq.com/cgi-bin/getpage?area=ctpfaq&page=index.htm` from Michael Still, in it section *OS: Linux specific*.

4 – PDF format generation and viewvers

- PostScript format and PostScript files viewers are not well adapted to screen presentations. PDF format and PDF files viewers are more well suited.
- There are many PDF generators from (LA)TeX source files. Nevertheless, nor *pdfLaTeX*[a] nor *dvipdfm*[b] have the required feature (a PostScript interpreter) to support an interesting usage of **Seminar**. You must rather use:
  - Acrobat (Macintosh and Windows systems only)[c]
  - Bakoma TeX (Windows systems only)[d]
  - GhostScript (all major platforms) (today the easiest solution in most circumstances)[e]
  - PStill (all major platforms)[f]
  - VTeX (Linux, Windows and some other systems) (today the most powerful solution)[g]

[a] See `http://tug.org/applications/pdftex`
[b] See CTAN:dviware/dvipdfm and `http://gaspra.kettering.edu/dvipdfm`
[c] See `http://www.adobe.com/products/acrobat`
[d] See CTAN:nonfree/systems/win32/bakoma and `http://www.tex.ac.uk/tex-archive/systems/win32/bakoma`
[e] See `http://www.cs.wisc.edu/~ghost`
[f] See `http://www.pstill.com`
[g] See CTAN:systems/vtex, `http://www.micropress-inc.com` and `http://www.micropress-inc.com/linux.htm`

Seminar demonstration files: introduction 16

- I have personally tested these files with GhostScript versions 6.50, 6.51, 7.04, 7.05 and 7.20, PStill version 1.55.8[a] (both with files compiled with the Linux teTEX distribution) and with the Linux VTEX compiler versions 7.59F and 8.00a[b]
- To view the generated PDF files, use Acrobat Reader[c], GhostScript[d] or (on Unix systems only) xpdf[e]. Acrobat Reader is today from far the most powerful tool.
- I have personally tested these PDF output files with Acroread versions 4.05 and 5.05 on Linux, Acroread version 5.05 on Windows 98 and xpdf versions 1.0 and 1.1 on Linux

[a] But without success on these files, even without using the 'sem-helv' package.
[b] With some pitfalls: the example 6 on animated graphics does not compile due to lack of memory (you must split it in several files), the example 7 contain some usage of 'hyperref' not defined in VTEX and few other minor problems in some of the other files. Also, with VTEX, better use the JPEG versions of the external files rather than the EPS ones.
[c] See `http://www.adobe.com/products/acrobat/readstep2.html`
[d] See `http://www.cs.wisc.edu/~ghost`
[e] See `http://www.foolabs.com/xpdf/home.html`

Seminar demonstration files: introduction 17

5 – Required packages

- Outside **Seminar**, the following packages can be considered as mandatory[a]:
  - 'fancyhdr' (1.99d), from Piet VAN OOSTRUM (see CTAN:macros/latex/contrib/supported/fancyhdr),
  - 'hyperref' (6.72s), from Sebastian RAHTZ, David CARLISLE and Heiko OBERDIEK (see CTAN:macros/latex/contrib/supported/hyperref and `http://www.tug.org/applications/hyperref`),
  - 'pstcol' (1.0a), from David CARLISLE (see CTAN:macros/latex/required/graphics),
  - 'PSTricks' (97), from Timothy VAN ZANDT (see CTAN:graphics/pstricks and `http://www.tug.org/applications/PSTricks`),
  - 'thumbpdf' (3.2), from Heiko OBERDIEK (see CTAN:support/thumbpdf).

[a] I put the number of the versions that I used when I generated the files that you read. If you have problems to compile these source files, first check that your own packages are at least at this level, comparing to these informations.

Seminar demonstration files: introduction 18

- In these examples, I use many other packages, but just for some needs in some examples or only for convenience, and they are not mandatory:
  - 'arrayjob' (1.03), from Zhuhan JIANG (see CTAN:macros/generic/arrayjob),
  - 'calc' (4.1b), from Kresten Krab THORUP, Frank JENSEN and Chris ROWLEY (see CTAN:macros/latex/required/tools),
  - 'caption2' (2.0), from Axel SOMMERFELDT (see CTAN:macros/latex/contrib/supported/caption2),
  - 'colortbl' (0.1h), from David CARLISLE (see CTAN:macros/latex/required/graphics),
  - 'fancybox' (1.3), from Timothy VAN ZANDT (see CTAN:macros/latex/contrib/supported/fancybox),
  - 'fancyvrb' (2.6), from Timothy VAN ZANDT (see CTAN:macros/latex/contrib/supported/fancyvrb),
  - 'fvrb-ex' (1.7), from Denis GIROU (see CTAN:macros/latex/contrib/supported/fancyvrb/contrib),
  - 'graphicx' (1.0f), from David CARLISLE and Sebastian RAHTZ (see CTAN:macros/latex/required/graphics),
  - 'listings' (1.0 and 0.21), from Carsten HEINZ (see CTAN:macros/latex/contrib/supported/listings and `http://www.atscire.de/products`),

Seminar demonstration files: introduction 19

- 'multido' (1.4), from Timothy VAN ZANDT (see CTAN:macros/generic/multido),
- 'multirow' (1.5), from Piet VAN OOSTRUM (see CTAN:macros/latex/contrib/supported/fancyhdr),
- 'pifont' (7.2), from Sebastian RAHTZ (see CTAN:macros/latex/required/psnfss/psfonts.dtx),
- 'pst-eucl' (0.5β), from Dominique RODRIGUEZ (see `http://dominique.rodriguez.9online.fr/pst-eucl`),
- 'pst-fr3d' (1.0), from Denis GIROU (see CTAN:graphics/pstricks/contrib/pst-fr3d after it public release and `http://www.tug.org/applications/PSTricks/More`),
- 'pst-key' (1.11), from David CARLISLE (see CTAN:graphics/pstricks/contrib/misc),
- 'pst-lens' (1.0), from Denis GIROU and Manuel LUQUE (see CTAN:graphics/pstricks/contrib/pst-lens and `http://www.tug.org/applications/PSTricks/Lens`),
- 'pst-slpe' (1.0), from Martin GIESE (see CTAN:graphics/pstricks/contrib/pst-slpe),
- 'truncate' (3.6), from Donald ARSENEAU (see CTAN:macros/latex/contrib/other/misc),
- 'url' (1.4), from Donald ARSENEAU (see CTAN:macros/latex/contrib/supported/url).

Seminar demonstration files: introduction

6 – Contents

- Each file emphasize a different aspect:
  - backgrounds: sem-dem1.pdf
  - transition effects: sem-dem2.pdf
  - overlays (I): sem-dem3.pdf
  - overlays (II): sem-dem4.pdf
  - navigation bars and panels: sem-dem5.pdf
  - animated graphics: sem-dem6.pdf
  - movies, sounds, annotations and miscellaneous: sem-dem7.pdf
- Obviously, everything can be mixed together...